Design of a proper set-up for low current measurements of insulation systems

Master thesis

Jordi Rovira Buhigas

Supervisor: M.Sc. Farzaneh Vahidi
1. Motivation and scope of work ........................................................................................................... 9

2. Methodologies for low current measurements................................................................. 10
   2.1. Shunt ammeter......................................................................................................................... 11
   2.2. Shunt Ammeter with buffer amplifier................................................................................. 12
   2.3. Feedback Ammeter ............................................................................................................. 13
   2.4. Differences between shunt and feedback.......................................................................... 14

3. Test set-up................................................................................................................................... 14
   3.1. Definitions of technical words designing a low current measurement circuit .......... 14
   3.2. Isolating material.................................................................................................................. 15
   3.3. Arduino UNO .................................................................................................................... 17
      3.3.1. Power pins..................................................................................................................... 18
      3.3.2. Input and Output.......................................................................................................... 19
      3.3.3. Communication ............................................................................................................ 20
      3.3.4. Programming............................................................................................................... 20
      3.3.5. Physical Characteristics ............................................................................................. 21
      3.3.6. ATmega328.................................................................................................................. 21
   3.4. Programming the software ............................................................................................... 23
      3.4.1. Arduino’s software ....................................................................................................... 23
      3.4.2. Storage of the data: “Processing 2” Software.............................................................. 27
   3.5. Testing Arduino.................................................................................................................... 30
      3.5.1. Comparison of accuracy between Arduino, multimeter and Electrometer. ..... 30
      3.5.2. Experiment changing the reference value of Arduino................................................ 34

4. Description of the circuits ......................................................................................................... 37
   4.1. First design............................................................................................................................ 37
      4.1.1. Variation of output depending on the isolating system’s resistance ................. 39
      4.1.2. Operational amplifiers ............................................................................................... 41
      4.1.3. Power supply................................................................................................................. 46
4.2. Modified designs ................................................................. 47
  4.2.1. Update 1: Three ranges and multimeter option .................. 48
  4.2.2. Update 2: Use of another Amplifier .................................. 55
  4.2.3. Update 3: Adapting for negative values .............................. 64
  4.2.4. Update 4: Design to fit into a shielding box ..................... 67

5. Eagle design........................................................................ 71

6. Components and costs estimation........................................ 76

7. Resolution .............................................................................. 77

8. Accuracy of the results with constant resistances ................... 78
   8.1. Positive-Negative Error ..................................................... 83

9. Accuracy of the results with insulating liquid ....................... 86

10. Summary and outlook ........................................................... 89

11. References ............................................................................ 91

A Appendix ................................................................................ 92
Declaration

I hereby declare that I prepared this thesis independently and without outside help or unauthorized aids, nor used other than the stated sources and aids and the sources points extracted have been marked as such.

Stuttgart, October 22, 2013

Signature:______________________________.
Acknowledgements

I want to thank my tutor M. Sc. Farzaneh Vahidi for letting me do this thesis under her supervision and all the help and guidance she has given me throughout the length of the thesis. Also Dipl.-Ing. Mark Jovalekic, who never had no as an answer to any question or doubt that I came to him with.

I want to also thank my girlfriend, Kristin N. Osborn, who gave me all her support and motivation and helped with checking the correctness of the English language.

I want to thank Sophia, Laura and Daniel, with whom I shared the lab most of the days. I thank them for their support in the days in the lab as colleagues and friends.

I also want to thank IEH and the University of Stuttgart for the opportunity to develop my thesis in their department and enjoy the great experience surrounded by great people and professional workers.

Finally, I thank my parents and my sister for giving me the opportunity to go on the ERASMUS program, as well as their support through all the days.
1. Motivation and scope of work

Low-level measurements are an important part of electrical studies. Due to the small values of current and voltage used, the precision of the values becomes very important. For example, a change of 1 nA or 1 nV can make a huge difference.

For that reason, measuring those values requires special study and therefore, special attention, procedures, equipment and designs.

In this thesis, different ways to measure low current, values of pA and nA, will be analyzed and compared. Currently in the market, devices capable of doing this task do exist, but for a very expensive price. The purpose of this study will be to design an economically affordable device capable of producing acceptable results.

The steps outlined in the following pages will focus on measuring polarization current through an isolating system applying a voltage up to 20 kV.

Different processes of measuring will be evaluated. In the end, one will be chosen to take the measurements and show them over a multimeter device while simultaneously logging the values in a computer.
2. Methodologies for low current measurements

This section will outline and describe the basic circuits on which the ammeters and electrometer are based as well as the basics of the operational amplifiers.

An operational amplifier is a device which can have a varying amount of pins. The standard ones are non-inverting input (V+), inverting input (V-), output (V_{out}) and positive and negative voltage supply inputs (V_{s+}, V_{s-}). In addition to those, there can be pins for offset cancelling, clock synchronizing, and many others.

The main equation for the operational amplifier is:

$$V_0 = A \cdot V_+ - V_-$$  \hspace{1cm} (2.1)

The amplifier amplifies the difference of voltage between both inputs multiplied by the open loop gain A. The gain is different at every operational amplifier and cannot be modified.

The range of the gain will typically be approximately 10,000 to 100,000. As it is so big, a very small difference between the inputs creates a big output, usually at the level of the power supply (saturation). Due to this big gain, it is really difficult to control the behavior in an open loop. The most usual way to operate using amplifiers is using a feedback loop. It usually connects the inverting input to a portion of the output.
By using 2 resistances the gain is highly reduced. The transfer function applied in this situation is:

\[ V_{out} = V_{in} \cdot (1 + \frac{R_f}{R_G}) \]  \hspace{1cm} (2.2)

The 2 resistances work as a voltage divider. It is acceptable to make the assumption that \( V_s = V \). For that reason, the voltage between both resistors will have to be the same as the \( V_{in} \), therefore, the relation between the resistors will give the new gain.

For example, if both resistors have the same value, and \( V_{in} = 5 \text{ V} \), the output will be 10 volts. That is because the node between the resistors connected to the inverting input has to also be 5 V, so the output has to be 10 V. It proves the equation:

\[ V_{out} = V_{in} \cdot (1 + \frac{R_f}{R_G}) \]  \hspace{1cm} (2.3)

\[ \Rightarrow 10 = 5 \cdot (1 + \frac{X}{X}) \]  \hspace{1cm} (2.4)

with \( X \) as the value of the resistors.

2.1. **Shunt ammeter**

![Shunt ammeter schematic](image3.png)
The shunt ammeter works by measuring the voltage created over a resistance \( R_s \), through which the current desired to measure circulates. Knowing the values of the shunt resistance \( R_s \) and \( R_A \) and \( R_B \), the current can be calculated from the voltage of the output with the following equation:

\[
V_o = I_{IN} \cdot R_s \cdot \left( 1 + \frac{R_A}{R_B} \right)
\]  \hspace{1cm} (2.5)

The shunt resistance should be as small as possible, for reasons such as: better stability with time and temperature, reduced input constant time and reduced voltage burden. However, if it is too small, it could lead to degrading of the signal-to-noise ratio and introducing some noise and zero drift.

### 2.2. Shunt Ammeter with buffer amplifier

This case is similar to the shunt ammeter but with a buffer amplifier between the shunt resistance and the operational amplifier. This case is used when the current measured is really low and the input bias current can be around the same range of the desired measured values. For that reason, a special ultralow input bias current operational amplifier is used. It is mounted in a follower configuration, without adding gain but providing stability. The buffer has ideal infinite input impedance and zero output impedance which allows to transfer voltage from first to second part of the circuit.

A voltage follower configuration is where the output is connected with the inverting input with a unity gain, giving the following transfer function:

\[
V_{out} = V_{in}
\]  \hspace{1cm} (2.6)

![Image 4 Voltage follower with ultralow input bias current](image)
Currently exists ultralow input bias current as low as 40fA like AD549L

![Shunt ammeter schematic with buffer amplifier](image5)

**Image 5** Shunt ammeter schematic with buffer amplifier

### 2.3. Feedback Ammeter

In the feedback ammeter, there is a resistor connected between the output and the negative input. The positive input is connected to the ground. The equation is:

\[ V_O = -I_{IN} \cdot R_F \]  

(2.7)

![Feedback Ammeter schematic](image6)

**Image 6** Feedback Ammeter schematic
The input current flows through the feedback resistor. It is suitable that the operational amplifier has a low input bias current, so the full current flows through the feedback resistor and not inside the operational amplifier. That way, the measurement won’t be affected by the own operational amplifier and will be as accurate as possible so the measurement is as accurate as possible.

The commercial picoammeter is an example of hardware using the Feedback ammeter situation.

2.4. **Differences between shunt and feedback**

Shunt measurement is the most common type of measuring for the wide range they can achieve. Feedback ammeters are better in cases of really low current measurements [2].

However, as long as the shunt resistor is significantly smaller than the resistance of the DUT and the currents to be measured are not very small (not much lower than microamp level \(10^{-6} \mu A\)), shunt ammeters work fine. [2]

In shunt ammeter, the voltage drop is created between the output and the input of the high gain amplifier instead of between the inputs like the shunt. The main characteristic for the voltage feedback ammeter is the low voltage burden, which makes it more appropriate for low level measurements. The input voltage is equal to the output voltage divided by the op-amp gain (typically 100,000), so the voltage burden has now typically been reduced to microvolts. [2].

3. **Test set-up**

3.1. **Definitions of technical words designing a low current measurement circuit**

Before starting to describe the different parts and elements of the circuits, it is important to get familiar with the technical words that are going to be used frequently throughout the study and the differences between them.
• **Accuracy**
  The definition of accuracy is the closeness of agreement between the measurement and its true value [3]. It is often specified as ± (% of reading + count) or ± (ppm of reading + ppm of range). The percentage of reading is more significant when the reading is close to full scale and the percent of range is most significant when the reading is a small fraction of full scale [3].

• **Error**
  The deviation of a measurement from its true value. It can be expressed as the difference or as a ratio [3].

• **Resolution**
  The smallest portion of the signal that can be observed [3]. This means the number of counts that can be displayed, based on the number of digits. An example is: ±(0,05% + 1 count) reading 10.000V equals a total error of ±(5mV + 1 mV) out of 10V. Usually, the higher resolution, higher accuracy.

• **Sensitivity**
  The smallest change in the signal that can be detected [3]. That means that a change bellow the sensitivity value won’t be reflected on the reading. The sensitivity will depend on the display resolution and the measurement range.

3.2. **Isolating material**

The isolating system consists of a metal plate test cell device with a surface of 300 mm, with 3 electrodes (High, Low and Guard) inside containing an oil or paper between them. The distance between the 2 electrodes can be changed between 1 and 5 mm.

On the system a voltage between 5 kV and 20 kV will be applied.
The resistance of the isolating system depends on the temperature and the strength of the electrical field applied. It can have a range between 100 MΩ and 100GΩ depending on the conditions.

Image 7 and 8 show the test cell used for current measurement. The insulation system goes placed inside, either oil or paper.
3.3. **Arduino UNO**

Arduino UNO is the device that will be used to read the values and send them to the computer to be logged. It will take some time and processes to adjust and program the device so the results are the expected ones.

Arduino is a prototype board, easy to use and without the need to add significant hardware. It can perform many different operations. All the components are mounted on a board, with the input and output connections defined.

The platform can also communicate with other devices through digital IN/OUT pins or through the USB. In this case it will be communicating with a computer that will receive and log the data coming from it through a USB.

The Arduino operations are made through the Arduino development environment. It is based on C computer language and will be later discussed and explained.

Referring to specifications, the Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.

The most common specifications are:

<table>
<thead>
<tr>
<th>Table 1 Arduino specifications [4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
</tr>
<tr>
<td>Input Voltage (limits)</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
</tr>
<tr>
<td>Analog Input Pins</td>
</tr>
<tr>
<td>DC Current per I/O Pin</td>
</tr>
<tr>
<td>Flash Memory</td>
</tr>
<tr>
<td>SRAM</td>
</tr>
<tr>
<td>EEPROM</td>
</tr>
<tr>
<td>Clock Speed</td>
</tr>
</tbody>
</table>
3.3.1. **Power pins**

The Arduino Uno can be powered via the USB connection or with an external power supply. External (non-USB) power can come from either an AC-to-DC adapter or a battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board’s power jack. Lead from a battery can be inserted in the GND and Vin pin headers of the POWER connector. By default it is provided 5V from the USB. As a computer will be used to show the results, in this case power will be powered via the USB connection.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it’s using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

- **5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V).

- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

- **GND.** Ground pins.
3.3.2. **Input and Output**

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and have an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial**: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

- **External Interrupts**: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

- **PWM**: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the `analogWrite()` function.

- **SPI**: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function.

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with `analogReference()`.
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

### 3.3.3. Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer.

### 3.3.4. Programming

The Arduino Uno can be programmed with the Arduino software. This software can be downloaded for free from the Arduino webpage. The ATmega328 on the Arduino Uno comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer.
3.3.5. **Physical Characteristics**

The maximum length and width of the Uno PCB are 68.58 mm and 53.34 respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (4,064 mm), not an even multiple of the 100 mil (2.54mm) spacing of the other pins.

![Arduino front and back sides](image)

**Image 10** Arduino front and back sides

3.3.6. **ATmega328**

The microcontroller ATmega328 is the main part of the Arduino board. It has 32 pins and a CPU of 8 bits. The main characteristics are the following:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash (kbytes)</td>
<td>32 Kbytes</td>
</tr>
<tr>
<td>Pin Count:</td>
<td>32</td>
</tr>
<tr>
<td>Max. Operating Frequency:</td>
<td>20 MHz</td>
</tr>
<tr>
<td>ADC channels</td>
<td>8</td>
</tr>
<tr>
<td>ADC Resolution</td>
<td>10 bits</td>
</tr>
<tr>
<td>ADC Speed</td>
<td>15 ksps</td>
</tr>
<tr>
<td>EEPROM</td>
<td>1024 Bytes</td>
</tr>
<tr>
<td>RAM</td>
<td>2 KBytes</td>
</tr>
<tr>
<td>Operating Voltage (Vcc)</td>
<td>1.8 to 5.5</td>
</tr>
</tbody>
</table>

**Table 2** ATmega328 Specifications [5]

Inside the ATmega328, an important part is the Analog-to-Digital converter.
The ADC is connected to an 8-channel Analog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port A. The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents GND and the maximum value represents the voltage on the AREF or USB supply.

A simple ended conversion is:

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$  \hspace{1cm} (3.1)

Where $V_{IN}$ is the voltage at the input pin and $V_{REF}$ is the voltage selected as a reference.

For example, if $V_{REF}$ is 5V, an example of conversion would be:

<table>
<thead>
<tr>
<th>$V_{IN}$ (V)</th>
<th>Conversion value</th>
<th>Digital Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0000</td>
<td>0</td>
</tr>
<tr>
<td>0.001</td>
<td>0.2048</td>
<td>0</td>
</tr>
<tr>
<td>0.002</td>
<td>0.4096</td>
<td>0</td>
</tr>
<tr>
<td>0.003</td>
<td>0.6144</td>
<td>0</td>
</tr>
<tr>
<td>0.004</td>
<td>0.8192</td>
<td>0</td>
</tr>
<tr>
<td>0.005</td>
<td>1.0240</td>
<td>1</td>
</tr>
<tr>
<td>0.01</td>
<td>2.0480</td>
<td>2</td>
</tr>
<tr>
<td>0.02</td>
<td>4.0960</td>
<td>4</td>
</tr>
<tr>
<td>0.05</td>
<td>10.24</td>
<td>10</td>
</tr>
<tr>
<td>0.1</td>
<td>20.48</td>
<td>20</td>
</tr>
<tr>
<td>0.2</td>
<td>40.96</td>
<td>40</td>
</tr>
<tr>
<td>0.5</td>
<td>102.40</td>
<td>102</td>
</tr>
<tr>
<td>1</td>
<td>204.80</td>
<td>204</td>
</tr>
<tr>
<td>2</td>
<td>409.60</td>
<td>409</td>
</tr>
<tr>
<td>4</td>
<td>819.20</td>
<td>819</td>
</tr>
<tr>
<td>5</td>
<td>1024.00</td>
<td>1024</td>
</tr>
</tbody>
</table>
3.4. **Programming the software**

The process to program the Arduino consists of 2 parts. First of all the process of making Arduino read the values and show them on the screen and second part, to log the values appropriately for representing them after.

3.4.1. **Arduino’s software**

The first step when attempting to work with Arduino is to program it so it can do the needed functions. Arduino is based in C/C++ programing. It has its own library of functions, which allows many actions to be made. The library and other tutorials found on the internet can be consulted.

The complete code used is the following:

```c
#define NUM_SAMPLES 30   // set a number of samples constant
#include< stdlib.h >     // import de library containing the function “dtostrf()”

int sum = 0;           // create and initialize variables
unsigned char sample_count = 0;
float voltage = 0.000;
float current = 0.000;
float correction = 0.000;

void setup() {         // start the communication with the computer
  Serial.begin(9600);
}

void loop(){          // add all the reads until the samples number
  while (sample_count < NUM_SAMPLES) {
    sum += analogRead(A3);   // read pin A3 but could be any other pin
    sample_count++;
    delay(10);               // make a read every 10 ms
  }

  voltage = ((float)sum / (float)NUM_SAMPLES * 5.06) / 1024.000;   // make the average of the samples, and reference voltage and divide by the arduino number of values

  Current = voltage*10000/400;  // multiply or divide the result value by the internal resistance
  Correction = Current + (current * -0.0085 + 5.25);  // applies the correction depending on the range

  Serial.print(Current);   // Show the result throught the serial port
  Serial.println(correction);

  sample_count = 0;       // restarts the variables
  sum = 0;

  Serial.print("#S|PROVAPRI1|[");  // Command for making the program “getwingo” record the values
  Serial.print(dtostrf(total,6,4,buffer));
  Serial.println("]#");
}   // end of the program
```
This algorithm has to be implemented through the Arduino software and uploaded to the board through the USB.

This program starts defining a constant value, NUM_SAMPLES, which will be used later to make the average of the values read from the input pin. This is used to eliminate single interferences by grouping them together.

Following that, it is important to include any library of which a function is going to be used. In this case, the function dtostrf() is included in the library stdlib.h. The way to include the library to the logarithm is #include.

```
#define NUM_SAMPLES 30  // set a number of samples constant
#include<stdlib.h>  // import de library containing the function “dtostrf()”
```

There are going to be some variables needed, which have to be declared and initialized. Sum and sample_count are going to be used for making the average of numbers, total and voltage will represent real values of voltages. Buffer is going to make space in the buffer for the subaction dtostrf()

```
int sum = 0;  // create and initialize variables
unsigned char sample_count = 0;
float voltage = 0.000;
char buffer[8];
float total = 0.000;
```

With the function Serial.begin(9600) the Arduino begins to communicate with the computer. 9600 sets the speed. It has to be the same in both the computer and the Arduino board.

The main program is found in the void loop (). First, with a while and the condition of “until the number of reads is equal as the expected number of samples, keep reading”, the program reads the value on the analog pin number 3 (A3) with the function analogRead(A3). Delay(10) indicates that every 10 ms Arduino will repeat the loop.
At the end of the `while` there is going to be the sum of `NUM_SAMPLE` values. The `voltage` variable is going to take that number, divide it by `NUM_SAMPLE` values, and scale it to the Arduino. This scaling is as simple as taking the value and multiplying for the maximum value input range 5V (in our case, measured before and it is 5,06V) and divide it by the number of bits it operates, in our case 1024.

```
voltage = ((float)sum / (float)NUM_SAMPLES * 5.06) / 1024.000; //make the average of the samples, and reference voltage and divide by the arduino number of values

Current = voltage*10000/400; //multiply or divide the result value by the internal resistance
Correction = Current + (current * -0.0085 + 5.25); //applies the correction depending on the range
```

The result of these operations is the value of `voltage` is going to be the real voltage at the middle of both resistances. For that reason, it is necessary to multiply the value for the voltage divisor rate, so the resulting value is the voltage at the source.

Next, the values need to be sent to the computer. `Serial.print()` will send through the serial the values and they will be able to be seen at the serial monitor.

```
Serial.print(Current); //Show the result throught the serial port
Serial.println(Correction);
```

To finish the program, the variables need to be restarted, so the program can start again.

```
sample_count = 0; //restarts the variables
```
In this case, it would be convenient to log those values into a text file so afterwards they can be later analyzed. To accomplish the logging operation, there are many different options. The one used here is the program “Gobetwino” which works in tandem with the Arduino program.

Gobetwino function of logging the data requires a special kind of data. The values sent have to be string type as the program doesn’t accept float numbers. For this reason, it is necessary to use the function `dtostrf(VALUE; PRECISION; DECIMAL; BUFFER)`. This function will transform the float number to string with the parameters precision, decimal, buffer. These values have to be big enough so they don’t differ from the float values.

To make “getwingo” record the values, it needs to be sent through `serial.print("#S|name_function_getwingo|[values_to_be_recorded]#")`.

Once we close the program “getwingo”, the values will be recorded in a .txt document with or without the date and time.

```cpp
Serial.print("#S|PROVAPRI1["; //Command for making the program “getwingo” record the values
Serial.print(dtostrf(total,6,4,buffer));
Serial.println("]#"); //end of the program
```

To finish the entire program, we just need to use the closing sign `}

On the Arduino software, there is a button to check for error prior to uploading the program onto the board.
3.4.2. Storage of the data: “Processing 2” Software

Once the Arduino sends the measured value over the serial port, and it is shown on the screen, it is important to be able to save those values. There are many ways to do it, but in this situation what is wanted is that the values get logged in .CSV type, so they can be easily uploaded to Matlab or Excel to represent the values.

The software Processing 2, based on the Arduino’s programming language, can perform this task.

To make Arduino software and Processing 2 software work with each other, there has to be some modifications to the original design.

On the Arduino code, it is very simple, just so the variables printed on the serial port will have to be show the following way:

```java
Serial.print(var1,4);
Serial.print (" , ");
Serial.print(var2,4);
Serial.print (" , ");
Serial.println(var3,4);
```

The variables being measured in this case are var1, var2 and var3. They are float numbers with 4 decimal values. The important part of this procedure is the printing of the comma between the variables and the `Serial.println` at the end. This command makes a new line, so the three variables go on the same line separated by commas and the next time the code makes the loop, it will be stored on the following line.

The part of the Processing 2 will have to be edited from the beginning. The language is similar to Arduinos, but some functions and commands are special for Processing 2.
The code to be written on the Processing page is:

```java
import processing.serial.*;  //import the library

Serial myPort;  //create a variable of the type “Serial”
PrintWriter output;  //creates “printwriter” type variable
int lf = 10;  // created and assigns 10 as an integer value

void setup() {
  size(500,500);  //opens the window
  println(Serial.list());  //shows on screen the ports available
  myPort = new Serial( this, Serial.list()[1], 9600 );  //chose port [1] at 9600 of speed
                                          // (same one as arduino)
  output = createWriter( "measure.csv" );  //creates a “.CSV” file
}

void draw() {
  if (myPort.available() > 0 ) {  //as long as there is serial port data:
    String value = myPort.readStringUntil(lf);  // read until a change of line
    if ( value != null ) {
      //if the value read is not null
      output.println( value );  //save the value the value
      println(value);  //show the value on the screen
    }
  }
}

void keyPressed() {  //when pressing any key:
  output.flush();  //writes the remaining data
  output.close();  //closes the file and save it
  exit();  //quit the program
}
```

There are some special functions that require a little bit of attention:

`Printwriter()` allows to print a text-output stream. It requires the function `createWriter()` to generate the file and ideally it should be flushed and closed with `flush()` and `close()`. 
When using the function `newSerial(this,Serial.list[],9600)`, it is important that the speed of the serial is the same in Arduino and processing 2. In this case, it is 9600. This value can be another one, but it must be the same in both Arduino and Processing 2.

Notice the use of the integer “1f” which has a value of 10. This is because in ASCII, 10 equals to “new line”. Therefore, when the program is using the function `readStringUntil()`, it will read until the a new line is made, which was set at the Arduino program with the `serial.println()`. By doing these commands, the results are just as wanted: a certain number of values on the same line for each iteration of the program and separated by commas.

The function `createWriter()` can create either a `.CSV` file to open with a datasheet program or a `.txt`.

The steps to set are: first connect Arduino to the computer, upload the code and run it. The serial monitor has to be closed because the program just allows one connection. Then, open the processing 2 software, and run the code. Once the program is closed, there will be in the same folder of the specified one, the document with the values saved and ready to open with excel or text editor.
3.5. **Testing Arduino**

The first step before taking any measurements is to check the accuracy of the measuring device. In this case, the Arduino will have to be checked and evaluated, so that the values, specifications and operation are as expected.

The following experiments will check the response from the Arduino known values, checking for accuracy and evaluating for efficiency.

3.5.1. **Comparison of accuracy between Arduino, Multimeter and Electrometer.**

For this experiment, the circuit made consists of a voltage source, from 0 to 25V, a resistance of 2MΩ, and a resistance of 300kΩ.

The measuring equipment is the Arduino, a standard multimeter and the Keithley electrometer.

The Arduino will measure the voltage drop over the 300kΩ resistance and then calculating what it should be at the voltage source. The multimeter and electrometer will measure directly over the voltage source. In addition, there will be an ammeter measuring the current going through the circuit. Knowing the current and the total resistance (2.3MΩ), it is easy to calculate the voltage drop at the source.

![Image 11 Circuit to test Arduino's measuring abilities](image-url)
### Table 4: Table of measured values with different devices

<table>
<thead>
<tr>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
<th>(7)</th>
<th>(8)</th>
<th>(9)</th>
<th>(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25,000</td>
<td>24,942</td>
<td>0,232</td>
<td>25,070</td>
<td>-0,280</td>
<td>24,957</td>
<td>0,172</td>
<td>10,634</td>
<td>24,458</td>
<td>2,167</td>
</tr>
<tr>
<td>22,000</td>
<td>21,880</td>
<td>0,545</td>
<td>22,030</td>
<td>-0,136</td>
<td>21,924</td>
<td>0,345</td>
<td>9,341</td>
<td>21,484</td>
<td>2,344</td>
</tr>
<tr>
<td>18,500</td>
<td>18,422</td>
<td>0,422</td>
<td>18,560</td>
<td>-0,324</td>
<td>18,470</td>
<td>0,162</td>
<td>7,868</td>
<td>18,096</td>
<td>2,182</td>
</tr>
<tr>
<td>16,200</td>
<td>16,208</td>
<td>-0,049</td>
<td>16,320</td>
<td>-0,741</td>
<td>16,249</td>
<td>-0,302</td>
<td>6,922</td>
<td>15,921</td>
<td>1,725</td>
</tr>
<tr>
<td>13,800</td>
<td>13,703</td>
<td>0,703</td>
<td>13,830</td>
<td>-0,217</td>
<td>13,766</td>
<td>0,246</td>
<td>5,864</td>
<td>13,487</td>
<td>2,267</td>
</tr>
<tr>
<td>11,500</td>
<td>11,519</td>
<td>-0,165</td>
<td>11,650</td>
<td>-1,304</td>
<td>11,596</td>
<td>-0,835</td>
<td>4,939</td>
<td>11,360</td>
<td>1,220</td>
</tr>
<tr>
<td>8,150</td>
<td>8,066</td>
<td>1,031</td>
<td>8,180</td>
<td>-0,368</td>
<td>8,138</td>
<td>0,147</td>
<td>3,465</td>
<td>7,970</td>
<td>2,215</td>
</tr>
<tr>
<td>6,500</td>
<td>6,399</td>
<td>1,554</td>
<td>6,510</td>
<td>-0,154</td>
<td>6,481</td>
<td>0,292</td>
<td>2,758</td>
<td>6,343</td>
<td>2,409</td>
</tr>
<tr>
<td>4,700</td>
<td>4,576</td>
<td>2,638</td>
<td>4,690</td>
<td>0,213</td>
<td>4,673</td>
<td>0,570</td>
<td>1,988</td>
<td>4,572</td>
<td>2,715</td>
</tr>
<tr>
<td>2,800</td>
<td>2,637</td>
<td>5,821</td>
<td>2,763</td>
<td>1,321</td>
<td>2,750</td>
<td>1,786</td>
<td>1,169</td>
<td>2,689</td>
<td>3,975</td>
</tr>
<tr>
<td>1,900</td>
<td>1,823</td>
<td>4,053</td>
<td>1,941</td>
<td>-2,158</td>
<td>1,930</td>
<td>-1,579</td>
<td>0,819</td>
<td>1,883</td>
<td>0,878</td>
</tr>
<tr>
<td>1,180</td>
<td>1,080</td>
<td>8,475</td>
<td>1,172</td>
<td>0,678</td>
<td>1,168</td>
<td>1,017</td>
<td>0,496</td>
<td>1,141</td>
<td>3,322</td>
</tr>
<tr>
<td>0,640</td>
<td>0,543</td>
<td>15,156</td>
<td>0,642</td>
<td>-0,313</td>
<td>0,637</td>
<td>0,547</td>
<td>0,270</td>
<td>0,621</td>
<td>2,969</td>
</tr>
<tr>
<td>0,330</td>
<td>0,233</td>
<td>29,394</td>
<td>0,327</td>
<td>1,061</td>
<td>0,325</td>
<td>1,576</td>
<td>0,137</td>
<td>0,316</td>
<td>4,219</td>
</tr>
<tr>
<td>0,250</td>
<td>0,155</td>
<td>38,000</td>
<td>0,254</td>
<td>-1,680</td>
<td>0,253</td>
<td>-1,144</td>
<td>0,107</td>
<td>0,246</td>
<td>1,744</td>
</tr>
<tr>
<td>0,140</td>
<td>0,039</td>
<td>72,143</td>
<td>0,135</td>
<td>3,857</td>
<td>0,134</td>
<td>4,593</td>
<td>0,056</td>
<td>0,129</td>
<td>8,131</td>
</tr>
<tr>
<td>0,120</td>
<td>0,003</td>
<td>97,500</td>
<td>0,119</td>
<td>1,167</td>
<td>0,118</td>
<td>1,917</td>
<td>0,049</td>
<td>0,113</td>
<td>5,777</td>
</tr>
<tr>
<td>0,020</td>
<td>0,000</td>
<td>100,000</td>
<td>0,018</td>
<td>10,000</td>
<td>0,017</td>
<td>13,450</td>
<td>0,006</td>
<td>0,015</td>
<td>26,400</td>
</tr>
<tr>
<td>0,015</td>
<td>0,000</td>
<td>100,000</td>
<td>0,016</td>
<td>-6,667</td>
<td>0,015</td>
<td>-3,067</td>
<td>0,006</td>
<td>0,013</td>
<td>12,600</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
<th>(6)</th>
<th>(7)</th>
<th>(8)</th>
<th>(9)</th>
<th>(10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S=8,224</td>
<td>S=3,104</td>
<td>S=3,387</td>
<td>S=55,943</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1): Ideal value (V)

(2) Arduino measured values (V)

(3) Relative error or Arduino (%)

(4) Multimeter measured values (V)

(5) Relative error Multimeter values (%)

(6) Electrometer measured values (V)

(7) Relative error electrometer (%)

(8) Current measured with pico ammeter (µA)

(9) Voltage calculated from the pico ammeter and resistance (2,3MΩ) (V)

(10) Relative error for pico ammeter (%)

Notice that the last column is the ideal voltage calculated by the current measured by the pico ammeter and the resistance values (2,3 MΩ).
\[ V = I \cdot R \] (3.2)

The last row shows the standard deviation, for each of the individual errors.

The standard deviation has been calculated as:

\[ S = \frac{x_i - \bar{x}}{n - 1} \]

The electrometer and the multimeter show the smallest ones. Arduino is bigger, because at low voltages it is not able to read appropriately, therefore, it shows a big error, which affects the deviation. The calculated voltage from the current, it has a certain error due the value of the resistors are not exact 2,3 MΩ.

**Image 12** Comparison of measurements between Arduino, Electrometer, Multimeter and Ammeter
Image 12 shows the evolution of the voltage from 25V to 0V. The only small difference seen appears around the values of 25 V. The "calculated values" line is the one that shows difference from the other 3 lines. The reason for this change is that the calculated measurement is measured from the current and multiplied, not using any voltage measurement. In this study, there won’t be a need to measure such elevated values, therefore it can be considered not of importance.

Either on Image 12 and Image 13, the X axes are “experiments”. Basically it go at a certain value and measure with the different devices, not giving important to the real value measured, just looked at the differences between them.

Image 13 is a magnification of the first graph, in the range between 2 and 0 V. There, it can be seen that at low voltages, Arduino presents “bad results”, starting at 2V and being very significant from 0,6V to 0V. Those values are the source values, which are not the same as the voltage applied on the Arduino. To know the ones applied on the input of the Arduino, it needs to be converted with the following formula:
According to this formula, 2V is actually 0,25V and 0,6V is 0,08V. That means that from now on, it should not be considered accurate the values below that point.

### 3.5.2. Experiment changing the reference value of Arduino.

Arduino has 5V as the default max reference value for analog input. That means Arduino divides the 5V into the 1024 bit.

But this is not the only option that Arduino can do. Arduino can be set up at other reference values using the function `analogreference(type)`, depending on the need of each measurement. There are different options:

- **DEFAULT**: the default analog reference of 5 volts
- **INTERNAL**: an built-in reference, equal to 1,1 volts on the ATmega328
- **EXTERNAL**: the voltage applied to the AREF pin (0 to 5V only) is used as the reference.

In this experiment, the value used will be the 1,1 volts. It will be set with the function `analogreference(INTERNAL)`.

The circuit used in this experiment will be exactly the same as what was used before in section 3.4.1.

<table>
<thead>
<tr>
<th>Arduino (V)</th>
<th>Multimeter (V)</th>
<th>Keithley (V)</th>
<th>Keithley (µA)</th>
<th>Voltage calculated by current * 2,3 (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8,43</td>
<td>10,06</td>
<td>10,0137</td>
<td>4,2675</td>
<td>9,81525</td>
</tr>
<tr>
<td>6,25</td>
<td>6,03</td>
<td>6,0078</td>
<td>2,5595</td>
<td>5,88685</td>
</tr>
<tr>
<td>5,2</td>
<td>5,03</td>
<td>5,0072</td>
<td>2,1328</td>
<td>4,90544</td>
</tr>
<tr>
<td>3,102</td>
<td>3,021</td>
<td>3,0057</td>
<td>1,2805</td>
<td>2,94515</td>
</tr>
<tr>
<td>2,053</td>
<td>2,016</td>
<td>2,0053</td>
<td>0,85397</td>
<td>1,964131</td>
</tr>
<tr>
<td>1,529</td>
<td>1,513</td>
<td>1,50515</td>
<td>0,64066</td>
<td>1,473518</td>
</tr>
<tr>
<td>1,001</td>
<td>1,008</td>
<td>1,00254</td>
<td>0,42646</td>
<td>0,980858</td>
</tr>
</tbody>
</table>
Image 14 shows the representation of the same measurements as before. Notice that now the limit for the Arduino input analog is 1,1 V.

In other words, if the limit for the Arduino is 1,1 V at the input pin, through the conversion it will be:

$$V_{max} = 1,1 \cdot \frac{2,3 \, \text{M} \Omega}{300 \, \text{k} \Omega} = 8,43 \, V \quad (3.4)$$

That means that at more than 8,43 volts on the power source, the Arduino will not be able to measure. This is the example of the first value shown on the graph and in the data. The other devices show a value of approximately 10 V, but instead, Arduino shows 8,43 V which
is the limit of the Arduino, the 1,1V. Obviously, this should not be considered in any calculation but should help to understand the idea of the reference value.

Following the idea of the previous graph, image 15 is a magnification of image 14. It goes from approximately 1 to 0 volts.

This graph is very important as it shows the difference in the deviation of the measurement. It follows the expected values until 0,3V where it differs from it. Following the formula shown before, that corresponds to a voltage of 0,039 V.

That change increases the accuracy of those measured, as it was expected, making the Arduino reliable until values of 0,05V as opposite of the 0,08V of the 5V reference.
4. **Description of the circuits**

4.1. **First design**

This section will explain the steps that have been taken to make the final circuit.

The first schematic about the circuit is simple using the idea a shunt ammeter. This means measuring the voltage created by a resistor and knowing the current flowing through it.

![Image 16](image.png) Voltage measuring with only 1 operational amplifier with unity feedback

It only requires an operational amplifier and the shunt resistance. The output of the operational amplifier should show the voltage between the common point of $R_1$ and $R_2$ and the ground, making the current flow though the shunt resistance and not damaging Arduino. Knowing the shunt resistance value, it is easy to know the current.

First, on the left side of the image there is a power source (V1) and a resistor (R1). These are the power source and the isolating system. As a starting value, 100 GB will be taken as the value of the isolating system and 10.000V in the power source. As a reminder, the voltage will be in a range of 5 to 20 kV and the resistance between 100GB and 1TB.

From now on, the isolated system will be also referred as R1.
The current that is being measured is the current that goes through R1. On the end not connected to the voltage source of R1, there will be a resistance (R2) connected and grounded to the floor. The functionality of this resistance is to create a voltage drop so it can be later measured. The value of this resistance has to be big enough so that the voltage created is big enough for the operational amplifier to read it, while at the same time not too big so it doesn’t interfere with the measurement. The proposed value will be 500 kΩ.

The voltage obtained at the output is around 50 mV, which is too small for the Arduino to measure properly.

For that reason, the amplifier needs a gain, which will be created using 2 resistors. The ideal values for Arduino should be an output between 0.8 and 5 V, as that is the range values that Arduino can read with more accuracy.

The operational amplifier will be set in the non-inverting configuration.

\[ V_{out} = V_{in} \cdot \left(1 + \frac{R_2}{R_1}\right) \quad (4.1) \]

A good value for Arduino to be measured would be around 3 V at the output. Therefore, a value of 50 mV is assumed as \( V_{in} \), the relation between R2 and R1 have to be of 59. Then, one option could be 100Ω for R1 and 5900Ω for R2.

Finally, the formula to know the value of the current thought the voltage will be:
Along with the design, it is important to know what situations could be expected, and know if those values will meet the specifications. In other words, we have to make sure that the Arduino doesn’t get over tension and, at the same time, the Arduino will be able to read all the values.

For that reason, there is a theoretical approximation of the expected output depending on the value that the isolating system takes.

R1 is the Isolating system’s resistance and the “Simulated Output” is the voltage expected at the output of the operational amplifier. R2 will have a constant value of 500kΩ and R3/R4 will be 59. The voltage source is 10kV.
Table 6 Variation of the output depending on the resistance value

<table>
<thead>
<tr>
<th>R1 (Ω)</th>
<th>Calculated Output (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0E+11</td>
<td>2.999</td>
</tr>
<tr>
<td>1.1E+11</td>
<td>2.727</td>
</tr>
<tr>
<td>1.2E+11</td>
<td>2.499</td>
</tr>
<tr>
<td>1.3E+11</td>
<td>2.307</td>
</tr>
<tr>
<td>1.4E+11</td>
<td>2.142</td>
</tr>
<tr>
<td>1.5E+11</td>
<td>1.999</td>
</tr>
<tr>
<td>1.6E+11</td>
<td>1.874</td>
</tr>
<tr>
<td>1.7E+11</td>
<td>1.764</td>
</tr>
<tr>
<td>1.8E+11</td>
<td>1.666</td>
</tr>
<tr>
<td>1.9E+11</td>
<td>1.578</td>
</tr>
<tr>
<td>2.0E+11</td>
<td>1.499</td>
</tr>
<tr>
<td>2.1E+11</td>
<td>1.428</td>
</tr>
<tr>
<td>2.2E+11</td>
<td>1.363</td>
</tr>
<tr>
<td>2.3E+11</td>
<td>1.304</td>
</tr>
<tr>
<td>2.4E+11</td>
<td>1.249</td>
</tr>
<tr>
<td>2.5E+11</td>
<td>1.199</td>
</tr>
<tr>
<td>2.6E+11</td>
<td>1.153</td>
</tr>
<tr>
<td>2.7E+11</td>
<td>1.111</td>
</tr>
<tr>
<td>2.8E+11</td>
<td>1.071</td>
</tr>
</tbody>
</table>

The table above shows an increase of R1 from 100 GΩ to 280GΩ. What maintains constant is the relation R3/R4, the value of R2, and the voltage source. The result is the value that the output of the operation amplifier will have at every situation.

As a result of this table, it proves that even as the resistance value changes, the whole range of numbers is completely functional for the Arduino (between 1 and 5 Volts).

The formula used to calculate it is:

\[
V_{output} = \frac{R2 \cdot V_{source}}{R1 + R2} \cdot \frac{R3}{R4 + 1}
\]  
\( (4.3) \)
4.1.2. **Operational amplifiers**

In the design 2 chips will be used. The LTC1050 and LT1013. The LT1013 will be used to create the virtual ground and the power supply. The LTC1050 will be used to amplifying and acquiring the main measuring signal.

4.1.2.1. **LTC1050**

The LTC is a precision zero-drift operational amplifier with internal capacitors. The main features of it, important in this thesis are:

<table>
<thead>
<tr>
<th>Specifications of the LTC1050 [6]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input offset voltage</td>
</tr>
<tr>
<td>Input offset current</td>
</tr>
<tr>
<td>Input bias current</td>
</tr>
<tr>
<td>Slew Rate</td>
</tr>
<tr>
<td>Total Supply voltage</td>
</tr>
</tbody>
</table>

The operational amplifier is mounted on a SO-8 package.

Outside of the specifications of the chip, it is important to see the actual response of the operational amplifier.

- **Response of the operational amplifier LTC1050**

It is important to know the performance of the operational amplifier beside the datasheet specifications. In the following graphs, it can be seen the response of the operational amplifier when the input is a step signal at different frequencies. It is important to check the
frequency response of the amplifier, thus it could affect the results and is the most important part of the circuit.

**Image 19** Response at 30 kHz

**Image 20** Response at 160 kHz
From these graphs, the different responses at every frequency can be seen. The response is the same in all cases, with the same space between peaks and rising time. The problem occurs when the frequency is higher than 425 kHz, in which case the step is too short for the
signal to reach the final value. In that case, as shown in Image 34 the output can’t follow the input signal and turns into a signal not dependent of the input.

For our set-up, we never get to reach those frequencies at any moment, so the response of the amplifier is not a variable likely to give errors on the results.

4.1.2.2. LT1013

The LT1013 is a dual precision operational amplifier, but only one of them will be used in the circuit. The main features are:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input offset voltage</td>
<td>60 µV</td>
</tr>
<tr>
<td>Input offset current</td>
<td>0,2 nA</td>
</tr>
<tr>
<td>Input bias current</td>
<td>15 nA</td>
</tr>
<tr>
<td>Slew rate</td>
<td>0,4 V/µS</td>
</tr>
<tr>
<td>Total supply voltage</td>
<td>±22 V</td>
</tr>
<tr>
<td>Drift</td>
<td>2 µV/°C</td>
</tr>
</tbody>
</table>

It also comes in a SO-8 configuration with the following distribution:
Response of the operational amplifier LT1013

![Image 24](image24.png) Response at 30 kHz

![Image 25](image25.png) Response at 150 kHz

In the case of this operational amplifier, it reaches the value perfectly at 30 kHz but, as seen on image 24, at a range of 150kHz it reaches its limit not being able to reach the expected value for higher frequencies. It is not really important because this chip is connected to the
voltage source, which has really small variations and the output is not required to be very exact.

As well as the LTC1050, the response of the amplifier doesn’t affect the system and can be considered fast enough at all time.

### 4.1.3. Power supply

The power supply of the device depends only on the operational amplifiers used. There is the option of using single batteries, as low as 3 volts, or using an external power source of any value. In this study, a power supply will be used connected to the electrical line, 230 V (Europe), and transformed to 12V outside the board. In this thesis, the one used will be a power source with a floating voltage of 12V.

![Image 26 Schematic of the voltage supply circuit](image)

This is the schematic of the supply circuit. It starts with the external floating supply of 12 volts. The aim of this part is to provide, on one part, a virtual ground, and on the other part, a power supply for the main operational amplifier, as a positive and negative supply.

The operational amplifier is configured as a follower, with a feedback between the output and the inverted input.
The pins represented on image 26 are:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 1</td>
<td>Output</td>
</tr>
<tr>
<td>Pin 2</td>
<td>Inverting input</td>
</tr>
<tr>
<td>Pin 3</td>
<td>Non-inverting input</td>
</tr>
<tr>
<td>Pin 4</td>
<td>Negative supply input</td>
</tr>
<tr>
<td>Pin 8</td>
<td>Positive supply input</td>
</tr>
</tbody>
</table>

Remember that the transfer function of an operational amplifier as:

\[
V_{OUT} = A \cdot (V_{IN+} - V_{IN-})
\]

(4.3)

If: \(V_{IN-} = V_{OUT}\)

(4.4)

Then: \(V_{IN+} = V_{IN-} = 0\)

(4.5)

On the other hand, both resistances have to guarantee that the voltage drop over one is the same as the voltage drop over the other, as they both have the same value.

If the center of both resistances is connected to \(V_{IN+}\) and it has to be 0, and the voltage drop on each resistance has to be the same, this means that the initial 12 volts will be +6 and -6.

By this method, now the board is supplied with a constant \(V_+ = 6V\), \(V_- = -6V\) and a virtual ground \(VGND\) of 0V.

It has also included a capacitor to stabilize and a resistor of 100\(\Omega\) at the output as a protection.

### 4.2 Modified designs

From the idea of the first design, diverse updates can be done to make the circuit more complex and adapt better to the difference expectations. Each update includes one or different new parts to add functionality to the first ideal design.
4.2.1. Update 1: Three ranges and multimeter option

The problem with the first design was that with only 1 resistance, the range of values was very specific.

In the first design, the shunt resistor was of 500kΩ, a voltage applied of 10.000V and 100GΩ of starting resistance of the insulator capacity and an amplification of 60 times to get the results with an output of 3V for the Arduino. That means the ideal current flowing was 100nA.

Table 9 shows with the standard parameters, what would be the output depending on the input.

<table>
<thead>
<tr>
<th>Input current (nA)</th>
<th>Output (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0,3</td>
</tr>
<tr>
<td>15</td>
<td>0,45</td>
</tr>
<tr>
<td>20</td>
<td>0,6</td>
</tr>
<tr>
<td>25</td>
<td>0,75</td>
</tr>
<tr>
<td>30</td>
<td>0,9</td>
</tr>
<tr>
<td>50</td>
<td>1,5</td>
</tr>
<tr>
<td>70</td>
<td>2,1</td>
</tr>
<tr>
<td>90</td>
<td>2,7</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>125</td>
<td>3,75</td>
</tr>
<tr>
<td>150</td>
<td>4,5</td>
</tr>
<tr>
<td>166</td>
<td>5</td>
</tr>
<tr>
<td>175</td>
<td>5,25</td>
</tr>
</tbody>
</table>

The device would actually be working on a range between 20nA and 166nA approximately. Obviously that is a huge limitation that doesn’t match with the expected range of values.

So in the first update, the idea is to have 3 different resistances that can focus in 3 measuring ranges: micro amps, nano amps and pico amps. The transition between the resistors will be made through a rotatory switch.
The basic idea of the circuit is the following:

![Image 27 Representation of the idea of the 3 switchable resistances.](image)

Based on Ohm law, $V = I \cdot R$ the voltage drops created will be the following:

<table>
<thead>
<tr>
<th>Resistance</th>
<th>Current</th>
<th>Voltage drop created</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Ω</td>
<td>1 µA</td>
<td>0,01 mV</td>
</tr>
<tr>
<td></td>
<td>1000 µA</td>
<td>10 mV</td>
</tr>
<tr>
<td>10 kΩ</td>
<td>1 nA</td>
<td>0,01 mV</td>
</tr>
<tr>
<td></td>
<td>1000 nA</td>
<td>10 mV</td>
</tr>
<tr>
<td>10 MΩ</td>
<td>1 pA</td>
<td>0,01 mV</td>
</tr>
<tr>
<td></td>
<td>1000 pA</td>
<td>10 mV</td>
</tr>
</tbody>
</table>

The reason for using these values is to allow, by just switching the resistance, to use all of the other components in the board without having to change any other part. This is very practical, especially at the moment of optimizing the amount of components on the board.

At the same time, the amplification of the resistors will be change to 100x instead of 60x. The reason to do that is to adapt to the 10, 10K and 10M, and their voltage drop created, 0,01mV to 10mV, multiply it 100 times to ranges of 1mV to 1000mV, where can be used a voltmeter on their if wanted. The voltmeter will read from 1 mV to 1000 mV for 1 µ/n/p A to 1000 µ/n/p A.
4.2.1.1. **Resistor and switch**

The switch is a very important part of the circuit. It is responsible for selecting what range will be used at any moment and connects to the operational amplifier. Many different switches can be used, but the one used at least needs to have 3 poles and 3 ways. Standard rotatory switches usually come as 4 poles 3 ways or 3 ways 4 poles.

The function is the same: At the same time, each pole will be connected to one of the ways. For example, in the 3 Pole 4 Way switch, Pole A can be only connected to way 1, 2, 3 or 4 as well as Pole B can be only connected to 5, 6, 7 or 8 at a time. As it rotates, all the connections will rotate at the same time in 30\(^\circ\), for example, the configurations A-1, B-5, C-9 or A-3, B-7, C-11.

The resistors used will be all 0.1%, so they are as accurate as possible. The switch will be the CK1027 with 4 pole and 3 way, even 1 pole will not be used.
Among different configurations, this picture shows how the connections are made. PAD1 and PAD2 are the jacks where the current will flow, input and output. PAD5 and PAD6 are the connections of the external power supply. This switch has 4 poles and 3 ways.

The switch is situated in positions: D1-C1-B1-A1. The current will flow through the 10MΩ resistance, making a voltage drop, which will be connected through D1 and C1 to the resistance of 100Ω used as a protection before the non-inverting input of the operational amplifier. If the values measured were micro or nanoamps, the same procedure would exist with D1-C1 and D2-C2.

At the same time, on pole A, the LED system will be connected. It will indicate what mode is selected at any time by illuminating one of the three LEDs. The LEDs will be connected between +V and –V of the power supply though a resistance of 2kΩ to limit the current.

4.2.1.2. **Adapting the measuring circuit and functionality for a multimeter.**

This part of the circuit is where the value is measured and amplified to millivolt output range. It consists of an operational amplifier with a feedback.
It is intended to have an amplification of 100 times. That means:

\[
\text{If } V_{\text{OUT}} = V_{\text{IN}} \cdot (1 + \frac{R_1}{R_2});
\]

\[
1 + \frac{R_1}{R_2} = 100;
\]

\[
\frac{R_1}{R_2} = 99
\]

Within an infinite amount of combinations, one solution could be:

\[
R_1 = 99\,k\Omega \quad R_2 = 1\,k\Omega
\]

As a resistor of 99kΩ is difficult to find, the conversion will be made with 2 resistors in series of 24kΩ and 75kΩ.

With this amplification, the values on the output will have a range from 1mV to 1000mV.

![Schematic of the measuring part](Image 30)
The voltage drop created over the resistors comes through $R_2$ (100Ω) into the non-inverting pin. That pin has the feedback with the resistors and is also connected to the virtual ground through the 1kΩ.

The $+V$ and $-V$ come from the power supply and 2 capacitors of 100nF are added on both positive and negative inputs and the ground for stabilization. The output is protected with a 100Ω resistor.

Between the virtual ground and the end of the resistor is where the voltage will be measured with a multimeter or connected to another amplifier to increase the amplifier to make it suitable for the Arduino or other measuring device.

The resistors used are all 0,1%. That is because there is a need to have the most accurate value at any time. If instead of 75k, the resistor was 76k, the final result wouldn’t be the same, as the whole board has many steps and resistances, the error created would sum up, making a considerable error at the end. For that reason, the most exact the resistances are, the better will be to have the expected value at the end.
Image 31 Second design board, front and back

Image 32 Complete schematic of second design
4.2.2. **Update 2: Use of another Amplifier**

At the output of the previous device, the values will be shown in mV, from 0 to 1000 corresponding to each of the selected values before, µA, nA or pA. That is appropriate if it is going to be read on a standard multimeter of 3,5 digits. Any other device able to read millivolts is also appropriate for the measuring, always being aware of the accuracy and resolution of it.

In this project, there is not just the intention to read the values, but also to log them. For that reason, Arduino will be used to read the values and transfer them through the serial port to a computer or a suitable recording device. As seen before Arduino doesn’t have a high accuracy at low values, between the 0 at 500 mV approximately. That is obviously a burden and will not record the real values. From 500 mV to 5V it will be considered as an ideal function, even though additional corrections could be made later.

Because of this, the input voltage at the Arduino has to previously be amplified. What is looked for with the amplification is that instead of going from 0 to 1 volt on the output, it goes from 0 to 5. That requires an amplification of 5 times. With it, the amount of values under the 500mV will be reduced, making the range of readable values bigger.

For example: Without amplification only 50% (500 mV of the 1000mV) were able to be read properly (over 500mV). That means a 50% of acceptable range. With the amplification, the output of 500mV will correspond to 100mV at the input ($500 = 100 \cdot 5$).

That means only 100 mV out of the 1000 will be not be in the acceptable range. That corresponds to the 90% of the values.

So it is clear that by using this amplification, it can be measured up to the 90% of the values in the range.
Image 33 shows how the output point of 500mV is at the 50% without amplification and at 10% and 12% with 4 and 5 times amplification.

But still, there are 10% of the values that will be lost in each range. This is a problem that has to be solved. One way to solve that problem is to overlap measuring values.

Another solution is to add other resistors between the 10MΩ and the 10KΩ, as could be 1MΩ or 100KΩ. Also, instead of amplifying 5 times, if amplified 4 times, the range of values overlaps within the different resistors.

The following table and graph shows the ranges in pico amperes that will be able to be read by each resistor. For example, 1MΩ should ideally to read from 10 pA to 12500 pA (12,5 nA).

<table>
<thead>
<tr>
<th>Resistor</th>
<th>Ideal Min. Value</th>
<th>Real Min. Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>10MΩ</td>
<td>1 pA</td>
<td>10 pA</td>
<td>1250 pA (1,2 nA)</td>
</tr>
<tr>
<td>1MΩ</td>
<td>10 pA</td>
<td>100 pA</td>
<td>12500 pA (12,5 nA)</td>
</tr>
<tr>
<td>10kΩ</td>
<td>1nA</td>
<td>10 nA</td>
<td>1250 nA</td>
</tr>
</tbody>
</table>
Image 34 represents the values of “table 6”. The blue part represents the values that cannot be read by Arduino. The red range is the range that can actually be properly read.

For example, in the new configuration, if it is desired to measure 2 nA, ideally it should be measured over the 10kΩ resistance, creating 0.02mV of voltage drop, then amplified 100 times to 2mV. With the next amplification of 4 times, it would go up to 8mV. That value is still in the 10% of values not able to be read by Arduino.

If it is measured over the 1MΩ resistor, it creates 2mV of voltage drop. Then amplified 100 times to 200mV. To finish, 4 times amplified to a final value of 800mV, where it can be completely read by Arduino.

The method to amplify will be to use an operation amplifier, selecting the appropriate values for the resistances.
It’s important to remember how this circuit works. The transfer function will be:

\[ V_{OUT} = V_{IN} \cdot \left( \frac{R_2}{R_1} + 1 \right) \quad (4.9) \]

In that case, the relation between resistors has to be 4, independent of the value of the resistances.

The operational amplifier used for the amplification will be the same one used previously, the LTC1050 with low offset and zero drift. A different one could be used as long as the pin connections were the same once the board is made.
In this figure, it a 3kΩ and 1kΩ has been used. The power supply is the same for the whole board and it is connected through pins 7 and 4. There is a resistance of 100Ω at the entrance of the op-amp for protection. Pad 7 and 8 are where the Arduino will be connected. Pad 8 is just the ground, so it can be situated anywhere on the board.

4.2.2.1. **Correction of the amplifier**

Even with the best of the amplifiers being used, there will be some error within it. Also it is important to remember the accuracy of the Arduino reading and the change of an offset.

For that reason, before starting the measurements, it is important to check the values transferred.

For this experiment, the same output value will be measured with the Arduino and the Keithley electrometer for a range between 0 and 5 volts.
Table 13 shows the values of the Arduino, the electrometer, the difference between both of them (K-A) in Volts and the relative variation between them as \((K-A)/K*100\).

It is clear that the Arduino doesn’t give the same values as the electrometer (ideal ones) throughout. As said previously, the values near to 0 are not readable for Arduino, making an error or around 10% until 0.5 Volts. On the other side, at high values, Arduino reads slightly higher values than the reals ones, inverting the previous situation.

Avoiding the first 2 measures, the 0.01 and 0.02 V, the following graphs show the results:
Image 48 shows how the difference between Arduino and the electrometer measures, \((K-A)\), in absolute values in volts. It reaches the common point at 2 Volts. It has also the regression line, and its equation.

Image 39 shows the relative error, as \(\frac{K-A}{K} \cdot 100\). It is very clear that at the small values, the error is much bigger.
The solution to solve this problem is correcting the values. The mechanism will be to add the difference value according to the regression line showed at image 34. This way, the resulting value will be the first one plus the difference of what it should be.

The regression formula is:

\[ y = -0.0171x + 0.0357 \]  \hspace{1cm} (4.10)

with \( x \) for the measured values and \( y \) showing the difference between the electrometer and Arduino.

Therefore, if the correcting formula calculates the difference to the ideal value, to get the corrected one the formula is:

\[
Corrected\ value = Measured\ value + (-0.0171 \cdot Measured\ value + 0.0357)
\]  \hspace{1cm} (4.11)

Once applied the correction, the difference between the ideal value and the value measured with the Arduino is reduced to almost zero.

\[ K-(A+y) \]

\begin{center}
\includegraphics[width=0.8\textwidth]{image40.png}
\end{center}

Image 40 Difference between Electrometer and corrected values\(^1\)

\(^1\) Values attached at appendix A.b
This graph shows the difference between the ideal values and the corrected ones. Notice that the values reach a maximum difference of 0.0027 Volts to the ideal one instead of 0.035 without the correction. The regression line in this case is almost \( Y=0 \), which means that there is no further adjustment possible.

At the percentage image, there is clearly one value higher than the rest, with a 2.5% difference but the rest stay within 0 and 0.5%. It is good to notice that the peak value corresponds to 100mV, said in another way, in the range of values that were considered not appropriate for Arduino. It reduces from a 32% error to a 2.5%.

This correction depends on the resistance being used, that means that for each range and each isolating system it will have to be adjusted for its values.
4.2.3. **Update 3: Adapting for negative values**

Arduino can only read positive values. That makes it an inconvenience and for that reason it would be suitable to have some kind of additional circuit to convert the negative values to positive ones.

The best solution is to have an operational amplifier in the inverting configuration.

![Image 42 Operational amplifier in inverting configuration](Image)

The intention of using this configuration is to have the following statement:

\[ V_{OUT} = -V_{IN} \]  \hspace{1cm} (4.12)

For this purpose, \( R_{in} \) must be equal to \( R_f \). The current will flow though both resistances from \( V_{IN} \) to \( V_{OUT} \). As the non-inverting pin is connected to ground the inverting one will try to have 0 V. If both resistors are equal, it means that the voltage drop on both will be the same, with 0V as the middle point. Therefore, if the \( V_{in} = -1 \), \( V_{out} = 1 \).

The operational amplifier should also have a low input bias, so the values are as exact as possible and the lowest current possible is lost in the inverting process.

The following step is to design how will be how to implement the circuit that will let to choose between positive and negative values. The circuit will consist of a switch of 2 positions, one for positive currents and one for negative. If the chosen one is the positive,
the output will go directly to the amplifier. If the negative one is chosen, the output will go to the inverter circuit and then to the amplifier.

In addition to that, there will also be 2 LEDs added to indicate in which situation the switch is positioned. A green LED will indicate positive values. Red LED will indicate negative values. It will also be commanded by the same switch.

The LED is connected with a 2kΩ resistor between the +V and –V of the power supply.
Image 45 Image of the third design front and back

Image 46 Schematic of the third design with parts labelled

Image 46 shows the schematic of the design. It includes the main parts such as the resistor and LED switch, the voltage supply and ground generator, the Inverter Op-Amp and LEDs with the other switch and the second amplifier with the output for the Arduino.
4.2.4. **Update 4: Design to fit into a shielding box**

The fourth update is the last one made. Even though update 3 was working properly, this one is designed with the idea of putting the board inside of a metal box in order to protect the components and also for isolating.

It was also made with a ground layout on the top and the bottom of the board, as a shielding protection.

It also includes one capacitor for each positive and negative supply pin of each operational amplifier, situated as closely as possible to the pins.

- **Electrostatic Interference and Shielding**

Electrostatic coupling appears when an electrically charged object approaches the input circuit under test. This happens at high resistance materials because they don’t allow the charge to dissipate quickly, and might result in bad measurements. It can be an AC or DC electrostatic field, so shielding will help minimize the effects of those fields.

In the case of DC fields, effects can be detected when there is movement around the experiment, for example a person, and the results will fluctuate. If the simple fact of moving around the device creates the measures to change considerably, there is not enough shielding.

AD fields are caused by power lines and RF fields. The way to know if they exist is looking at the signal through an oscilloscope. They can be created by transmitters, contactors or even cellular phones.

The shielding can be built to enclose the circuit by making metal box surrounding the device. It should be enough to protect against most electrostatic AC interferences.

The shield has to be connected to the low point, in this case the ground. The cabling between the HI terminal of the meter and the device under test should also be shielded, especially when measuring under 1 nA.
The main difference between shielding and guarding is that shielding implies the use of a metallic enclosure to prevent electrostatic interference. Guarding implies use of an added low impedance conductor at the same potential as the high impedance with the purpose to intercept any voltage or current.

4.2.4.1. **Comparison between grounded board with shielding and without shielding**

In the fourth update, it was important to take care of the shielding as well as the leakage currents inside the board, due to the parasite capacitances and junctions between them.

First of all, the board was design making both top and bottom of the board to be at a ground potential.
To see how the new box and printing system works, the same measure will be taken with the board without (update 3) and this board (update 4). Additionally, it will also be measured with the electrometer.

What will be measured is 5 volts over a resistance of 16GB, and making no movement around the board while measuring.
In image 50 it can be seen how it affects the internal leakage current. Update 3 has a loose of 15pA on average respect the board number 4. Electrometer is considered the ideal one and it has 10 pA over the board with shielding. Also important to consider that to measure with the electrometer, a low noise cable has been used and with the board, standard cables are the ones being used.

In this case, the same experiment was made, with the same variables, but this time, there was movement around the measures. The movement consisted walking in and out around it and move a hand without touching anything. The measures were taken one after another, but trying to repeat the same movements in both.

What can be seen is that with the shield, the noises because of the interference and external fields are highly reduced, proving the convenient and adequate use of a shield.

[Image 51] Experiment to prove the influence of the shield while movement around
5. **Eagle design**

Eagle is the software used to implement the design of the layout. The first step is to draw the full schematic, with all the pieces from the components library. It is important to choose the correct components, because they will define the size of the components and the space between its pins.

![Image 52 Schematic of the final design on eagle software](image.png)
Image 52 shows the final schematic of the circuit. Each single part has been described previously separately.

This picture also shows how the connections between the board and the outside are made.

PAD 1 and 2 are the input and output of the current desired to be measured. Pad 3 and 4 are jack connections for a multimeter. In case there is no option of having a computer and Arduino available in that moment, these connections will give the results without amplification, inverted or not inverted so they can be read in a multimeter in mV. 1 µA, nA or pA will be equivalent to 1mV.

The power circuit has 4 input pads. X1-1 and X1-2 are the connections for the supply converter used in this experiment. In addition, there are PAD5 and PAD6 as banana jack connections, in case there is the need to use a different voltage supply.

X2-1 and X2-2 will be the connections to the Arduino. They are simple cable connections and also fit in the Arduino slots.

Once the schematic is completed, all the components need to be set up on the board. Initially, the components are lined by lines, showing which parts are connected with the other parts.
Once the pieces are placed in the desired locations the connections between points can be changed, making the shortest possible distances.

**Image 54** Circuit before optimizing the connections and positions

**Image 55** Circuit after optimizing the connections and position
Once the connecting points are the optimal ones, the function “AutoRoute” defines the circuit lines on the board. The color red means on top of the board and blue is on the bottom.

At this point, it is advisable to manually modify the lines a little bit, making it easier to weld and mount it. For example, leaving some space between parallel lines or the connections to the chips chassis not too close.

Image 56 shows exactly how the components are placed in the layout. The size of the layout is 100x80mm.
The final thing to be done is adding a layout with the ground on top and bottom of the board. That is a way to create some guarding over the board. Basically consist in making the part of the board which has no lines to be the ground.

Image 57 Forth design with the ground layover on top and bottom

Image 58 Complete Schematic diagram
6. Components and costs estimation

The components used in all of the boards can be easily bought through the internet at an affordable prize. The following table shows the components needed for the last design made and their prices.

<table>
<thead>
<tr>
<th>Component</th>
<th>Amount</th>
<th>Unity price €</th>
<th>Total price €</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply 230/12 V</td>
<td>1</td>
<td>8,83</td>
<td>8,83</td>
</tr>
<tr>
<td>10 MΩ 1%</td>
<td>1</td>
<td>0,06</td>
<td>0,06</td>
</tr>
<tr>
<td>1 MΩ 0,1%</td>
<td>1</td>
<td>0,24</td>
<td>0,24</td>
</tr>
<tr>
<td>10 kΩ 0,1%</td>
<td>3</td>
<td>0,98</td>
<td>2,94</td>
</tr>
<tr>
<td>10 Ω 0,1%</td>
<td>1</td>
<td>5,18</td>
<td>5,18</td>
</tr>
<tr>
<td>1 kΩ 0,1%</td>
<td>2</td>
<td>0,68</td>
<td>1,36</td>
</tr>
<tr>
<td>2 kΩ 1%</td>
<td>2</td>
<td>0,22</td>
<td>0,44</td>
</tr>
<tr>
<td>3 kΩ 0,1%</td>
<td>1</td>
<td>0,94</td>
<td>0,94</td>
</tr>
<tr>
<td>24 kΩ 0,1%</td>
<td>1</td>
<td>0,71</td>
<td>0,71</td>
</tr>
<tr>
<td>75 kΩ 0,1%</td>
<td>1</td>
<td>0,28</td>
<td>0,28</td>
</tr>
<tr>
<td>100 Ω 0,1%</td>
<td>5</td>
<td>0,86</td>
<td>4,30</td>
</tr>
<tr>
<td>200 kΩ 1%</td>
<td>2</td>
<td>0,04</td>
<td>0,08</td>
</tr>
<tr>
<td>100nF Capacitor</td>
<td>9</td>
<td>0,20</td>
<td>1,8</td>
</tr>
<tr>
<td>220nF Capacitor</td>
<td>1</td>
<td>0,53</td>
<td>0,53</td>
</tr>
<tr>
<td>LT1013</td>
<td>1</td>
<td>1,84</td>
<td>1,84</td>
</tr>
<tr>
<td>LTC1050</td>
<td>3</td>
<td>5,15</td>
<td>15,45</td>
</tr>
<tr>
<td>Green LED</td>
<td>4</td>
<td>0,15</td>
<td>0,6</td>
</tr>
<tr>
<td>Red LED</td>
<td>1</td>
<td>0,09</td>
<td>0,09</td>
</tr>
<tr>
<td>Rotary Switch CK-1027</td>
<td>1</td>
<td>1,86</td>
<td>1,86</td>
</tr>
<tr>
<td>2 Positions switch APEM - 5646A9</td>
<td>2</td>
<td>6,39</td>
<td>12,78</td>
</tr>
<tr>
<td>Xip chassis</td>
<td>4</td>
<td>0,53</td>
<td>2,12</td>
</tr>
<tr>
<td>Screw Connector</td>
<td>2</td>
<td>0,7</td>
<td>1,4</td>
</tr>
<tr>
<td>4mm Red Banana Jack</td>
<td>3</td>
<td>2,76</td>
<td>8,28</td>
</tr>
<tr>
<td>4mm Black Banana Jack</td>
<td>3</td>
<td>2,76</td>
<td>8,28</td>
</tr>
<tr>
<td>Metal box</td>
<td>1</td>
<td>11,71</td>
<td>11,71</td>
</tr>
<tr>
<td>Board manufacturing</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>56</strong></td>
<td></td>
<td><strong>92,07 €</strong></td>
</tr>
</tbody>
</table>
7. Resolution

It is important to know the resolution of the whole measurement device, because it will determine how precise can the ammeter measure.

First of all, the resolution will be determined by the Arduino board, specially the ADC converter inside the chip ATMega328. It has a resolution of 10 bits.

\[ \text{10 bits resolution} = 2^{10} \text{ bits} = 1024 \text{ bits} \]

It is also known that the Arduino goes from 0 to 5 volts (5,02 on this board). That means that it can be calculated what the smallest change on the input in volts is needed to increase 1 bit of the converter:

\[ 1 \text{ bit} = \frac{5.02 \text{ Volts}}{1024 \text{ bit}} = 0.004902 \text{ V} = 4.902 \text{ mV} \]

So for every 4.902mV increased at the input of the Arduino board, the output will show one bit more.

Then it is important to remember that the hardware device has 2 amplifiers, one with 4 times amplification and one with 100 times amplification. It can be known the voltage previous the amplifications that result in the 4,902mV:

\[ \frac{4,902mV}{4} = 1,2255 \rightarrow \frac{1,2255}{100} = 0,012255 \text{ mV} \]

So a voltage variation on the shunt resistance of 0,012255 mV will lead to a change of bit at the output of the Arduino board.

To look exactly what current will make that change on each resistance, this voltage has to be divided by each of the shunt resistances.

\[ 10\text{M}\Omega \rightarrow I = \frac{12,255 \cdot 10^{-6} \text{ mV}}{1 \cdot 10^6 \Omega} = 1,2255 \cdot 10^{-12} \text{ A} = 1,2 \text{ pA} \]

\[ 1\text{M}\Omega \rightarrow I = \frac{12,255 \cdot 10^{-6} \text{ mV}}{1 \cdot 10^6 \Omega} = 12,255 \cdot 10^{-12} \text{ A} = 12,2 \text{ pA} \]

\[ 10 \text{ k}\Omega \rightarrow I = \frac{12,255 \cdot 10^{-6} \text{ mV}}{10 \cdot 10^3 \Omega} = 1,2255 \cdot 10^{-9} \text{ A} = 1,2 \text{ nA} \]
8. **Accuracy of the results with constant resistances**

Measure the accuracy on constant value is relatively easy, because the time is not a variable, therefore what matters is the constant final value and what it should be compared to the ideal one.

The most critical range is the pico range, with the shunt resistor of 10MΩ. It is the one where there will be need more correction

To reproduce this experiments, a standard resistance of 16GΩ (16,82Ω) will be used, and the voltage applied to it is the one that will be changed, reaching values from 0 to 1250 pico amps.

Image 58 shows the values of the Arduino are the ones directly obtained from it, without applying any correction using the 10MΩ shunt resistance (pico range).

To these values it is needed to apply the correction formula, which will take out the error made by the device, as explained in chapter 4.2.2.1. The formula for the pico range is:

\[
Corrected\ value = Measured\ value + (-0.0091 \cdot Measured\ value + 14,906)
\]
Table 15 shows the numeric values for image 58 and 59. It shows the average values with and without correction and what should be the ideal value. It also includes the difference between the ideal and the measured values and the ideal and corrected values.

It is important to see that as said before, the Arduino doesn’t work properly for low values. Once the correction is applied, those values don’t correspond with the ideal ones, because the correction is optimized for the values that the Arduino can actually read. For example, if the measure is 0 pA, the correction shows 14,906. Therefore, those values should be considered not appropriate though can be estimated.
Values for 297.27 pA and 594.53 pA

In image 60 and 61 it is shown the same as image 59, in this case for 297.27 and 594.53 pA. It compares before and after the correction. Image 61 shows it for 1189.06 pA.

Values for 1189.06 pA
Table 16 shows the values for 297.27, 594.53 and 1189.06 pA. It shows the measured values, the corrected ones and the ideal ones. It also shows the difference between the ideal and real values and ideal and corrected values.

Now it is easy to see how the correction really works and deletes the error created by the device. Image 62 proves how the difference of values between measured and corrected compared to the ideal ones changes. In the corrected ones, the difference is reduced significantly.

For the nano range, with the shunt resistor of 10kΩ, another correcting formula will be used. It is:

\[
Corrected\ value = Measured\ value + (-0.007 \times Measured\ value + 5.75)
\]

In this case the experiments are done with a variable voltage over 10kΩ resistance, measuring values from 27nA to 1103nA.
In this case, the error made is not that big, also because there is no need to go for really small values, as the small ones will be covered by the 1MΩ resistor. Therefore, it can be calculated from 30 nA, which is big enough.

Still the values are similar, a little correction adjust the values, decreasing the difference between Ideal ones and measured.

**Table 17 Values for the nano-range with 10k resistor**

<table>
<thead>
<tr>
<th>Ideal value (nA)</th>
<th>Average without correction (nA)</th>
<th>Average with correction (nA)</th>
<th>Ideal-real (nA)</th>
<th>Ideal -corrected (nA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27.57</td>
<td>22.36</td>
<td>27.95</td>
<td>5.209</td>
<td>-0.384</td>
</tr>
<tr>
<td>64.34</td>
<td>59.07</td>
<td>64.41</td>
<td>5.265</td>
<td>-0.071</td>
</tr>
<tr>
<td>182.82</td>
<td>179.18</td>
<td>183.68</td>
<td>3.637</td>
<td>-0.859</td>
</tr>
<tr>
<td>459.59</td>
<td>456.84</td>
<td>459.39</td>
<td>2.751</td>
<td>0.199</td>
</tr>
<tr>
<td>735.31</td>
<td>734.41</td>
<td>735.01</td>
<td>0.904</td>
<td>0.295</td>
</tr>
<tr>
<td>919.12</td>
<td>920.2</td>
<td>919.51</td>
<td>-1.083</td>
<td>-0.391</td>
</tr>
<tr>
<td>1102.94</td>
<td>1105.42</td>
<td>1103.43</td>
<td>-2.478</td>
<td>-0.490</td>
</tr>
</tbody>
</table>
Image 64 shows how the error is deleted with the correction, from a range of 5 nA to slightly less than 1 nA error.

The range of 1MOhm is supposed to covert he space between the 2 previous ones, so between 1 and 10 nA. That range is very small and centered in the possible ranges, so it basically doesn’t need correction.

<table>
<thead>
<tr>
<th>Measured (nA)</th>
<th>1.06</th>
<th>1.96</th>
<th>3.17</th>
<th>5.78</th>
<th>5.25</th>
<th>9.54</th>
<th>7.36</th>
<th>10.54</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ideal (nA)</td>
<td>1.09</td>
<td>1.93</td>
<td>3.17</td>
<td>5.8</td>
<td>5.29</td>
<td>9.66</td>
<td>7.41</td>
<td>10.58</td>
</tr>
<tr>
<td>Difference (nA)</td>
<td>-0.028</td>
<td>0.031</td>
<td>0.000</td>
<td>-0.022</td>
<td>-0.044</td>
<td>-0.116</td>
<td>-0.046</td>
<td>-0.044</td>
</tr>
</tbody>
</table>

8.1. **Positive-Negative Error**

Another important part to look at the accuracy is to look at the response when the board is working with negative values. To look at this, the same current is going to be measured with positive values and negative ones.
Comparison between Positive and Negative measurement for 1V input over 16GΩ

Image 65 Comparison between Positive and Negative measurement for 1V input over 16GΩ.

Comparison between Positive and Negative measurements for 5V input over 16GΩ previous to the correction.

Image 66 Comparison between Positive and Negative measurements for 5V input over 16GΩ previous to the correction.

Image 65 and 66 show the comparison of measuring positive and negative values. On the board that means changing the switches and using another operational amplifier. The
negatives values are always higher than the positives. That can be because the board is using another operational amplifier, it includes more offset to the final value.

The average values are:

<table>
<thead>
<tr>
<th>positive (current pA)</th>
<th>negative (current pA)</th>
<th>difference (pA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1V input</td>
<td>46.86</td>
<td>57.5</td>
</tr>
<tr>
<td>5V input</td>
<td>284.28</td>
<td>293.95</td>
</tr>
</tbody>
</table>

For that reason, it is important to know that there is a difference around 10 pA between reading positive and negative values, therefore, a different correction will be needed when measuring either way. The error between positive and negative doesn’t depend on the current measured. The inverter circuit can add some leakage current, offset and error, as well as reduce the accuracy. All this factors can be seen as the difference between positive and negative comparison.
9. **Accuracy of the results with insulating liquid**

The device is designed to measure the current over the insulating system, which can be oil. The oil goes inside the plate's electrodes and the current measured is the one flowing between both plates.

It is not intended in this thesis to study the behavior of the oil, but it is important to compare the results to another device, in this case the electrometer measures.

For this reason, the experiments made will be with 1kV over a 2mm gap applied between the 2 electrodes.

Image 67 is the results of measuring over the 10MΩ, the pico range. It can be clearly seen how the current value goes over the 1250pA, the reading stops at that value. That is because the 1250pA is equal to the 5V on the input of the Arduino, which is high limit value. Therefore, it will show 1250pA until it the value decreases again.

The values shown are values without and correction, just showing directly as they are read from the Arduino. For correction process, refer to chapter 4.

![Image 67](image67.png)

**Image 67** Measures of 1kV with 2mm gap on the insulating liquid with pico range (10MΩ)
Image 68 is the same experiment, with the same oil and temperature conditions but in this case measured with the electrometer. The electrometer doesn’t have a limit value, therefore in this case it can be seen what value it reaches. It is 10,000 pA (10 nA) before going down as expected.

![Image 68](image68.png)

**Image 68** Full range with Electrometer

Image 69 has the same values as image 68, but in this case the limit of the graph has been set at 1250 pA, so it matches with the experiment made with the shunt resistance. This way, the graphs can be compared easier.

![Image 69](image69.png)

**Image 69** Range limited (1250 pA) with electrometer
Between the electrometer and the shunt resistance system, there is one main difference. It is the falling time between 1250pA and the end. Electrometer is faster decreasing the values and the Arduino takes a little bit longer, but after some seconds they reach the expected values.

![Image 70 Arduino and Electrometer for 2kV and 2mm](image)

Image 70 shows another experiment, this time with 2 kV/mm and 2mm gap between the electrodes. In this case the end values are 552 pA for Arduino and 543,66 pA for Electrometer. The difference between them is completely acceptable considering how much variation there are in the measures.
10. Summary and outlook

The summary will review all of the goals and objectives that were expected and set in place at the beginning of the project and how many and what was accomplished at the end.

The main goal was to build an ammeter that would be able to measure low level currents over an insulting system. This was successfully completed and the result has been the creation of a device capable of measuring that current.

One of the most important parts to consider when designing the device was that the device would need to be able to log the values and work with the data afterwards. This requirement was significant because it was necessary to keep this in consideration as with a multimeter the values would only be seen on screen. The final device is able to read and log a large amount of values, even more than one at a time, and stored in the .CSV file they are already showing the final values. In addition to storing values, the device also has also the output pins for a multimeter in the voltmeter function. The results can be seen on the screen of the multimeter at the same time, as for example, 100 pA = 100mV.

Another important part of this thesis was to design hardware that would be easy to repair or replace. The components used have been standard-size components. The welding and the process of designing and printing the board had also been done completely by hand. Welding by hand requires leaving enough space for the potential error created by human hands, and considering that mistakes could be made. This means that the spaces between components, as well as the separation between holes, lines and ground points is bigger than what it would be if the components had been placed by a machine.

Being inside a box protects the device from electric fields, distortions, and from external causes that could result in it being broken or malfunctioning.

In addition to the previous goals and requirements, the cost of all the components had to remain low. The total cost is approximately 125€, including the Arduino board. This cost doesn’t include the manufacturing time or cost.

Simplicity of use was also something important to consider through the development of the device. The whole process of measuring with the device consists of 3 switches to select the
measuring range, 2 switches for positive or negative values, and the software. It is really intuitive and easy to use, with the LED showing the switches and the software already prepared for each measurement. The only thing to do is run the test and measure the values.

In addition to the main goal of designing the device, it was also important that the device was able to measure a certain range of values. The micro amps range was discarded, as those values were too high for what the device was supposed to measure. Therefore, the special range of values was the pico and nano range.

The device has a resolution of 1,2pA between 10 and 1200 pA (10MOhm shunt resistor), 12,2 pA between 100 and 12,5 nA (1MOhm shunt resistor) and 1,2 nA between 10 and 1200 nA (10kOhm shunt resistor). These resolutions surpass the expected ones, as they are so small. A smaller resolution would have worked for the purpose of the device.

The accuracy of the results was more difficult and created a bit of trouble, but by the end of the project reached an acceptable result. There was an error that couldn’t be avoided due to the handmade board and components. To account for this error, correction formulas have been created to minimize that error for each range and approximate to the ideal values.

For future improvements, it would be good to consider installing automatic switches instead of manual ones. The benefit would be that automatic switches could be controlled by the computer and the Arduino, with the serial port and digital in/out pins. Another improvement would be to include the Arduino inside the metal box, if desired, for protection.

Finally, it would be good to consider installing a display on the metal box. This would substitute the function of the multimeter and show the current on the screen. There would be no need for a computer if the values were only wanted to be seen in real time.
11. References


## Appendix

### a. Correction arduino

<table>
<thead>
<tr>
<th>Keithley</th>
<th>Arduino</th>
<th>A+y</th>
<th>K-(A+y)</th>
<th>(K-(A+y))/K (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0</td>
<td>0.0357</td>
<td>-0.0257</td>
<td>-257</td>
</tr>
<tr>
<td>0.02</td>
<td>0</td>
<td>0.0357</td>
<td>-0.0157</td>
<td>-78.5</td>
</tr>
<tr>
<td>0.05</td>
<td>0.015</td>
<td>0.0504435</td>
<td>-0.0004435</td>
<td>-0.887</td>
</tr>
<tr>
<td>0.07</td>
<td>0.035</td>
<td>0.0701015</td>
<td>-0.0001015</td>
<td>-0.145</td>
</tr>
<tr>
<td>0.1</td>
<td>0.068</td>
<td>0.1025372</td>
<td>-0.0025372</td>
<td>-2.5372</td>
</tr>
<tr>
<td>0.15</td>
<td>0.119</td>
<td>0.1526651</td>
<td>-0.0026651</td>
<td>-1.776733333</td>
</tr>
<tr>
<td>0.2</td>
<td>0.168</td>
<td>0.2008272</td>
<td>-0.0008272</td>
<td>-0.4136</td>
</tr>
<tr>
<td>0.25</td>
<td>0.22</td>
<td>0.251938</td>
<td>-0.001938</td>
<td>-0.7752</td>
</tr>
<tr>
<td>0.3</td>
<td>0.271</td>
<td>0.3020659</td>
<td>-0.0020659</td>
<td>-0.688633333</td>
</tr>
<tr>
<td>0.35</td>
<td>0.321</td>
<td>0.3512109</td>
<td>-0.0012109</td>
<td>-0.345971429</td>
</tr>
<tr>
<td>0.4</td>
<td>0.371</td>
<td>0.4003559</td>
<td>-0.0003559</td>
<td>-0.088975</td>
</tr>
<tr>
<td>0.45</td>
<td>0.419</td>
<td>0.4475351</td>
<td>0.0024649</td>
<td>0.547755556</td>
</tr>
<tr>
<td>0.5</td>
<td>0.47</td>
<td>0.497663</td>
<td>0.002337</td>
<td>0.4674</td>
</tr>
<tr>
<td>0.6</td>
<td>0.572</td>
<td>0.5979188</td>
<td>0.0020812</td>
<td>0.346866667</td>
</tr>
<tr>
<td>0.7</td>
<td>0.673</td>
<td>0.6971917</td>
<td>0.0028083</td>
<td>0.401185714</td>
</tr>
<tr>
<td>0.8</td>
<td>0.777</td>
<td>0.7994133</td>
<td>0.005867</td>
<td>0.0733375</td>
</tr>
<tr>
<td>0.9</td>
<td>0.877</td>
<td>0.8977033</td>
<td>0.0022967</td>
<td>0.255188889</td>
</tr>
<tr>
<td>1</td>
<td>0.98</td>
<td>0.998942</td>
<td>0.001058</td>
<td>0.1058</td>
</tr>
<tr>
<td>1.2</td>
<td>1.185</td>
<td>1.2004365</td>
<td>-0.0004365</td>
<td>-0.036375</td>
</tr>
<tr>
<td>1.3</td>
<td>1.287</td>
<td>1.3006923</td>
<td>-0.0006923</td>
<td>-0.053253846</td>
</tr>
<tr>
<td>1.4</td>
<td>1.388</td>
<td>1.3999652</td>
<td>3.48E-05</td>
<td>0.002485714</td>
</tr>
<tr>
<td>1.5</td>
<td>1.49</td>
<td>1.500221</td>
<td>-0.000221</td>
<td>-0.014733333</td>
</tr>
<tr>
<td>1.6</td>
<td>1.692</td>
<td>1.6987668</td>
<td>0.0012332</td>
<td>0.072541176</td>
</tr>
<tr>
<td>1.7</td>
<td>1.899</td>
<td>2.0006171</td>
<td>-0.0006171</td>
<td>-0.025855</td>
</tr>
<tr>
<td>1.8</td>
<td>2.020</td>
<td>2.0200458</td>
<td>-4.58E-05</td>
<td>-0.002081818</td>
</tr>
<tr>
<td>1.9</td>
<td>2.405</td>
<td>2.3995745</td>
<td>0.0004255</td>
<td>0.017729167</td>
</tr>
<tr>
<td>2.0</td>
<td>2.608</td>
<td>2.59960432</td>
<td>0.0008968</td>
<td>0.034492308</td>
</tr>
<tr>
<td>2.1</td>
<td>2.811</td>
<td>2.79868319</td>
<td>0.0013681</td>
<td>0.048860714</td>
</tr>
<tr>
<td>2.2</td>
<td>2.913</td>
<td>2.8988877</td>
<td>0.0011123</td>
<td>0.038555172</td>
</tr>
<tr>
<td>2.3</td>
<td>3.014</td>
<td>2.9981606</td>
<td>0.0018394</td>
<td>0.061333333</td>
</tr>
<tr>
<td>2.4</td>
<td>3.22</td>
<td>3.200638</td>
<td>-0.000638</td>
<td>-0.0199375</td>
</tr>
<tr>
<td>2.5</td>
<td>3.424</td>
<td>3.4011496</td>
<td>-0.0011496</td>
<td>-0.033811765</td>
</tr>
<tr>
<td>2.6</td>
<td>3.628</td>
<td>3.6016612</td>
<td>-0.0016612</td>
<td>-0.046144444</td>
</tr>
<tr>
<td>2.7</td>
<td>3.829</td>
<td>3.7992241</td>
<td>0.0007759</td>
<td>0.020418421</td>
</tr>
<tr>
<td>2.8</td>
<td>4.033</td>
<td>3.9997357</td>
<td>0.0002643</td>
<td>0.0066075</td>
</tr>
<tr>
<td>2.9</td>
<td>4.237</td>
<td>4.2002473</td>
<td>-0.0002473</td>
<td>-0.005888095</td>
</tr>
<tr>
<td>3.0</td>
<td>4.443</td>
<td>4.4027247</td>
<td>-0.0027247</td>
<td>-0.061925</td>
</tr>
<tr>
<td>3.1</td>
<td>4.646</td>
<td>4.6022534</td>
<td>-0.0022534</td>
<td>-0.048986957</td>
</tr>
</tbody>
</table>
b. Values for electrometer when testing the insulating system with 1kV and 2mm

<table>
<thead>
<tr>
<th>Current (pA)</th>
<th>Time (s)</th>
<th>78.165</th>
<th>7.6</th>
<th>79.025</th>
<th>12.7</th>
<th>75.591</th>
<th>17.4</th>
<th>75.735</th>
<th>22.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.0</td>
<td>76.102</td>
<td>7.9</td>
<td>76.804</td>
<td>12.8</td>
<td>81.69</td>
<td>17.6</td>
<td>74.95</td>
<td>22.5</td>
</tr>
<tr>
<td>0.0417</td>
<td>2.3</td>
<td>77.483</td>
<td>8.0</td>
<td>77.585</td>
<td>12.9</td>
<td>78.284</td>
<td>17.7</td>
<td>72.324</td>
<td>22.6</td>
</tr>
<tr>
<td>2.6142</td>
<td>2.5</td>
<td>75.05</td>
<td>8.2</td>
<td>74.164</td>
<td>13.1</td>
<td>74.196</td>
<td>17.8</td>
<td>78.606</td>
<td>22.7</td>
</tr>
<tr>
<td>9311.1</td>
<td>2.6</td>
<td>74.484</td>
<td>8.3</td>
<td>79.446</td>
<td>13.2</td>
<td>74.184</td>
<td>18.0</td>
<td>77.84</td>
<td>22.9</td>
</tr>
<tr>
<td>6173.3</td>
<td>3.3</td>
<td>77.217</td>
<td>8.5</td>
<td>77.661</td>
<td>13.3</td>
<td>73.439</td>
<td>18.4</td>
<td>76.81</td>
<td>23.0</td>
</tr>
<tr>
<td>2601.4</td>
<td>3.6</td>
<td>75.982</td>
<td>8.6</td>
<td>79.096</td>
<td>13.5</td>
<td>71.209</td>
<td>18.7</td>
<td>77.658</td>
<td>23.2</td>
</tr>
<tr>
<td>1116.7</td>
<td>3.7</td>
<td>73.343</td>
<td>8.7</td>
<td>82.876</td>
<td>13.6</td>
<td>73.641</td>
<td>18.8</td>
<td>79.304</td>
<td>23.6</td>
</tr>
<tr>
<td>953.37</td>
<td>4.0</td>
<td>77.438</td>
<td>8.9</td>
<td>82.815</td>
<td>13.8</td>
<td>73.282</td>
<td>19.0</td>
<td>75.329</td>
<td>23.7</td>
</tr>
<tr>
<td>508.53</td>
<td>4.1</td>
<td>74.619</td>
<td>9.2</td>
<td>80.471</td>
<td>13.9</td>
<td>70.815</td>
<td>19.1</td>
<td>74.951</td>
<td>23.9</td>
</tr>
<tr>
<td>386.22</td>
<td>4.3</td>
<td>75.973</td>
<td>9.4</td>
<td>78.475</td>
<td>14.1</td>
<td>69.355</td>
<td>19.2</td>
<td>74.586</td>
<td>24.0</td>
</tr>
<tr>
<td>244.92</td>
<td>4.4</td>
<td>73.257</td>
<td>9.7</td>
<td>78.822</td>
<td>14.3</td>
<td>74.275</td>
<td>19.4</td>
<td>76.083</td>
<td>24.4</td>
</tr>
<tr>
<td>48.125</td>
<td>4.5</td>
<td>73.351</td>
<td>9.9</td>
<td>83.877</td>
<td>14.5</td>
<td>74.451</td>
<td>19.5</td>
<td>73.659</td>
<td>24.6</td>
</tr>
<tr>
<td>55.621</td>
<td>5.1</td>
<td>73.637</td>
<td>10.0</td>
<td>79.984</td>
<td>14.6</td>
<td>72.905</td>
<td>19.7</td>
<td>80.433</td>
<td>24.7</td>
</tr>
<tr>
<td>60.677</td>
<td>5.3</td>
<td>74.433</td>
<td>10.2</td>
<td>77.375</td>
<td>14.8</td>
<td>71.536</td>
<td>19.9</td>
<td>78.368</td>
<td>24.8</td>
</tr>
<tr>
<td>67.14</td>
<td>5.4</td>
<td>72.091</td>
<td>10.3</td>
<td>78.524</td>
<td>14.9</td>
<td>74.878</td>
<td>20.1</td>
<td>74.263</td>
<td>25.0</td>
</tr>
<tr>
<td>68.077</td>
<td>5.6</td>
<td>77.812</td>
<td>10.4</td>
<td>77.784</td>
<td>15.1</td>
<td>72.516</td>
<td>20.2</td>
<td>76.504</td>
<td>25.1</td>
</tr>
<tr>
<td>68.064</td>
<td>5.7</td>
<td>78.39</td>
<td>10.6</td>
<td>77.44</td>
<td>15.2</td>
<td>72.187</td>
<td>20.4</td>
<td>74.573</td>
<td>25.3</td>
</tr>
<tr>
<td>72.536</td>
<td>5.8</td>
<td>85.031</td>
<td>10.7</td>
<td>73.642</td>
<td>15.6</td>
<td>73.653</td>
<td>20.5</td>
<td>74.945</td>
<td>25.4</td>
</tr>
<tr>
<td>73.438</td>
<td>6.0</td>
<td>81.979</td>
<td>11.0</td>
<td>70.841</td>
<td>15.7</td>
<td>73.36</td>
<td>20.6</td>
<td>76.304</td>
<td>25.6</td>
</tr>
<tr>
<td>75.799</td>
<td>6.1</td>
<td>80.395</td>
<td>11.1</td>
<td>77.248</td>
<td>16.0</td>
<td>77.572</td>
<td>20.8</td>
<td>73.556</td>
<td>25.7</td>
</tr>
<tr>
<td>85.181</td>
<td>6.4</td>
<td>80.726</td>
<td>11.3</td>
<td>77.802</td>
<td>16.2</td>
<td>77.428</td>
<td>20.9</td>
<td>70.97</td>
<td>26.0</td>
</tr>
<tr>
<td>78.342</td>
<td>6.7</td>
<td>77.802</td>
<td>11.4</td>
<td>74.989</td>
<td>16.3</td>
<td>81.973</td>
<td>21.1</td>
<td>70.058</td>
<td>26.1</td>
</tr>
<tr>
<td>72.758</td>
<td>6.9</td>
<td>77.131</td>
<td>11.5</td>
<td>76.399</td>
<td>16.4</td>
<td>76.145</td>
<td>21.4</td>
<td>69.481</td>
<td>26.3</td>
</tr>
<tr>
<td>72.491</td>
<td>7.1</td>
<td>76.147</td>
<td>11.7</td>
<td>77.968</td>
<td>16.6</td>
<td>71.727</td>
<td>21.7</td>
<td>67.821</td>
<td>26.4</td>
</tr>
<tr>
<td>70.773</td>
<td>7.2</td>
<td>76.734</td>
<td>11.8</td>
<td>75.214</td>
<td>16.8</td>
<td>72.546</td>
<td>21.8</td>
<td>68.102</td>
<td>26.6</td>
</tr>
<tr>
<td>71.846</td>
<td>7.4</td>
<td>78.484</td>
<td>11.9</td>
<td>75.61</td>
<td>17.0</td>
<td>75.851</td>
<td>21.9</td>
<td>66.658</td>
<td>26.7</td>
</tr>
<tr>
<td>77.97</td>
<td>7.5</td>
<td>76.826</td>
<td>12.1</td>
<td>72.029</td>
<td>17.1</td>
<td>73.924</td>
<td>22.1</td>
<td>72.093</td>
<td>26.8</td>
</tr>
</tbody>
</table>
c. Values for Arduino when testing the insulating system with 1kV and 2mm

<p>| Voltage (V) | 0.00 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 | 1.4 | 1.6 | 1.8 | 2.0 | 2.2 | 2.4 | 2.6 | 2.8 | 3.0 | 3.2 | 3.4 | 3.6 | 3.8 | 4.0 | 4.2 | 4.4 | 4.6 | 4.8 | 5.0 | 5.2 | 5.4 | 5.6 | 5.8 | 6.0 | 6.2 | 6.4 | 6.6 | 6.8 | 7.0 | 7.2 | 7.4 | 7.6 | 7.8 | 8.0 | 8.2 | 8.4 | 8.6 | 8.8 | 9.0 | 9.2 | 9.4 | 9.6 | 9.8 | 10.0 | 10.2 | 10.4 | 10.6 | 10.8 | 11.0 | 11.2 | 11.4 | 11.6 | 11.8 | 12.0 | 12.2 | 12.4 | 12.6 | 12.8 | 13.0 | 13.2 | 13.4 | 13.6 | 13.8 | 14.0 | 14.2 | 14.4 | 14.6 | 14.8 | 15.0 | 15.2 | 15.4 | 15.6 | 15.8 | 16.0 | 16.2 | 16.4 | 16.6 | 16.8 | 17.0 | 17.2 | 17.4 | 17.6 | 17.8 | 18.0 | 18.2 | 18.4 | 18.6 | 18.8 | 19.0 | 19.2 | 19.4 | 19.6 | 19.8 | 20.0 | 20.2 | 20.4 | 20.6 | 20.8 | 21.0 | 21.2 | 21.4 | 21.6 | 21.8 | 22.0 | 22.2 | 22.4 | 22.6 | 22.8 | 23.0 | 23.2 | 23.4 | 23.6 | 23.8 | 24.0 | 24.2 | 24.4 | 24.6 | 24.8 | 25.0 | 25.2 | 25.4 | 25.6 | 25.8 | 26.0 | 26.2 | 26.4 | 26.6 | 26.8 | 27.0 | 27.2 | 27.4 | 27.6 | 27.8 | 28.0 | 28.2 | 28.4 | 28.6 | 28.8 | 29.0 | 29.2 | 29.4 | 29.6 | 29.8 | 30.0 | 30.2 | 30.4 | 30.6 | 30.8 | 31.0 | 31.2 | 31.4 | 31.6 | 31.8 | 32.0 | 32.2 | 32.4 | 32.6 | 32.8 | 33.0 | 33.2 | 33.4 | 33.6 | 33.8 | 34.0 | 34.2 | 34.4 | 34.6 | 34.8 |</p>
<table>
<thead>
<tr>
<th>53.86</th>
<th>72</th>
<th>58.09</th>
<th>75</th>
<th>56.07</th>
<th>78</th>
<th>56.87</th>
<th>81</th>
<th>54.97</th>
<th>84</th>
</tr>
</thead>
<tbody>
<tr>
<td>54.42</td>
<td>72.2</td>
<td>57.73</td>
<td>75.2</td>
<td>55.34</td>
<td>78.2</td>
<td>56.93</td>
<td>81.2</td>
<td>55.15</td>
<td>84.2</td>
</tr>
<tr>
<td>56.99</td>
<td>72.4</td>
<td>58.03</td>
<td>75.4</td>
<td>55.46</td>
<td>78.4</td>
<td>57.60</td>
<td>81.4</td>
<td>54.84</td>
<td>84.4</td>
</tr>
<tr>
<td>56.07</td>
<td>72.6</td>
<td>57.30</td>
<td>75.6</td>
<td>55.76</td>
<td>78.6</td>
<td>56.93</td>
<td>81.6</td>
<td>53.99</td>
<td>84.6</td>
</tr>
<tr>
<td>55.27</td>
<td>72.8</td>
<td>57.05</td>
<td>75.8</td>
<td>55.70</td>
<td>78.8</td>
<td>57.17</td>
<td>81.8</td>
<td>53.50</td>
<td>84.8</td>
</tr>
<tr>
<td>55.03</td>
<td>73</td>
<td>56.50</td>
<td>76</td>
<td>55.34</td>
<td>79</td>
<td>55.89</td>
<td>82</td>
<td>53.44</td>
<td>85</td>
</tr>
<tr>
<td>56.81</td>
<td>73.2</td>
<td>54.84</td>
<td>76.2</td>
<td>55.09</td>
<td>79.2</td>
<td>54.29</td>
<td>82.2</td>
<td>52.39</td>
<td>85.2</td>
</tr>
<tr>
<td>54.23</td>
<td>73.4</td>
<td>56.25</td>
<td>76.4</td>
<td>54.29</td>
<td>79.4</td>
<td>54.05</td>
<td>82.4</td>
<td>51.29</td>
<td>85.4</td>
</tr>
<tr>
<td>54.05</td>
<td>73.6</td>
<td>56.32</td>
<td>76.6</td>
<td>55.58</td>
<td>79.6</td>
<td>53.62</td>
<td>82.6</td>
<td>51.84</td>
<td>85.6</td>
</tr>
<tr>
<td>54.23</td>
<td>73.8</td>
<td>55.40</td>
<td>76.8</td>
<td>54.97</td>
<td>79.8</td>
<td>53.25</td>
<td>82.8</td>
<td>53.80</td>
<td>85.8</td>
</tr>
<tr>
<td>53.80</td>
<td>74</td>
<td>55.34</td>
<td>77</td>
<td>52.82</td>
<td>80</td>
<td>51.84</td>
<td>83</td>
<td>56.32</td>
<td>86</td>
</tr>
<tr>
<td>53.19</td>
<td>74.2</td>
<td>54.23</td>
<td>77.2</td>
<td>53.25</td>
<td>80.2</td>
<td>54.17</td>
<td>83.2</td>
<td>56.38</td>
<td>86.2</td>
</tr>
<tr>
<td>53.13</td>
<td>74.4</td>
<td>53.19</td>
<td>77.4</td>
<td>54.60</td>
<td>80.4</td>
<td>55.03</td>
<td>83.4</td>
<td>57.23</td>
<td>86.4</td>
</tr>
<tr>
<td>52.95</td>
<td>74.6</td>
<td>52.33</td>
<td>77.6</td>
<td>57.05</td>
<td>80.6</td>
<td>55.15</td>
<td>83.6</td>
<td>56.62</td>
<td>86.6</td>
</tr>
<tr>
<td>56.38</td>
<td>74.8</td>
<td>55.83</td>
<td>77.8</td>
<td>57.60</td>
<td>80.8</td>
<td>54.66</td>
<td>83.8</td>
<td>53.25</td>
<td>82.8</td>
</tr>
</tbody>
</table>