DIY LABS

MICROFLUIDIC SYSTEMS

BIOMICROSYSTEMS

MAKER

START BUILDING

PUBLISHED BY
UNIVERSIDAD DE LOS ANDES
FACULTY

EDITION
#1

Printed by Editorial Uniandes, in Colombia.

First Printing, 2021.

Editorial Uniandes
Cra. 1E #19A-44
Bogotá, Colombia.

AUTHORS

Johann F. Osma, PhD
Cargo/Info Importante

Alejandra Murillo
Cargo/Info Importante

Natalia Ramos
Cargo/Info Importante

Juan Manuel Rivera
Cargo/Info Importante

Gina Salazar
Cargo/Info Importante

Juan Velásquez
Cargo/Info Importante

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to

Olga Fuentes – Microfluidic systems simulations

Miguel Angel Aroca – Impedance Meter

Crhistian Segura – Magnetic Field, (Image Processing Co-Author)

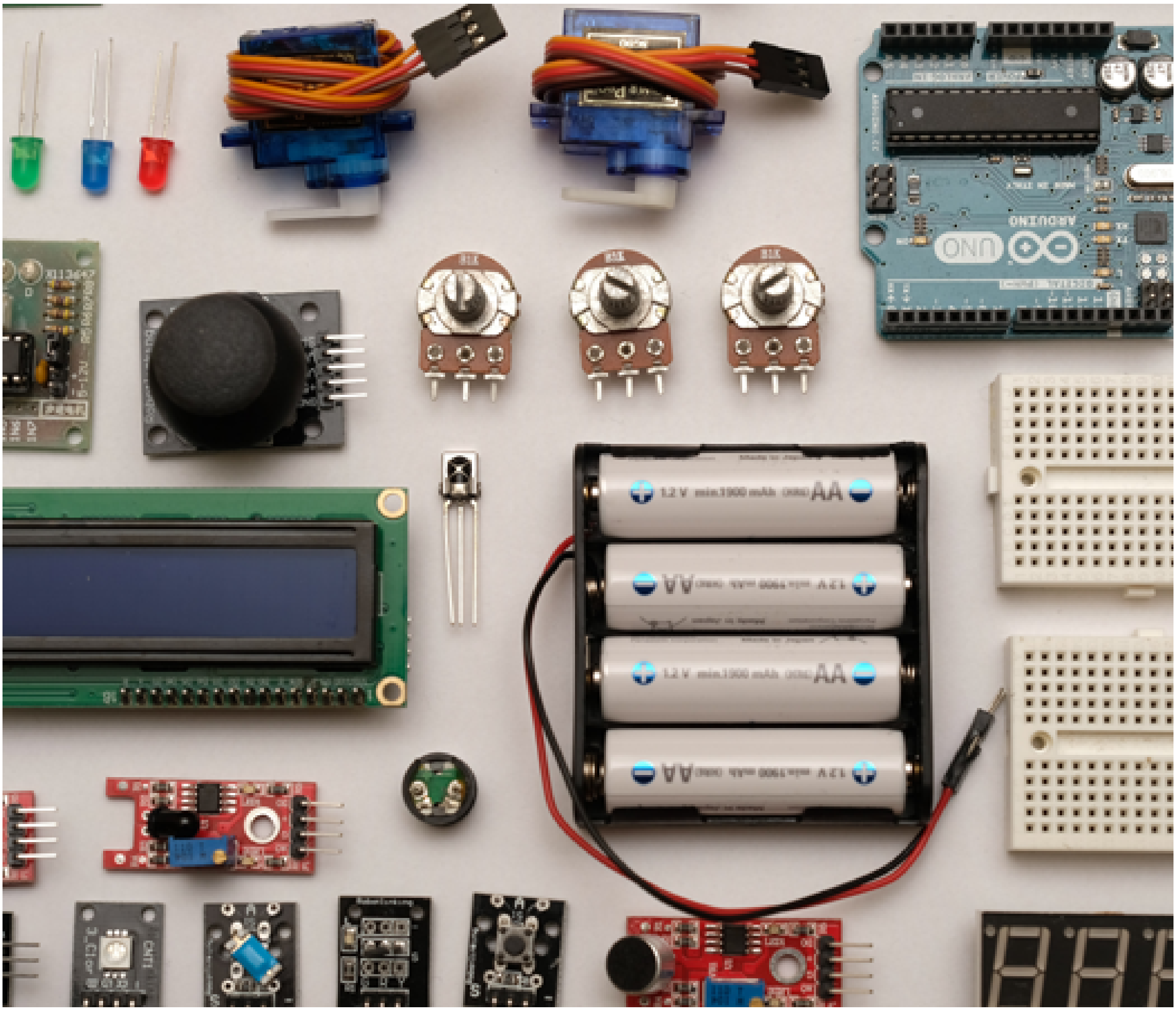
Ana Maria Forero Roa & Isabella Avendaño Cortés – Thermal Bed

Agregar a más personas

Agregar a más personas

for their invaluable input, which allowed us to

INDEX



Adam Savage said:

"Humans do two things that make us unique from all other animals; we use tools and tell stories. And when you make something, you're doing both at once."

That is a beautiful feeling. We create magic in every school lab, makerspace, and workshop we've visited. We love experiencing stories as oral tales, pictures and videos, or even in the form of tutorials. Here there is the start point for you to create your lab stories.

Start being a Lab Maker

DOWNLOADING THE FILES

| QR AND GITHUB

For each device, there is a file uploaded to Github that complements the information of the chapter. In each file you are going to find the technical drawings, coding, PCBs and any aditional file that you need to download to successfully build your own lab.

https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/

DIY LABS

LASER CUT ELEMENTS

BASIC INFORMATION

The following instructions are for the assembly of different elements that can help you build your own laboratory. Elements such as: racks, weight supports and stands

MATERIALS

Acrylic 2mm

Glue for acrylic (typically solvent-based glues).

Laser Cutter

PROCEDURE

Download cutting plans.

Cut the plans properly

Detach the pieces from the acrylic.

Allow to dry for 24 hours.

Follow the assembly instructions.

Allow the solvent to act for a few seconds softening the acrylic and then press the two parts together to obtain a firm contact. Hold the parts still for a few minutes (1-2 min) until they are self-supporting.

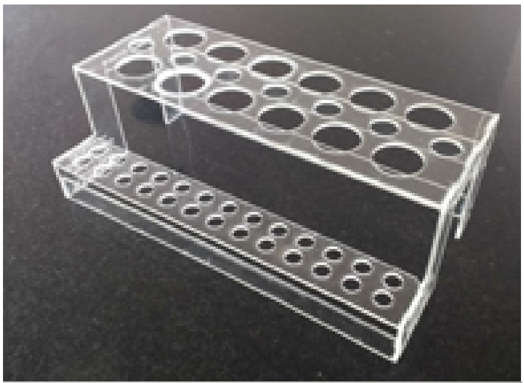
01. Materials
02. Procedure
03. Technical Drawings

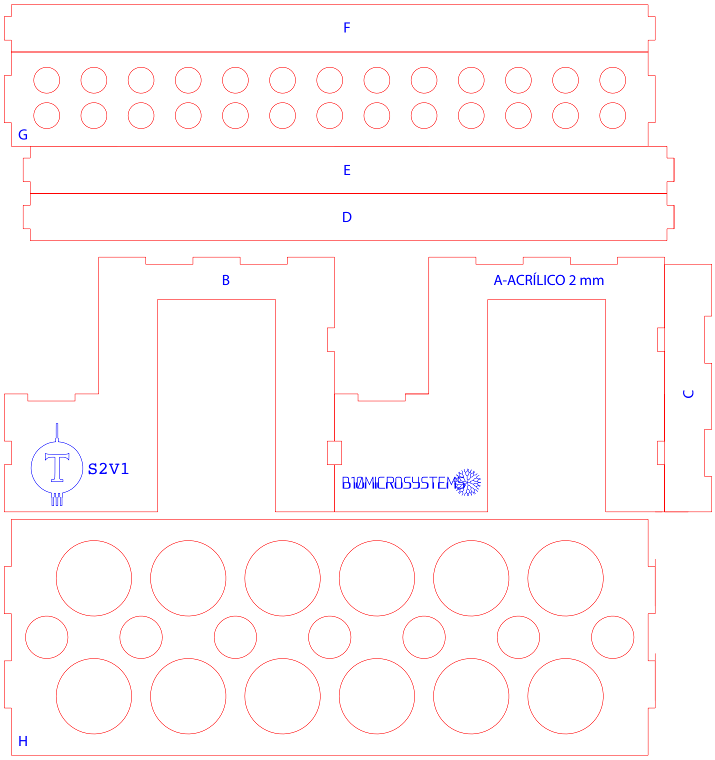
https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/Racks/Racks

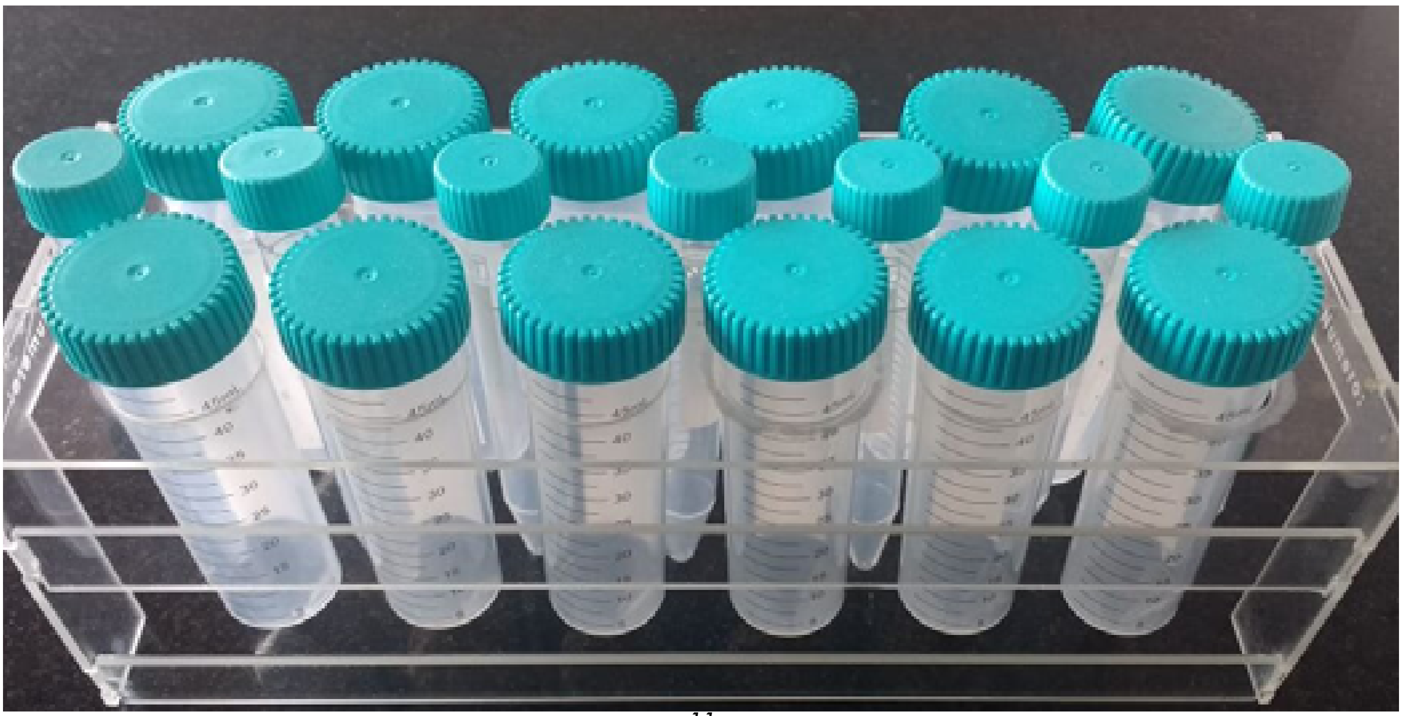
TECHNICAL DRAWINGS
RACKS

The rack is a laboratory tool (mainly in molecular biology, genetics and chemistry laboratories), which is used as a support to store and hold test tubes or sample tubes.

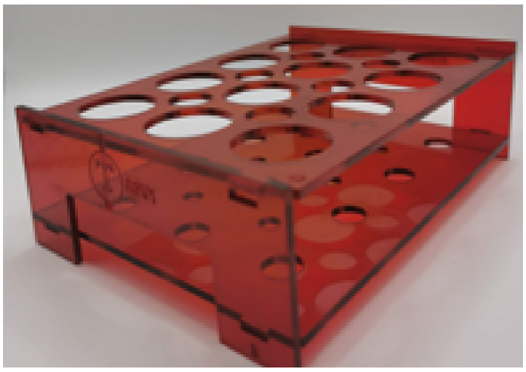
| MIXED RACKS

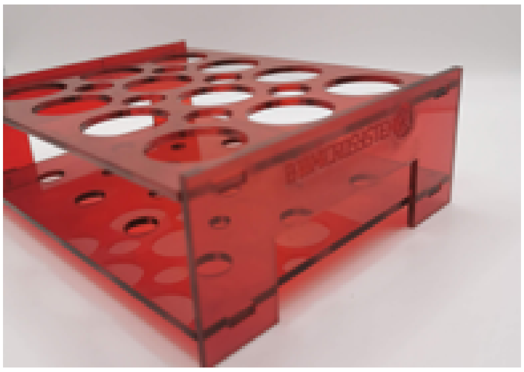


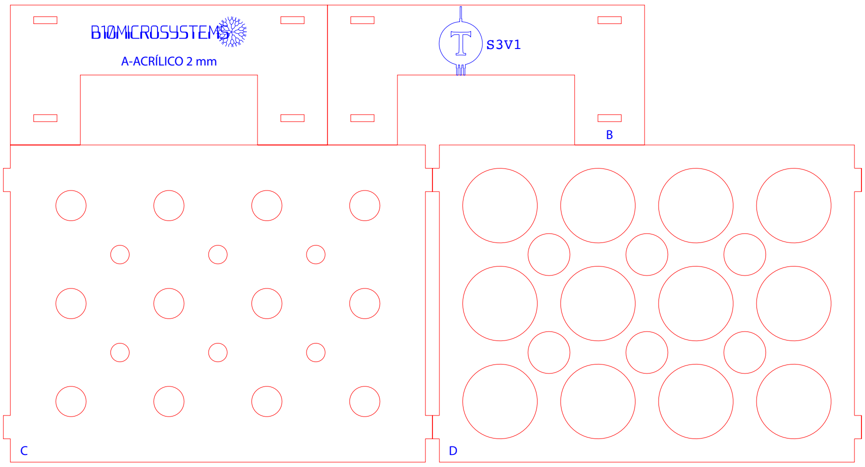




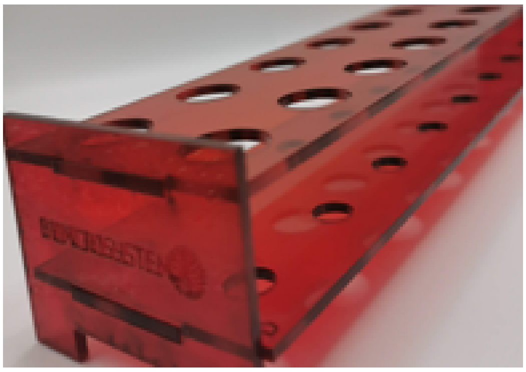
15 AND 50 ml RACKS

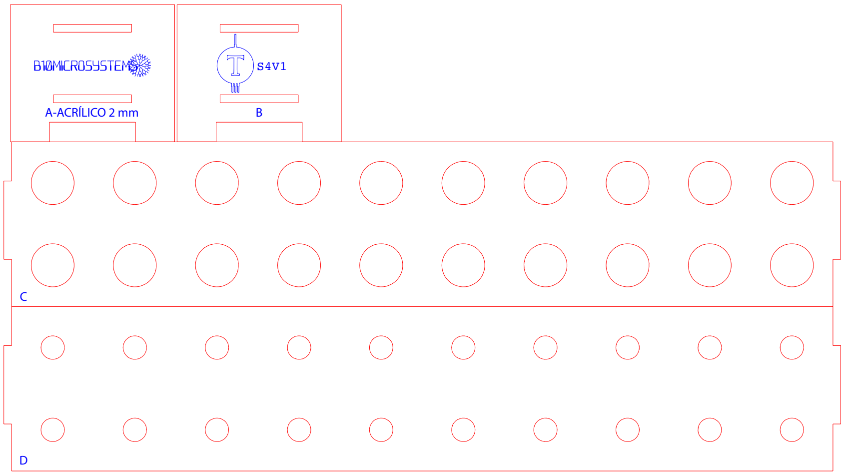


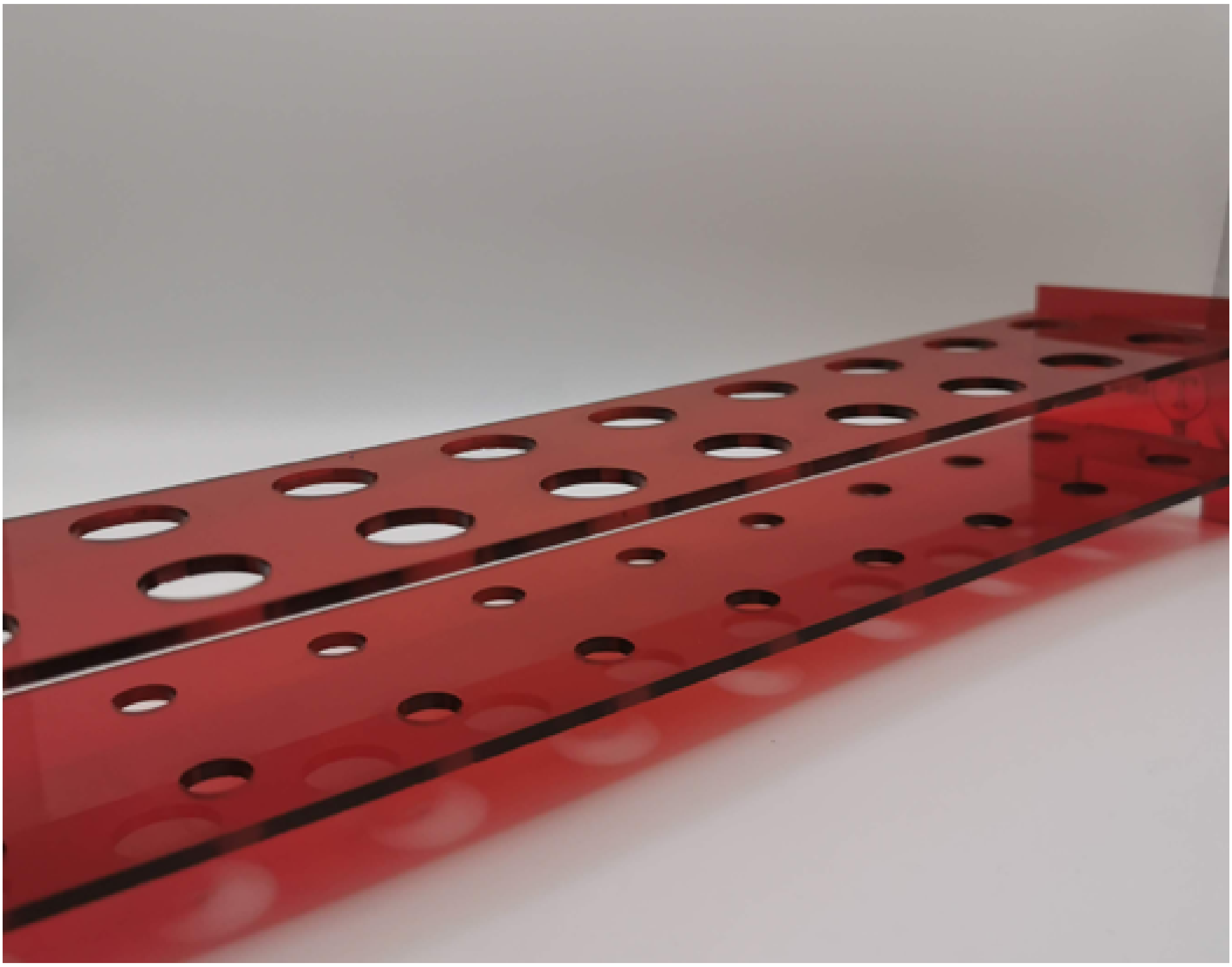




EPPENDORF RACKS



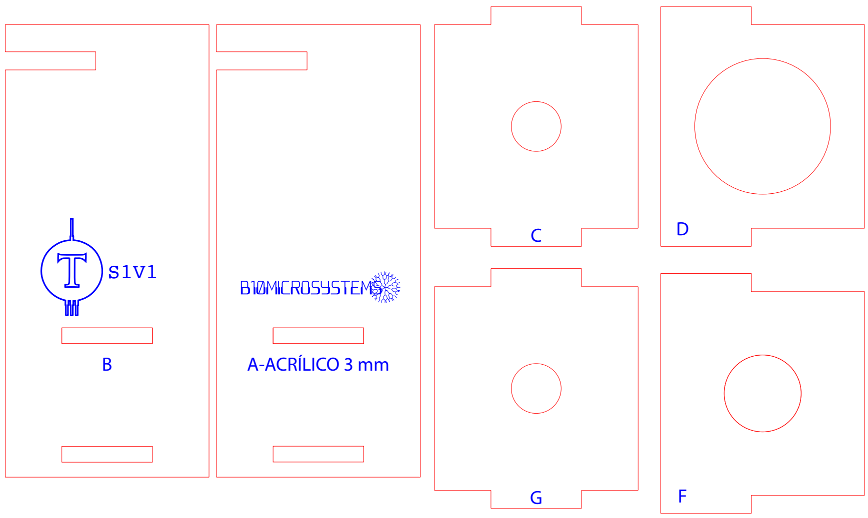


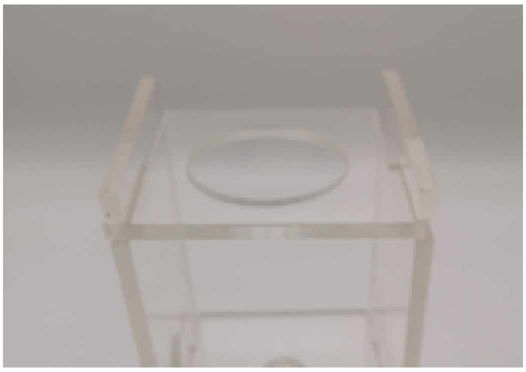


TECHNICAL DRAWINGS

WEIGHT SUPPORT

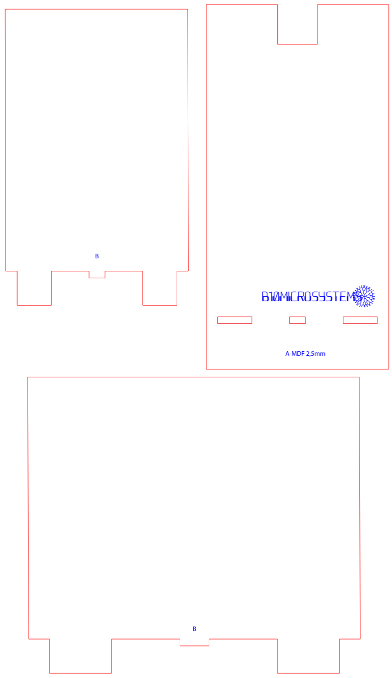
https://github.com/Biomicrosystems/Documentation/blob/main/DIY%20Book/Racks/weight%20support.ai

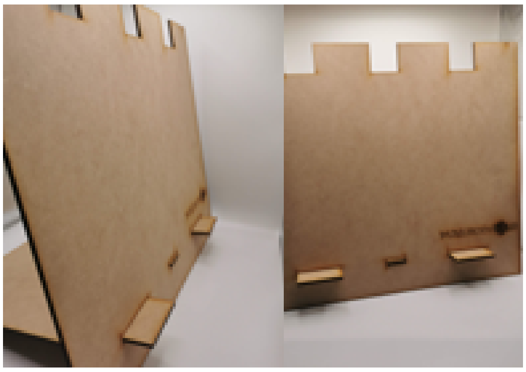




TECHNICAL DRAWINGS
PIPETTE STAND

https://github.com/Biomicrosystems/Documentation/blob/main/DIY%20Book/Racks/Pipette%20Stand.pdf



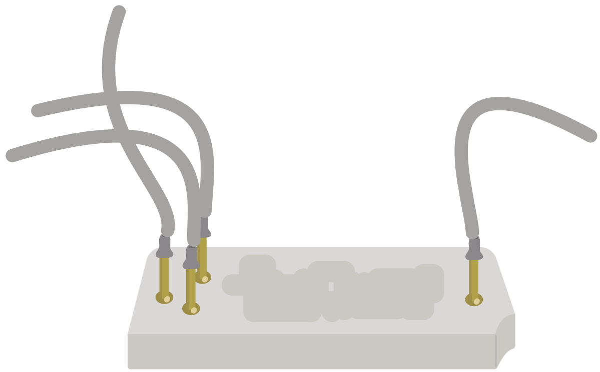


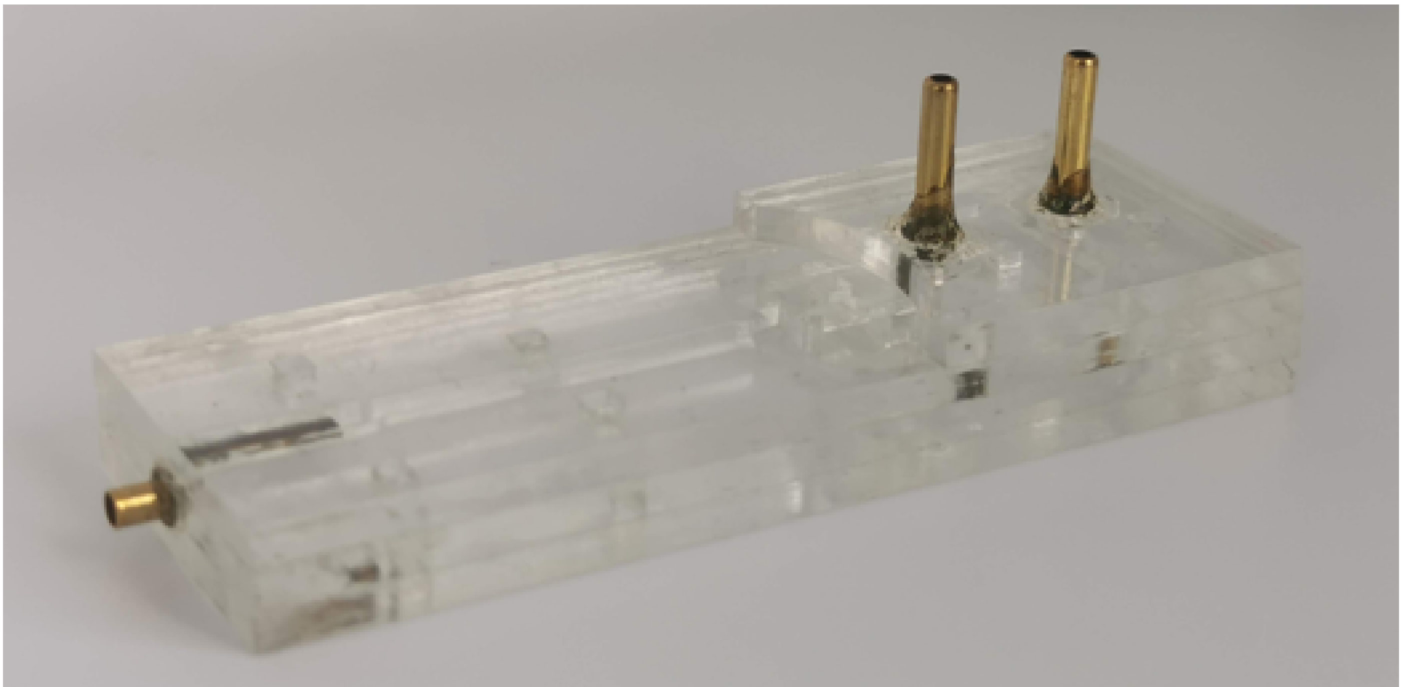
MICROFLUIDIC SIMULATION

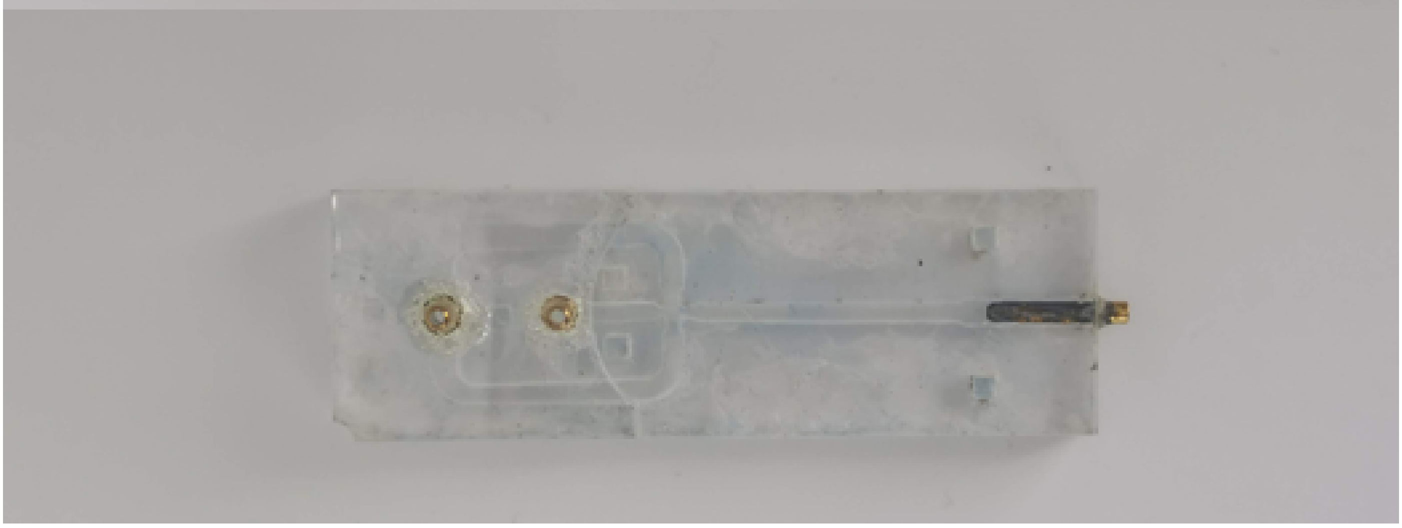
Understanding that this elements should serve with a propose. At Biomicrosystems, we use this elements to support the making of Microfluidic systems. These are systems that process or manipulate small amounts of fluidics (10−9 to 10−18L), using channels measuring from tens to hundreds of micrometers. [1]

Numerous effects can be exploited at this length scale that cannot (or hardly) be seen in fluidic systems at greater length scales [2].

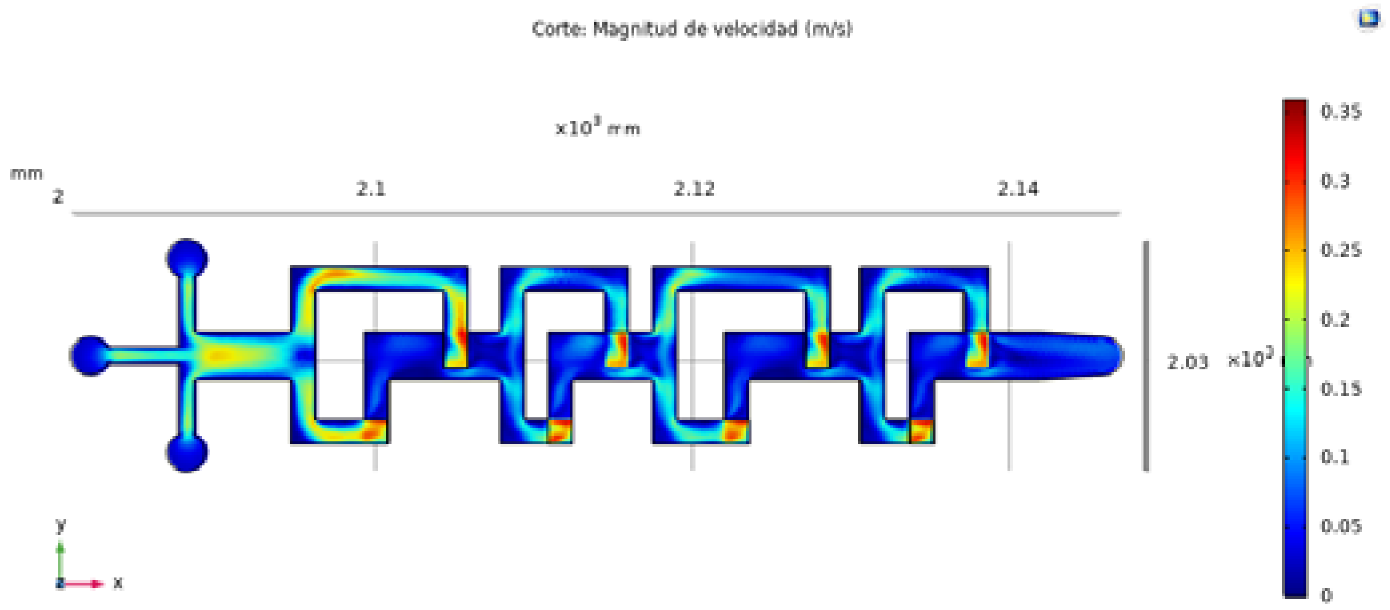
Here we present velocity profiles of a fluid flowing within a microfluidic system. Specifically, seen from a fluid that is know for most of us, waste waters. A promising way to tackle these issues is the use of enzyme-based and miniaturized biosensors for their electrochemical detection. [3]

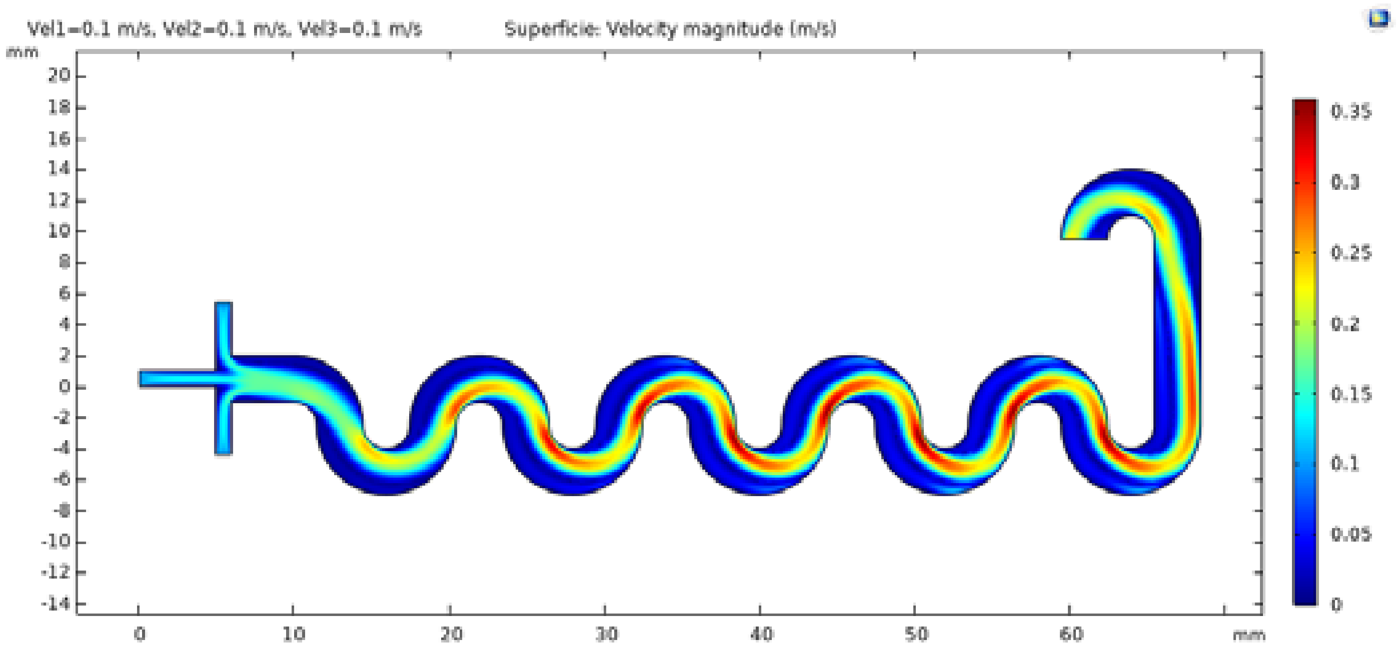




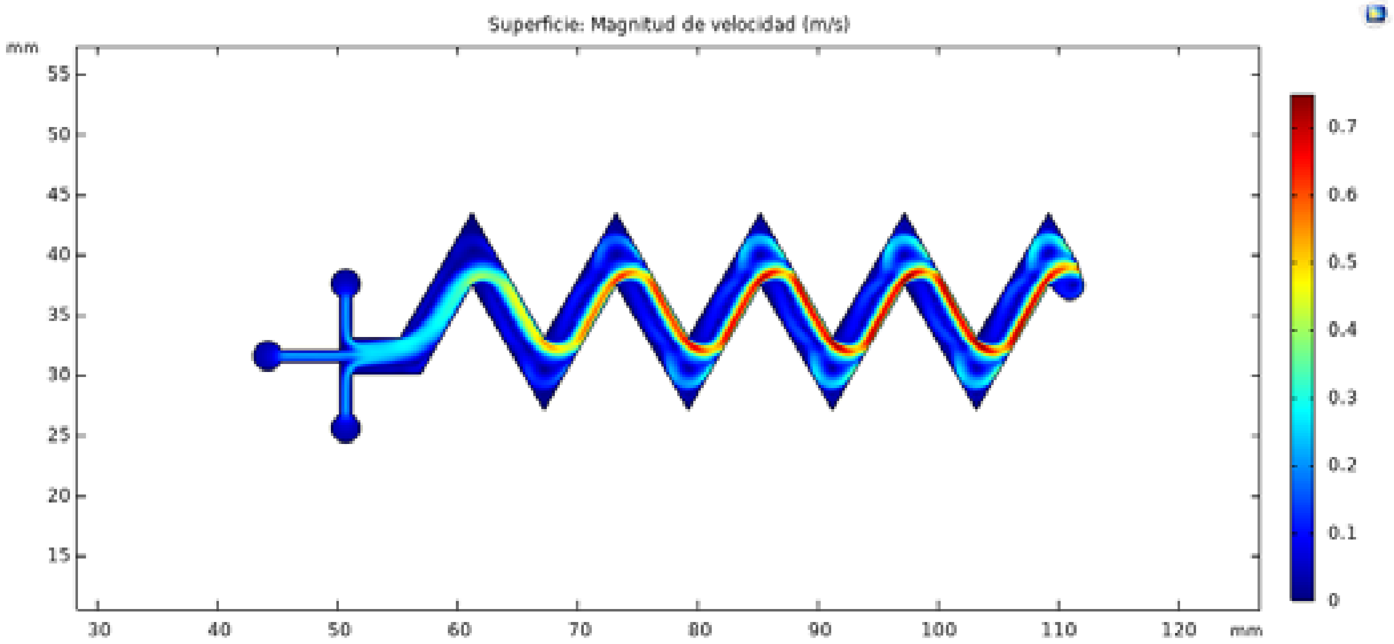


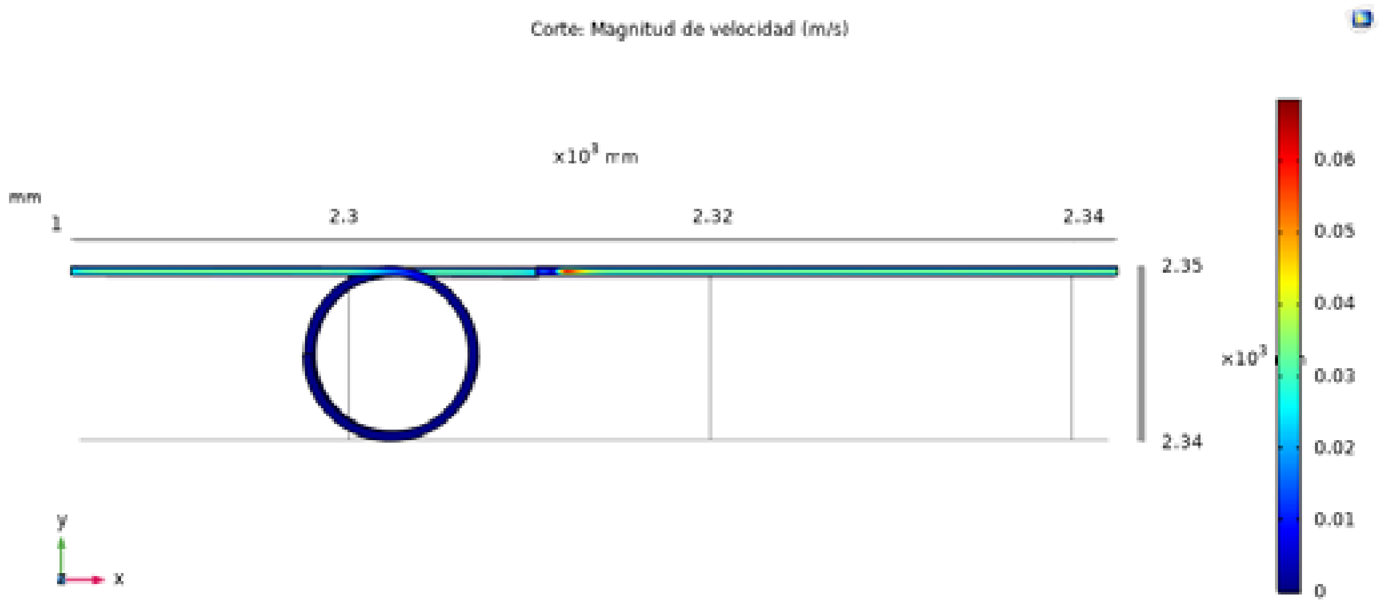
VELOCITY PROFILES



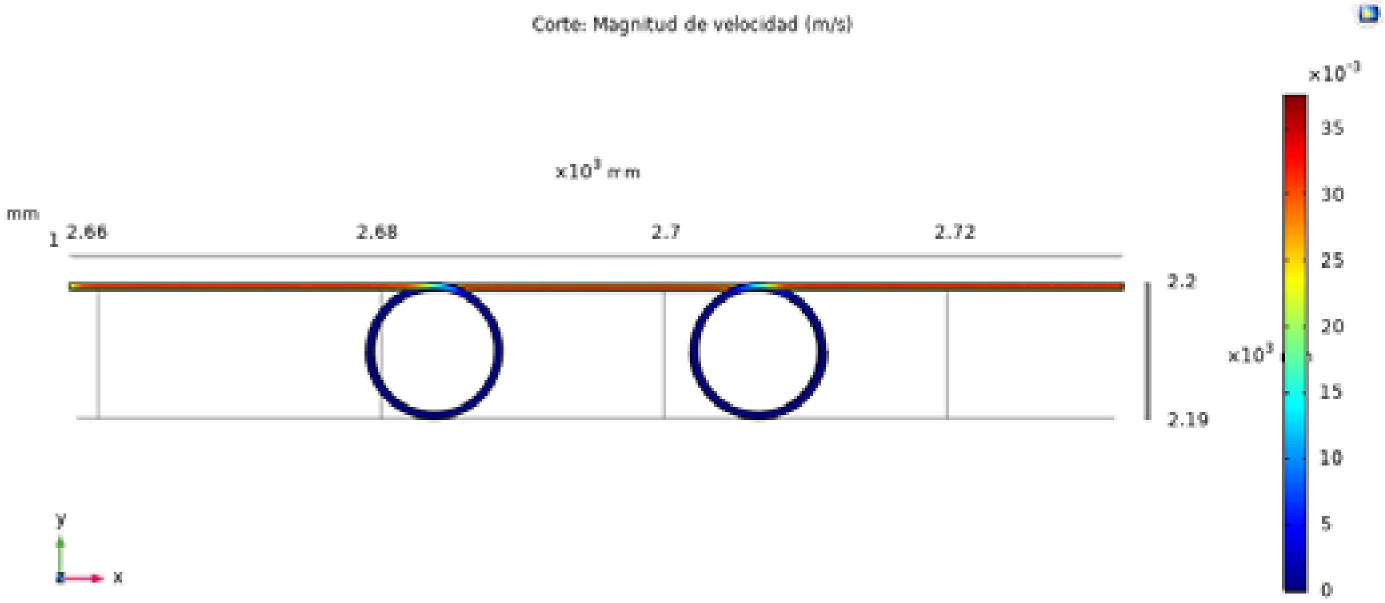


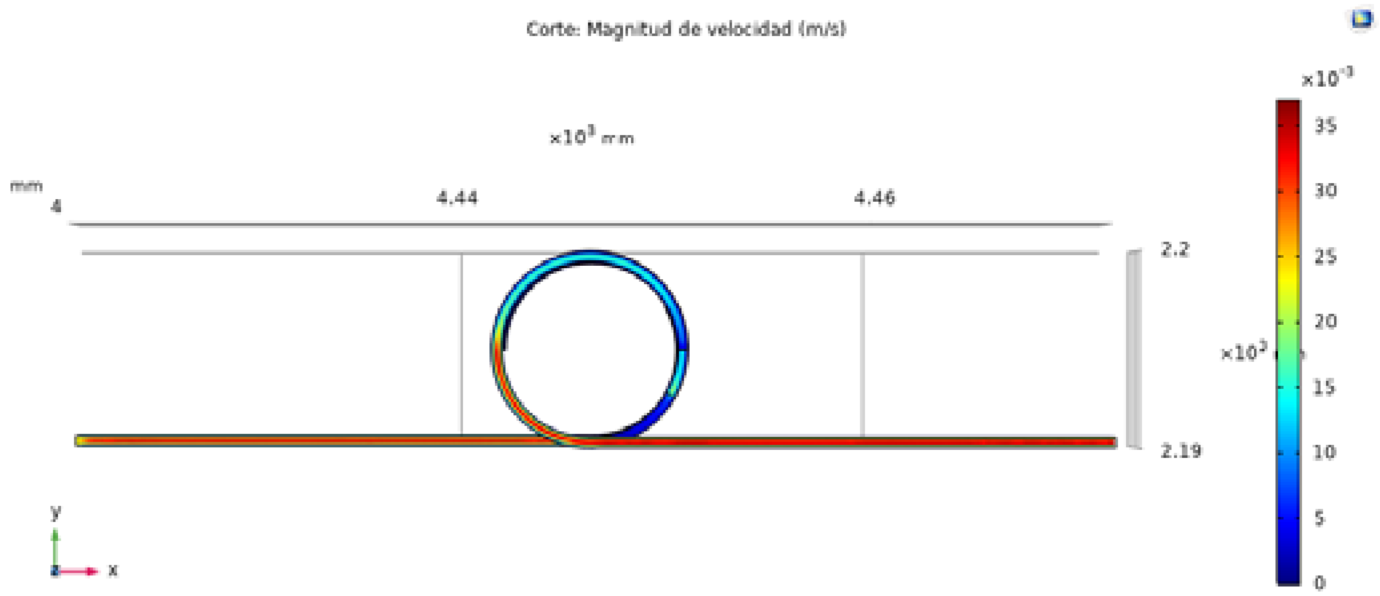
VELOCITY PROFILES



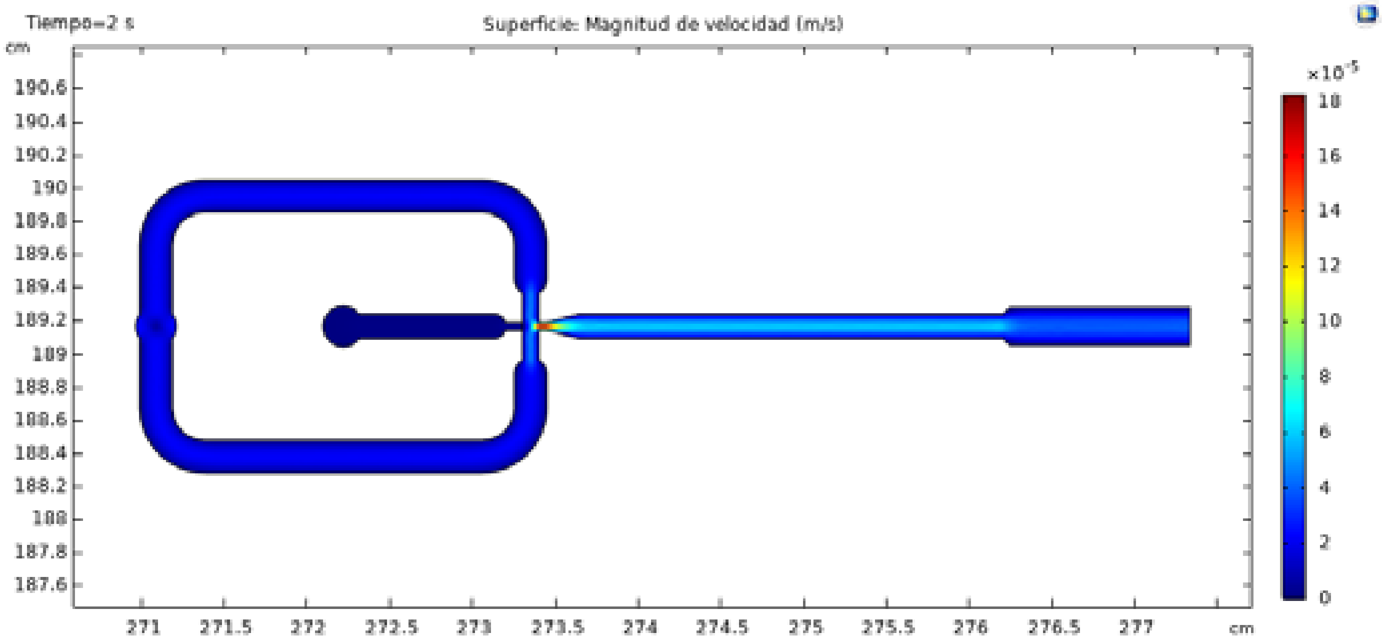


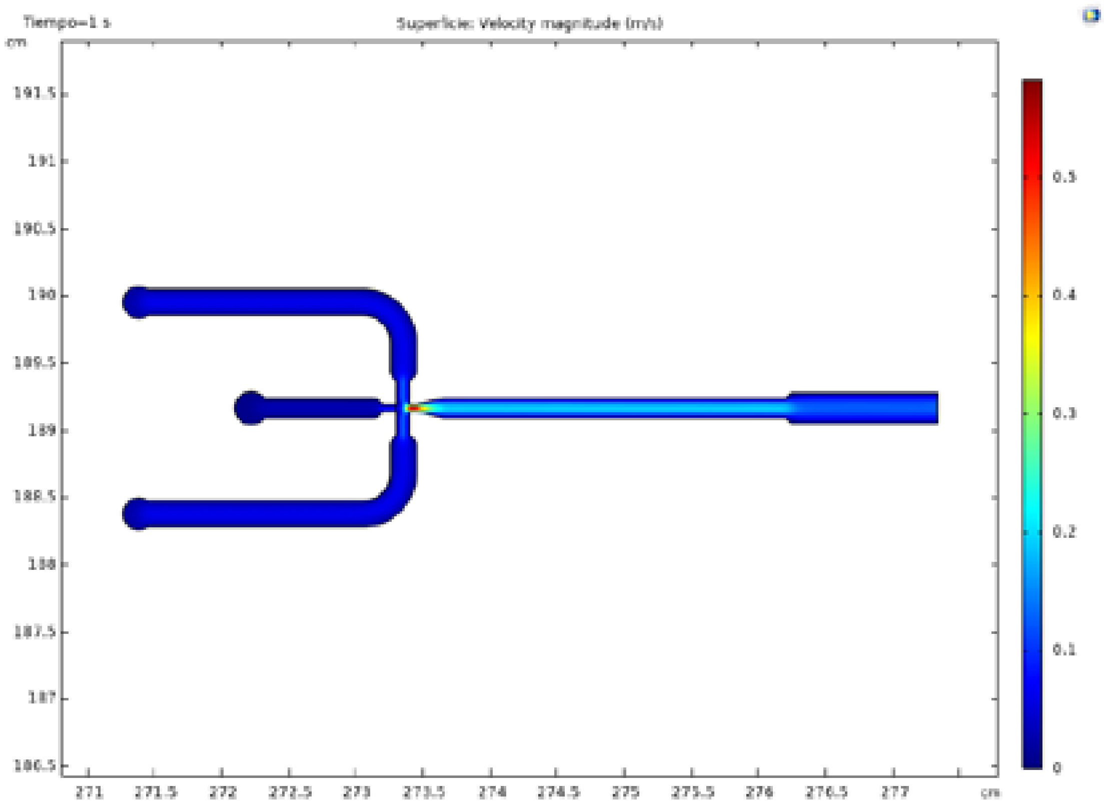
VELOCITY PROFILES





VELOCITY PROFILES





DIY LABS

ELECTRONICS

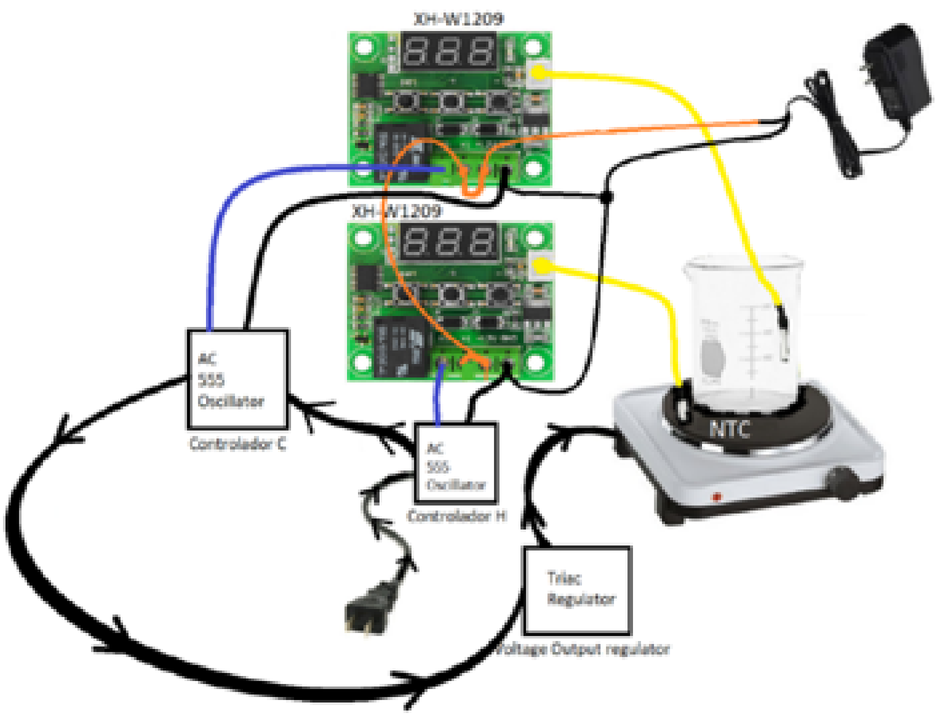
HEATING PLATE

A heating plate is a small, portable, and autonomous desktop appliance that has one or more electrical heating elements, and is used to heat containers with liquids, in a controlled way [1].

The methodology used for the development of the heating plate consists of the implementation of an ON-OFF controller, that is, a simple system which checks if the process variable is above or below a certain fixed point [2].

This implementation incorporates a feedback process, which allows to adjust the desired temperature of an electric stove and further adjust the precision given by the ON-OFF type controller. This implementation is carried out using two programmable digital thermostats that control the signal given by two astable oscillators, which control the work cycle of the electric stove. This implementation can also regulate the power used by the electric stove.

https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/Heating%20Plate



COMPONENT EXPLANATION

LM 555

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor [3].

The LM555 timer used in this implementation is used in an astable configuration, which allows for a square signal at the output. This square signal allows for the operation of relays that control the current flow to the electric stove [4].

XH - W1209

The implementation uses the XH-W1209 programmable digital thermostat which is a controller that allows a relay to be activated when it is below / above a desired temperature. The thermostat makes use of a 10K NTC thermistor (included with the thermostat) that allows to sense the current temperature, and thus, activate or deactivate its relay.

The thermostat works with a voltage of 12V, so it is recommended to use an AC-DC adapter of said voltage. When turned on, it shows the temperature currently sensed by the NTC. By pressing the SET button (left) the display flashes and shows the required temperature. Using the + (center) and - (right) buttons, the desired temperature is set.

Holding the SET button, the different settings of the digital thermostat appear. Using +, and - the user can go to the different settings, and by pressing SET they can be modified. When the desired setting is set, press SET again, and wait a couple of seconds to return to the current temperature. Here are all the thermostat settings:

P0 H / C
(Allows you to activate the relay when it is below the desired temperature / deactivate the relay when it is above the desired temperature)
P1 Backlash 0.1 -15
(It allows to adjust the trigger of the relay to the desired temperature + Backlash)
P2 Upper limit 110 110
(Sets the maximum measured temperature)
P3 Lower limit -50 -50
(Sets the minimum measured temperature)
P4 Correction -7.0 ~ 7.0
(Adjusts the resistance measured correction, adjust using the submersible thermometer)
P5 Delay start time 0 - 10
(Sets the relay trigger time in minutes)
P6 High temperature alarm 0 - 110
(Sets a maximum temperature, for the operation of the equipment, the equipment turns off when it reaches this limit, OFF default)

Pressing + and - for several seconds returns the values to their default values.

One of the XH-W1209 must be adjusted to P0 H, and another to P0 C. In this configuration, one triggers the relay when it is below the desired temperature, and the other triggers above the desired temperature. The correction values for P4 are adjusted using a submersible thermometer. The procedure for calibrating the values given by the thermostats is later explained.

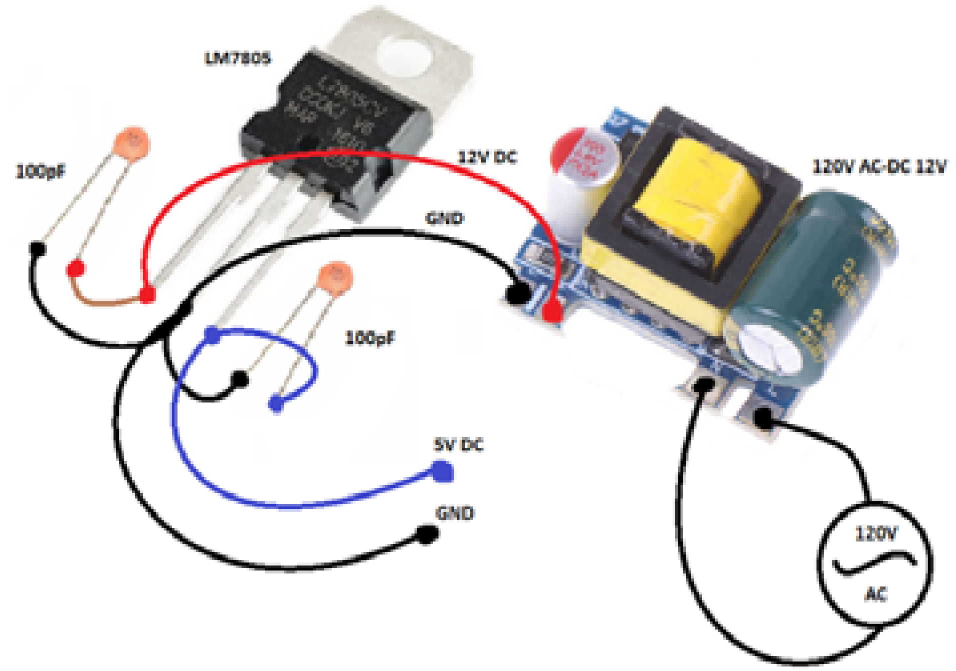
This link goes into more detail about the operation of the thermostat used for this implementation [https://www.youtube.com/watch?v=DBRI7ry0Ku8&ab_channel=Robojax]

OSCILLATOR H DEVELOPMENT

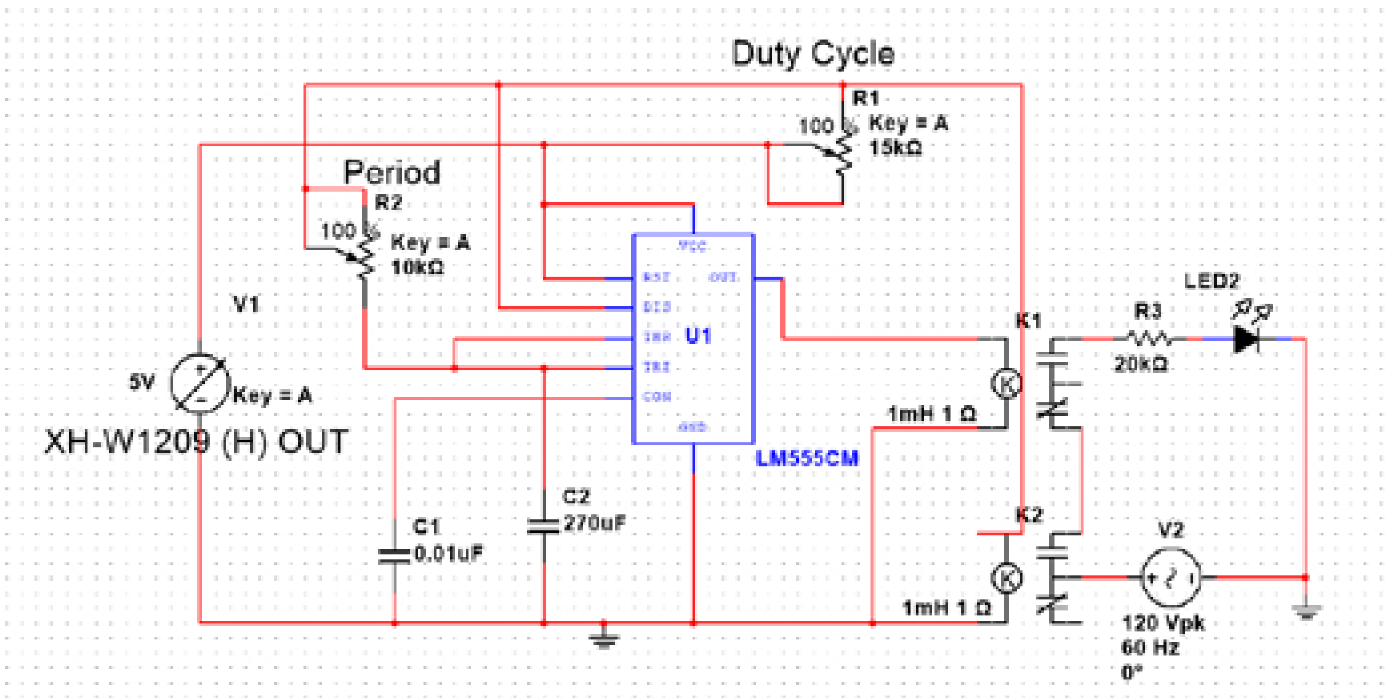
Oscillator H is implemented using a NE555 timer in astable configuration. This controller oversees regulating the flow of current to the electric stove using relays, which oscillate and allow the stove to be energized for a time T, which is defined by the timer's duty cycle. It should be taken into account that, in this case, the controller works with the inverse of the duty cycle. That is, for values that provide a 70% duty cycle, a 30% duty cycle will be used for the stove's current flow. In the same way, it must be taken into account that the operating voltage of the oscillator H is 5V; thus, the circuit requires an LM7805 regulator for its operation. Below, you can see both the materials for the development of the oscillator H, as well as the connections to obtain the 5V from the 12V AC-DC adapter:

| OSCILLATOR H DEVELOPMENT MATERIALS

- NE555
- 2x Cap. 10nF
- Cap. 270uF
- Cap. 1uF
- Trimmer 10kOhm
- Trimmer 50kOhm*
- 2X 5V Relay
- LM7805



In this astable configuration of the NE555 timer, the controller features a frequency of 0.153Hz through the K1 relay, with a duty cycle of 71.43%, so the current flow only reaches the electric stove for 1.871s. It is not recommended to modify the value given by the 'Period' trimmer as it may affect the device life of the electric stove. By modifying the 'Duty Cycle' trimmer, the LOW time of the controller remains the same, thus effectively modifying the duty cycle of the controller, making the stove always receive 1,871s of current flow. Relay K2 is activated whenever digital thermostat H triggers the relay. In the illustration above, the hot plate is represented using a 20kOhm resistor, followed by an LED. This setting does not yet take feedback from the system into account. The feedback is provided by oscillator C.
However, a simple implementation without feedback can be done by only using oscillator H.
*As with the other devices, the trimmers can be adjusted to meet the values in the images.



OSCILLATOR C DEVELOPMENT

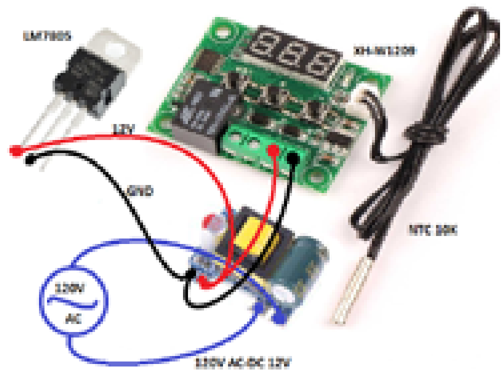
The implementation of this oscillator is virtually the same as that of oscillator H. However, the values are taken so that the operating frequency is 0.534Hz, that is 1.871s, which is the LOW operating time of oscillator H. This implementation has a 54% duty cycle, which serves to halve the flow of current that reaches the electric stove.

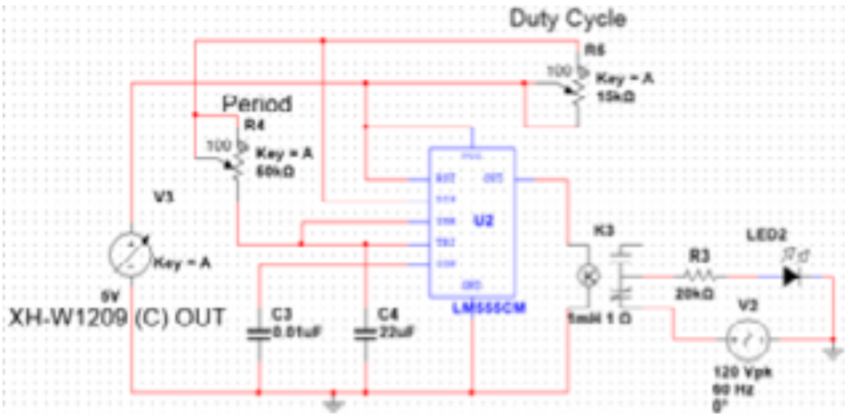
Materials:
- NE555
- 2x Cap. 10nF
- Cap. 22uF
- Cap. 1uF
- 2X Trimmer 50kOhm
- 5V Relay

In case of modifying the Duty Cycle trimmer of oscillator H, the operating frequency must be adjusted with the Period trimmer on oscillator C, so that they maintain the same operating time. The duty cycle can vary as much as 10%.

The feedback of the system is given by the oscillator C, which modifies the duty cycle of the oscillator H once the temperature exceeds a desired limit. This flattens the heating curve given by oscillator H. In the illustration above, the heating plate is represented using a 20kOhm resistor, followed by an LED.

*Unlike the illustration, the trimmer values must be at 100%.





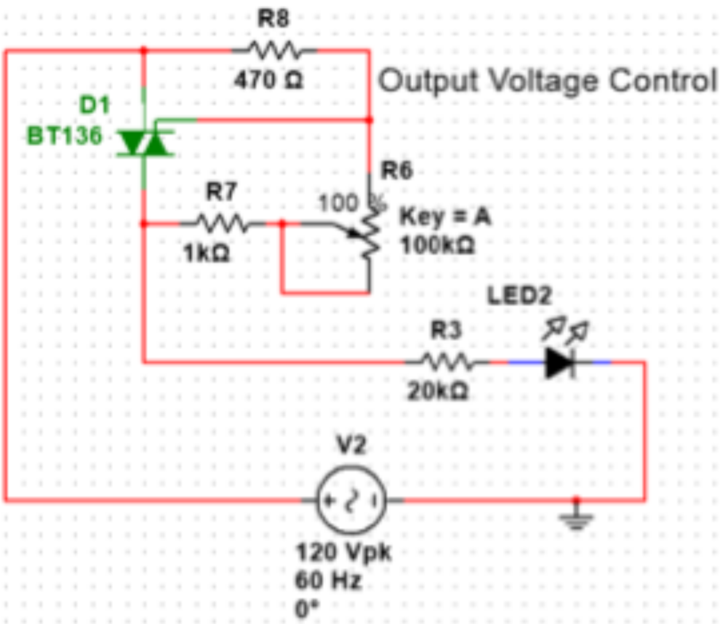
AC REGULATOR

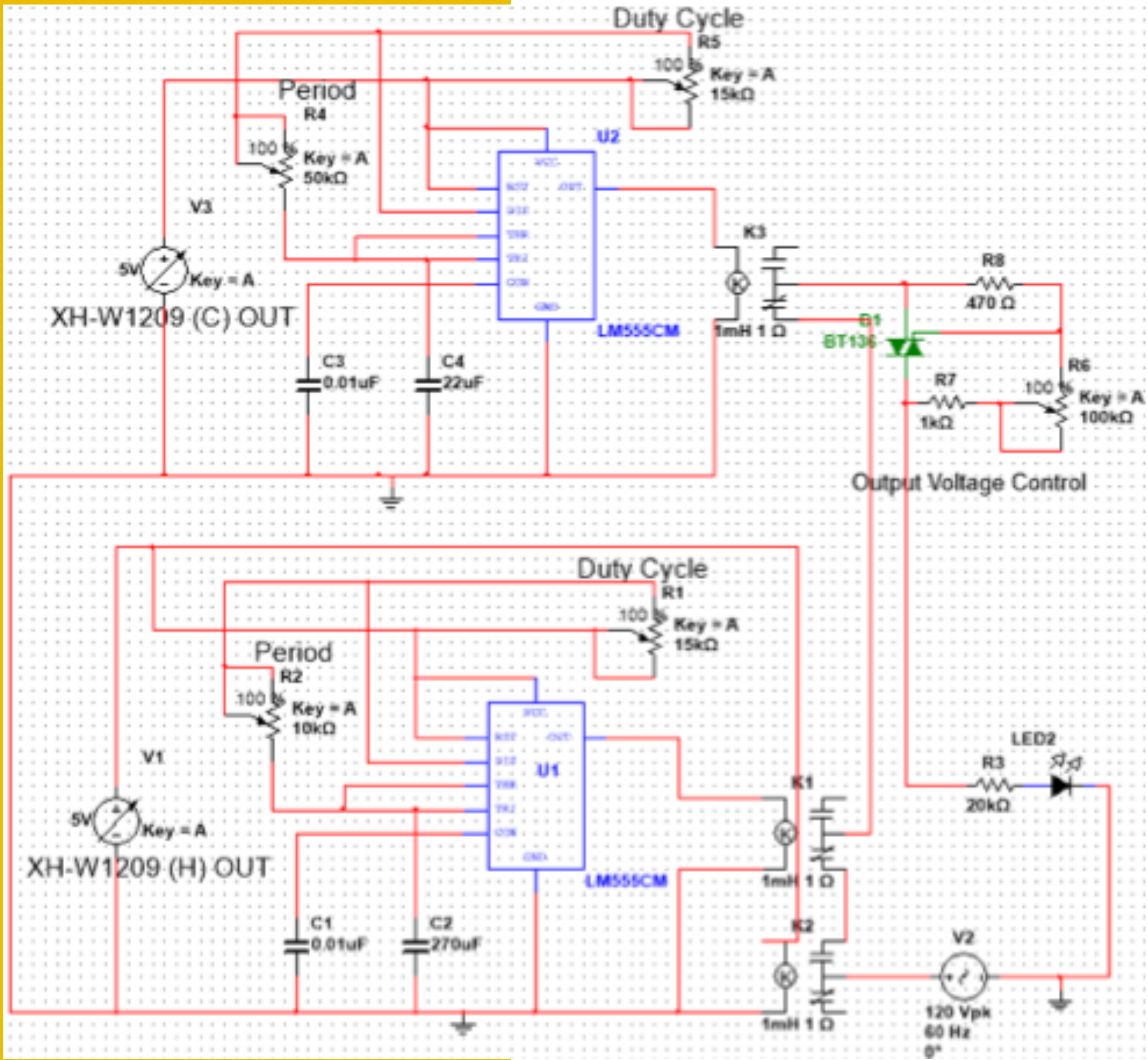
The voltage regulator implementation makes use of the BT136 Triac (an AC switch) and a potentiometer to regulate the output of the AC voltage. In this implementation, while having the potentiometer at 100% of its value, it is observed that all the current passes to the electric stove. By having the potentiometer at a value of 75%, a 50% decrease in the output voltage is observed, and for a 50% value of the potentiometer, a 100% decrease in the output voltage is observed. The behavior of the regulator can be observed in the following formula:

V(out)=
V(in)/(2*(100%-(Potentiometer operation %)))

It should be taken into account that the AC regulator can be placed before or after the H and C controllers shown above since the connection between them is in series. Similarly, the implementation of the AC regulator is optional, and does not have a great impact on the final implementation. That is, this regulator serves to have greater precision in adjusting the temperatures of the final implementation of the heating plate.

Materials:
- Triac BT136
- Res. 1kOhm
- Res. 470 ohm
- Pot. 100kOhm



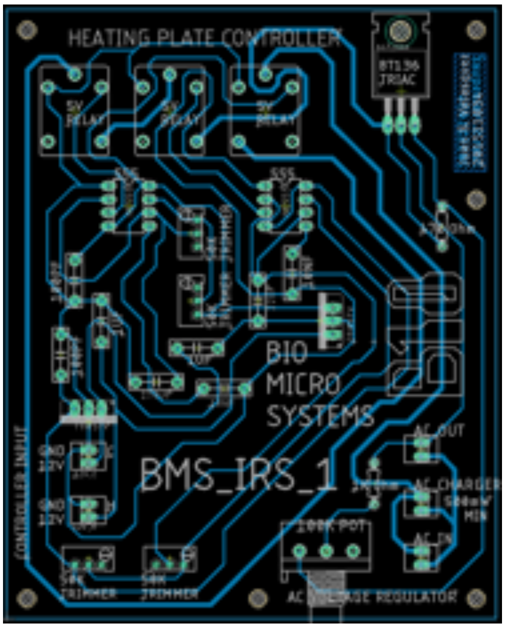


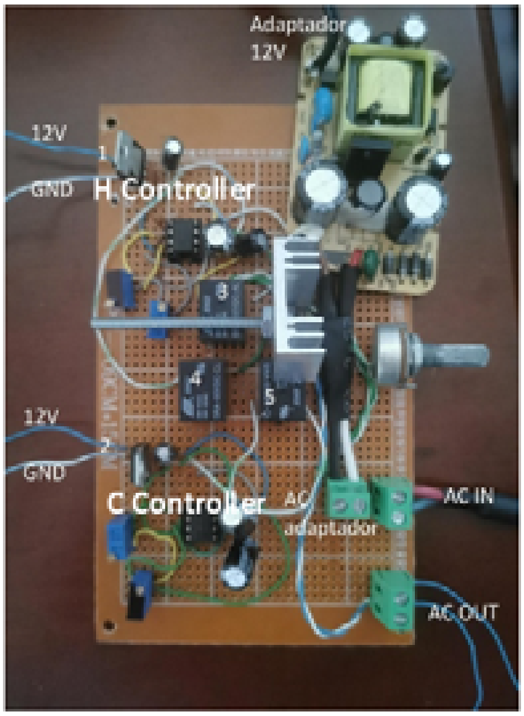
FINAL DEVICE

***It should be noted that the PCB generated using the previous diagram was manufactured using the laboratories at Universidad de Los Andes.**

Therefore, micrometer precision can be difficult to reproduce. It is recommended to take this into account before reproducing it, or generating a new PCB using the above diagram. The PCB can be seen on the left:

IMPLEMENTATION





For the development of the implementation, it is suggested to incorporate the AC-DC 12V adapter to the same connection of the electric stove. This, to have a single AC connection and simplify the integration of the different devices. Similarly, the AC-DC 12V adapter can be connected separately, keeping the 12V and ground connections on the corresponding inputs of the digital thermostats.

CONTROLLER OPERATION TEST

Next, an implementation made in a soldering board (8b) is presented to carry out a simple operation test. It should be noted that the final implementation is done using a PCB (8a). This implementation has the function of showing, through the sounds generated by the relays, the operation of the controllers.

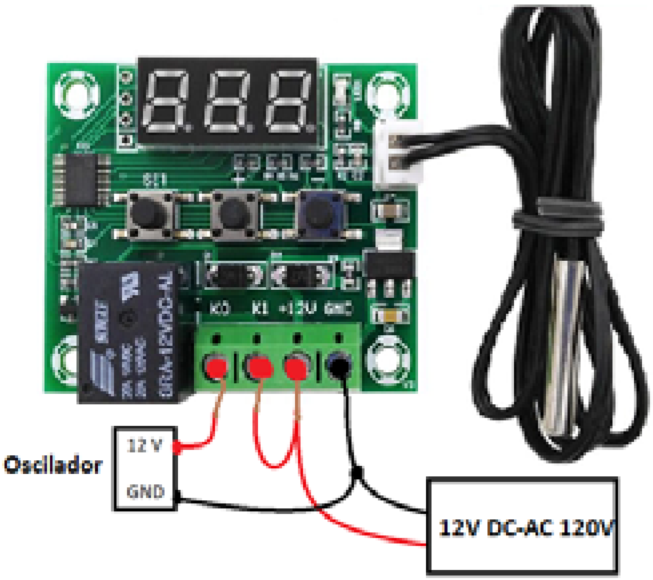
The correct operation of the relays can be observed in the following link, in which the connection (1) is responsible for the operation of the relays (3) and (4). The first relay of the oscillator H (3) must be triggered when connecting the source, and the second relay (4) must be triggered every 1.8s-4.7s.

The connection (2) is responsible for the operation of the relay (5). Oscillator C relay must be triggered every 900ms-1s. The link shows the operation of the device with both connections (1) and (2) connected. [https://youtu.be/ANu1v0kD7Ro]

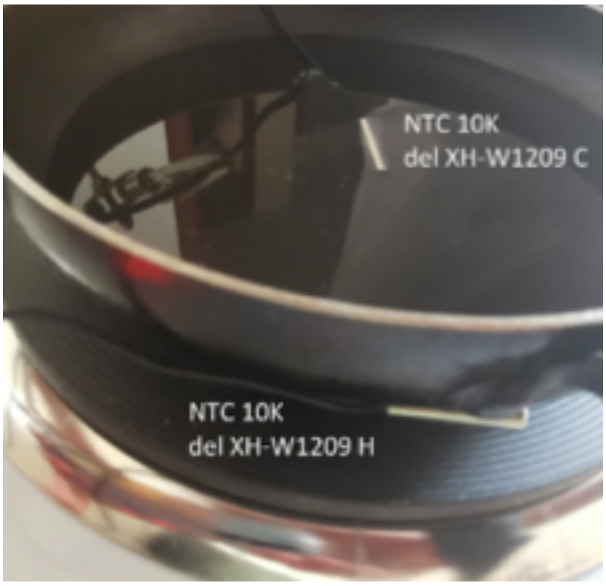
In the annexes, you can see a higher resolution image of the PCB generated for this implementation, as well as the finished product soldering the components onto the proposed PCB.

TEMPERATURE MEASUREMENT

For the correct sensing of the temperature of the device, both 10K NTCs are connected to their respective XH-W1209. Below is a connection diagram for the digital thermostat used in the H and C controllers. The voltage is regulated using the 12V AC-DC adapter. It must be taken into account that the 12V voltage that feeds both the thermostat, and the oscillators must pass through the LM7805 5V regulator to energize the oscillators.



To make the correct feedback through the XH-W1209 digital thermostats, the NTC 10K of the XH-W1209 H is placed on the electric stove, this to regulate the flow of current that reaches the stove. To improve thermal conduction, thermal paste can be used, which also improves the bond between the NTC and the stove. In the same way, the NTC 10K of the XH-W1209 C is placed inside the bath, to observe and regulate the circuit based on the bath's temperature. Controller C, feeds back the current flow given by controller H.



*The image above shows the thermal bath using a frying pan; however, glassware would provide the same results. For this implementation, a bath of approximately 100mL was used.

The setting of the temperature of the digital thermostat H is the value of the desired temperature, that is, the temperature that is desired for the thermostated bath. While the temperature setting of the digital thermostat C must be approximately 5°C lower than that of the XH-W1209 H for the controller to give the correct feedback. To start the calibration, the value of the voltage regulator is kept at 100%, that is, all the current flow reaches the stove.

CALIBRATION

It should be taken into account that the temperature of the electric stove can vary ± 1.2°C* since the temperature control system (taking into account the feedback) is based on an ON-OFF controller. Therefore, if higher precision is required, it is preferable to develop a PID-type controller.

*Value obtained for a Duty Cycle of 75% at 35°C

For device calibration, turn on the device and adjust the temperatures of both XH-W1209 as stated above. Once the temperature of the electric stove stabilizes, the submersible thermometer is used to make a precise measurement of the bath temperature. It is recommended to place the desired temperature between 35°C and 40°C since commercial submersible thermometers have more accurate measurement values for this temperature range. For this implementation, the thermometer used has an accuracy of ± 0.2°C in this range.

Using the P4 setting of the XH-W1209 thermostats, a correction of ± 7.0°C every 0.1°C can be made until the temperature given by the submersible thermometer is reached.

If a temperature calibration or correction is required on the XH-W1209 H, an infrared thermometer should be used at the maximum suggested distance for correct measurements (usually 15cm), at an angle of 45°. This measurement is made by pointing the thermometer directly at the electric stove.

Thermometer Data

SUNPHOR BZ-R6	Min	Max
Measurement range for minimum uncertainty °C	36	39
uncertainty °C	0,2	0,2
Measurement distance (cm)	5	15

AlfaSafe DT-11F	Min	Max
Measurement range for minimum uncertainty °C	35,5	42
uncertainty °C	0,1	0,1

Measurement Data °C

	BZ-R6*	DF-11F**
Subject1	36	36,3
Subject2	36,5	36,6
Subject3	36,4	36,4

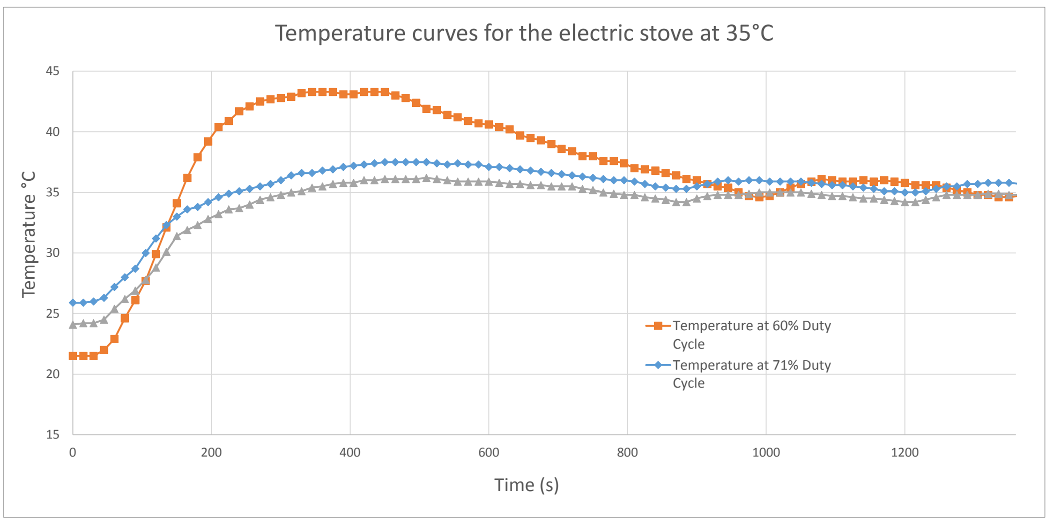
*Measurements done at 5cm

**All measurements done in the armpit

Once the XH-W1209 thermostats have been calibrated for 35°C, the value of the Duty Cycle trimmer of oscillator H is adjusted to observe the behavior of the heat curve with different percentages of the duty cycle. This to observe a curve that presents a stable behavior. For this implementation, the value of the voltage regulator is kept at 100%. The different temperature curves for the duty cycle settings are shown below:

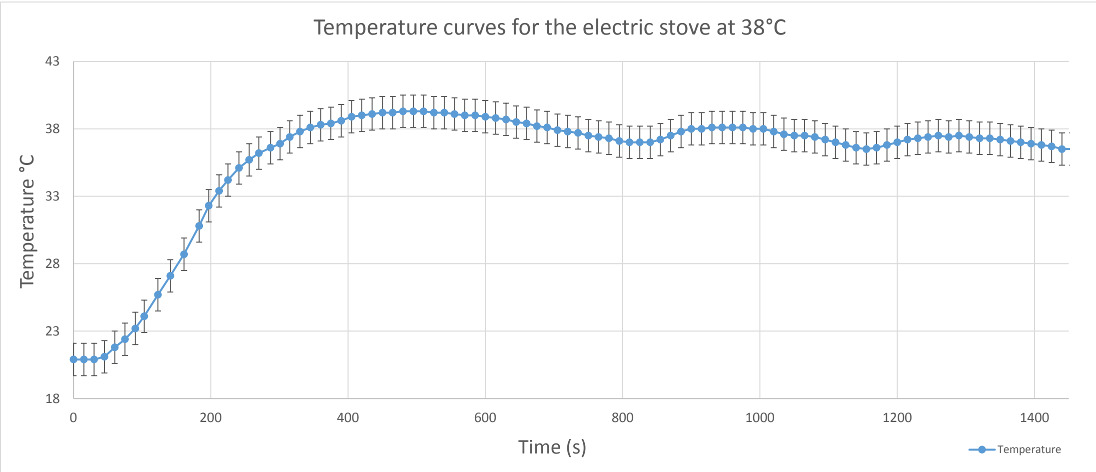
*The values of the percentages of the duty cycle are taken taking into account the operating times of the relays: link Duty Cycle 60% [https://youtu.be/cjP7TYS4oLI]; link Duty Cycle 71% [https://youtu.be/mK7zi6v_CN8]; link Duty Cycle 75% [https://youtu.be/RwB-2JGHj_3M]

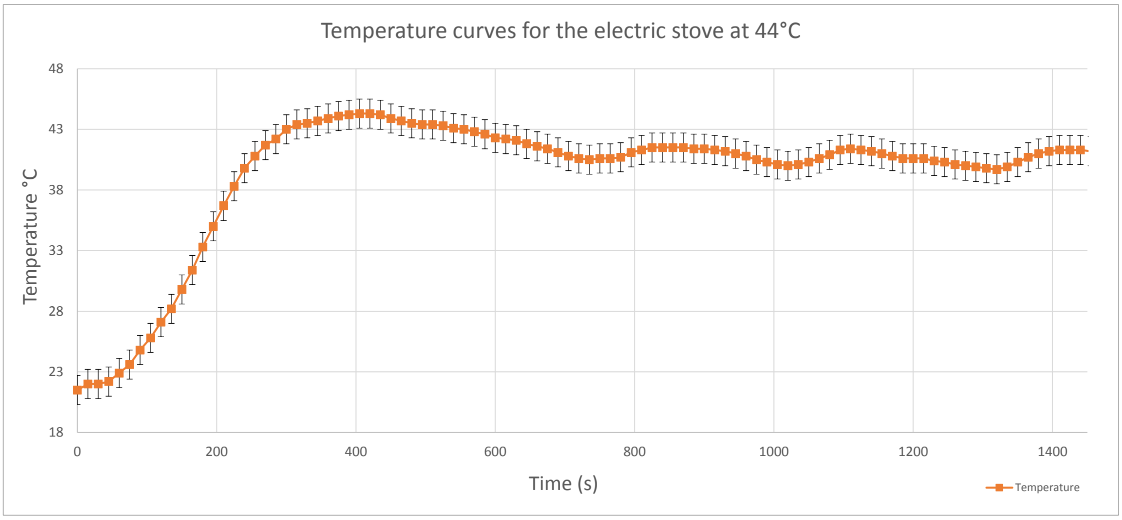
Using the trimmer setting for a 75% duty cycle, different temperature curves are taken to observe how the specific operating values for 35°C affect the other temperatures, and thus have a calibration curve for a range of higher temperatures. The curve with 75% duty cycle is used as it shows a more stable behavior. Similarly, using the temperature curve of 75% at 35°C, the calibration uncertainty of the device is generated (± 1.2°C, Standard Deviation = 0.588°C).

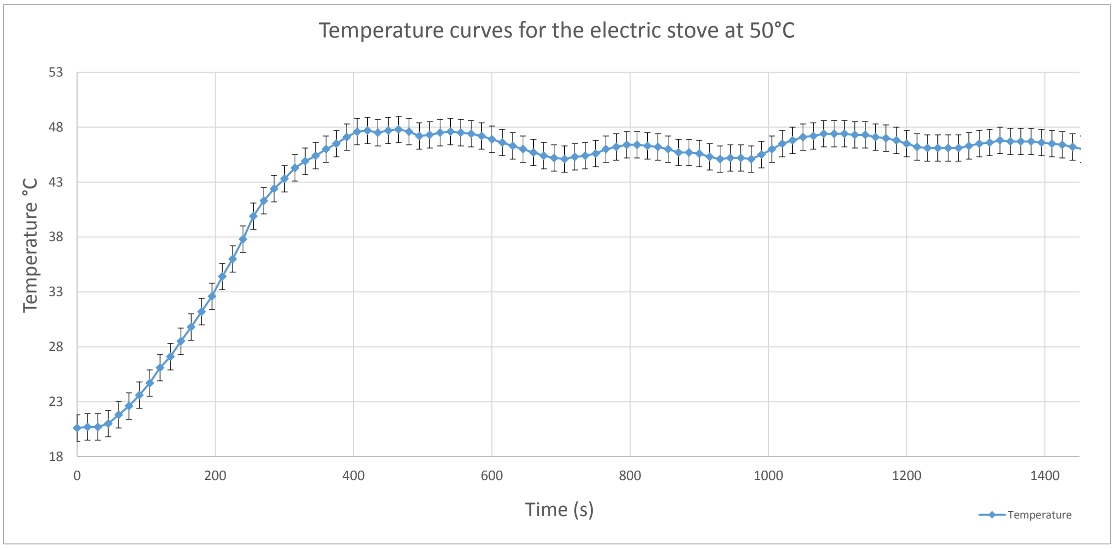


TEMPERATURE CURVES

Data was taken for temperatures of 38°C, 44°C and 50°C. That is, the temperature set on the XH-W1209 H thermostat is 38°C, 44°C and 50°C respectively, while the temperature set on the XH-W1209 C is 33°C, 39°C, and 45°C respectively for system's feedback. The data collection can be observed in the following images and in the following links:
Link curve 38°C [https://youtu.be/ohRhE2VJytl];
Link curve 44°C [https://youtu.be/vcO8OdUfcsQ];
Link curve 50°C [https://youtu.be/VLtTLCZcf1g]



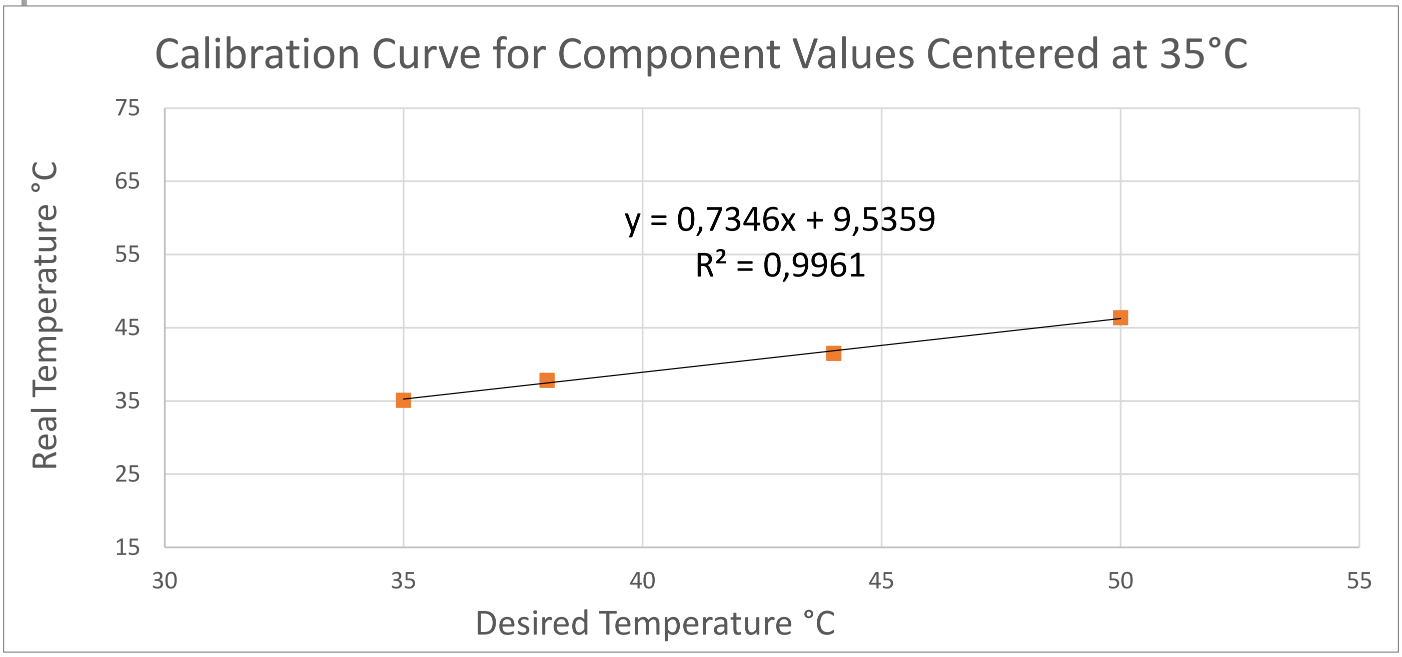




LINEAR REGRESSION

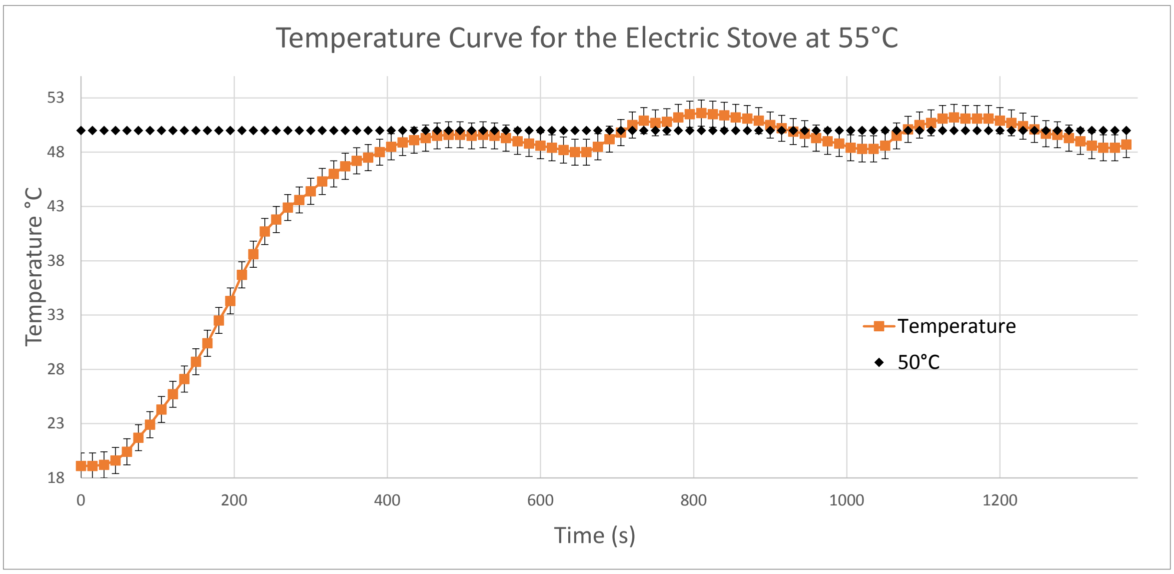
By getting an average temperature from minute 5 to the end of the temperature graph, the actual temperatures are obtained, based on the desired temperatures for the operating values for 35°C to create a temperature curve for the other values.

Desired Temperature H controller	Feedback Temperature C controller	Actual Temperature	Standard Deviation
35°C	30°C	35,089°C	0.588°C
38°C	33°C	37,809°C	0.820°C
44°C	39°C	41,528°C	1.287°C
50°C	45°C	46,389°C	0.777°C



To verify the linear regression obtained with the calibration data, the equation obtained is used. Since for this implementation the maximum temperature is 50°C, this value is placed in the equation to obtain the temperature of the H and C controllers.

(50°C - 9.5359)/0.7346=Controller H Temp.; CHT= 55.083°C



When adjusting the controllers (controller H 55°C, controller C 50°C) with the equation obtained using the linear regression (2) of the temperature curves with the trimmer settings for 35°C, the desired temperature is effectively obtained. The temperature curve can be observed in the 55°C curve link [https://youtu.be/MEb-PMuGhH5s]. Similarly, uncertainties of ± 1.2°C corresponding to the 35°C calibration curve are placed.

Using the same procedure as the previous temperature curves for the average temperature, a value of 49,494°C is obtained, which is within the uncertainty values obtained for the 35°C calibration curve. Using the data from minute 5 to the end of the measurement, a standard deviation of 1.33°C is also obtained.

Once the linear regression equation has been verified, an equation for the standard deviation with respect to the set temperature (35°C) and the current temperature of the system can be obtained. However, since the device is based on an ON-OFF controller, the standard deviation has a large degree of variance with respect to the expected values.

$$\textbf{Standard Deviation} = 0{,}0353x - 0{,}5228$$

$$R^2 = 0.4032$$

RECOMMENDATIONS

For the first operation, it is recommended to place a temperature with a value 1°C above the desired one in the thermostats H and C. This is because the charging time of the controllers' capacitors can impact the time response time, especially the response time of controller C (feedback process). Once it is observed that the temperature has begun to stabilize, the temperature value of the controllers can be modified to the expected values.

For a more precise measurement of both the temperature of the electric stove and the bath, it is recommended to set/glue the NTC 10K to the stove to avoid movements or oscillations caused by the environment.

OBSERVATIONS

Compared to some of the commercial hotplates, the developed device manages to have comparable uncertainty values for the measured temperatures. Thus, with proper calibration the device can be used in a laboratory environment.

Although the controllers have trimmers for both the duty cycle adjustment and the operating frequency, in this implementation only the duty cycle trimmers were used to adjust the temperature curve for 35°C. Using the other trimmers, it is possible to have a better temperature curve, which can have a better response to changes in the desired temperature of the oscillator H. Similarly, although the device has an AC voltage regulator at the output of oscillator C (feedback), for this implementation the value was not modified. That is, the voltage was maintained at 100% of the input value. By making modifications to this regulator, better temperature curves could be obtained that more closely match the desired temperatures.

It should be noted that all the temperature curves were made using a bath of approximately 100mL.

The measurements and temperature curves of this implementation were only performed for values in the range of ambient temperature up to 50°C. For higher temperature implementation, it is recommended to perform the initial calibration at a temperature close to that desired for device operation.

REFERENCES

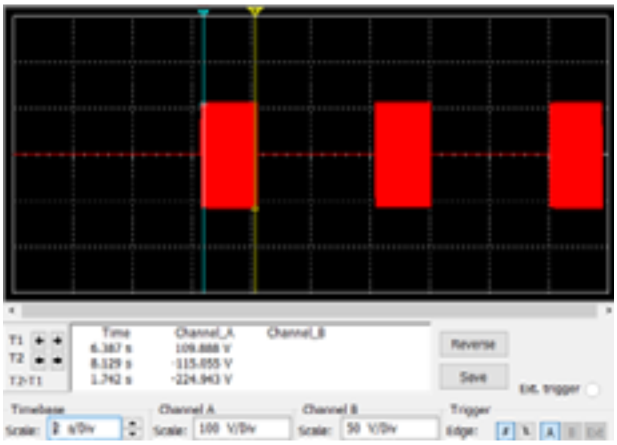
1. Wikimedia Foundation. (2021, August 13). Hot plate. Wikipedia. Retrieved September 9, 2021, from https://en.wikipedia.org/wiki/Hot_plate.
2. Villajulca, J. C. (2020, June 18). Control on/off o TODO/NADA. Control ON/OFF o Todo/Nada. Retrieved September 9, 2021, from https://instrumentacionycontrol.net/control-on-off-o-todo-nada/.
3. TI. (2015, January). LM555 Timer. Texas Instruments. Retrieved September 2021, from https://www.ti.com/lit/ds/symlink/lm555.pdf.
4. Gómez, E. (2021, May 6). 555 Astable, ¿QUÉ ES? ¿CÓMO CONFIGURAR? Rincón Ingenieril. Retrieved September 9, 2021, from https://www.rinconingenieril.es/555-astable/.

SHOPPING ITEMS LIST

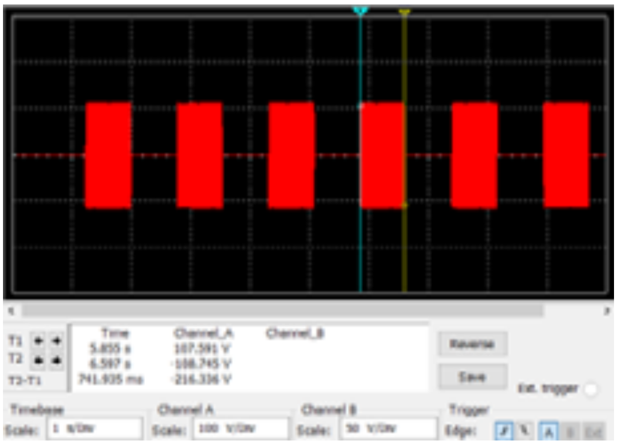
1. https://articulo.mercadolibre.com.co/MCO-562121780-estufa-electrica-un-puesto-1000w-_JM?quantity=1&variation=55608897828
2. https://articulo.mercadolibre.com.co/MCO-454743554-termostato-digital-xh-w1209-_JM?quantity=1#position=1&type=item&tracking_id=00422020-dc94-4206-a93b-18d22b06d4a2
3. https://articulo.mercadolibre.com.co/MCO-469275794-fuente-adaptador-12v-1a-1000ma-arduino-ac-dc-110v-ac-power-_JM?quantity=1#position=4&type=item&tracking_id=4ba5fac1-2e1f-4a40-bf10-89d48b3461f4

ANEXXES

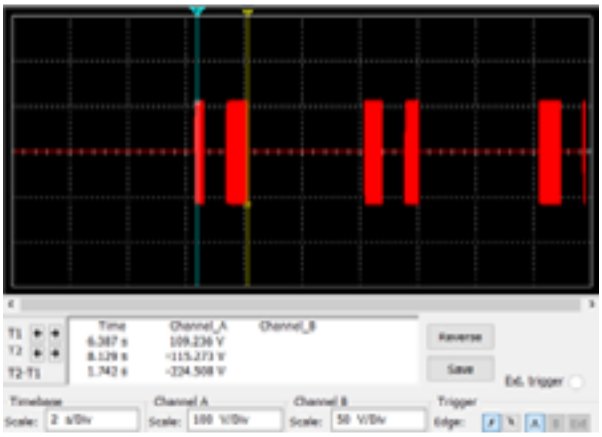
Controller H Duty Cycle 75%



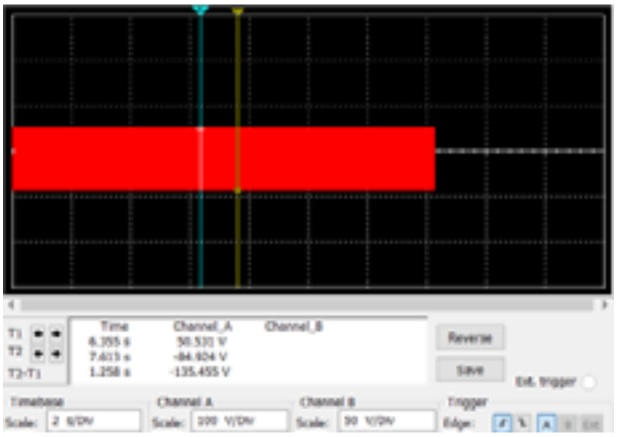
Controller H Duty Cycle 54%

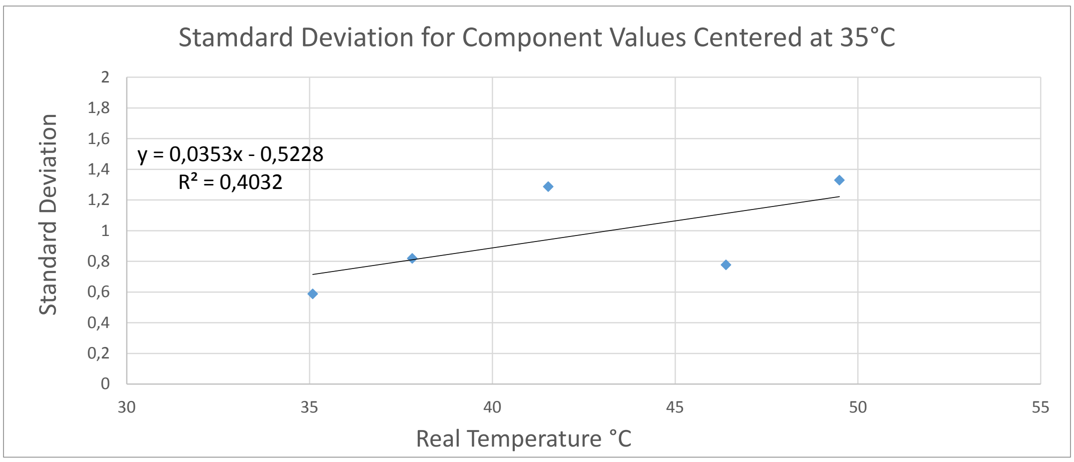


Controller H with Controller C

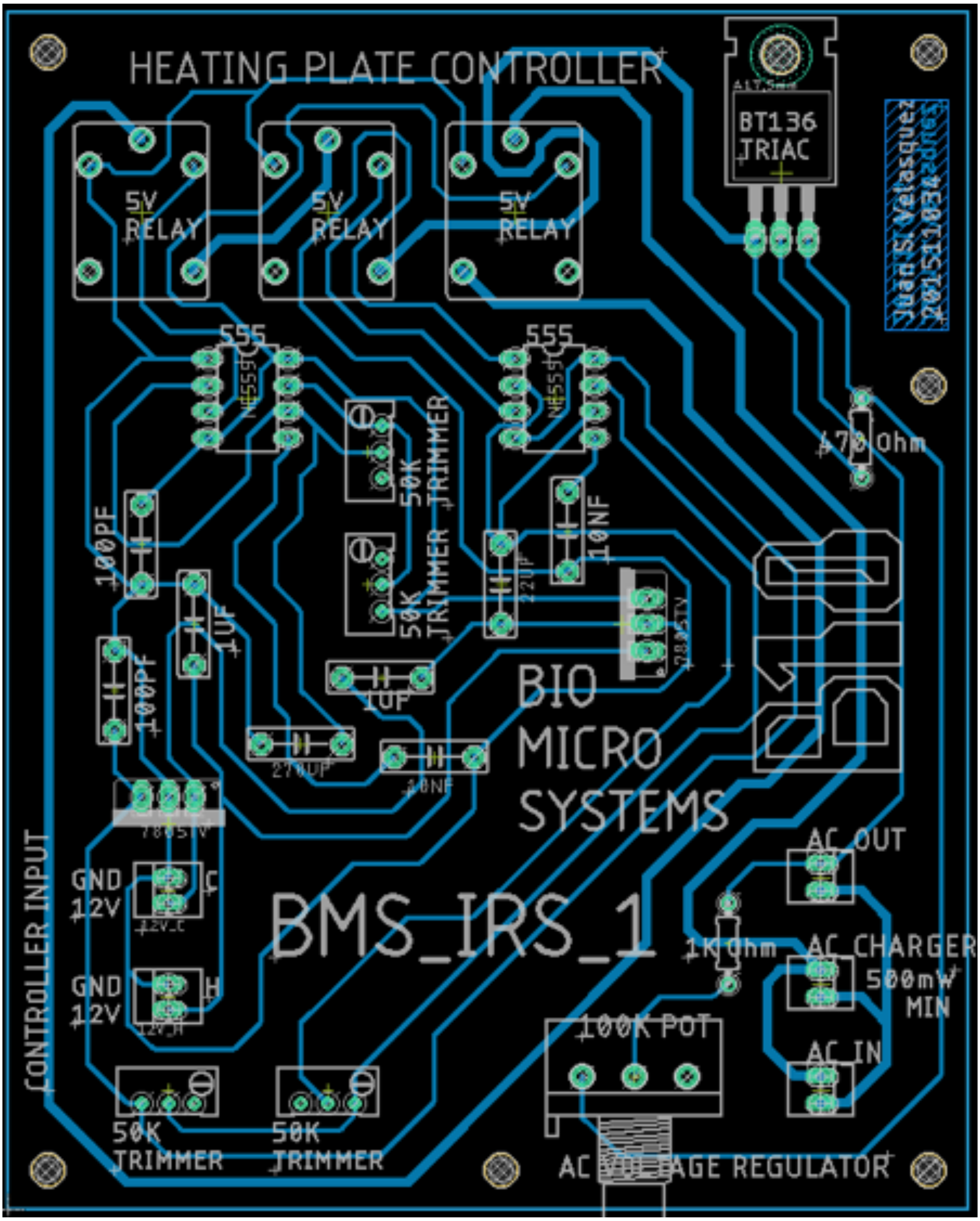


Voltage Regulator at 75%

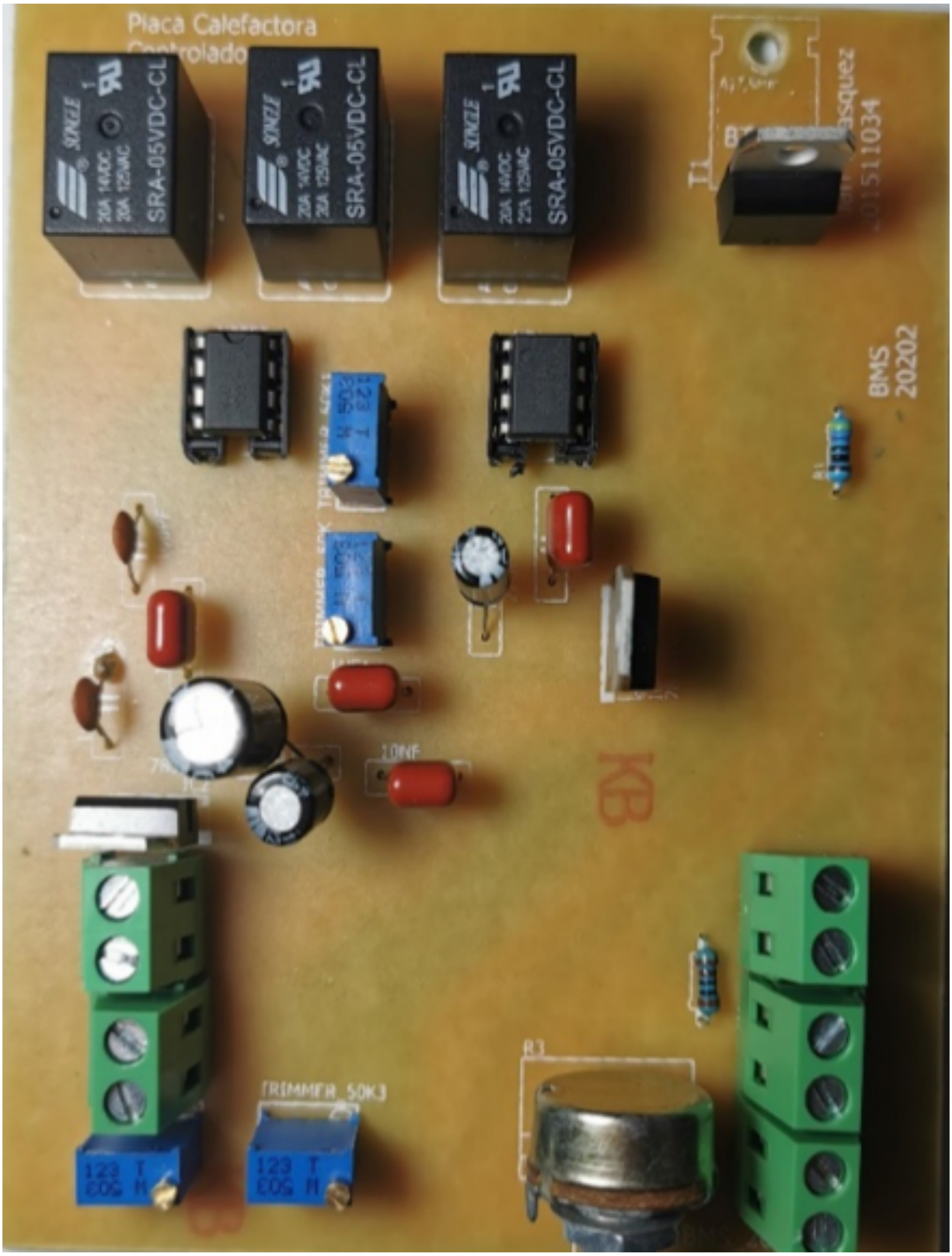




PCB DESIGN


HEATING PLATE CONTROLLER
5V
RELAY
5V
RELAY
5V
RELAY
BT136
TRIAC
Juan S. Velasquez
201511034
555
555
470 Ohm
100PF
1UF
50K TRIMMER
50K TRIMMER
10NF
22UF
7805TV
100PF
1UF
270UF
10NF
BIO
MICRO
SYSTEMS
7805TV
CONTROLLER INPUT
GND
12V
C
GND
12V
H
BMS_IRS_1
1K Ohm
AC OUT
AC CHARGER
500mW
MIN
AC IN
100K POT
50K
TRIMMER
50K
TRIMMER
AC VOLTAGE REGULATOR

PCB DESIGN


Placa Calefactora
SRA-05VDC-CL
SRA-05VDC-CL
SRA-05VDC-CL
BMS
20202
10NF
KB
TRIMMER_50K3

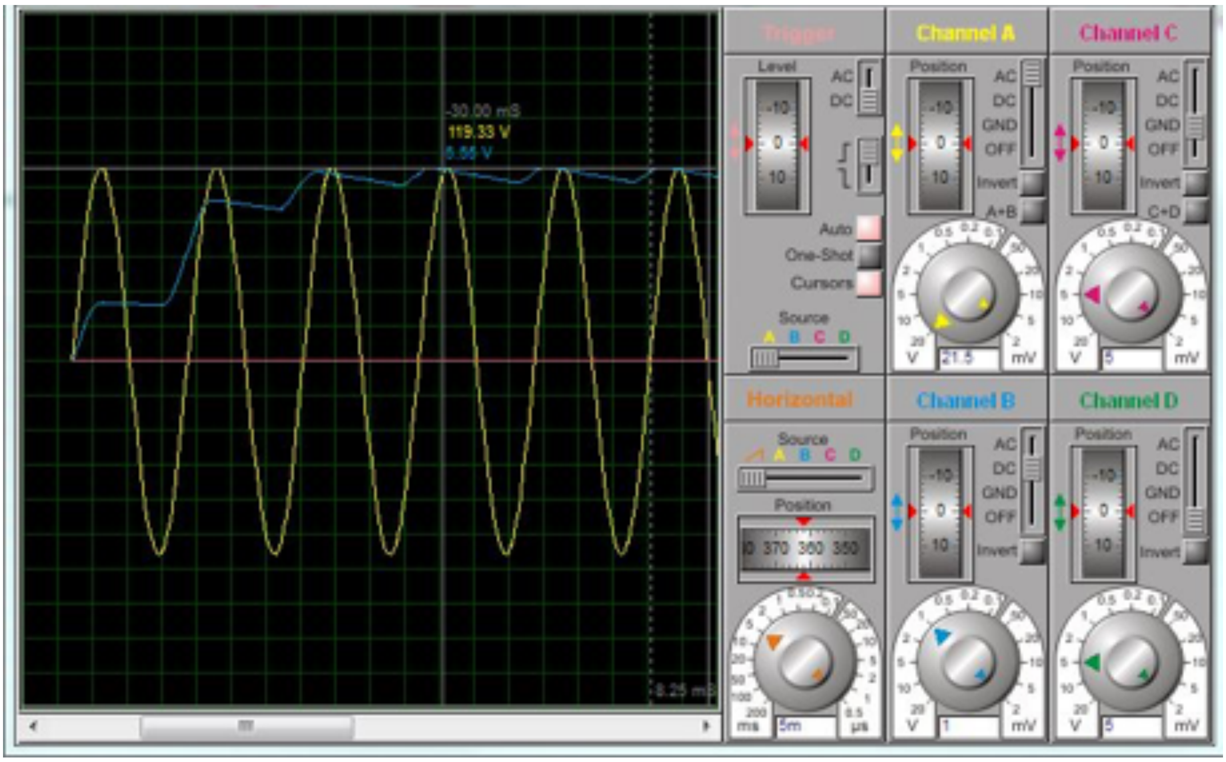
IMPEDANCE METER

An impedance meter is a device that allows for the measurements of impedance or resistance to the flow of an alternating current. An impedance meter designed for low magnitudes, allows for the measurements of impedance in chemical reactions or when working with enzymes and molecules.

The methodology used for the implementation of the low impedance meter consists of a power stage, a signal generation stage as well as a sensing and communication stage. The power stage implemented allows for the use of AC current as the power for both an arduino microcontroller as well as an HC-05 BT module to send the impedance sensed to a mobile device, or to the Arduino IDE monitor, using 2x Hi-Link ultra-compact power modules. These power modules allow for a constant output of 5V DC at 0.6A. The impedance meter for low magnitudes also features an LM317T voltage regulator as part of the signal generation stage, to adjust and amplify the voltage response from the impedance being measured. This implementation allows for the accurate measurements of impedances in the range of ones and tens of Ohms, although further tuning may be required. Another approach for the power stage is also proposed using an AC to DC capacitive transformerless configuration in case that the Hi-Link power modules are to not be used.

https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/Impedance%20Meter

01. Materials
02. Component explanation
03. Final Device Diagram
04. Implementation
05. Calibration
06. Recomendations and observations
07. References
08. Anexxes



COMPONENT EXPLANATION

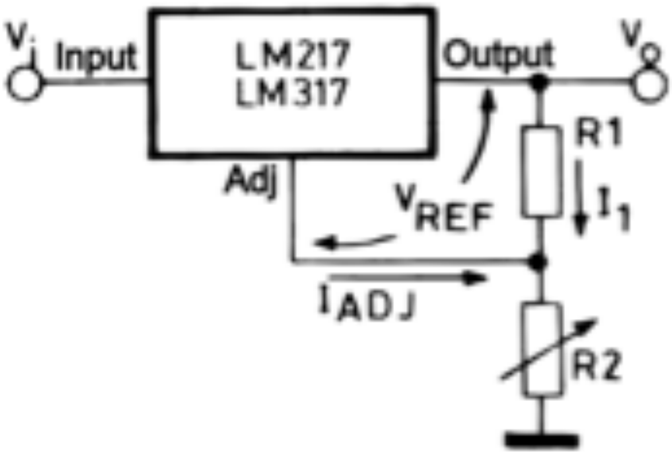
Hi-Link HLK-PM01

The 3W (5V DC, 600mA) ultra-small series module power supply is a small-volume, high-efficiency AC DC power module, which has the advantages of global input voltage range, low temperature rise, low power consumption, high efficiency, and high reliability [2]. The HLK-PM01 can operate with input voltages in the range 90-245VAC with an efficiency of 72%. If higher efficiency is needed out of the power module, the Hi-Link power module with reference HLK-20M05 features a 5V DC output and 20W. The operation voltage is the same as that of the HLK-PM01, however, it has an 80% efficiency. Furthermore, for this implementation, the HLK-PM01 3W power module is more than enough to power the signal generation stage as well as the sensing and communication stage.

LM317T

The LM317 is an adjustable linear voltage regulator capable of supplying at its output under normal conditions a range from 1.2 to 37 Volts and a current of 1.5 A. In its smallest configuration, a pair of resistors is enough to obtain the required voltage [3]. A basic adjustable regulator using the LM317T uses the following formula to adjust the output voltage, as can be observed in the following illustration

Vo= Vref(1+R2/R1)+(Iadj*R2)

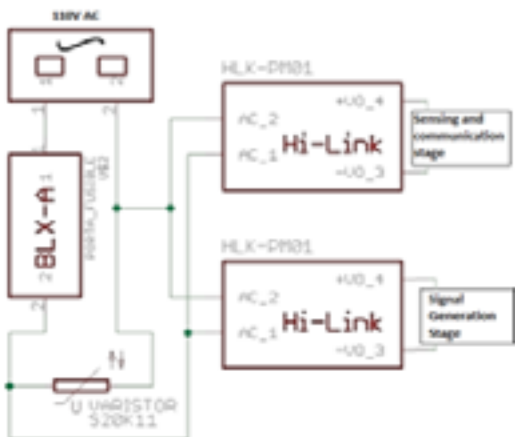


POWER STAGE

Since the device is to power a microcontroller and a BT module through AC-DC converters, it is necessary to implement both a fuse as well as a varistor to protect the device from power surges. The circuit has a BLX-A fuse holder for crystal fuses with a maximum amperage rating of 9 amps. However, a fuse of about 2-3 amps should suffice for this implementation.

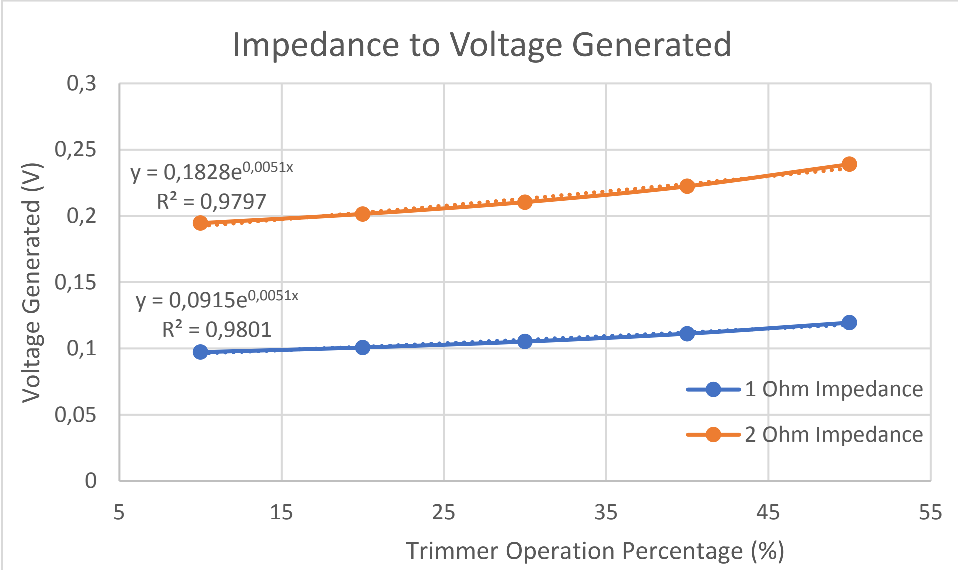
The varistor used is the S20K11, which has a maximum power rating of 10W. When a high voltage surge is applied, which surpasses the varistor voltage, the varistor suppresses the voltage to protect the circuit. It protects the circuit by lowering the resistance value as the voltage input rating increase; as such, for higher voltages, the resistance of the varistor lowers and resembles a short circuit. The S20K11 varistor used for this implementation has a maximum voltage rating of 11VAC as well as 14VDC. However, while the voltage input rating is under the specified threshold, the varistor works as a capacitor, or rather a filter to diminish high frequencies [4]. This behavior also helps protect the circuit from electronic noise generated by other components.

Now that the circuit is protected, two Hi-Link HLK-PM01 ultra-compact power modules are used to power each of the stages: the signal generation stage as well as the sensing and communication stage. A diagram of the power stage is shown in the following image. In case that the Hi-Link modules are to not be used, an implementation using a capacitive transformerless power supply is also shown in a later section.



SIGNAL GENERATION STAGE

The signal generation stage allows for the sensing of low impedances by converting the impedance in the sensing module to a voltage which can be then measured using the Arduino Nano microcontroller. This process is done by modifying the parameters of equation (1) once the impedance is connected to the impedance sensing module.

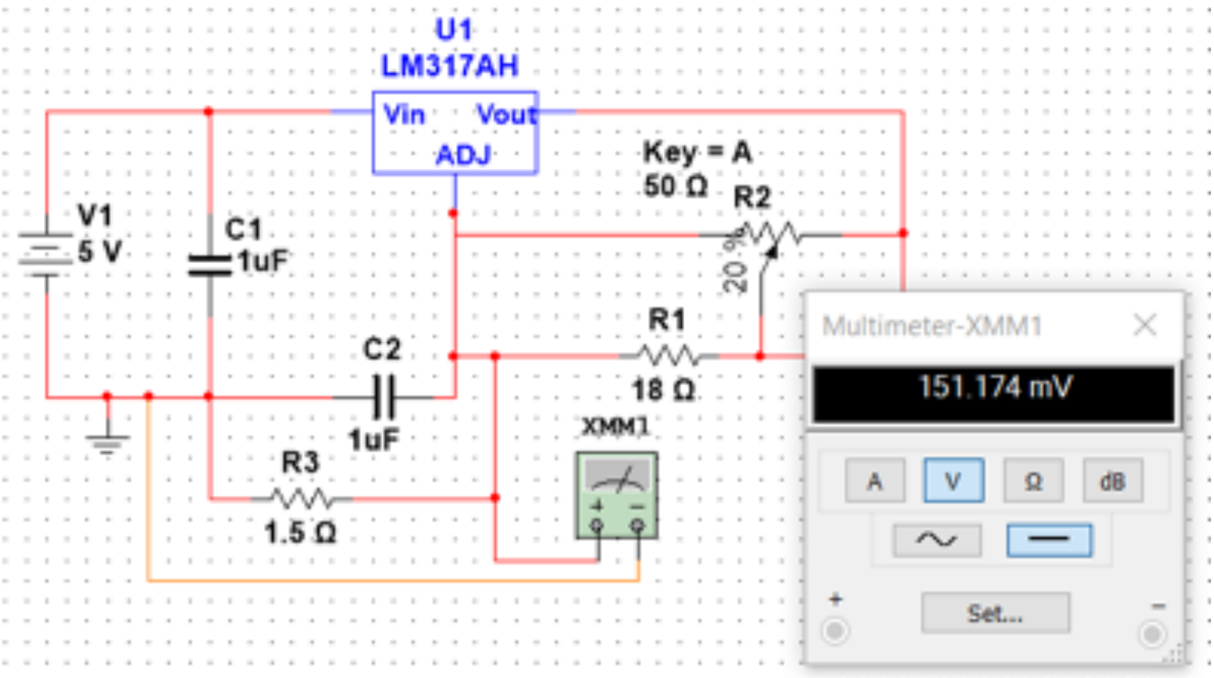


As such, the output value Vo of the LM317T is modified, which then allows for a voltage response to an impedance input. The LM317T generates a constant current I, since the 18 Ohm resistor's value does not change. Using V = I*R, the microcontroller is able to properly measure the device using the ADC.

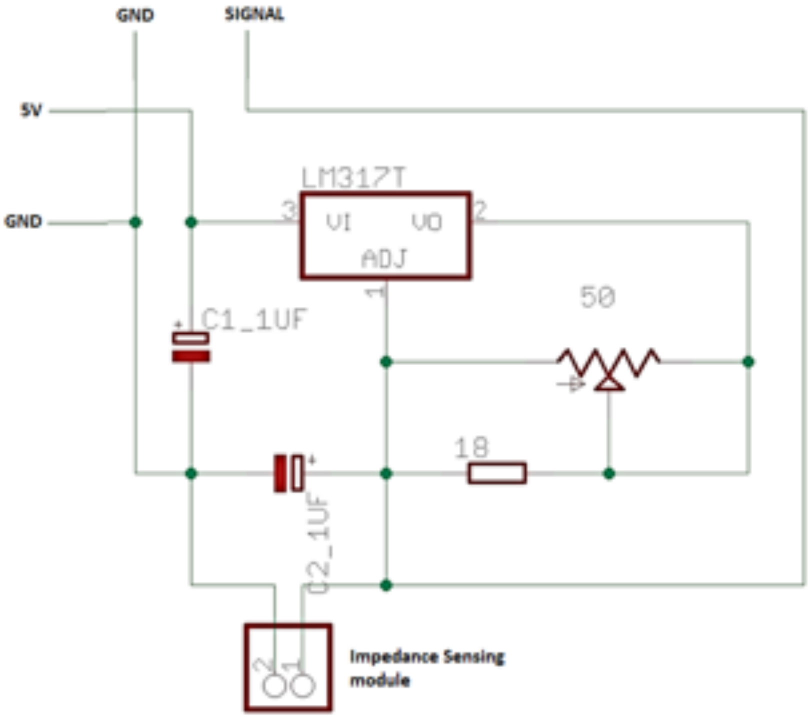
To properly adjust the voltage caused by the impedance sensing module, an LM317T voltage regulator is used. The voltage of the signal generated by the impedance can be observed in the following equation as well as in the following graph. The equation is obtained by taking into account values for the trimmer operation percentage from 0 to 50%. It should be noted that the following equation was created by taking into account impedance input values of 1 Ohm and 2 Ohm as the implementation was designed for values in this range. It should be noted that the implementation can take into account values of up to tens of Ohms, although further calibration may be needed. The following image shows the data obtained using the signal generation stage to create the exponential regression (2). The trimmer operation was kept below 50% since it also affects the voltage output of the LM317T in an exponential manner. As such lower trimmer operation values would provide a more linear regression equation, which in turn would make the signal generation regression easier to calculate.

$$\mathbf{V(signal)= (91.5mV*Impedance(Z))\ e^{(0.0051\ *\ (Trimmer\ Operation\ \%))}}$$

The following image shows the signal generated by using a 1.5 Ohm impedance at 20% trimmer operation. When the simulation is compared with equation (2), the is an absolute error of 0.000814 or 0.5%. It should be noted that values below the 1 Ohm and above the 2 Ohm threshold would have larger error values.



The diagram of the signal generation stage is shown below.

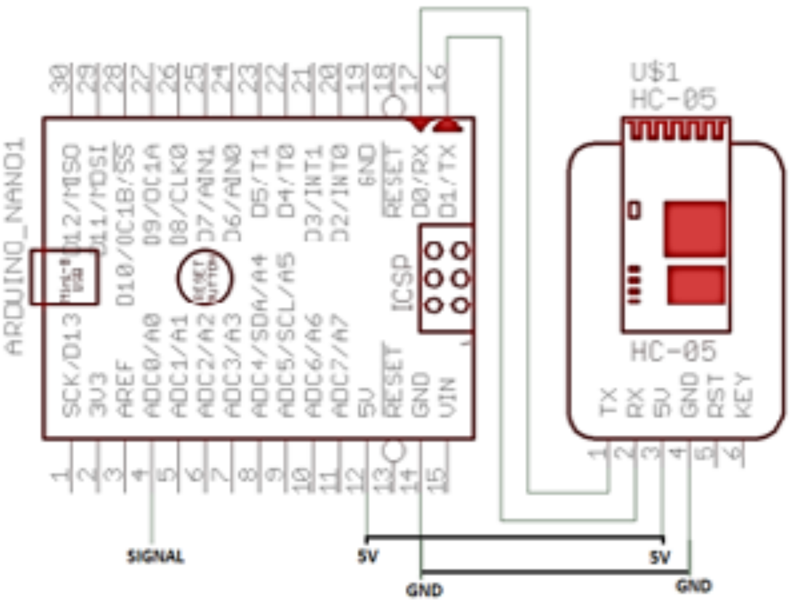


| SENSING AND COMMUNICATION STAGE

The sensing and communication stage consists of an Arduino Nano microcontroller as well as the HC-05 BT module. The BT module's serial communication is connected to the Arduino's serial port, thus, the communication between the two devices is as if they interacted through the Arduino IDE monitor. It was created this way as to allow the user to interact with the microcontroller even if the HC-05 BT module is not available or if the implementation done by the user does not require a BT module. The Arduino IDE is an open-source Java-based application that allows for a fast and easy way to upload code to the ATmega328 microcontroller chip [6].

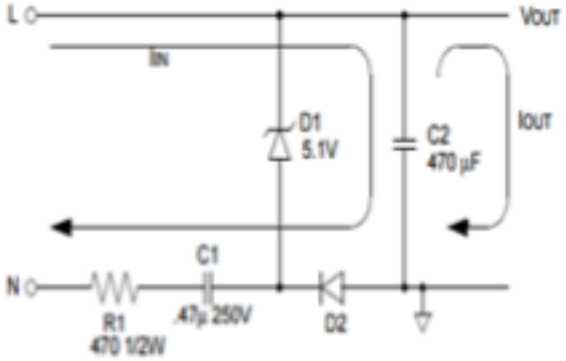
The Arduino Nano is a small board based on the ATmega328. It features a 6-20V unregulated power supply in pin 30 and a 5V regulated power supply in pin 27. The Hi-Link module is connected to pin 27 [5]. The HC-05 BT is an easy-to-use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Its communication is via serial communication which makes an easy way to interface with controller or PC. HC-05 Bluetooth module provides switching mode between master and slave mode which means it able to use neither receiving nor transmitting data [7].

A diagram of the sensing and communication stage is shown below.



CAPACITIVE TRANSFORMERLESS POWER STAGE

A different approach replacing the power stage previously shown with a capacitive transformerless power supply is proposed. The following configuration is obtained from [8]. An example diagram of the schematic of the power supply is shown in the following illustration. This power stage is proposed due to its cost. Although an AC-DC converter can often be created using a transformer and a rectifier circuit, if the device implemented only consists of a microcontroller and a few passive components, the cost of the AC-DC converter is not effective. This implementation also takes a lot of space when compared with the capacitive transformers power stage.



Variable	Value
Vrms (VAC)	110
Vz (VDC)	5.1
F (Hz)	59.5
C1 (uF)	6.8
Imin (mA)	150
Capacitor Tolerance (%)	20

Using the formulas found in [8], one can modify the values of the components to suit the needs necessary. The values of Vrms, Vz, f, C1 and Iman are assumed as to find the value of R1.

R1=(1/(2*(Imin/1000)))*(SQRT(2)*Vrms-Vz)-(1/(2*ϖ*f*((Cap/1000000)*(1-CapT))))

Getting a value of 9.84 Ohm, from (3), the value is set at 10 Ohm. After which, assuming both the lowest and largest values of the components, the Imin and Imax can be correctly calculated to make sure that the power ratings of the components will be properly taken into account. The values obtained are as follows.

Variable	Min Value	Max Value
Vrms (VAC)	110	120
Vz (VDC)	5.1	5
F (Hz)	59.5	60.1
C1 (uF)	6.8	6.8
R1	10	10
Capacitor Tolerance (%)	20	20
Resistor Tolerance (%)	5	5

$$Imin = \left(\frac{SQRT(2) \cdot VrmsMin - VzMax}{2 \cdot \left(\left(\frac{1}{2 \cdot \pi \cdot frecMin \cdot \left(\frac{CapMin}{1000000} \right) \cdot (1 - CapTmin)} \right) + \left(Rmax \cdot (1 + RTmax) \right) \right)} \right) \cdot 1000$$
$$= 149.8mA$$

$$Imax = \left(\frac{SQRT(2) \cdot VrmsMax - VzMin}{2 \cdot \left(\left(\frac{1}{2 \cdot \pi \cdot frecMax \cdot \left(\frac{CapMax}{1000000} \right) \cdot (1 + CapTmax)} \right) + \left(Rmin \cdot (1 - RTmin) \right) \right)} \right) \cdot 1000$$
$$= 246.5mA$$

After the minimum and maximum current is found, the values of the power ratings of the components are taken into account. This must be done to make sure that the power stage won't cause problems for the microcontroller and the BT module in the form of busted components that may cause variations in the current and as such damage the circuits. The following equations are used to find the power ratings of the components.

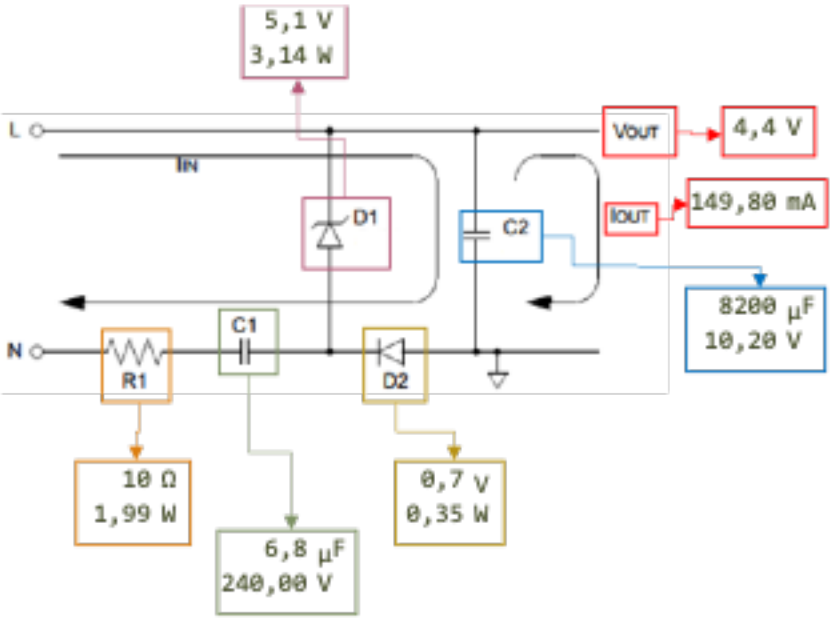
$$PR1 = \left(\left(VrmsMax \cdot 2 \cdot \pi \cdot frecMax \cdot \left(\frac{Cap}{1000000}\right)\right)^2\right) \cdot \left(Rmax \cdot (1 + RTmax)\right) = 1W$$

$$PD1 = \left(VrmsMax \cdot 2 \cdot \pi \cdot frecMax \cdot \left(\frac{Cap}{1000000}\right)\right) \cdot Vz = 1.57W$$

$$PD2 = \left(\frac{Imax}{1000}\right) \cdot Vd = 0.17W$$

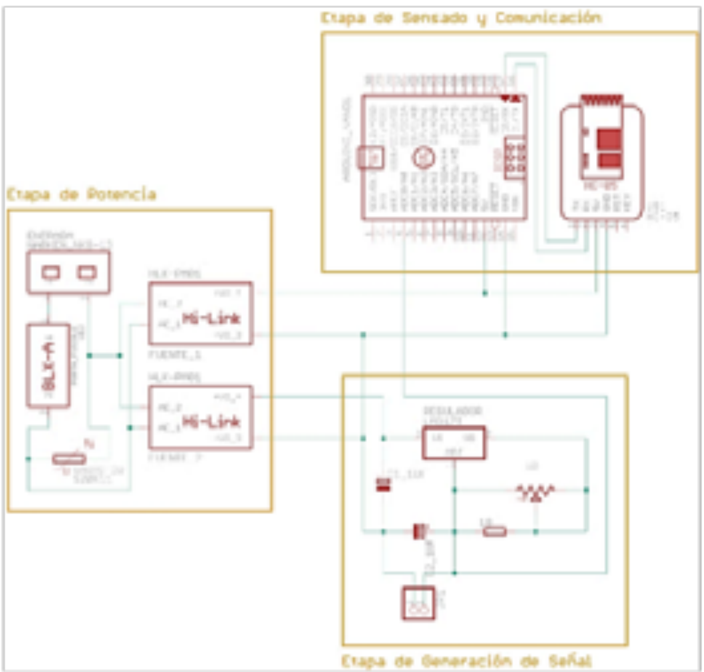
These values are then multiplied by 2, to account for possible power surges (It should be noted that although the circuit does possess both a fuse as well as a varistor, some components may be damaged in the case of a power surge). After which, the values obtained create the following schematic. It should be noted that the higher the capacitance of C2, the less ripple the power source will show, as such a value of 8.2mF was specified. This value may as well be larger if the resulting ripple of the power stage is still too much for the user. A rectifier bridge can also be added to further reduce the ripples at the output of the circuit.

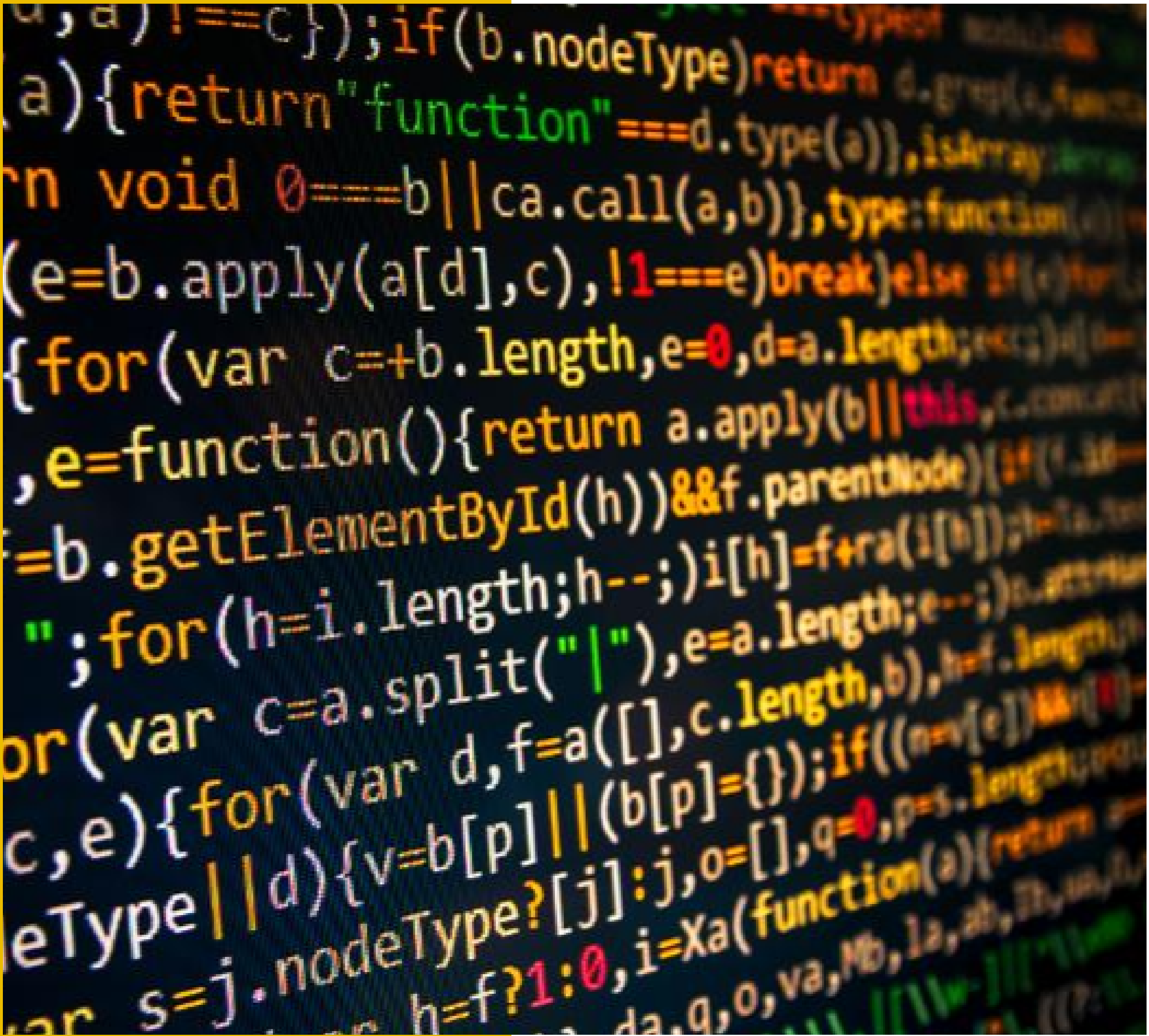
[https://youtu.be/r2ekn8KQhSQ] shows a video of the operation of the circuit, taking into account both considerations. A simulation of the operation of the module can be observed in Annex 2.



DIAGRAM

The overall diagram of the implementation of the three stages can be observed below. The implementation is done using the power stage with the Hi-Link ultra-compact power modules. It should be noted that this implementation is done using the Hi-Link modules instead of the capacitive transformerless power supply.



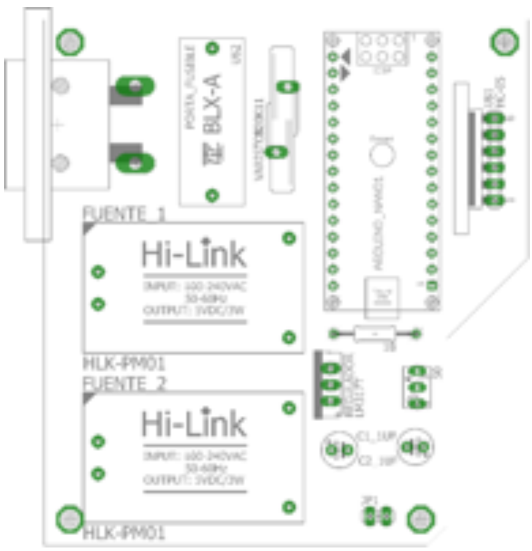


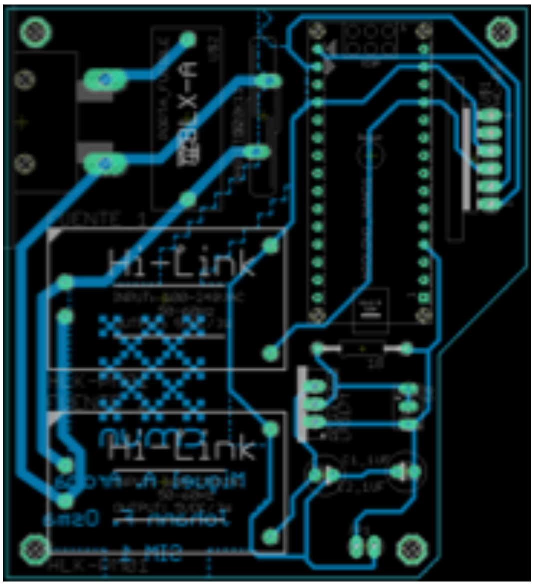
CODE

The code for the implementation can be observed below. It should be noted that the communication implemented with the device is strictly through BlueTooth; thus, if the implementation is to be done using the Arduino IDE monitor, the code should be modified to print the results without taking into account the BT module. It should be noted that this implementation makes use of the EEPROM library which allows the arduino to use non-volatile memory. The code can be observed in Annex 1. The implementation already uses the Serial communication pins of the Arduino to communicate with the BT module, as such the code modifications needed to be implemented are not extensive. It should also be noted that if a second serial connection is to be implemented through a USB to TTL serial, both the BT module as well as the Arduino IDE monitor could be used simultaneously. It should also be noted that the PCB diagram shown does not take this into account, thus if it were to be implemented, a new PCB diagram should be taken into account.

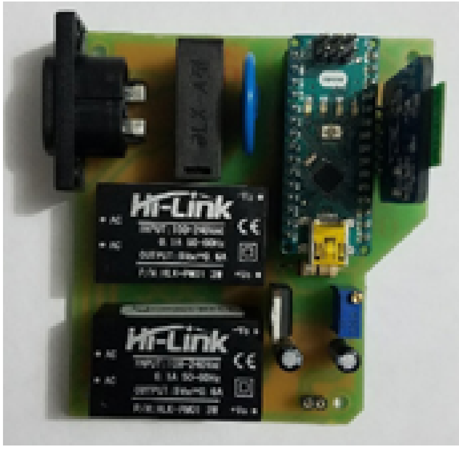
IMPLEMENTATION

For the implementation, a PCB is designed to allow for a more organized circuit. A higher resolution image of the PCB layers can be observed in Annex 3 and Annex 4. It should be noted that to make the implementation easier to use, the fuse holder should be placed in a spot with easy access, as to allow for the replacement of the fuse in case of a power surge.

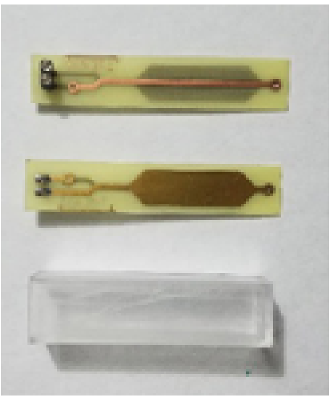




The final implementation using the PCB diagram above can be observed below.



For this implementation, an electrode was also employed to observe the impedance change when working with enzymes or molecules. The electrode implementation can be observed in illustration 14. In the following illustration, a picture of the electrode implemented is shown.



RESULTS

The validation of the circuit is done placing resistors in the impedance sensing module. It should be noted, that to obtain the most accurate results, the resistors themselves should be as precise as possible. As such, it is recommended to use 1% tolerance resistors for this validation. The validation is done using resistors in the 1 Ohm and 2 Ohm range. The following data was obtained.
The data collection process can be observed in Device Validation Process [https://youtu.be/2dv06NhYr4M].

Resistor Value	Measured Value	Percentage Error
1.5 Ohm	1.52 Ohm	1.33%
1.62 Ohm	1.64 Ohm	1.23%

OBSERVATIONS

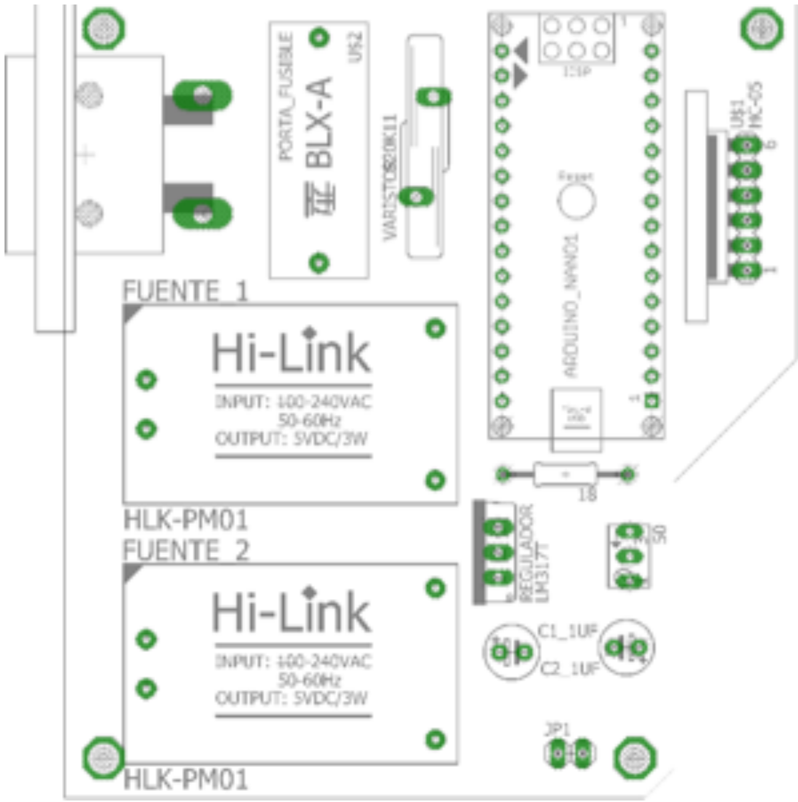
It should be noted that equation (2) is taken using impedances of 1 Ohm and 2 Ohm. Thus, the larger the impedance in the sensing module, the larger the error when compared with the theoretical values. The calibration process can be performed by measuring the voltage values caused by the impedance in the sensing module; it should be noted that the calibration is to be done near the values that the low impedance meter is to be used for. For this implementation, values of about 2 Ohms were sufficient. It should also be noted that the trimmer operation was kept below 50% as to allow for a more linear regression. The errors obtained in the results were always 0.02 Ohm off from the original value, which is a constant value in a lineal system. Thus, corrections in the software implemented would suffice to compensate for this uncertainty.

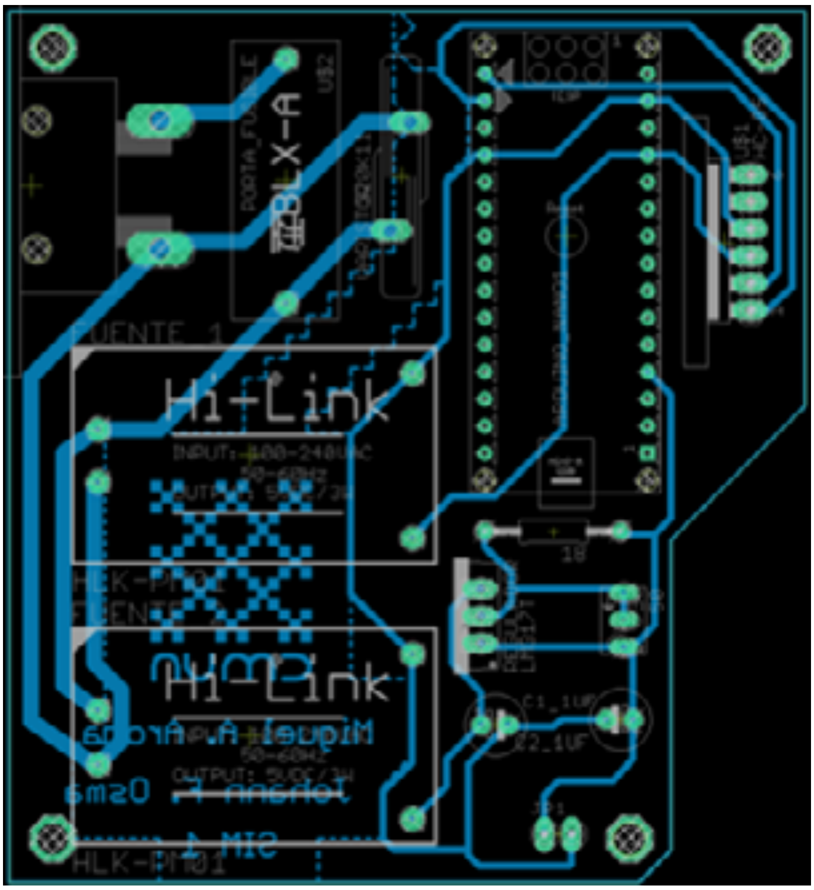
It should also be noted that if the power stage is to be replicated using the capacitive transformerless method, the capacitor used should be made from polyester. An example of the capacitive transformerless power stage using electrolytic capacitors can be observed in Capacitive Power Stage [https://youtu.be/EOQJMa6S264].

REFERENCES

1. HIOKI E.E. CORPORATION. (n.d.). How to use an impedance meter. HIOKI. Retrieved September 9, 2021, from https://www.hioki.com/global/learning/how-to/u-impedance-meters.html.
2. Ultra-compact power module 3W 5V HLK-PM01. Hi-Link HLKTech. (n.d.). Retrieved September 9, 2021, from http://www.hlktech.net/product_detail.php?ProId=54.
3. Regulador de voltaje ajustable L317 / LM317 / LM317T variable de 1.2V A 37V 1.5A. Tecnopura. (2020, October 14). Retrieved September 9, 2021, from https://www.tecnopura.com/producto/regulador-de-voltaje-ajustable-l317-lm317-lm317t-variable-de-1-2v-a-37v-1-5a/.
4. Varistor and metal Oxide Varistor Tutorial. Electronics Tutorials. (2018, July 9). Retrieved September 9, 2021, from https://www.electronics-tutorials.ws/resistor/varistor.html.
5. Electronilab. (2021, September 8). Arduino nano v3 - atmega328 5V + Cable USB Compatible. Electronilab. Retrieved September 9, 2021, from https://electronilab.co/tienda/arduino-nano-v3-atmega328-5v-cable-usb/.
6. Arduino IDE 1.8.16 Software. Arduino. (n.d.). Retrieved September 9, 2021, from https://www.arduino.cc/en/software.
7. HC-05 Bluetooth Module User's Manual V1.0. GME. (n.d.). Retrieved September 9, 2021, from https://www.gme.cz/data/attachments/dsh.772-148.1.pdf.
8. Condit, R. (2004). AN954 Transformerless Power Supplies: Resistive and Capacitive. MicroChip. Retrieved September 9, 2021, from http://ww1.microchip.com/downloads/en/AppNotes/00954A.pdf.

ANEXXES


PORTA_FUSIBLE
BLX-A
U$2
VARISTOR
FUENTE_1
Hi-Link
INPUT: 100-240VAC
50-60Hz
OUTPUT: 5VDC/3W
HLK-PM01
FUENTE_2
Hi-Link
INPUT: 100-240VAC
50-60Hz
OUTPUT: 5VDC/3W
HLK-PM01
ARDUINO_NANO1
ICSP
U$1
HC-05
REGULADOR
LM317T
C1_1UF
C2_1UF
JP1

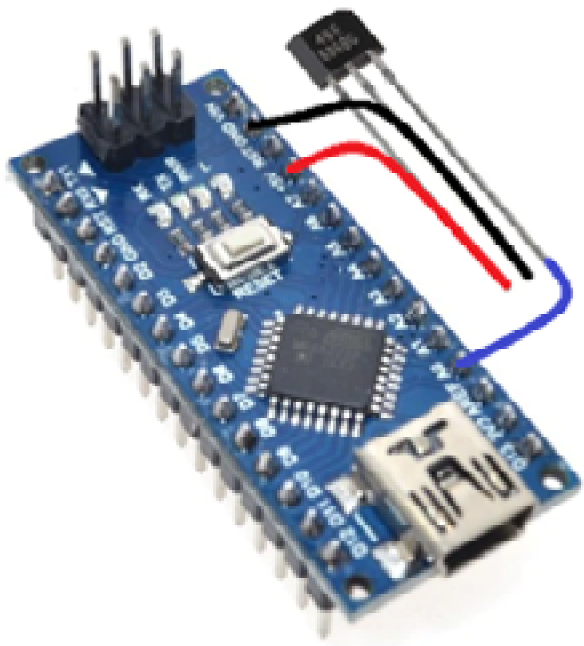

PORTA_FUSIBLE
BLX-A
U$2
FUENTE_1
Hi-Link
HLK-PM01
Hi-Link
C1_1UF
C2_1UF
JP1

MAGNETIC FIELD METER

Magnetic field meters are devices which allow to measure magnetic flux density and magnetic field exposure [1]. These are often used to determine the magnetic anomalies or dipole moments of various materials and magnets.

The methodology used for the development of the heating plate consists of the implementation of an ON-OFF controller, that is, a simple system which checks if the process variable is above or below a certain fixed point [2].

This implementation makes use of an Arduino microcontroller as well as the ss49e Hall effect sensor to output a voltage proportional to the strength of the magnetic field being measured. The implementation's software allows to connect a computer through USB with the Arduino, which through serial communication sends the data. The software displays the data in real time as it is sent by the microcontroller



MATERIALS

1x Arduino Nano
1x ss49e Hall Sensor

COMPONENT EXPLANATION

SS49E SENSOR

The ss49e is a hall effect sensor which outputs a voltage proportional to the external magnetic field. This sensor has a p-type semiconductor which has a current flowing through it. When the sensor is placed in proximity to a magnetic field, the field interacts with the semiconductor material deflecting the charge to one side of the material. As such, this creates a potential difference between the 2 sides of the semiconductor, resulting in a voltage proportional to the strength of the magnetic field [2].

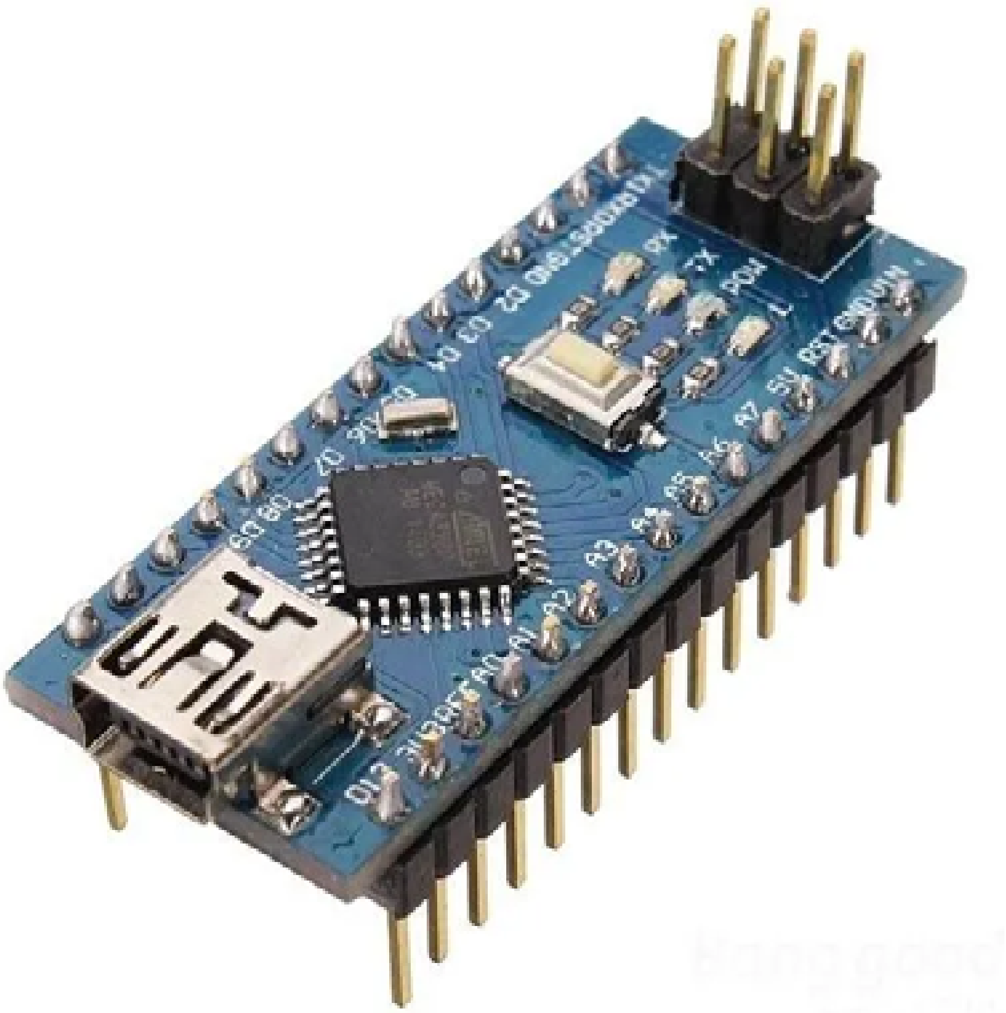
The Hall effect voltage output can be calculated using the following formula:

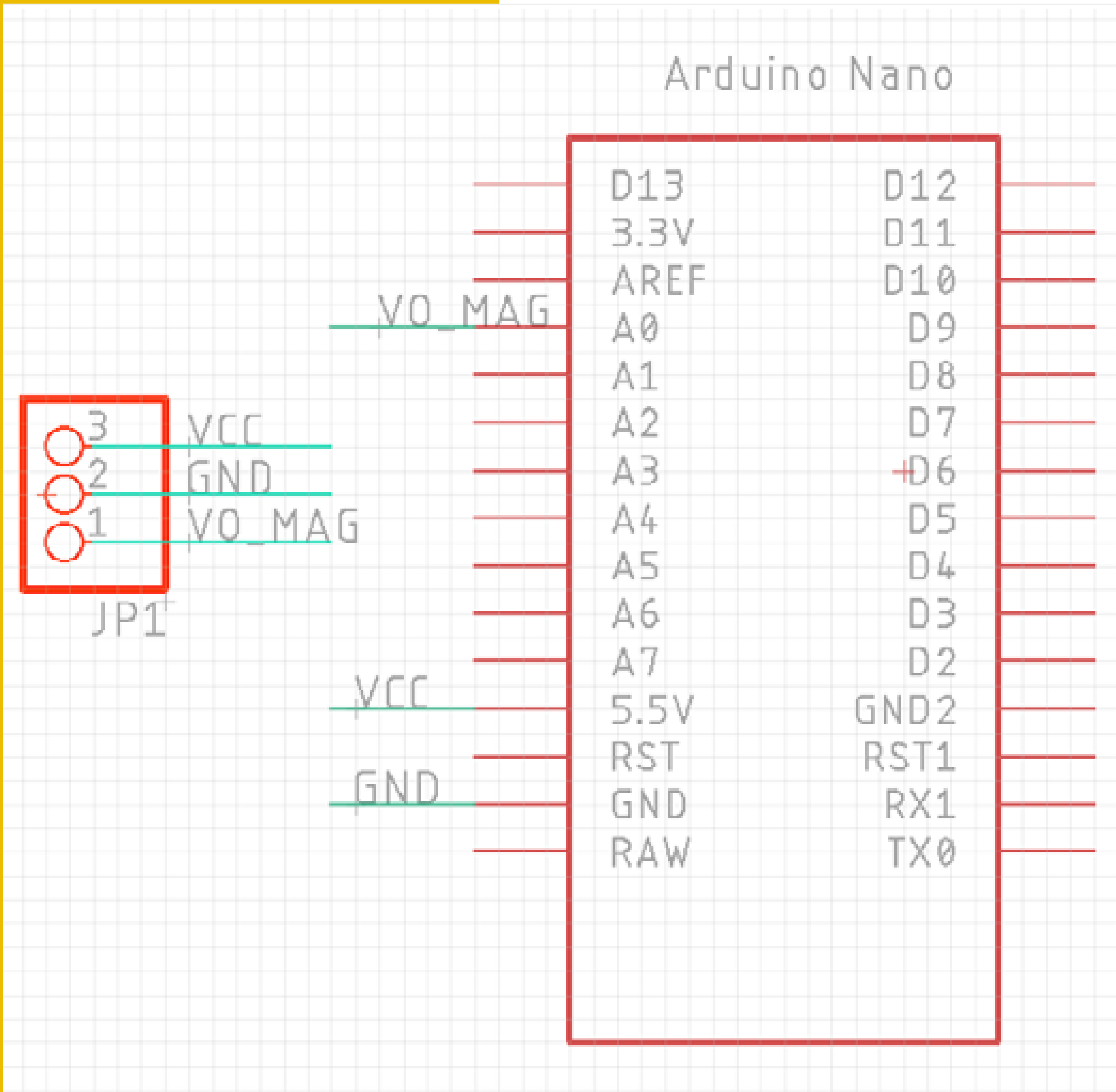
$$V_h = R_h * (I/t * B)$$

Where Vh is the Hall effect voltage output, Rh the Hall effect coefficient, I the current passing through the sensor in Amps, t the thickness of the sensor and B the magnetic flux density. The ss49e operate at 4.5-6V and provides a linear output of approximately 15mV/mT at room temperature.

ARDUINO

An Arduino is a microcontroller board based on the ATmega328P. It has an unregulated power source in the VIN pin which supports 7-12V, and a regulated 5V power source [3]. For this implantation the Arduino is the device which reads the analog output of the ss49e sensor and sends it through serial communication to the computer through USB.

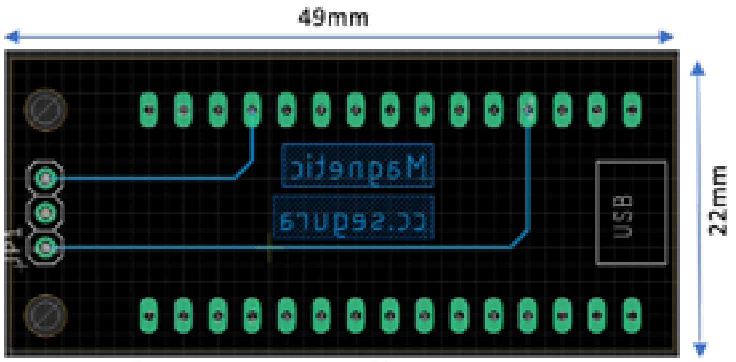




FINAL DEVICE

The device diagram is shown above:

IMPLEMENTATION



For the implementation, the following PCB is designed for fast and easy use and deployment of the device

CALIBRATION

Although no calibration is implemented, the methodology will be shown. It should be noted that for the device to be properly calibrated a magnet with a known magnetic field amplitude is needed.

The calibration methodology consists of a regression through which the magnet is measured at different distances to create a calibration equation. For point charges, the magnetic field strength follows Coulomb´s Law and as such is inversely proportional to D^2. For electric dipoles however, it is inversely proportional to D^3. Thus, data points should be measured to replicate an exponential behavior [4].

Once the calibration equation has been calculated, it can be implemented in the Arduino code. It should be noted that the magnetic field measurements should be conducted at a specified distance, which can be determined after the device's calibration.

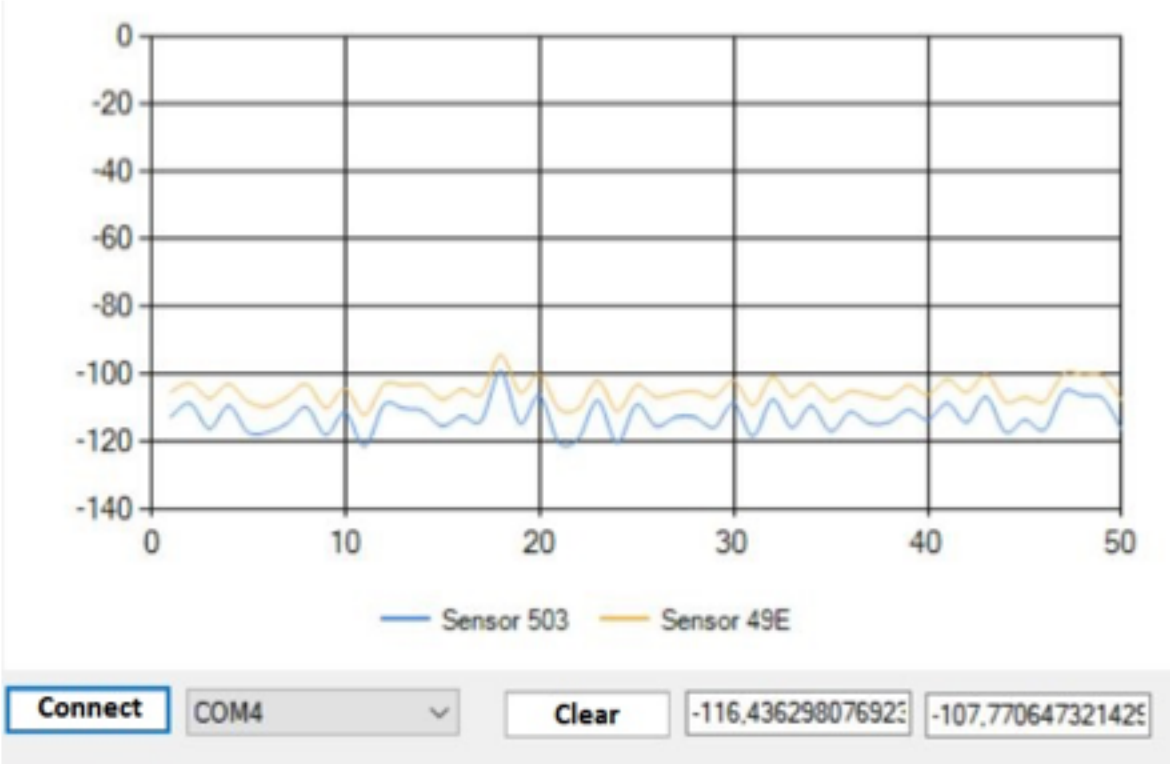
CODE

Arduino:
The Arduino code consists a loop which reads the analog voltage in analog pin 0, as well as in analog pin 1. For this implementation, analog pin 1 should be grounded, so that the readings from the A1 pin are 0. The Arduino code can be observed in Annex 1.

Computer:
The software consists of a chart which shows the data obtained by the Arduino microcontroller in real time. If the A1 pin in the Arduino is grounded, the chart will only show the data relevant for the device, which is the data obtained by the A0 pin. The software's code can be observed in Annex 2.

The software implemented scans for COMs in the USB ports of the computer to try to connect. Once an Arduino COM is selected and the Connect button is clicked, it starts the serial communication with the Arduino microcontroller, which starts to send the data to the computer. The software features two text boxes which show the voltage readings from the A0 and A1 pins; these values are then mapped in the chart in real time. The software also features a Clear button which deletes the data in the chart; the chart, however, will continue to load data in real time as long as the Arduino sends it.

The following illustration shows the software implementation for this device.



RECOMMENDATIONS

Regarding the device's operation, the Arduino microcontroller can be affected by the magnetic field measured by the ss49e sensor. If the magnetic field is strong enough, the microcontroller may experience voltage fluctuations and as such may restart or shut down. If the implementation is to measure large magnetic fields, approximately 1T, it may be necessary to account for these fluctuations by moving the sensor further away from the Arduino Nano to diminish the magnetic field strength.

It should also be noted that both the Arduino as well as the software monitor the voltage in pins A0 and A1. Thus, if only one of the pins is to be used, the other should be grounded to reduce the voltage variations measured by the microcontroller. This will also allow to properly see the data in the software chart, as voltage variations from an unused pin can cause visual noise in the chart.

REFERENCES

1. MAGNETIC FIELD HiTESTER FT3470-51. HIOKI. (n.d.). Retrieved September 9, 2021, from https://www.hioki.com/global/products/field-measuring/magnetic-field/id_5815#:~:text=Hioki%20magnetic%20field%20meters%20are,research%20on%20magnetic%20field%20exposure.
2. Hall Effect Sensor. Basic Electronics Tutorials. (2018, February 9). Retrieved September 9, 2021, from https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html.
3. ¿Qué es Arduino? Arduino. (2017, July 12). Retrieved September 9, 2021, from https://arduino.cl/que-es-arduino/.
4. MAGNETIC FIELDS VARYING AS AN INVERSE CUBE. Millersville University. (1970, September 9). Retrieved September 9, 2021, from https://www.millersville.edu/physics/experiments/023.

IMAGE PROCESSING SOFTWARE

When working with images taken through microscopy, it is often needed to take accurate measurements of objects within the image. Although approximations can be obtained by using a scale bar along the glass slides, this method does not provide an accurate reading of objects further away from the scale bar; furthermore, the measurement uncertainty increases as the distance between the measured object and the scale bar increases.

This implementation allows to take microscopy measurements having a measurement pattern with the optical settings of the microscope used.

Furthermore, the measurements made can be quantified using a user-defined scale. In the software, images of the measurements can be exported with the microscopy scale bar given by the user. Thus, it provides a faster and more accurate way of taking reliable measurements of microscopy images.

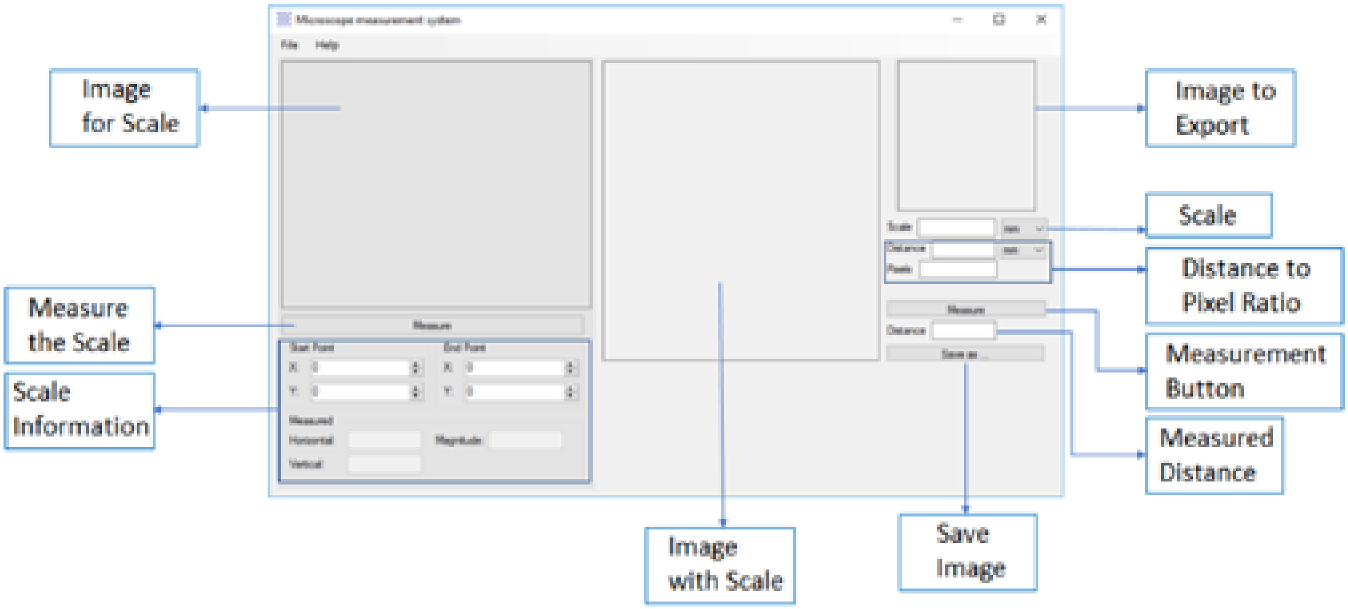
https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/Image%20Processing%20Software

SYSTEM REQUIREMENTS

The program requires Microsoft Windows 7 or newer with .net framework 4.5 o newer.

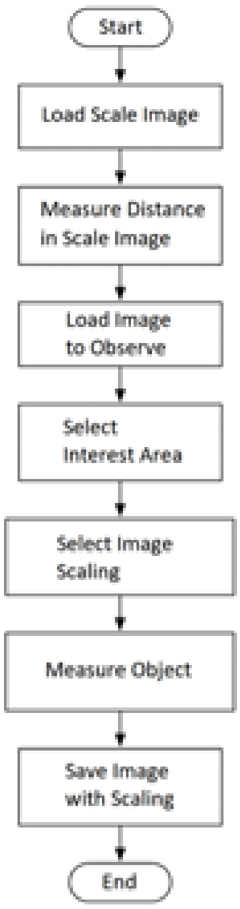
COMPONENT EXPLANATION

The software components can be observed in the following illustration.

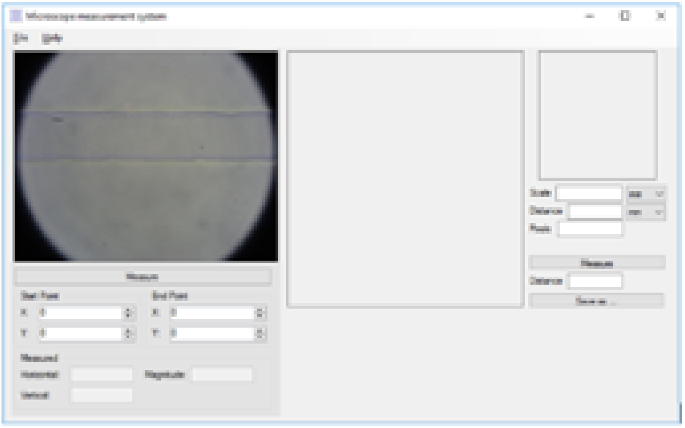


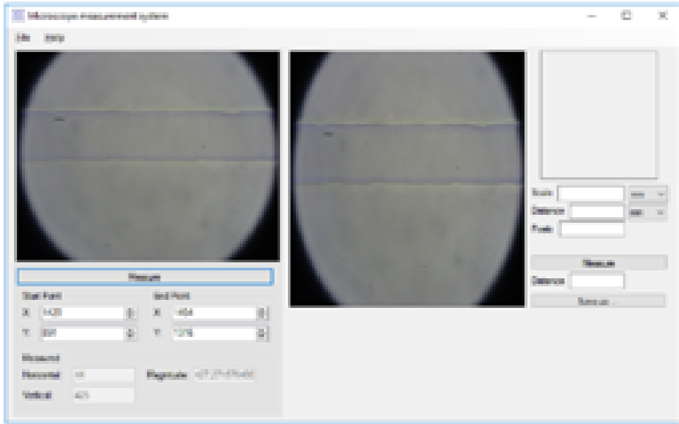
OPERATION

By executing the application, the interface shown in illustration 1 will appear. Illustration 2 shows the flow diagram for the correct operation of the software. The illustration can be observed below.

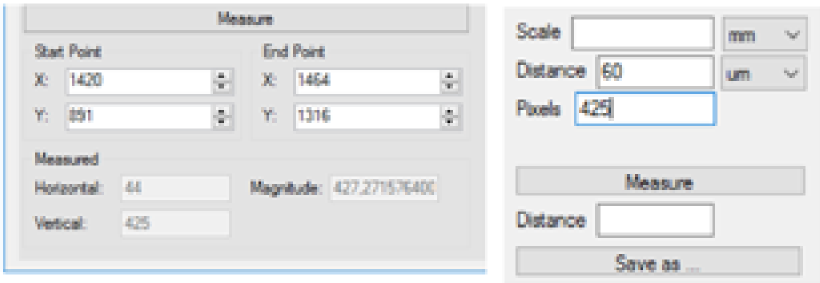


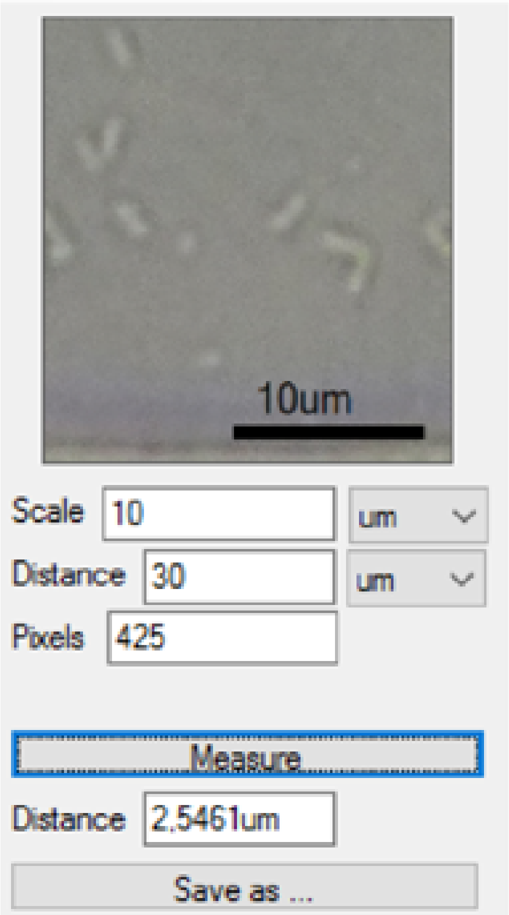
The images are loaded by dragging them to each of the spaces. The scale image must be one that has an element in which the length is known. It should be noted that for the proper operation, the measurements have to be known in both standard units as well as pixels.



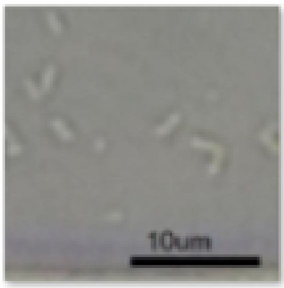


Perform the distance measurement in pixels of the element whose measurement is known, for this example the size of the channel is 60um, which is equivalent to approximately 425 pixels. For the measurement press the Measure button, and then click twice on the Image for Scale on the screen. The program calculates the horizontal and vertical distance and the magnitude of pixels between both points. The obtained size and measurement data are then loaded.





The area for calculation is selected by clicking and moving the mouse in the Image with Scale. After having creating the calculation area, the size of the scale bar is defined (10um for the example), and the measurement is carried out. This is observed in illustration X. For the actual measurement, press the Measure button, and then two clicks are made on the screen, the program calculates the distance in pixels between both points. The program also generates a distance value for the selected points, which can be observed in red.



02-03-2017
2,5461um.jpg

By clicking the Save As button, a file directory opens to save the image, the default name contains the date and the measurement value, the result is an image like the one presented in the illustration.

RECOMMENDATIONS

It should be noted that higher resolution images provide a better product, as the software scales up the image according to the calculation points of the Image with Scale.

THERMAL BATH

A thermal bath is an instrument used to maintain a solution at a constant temperature. It is a thermodynamic system with a heat capacity so large that the temperature of the reservoir does not change when a reasonable amount of heat is added or extracted [1].

The methodology used for this implementation consists of a PID controller which operates a submerged heater in a casing. The thermal bath implemented makes use of a water pump to move the water from the casing to the thermal bath, where the thermal system is to be placed. The PID controller is implemented through an Arduino microcontroller that gets the input through a DS18B20 submersible temperature sensor placed in the heated casing. The thermal bath itself also features a light box, which serves the purpose of illuminating the thermal system, for when photo and/or video capture is needed.

Although the implementation is controlled through the Arduino microcontroller, it can be controlled through a Raspberry PI. The code to control the thermal bath through the Raspberry PI is also provided.

https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/Thermal%20Bath

MATERIALS

1x Resistor Heater
1x Max6675 Module
1x MOC3020 Optocoupler
1x PC817 Optocoupler
1x Full Wave Rectifier
1x TIP120 Transistor
1x Res. 39Ohm 1/2W
2x Res. 1kOhm 1/2W
1x Res. 4.7kOhm 1/2W
1x Res. 10kOhm 1/2W
2x Res. 27kOhm 1W
1x Res. 330Ohm 1/2W
1x Res. 270Ohm 1/2W
1x Cap. 0.01uF 250V
1x Triac Q4015L5
1x 1N4007 Diode
1x LED Strip
1x Stainless Steel Casing
1x Non-submersible Water Pump 6-12V
1x DS18B20 Temperature Sensor

COMPONENT EXPLANATION

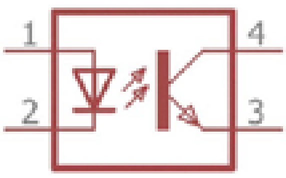
MAX 6675

The MAX6675 module serves to compensate and linearize the output of type K thermocouples. The module offers a 0.25°C resolution, with an accuracy of ±1.5°C. Since it has an operation voltage of 3-5.5V, it can be easily used with Arduino microcontrollers [2]. This module is used because the thermal bed is monitored using a type K thermocouple, and the module can output the data obtained in 12-bit resolution and it is SPI compatible.

OPTOCOUPLERS

The MOC3020 optocoupler is a device which emits and receives light and serves as a light-activated-switch. It is often used in implementations which require to trigger an isolated triac through a digital signal [3]. The MOC3020 has an isolation voltage of about 5.3kV. In comparison with the MOC3020, the PC817 is a general purpose optocoupler.

This component is most often used in implementations which require the isolation of currents, in this case the isolation of the DC part of the circuit with the AC part of the circuit, while still being able to have interactions between the two.



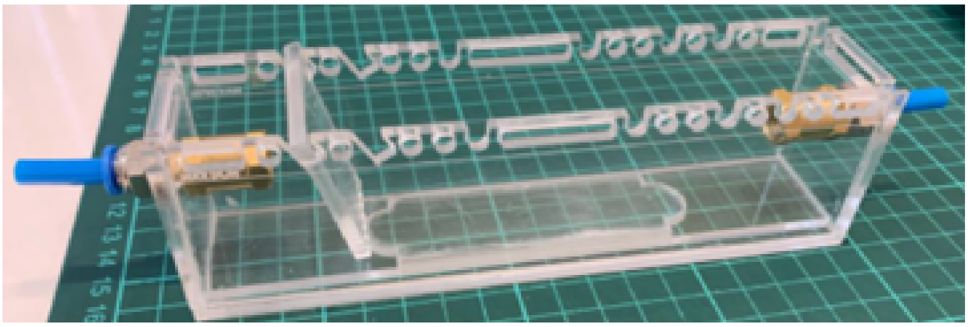
DS18B20

It is a submersible probe which allows for the measurement of temperatures in the range of -55°C to 125°C [5]. This temperature sensor operates using the 1-Wire protocol for data transmission. This allows for the use of only 1 pin to operate various DS18B20 temperature sensors. This sensor was taken into account due to its uncertainty values. For temperatures in the range of -10°C to 85°C, the sensor measures with an accuracy of ±0.5°C, while for temperatures in the ranges of -55°C to -10°C and 85°C to 125°C, it measures with an accuracy of ±2°C. Since the maximum temperature threshold for this implementation is 60°C, this sensor provides the required characteristics.

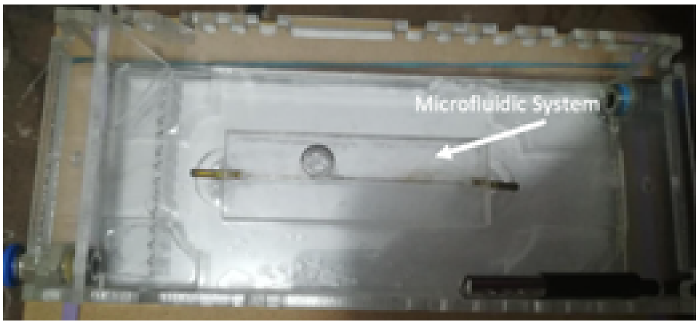
WATER PUMP

The R385 non-submersible was chosen for this implementation due to its low operation voltage as well as its large flux rate. The R385 has an inner and outer diameter of 6mm and 8.5mm respectively and can pump from 1.5 to 2.1 liters per minute. Its operation voltage can vary between 6-12V. It should be noted that the R385 water pump can withstand water up to 75°C. For this implementation, the water pump is powered by the Arduino microcontroller.
It should be noted that this water pump is non-submersible, as such it should be placed safely outside of the thermal bath and the heated reservoir. A case to hold the water pump and keep it dry should also be considered if the components are to be integrated.

THERMAL BED



The thermal bed implemented is created using 3mm acrylic. The entry and exit points for the water flow are located diagonally on opposite walls, to make sure that the water flow covers most of the thermal bed and thus allows for the desired temperature throughout the thermal system. The other openings for the temperature sensor and the probes are located above the water pump's entry and exit points to minimize leakage. The probe openings are 1.5mm in diameter, and the water pump's entry and exit points are 1cm in diameter. Another feature of the thermal bed is an embedded case in the bottom of the thermal bed which can hold a microfluidic system. This implementation is 3mm deep and 25mm wide. The following image shows the thermal bed implementation in 3mm acrylic.
It should be noted that the thermal bed is not thermally isolated.



LIGHT BOX

Another important element of this implementation is the light box, which allows for the proper illumination of the thermal bed and the microfluidic system when photo and/or video capture is needed. The light box consists of an MDF casing with an LED strip glued to the inside walls. The operation of the light box is done through the same implementation which controls the thermal bed. That's to say, a command is sent from either the Arduino microcontroller or the Raspberry PI. The following illustrations show the LED strip as well as the light box MDF implementation.

The light box serves as a base for the thermal bed, which is to be placed on the outline seen in illustration 7. This properly illuminates the acrylic thermal bed.

ARDUINO

Arduino is a microcontroller board based on the ATmega328P. It has an unregulated power source in the VIN pin which supports 7-12V, and a regulated 5V power source [6]. For this implantation the Arduino is the device which controls the thermal bath's temperature as well as the state of the light box. The control of this implementation is done either through the Arduino's IDE monitor or through the Raspberry PI, if the Arduino is connected to it through USB.

RASPBERRY PI

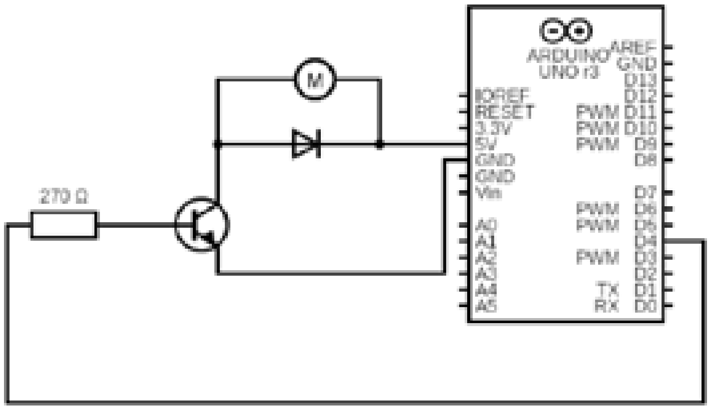
The Raspberry Pi is a low-cost, small-size computer which allows the user to run code [7]. For this implementation, the Arduino controller is implemented through python code, which communicates through serial communication with the microcontroller. In this implementation, the job of the Raspberry PI is that of the Arduino's IDE monitor. The Raspberry PI uses the Serial communication library to emulate the Arduino's monitor.

DIAGRAM

WATER PUMP

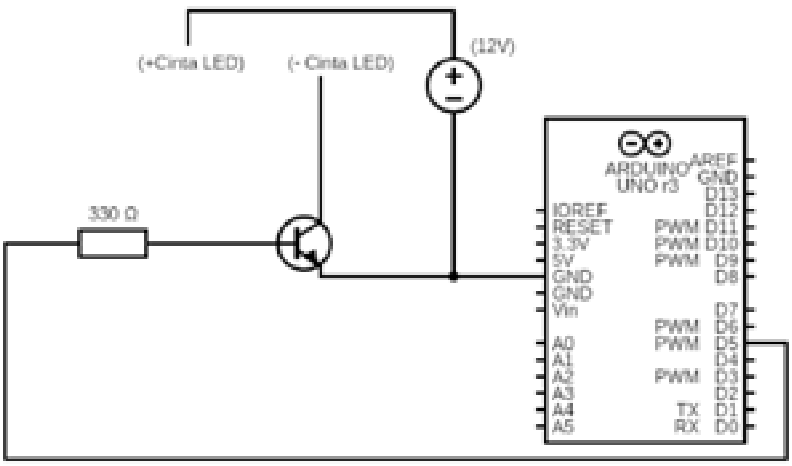
Although the water pump has a maximum flux rate of 2.1 liters per minute, it was found that this flux rate could cause disturbances within the microfluidic system. Thus, the electric current to the water pump was limited. Although some implementations using Darlington pairs were taken into account, the final implementation can be observed below [8].

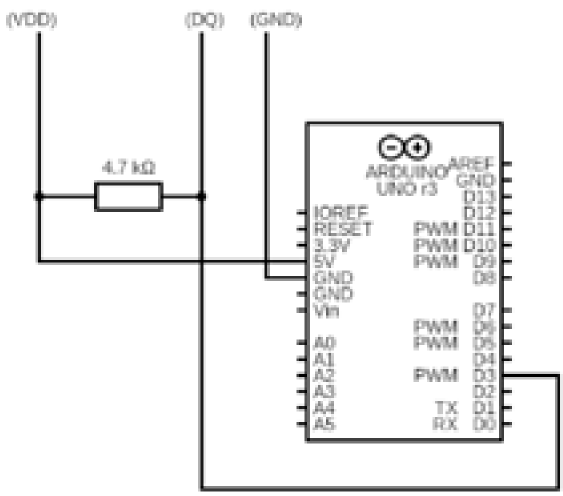
It should be noted that to protect the electronic components connected to the water pump, a 1N4007 rectifier diode is placed in parallel with the motor. This is done, because once the water pump is turned off, a magnetic field generated by back EMF, can cause problems within the circuit. The motor is controlled through the D4 pin in the Arduino, using an NPN transistor as a switch to close the circuit.



LIGHT BOX AND DS18B20

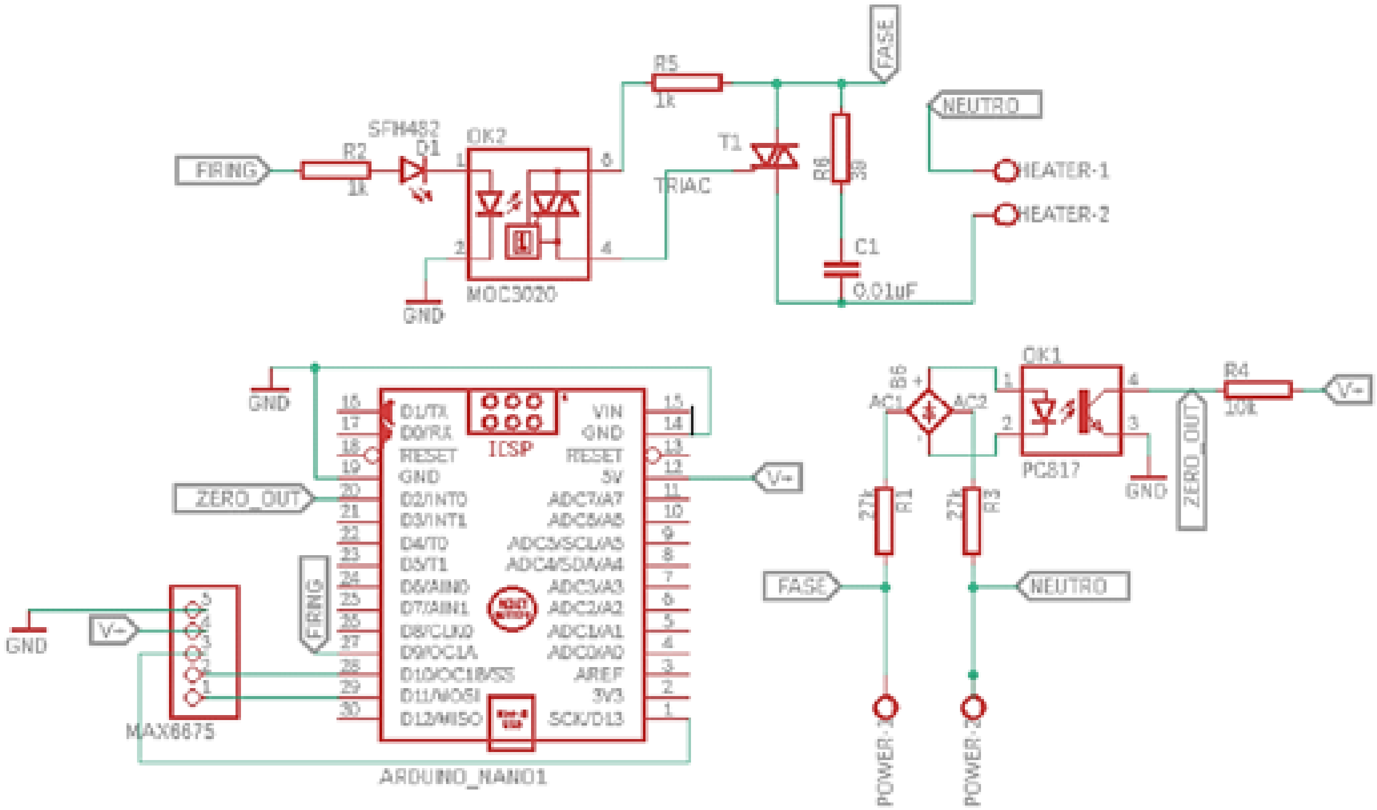
The light box's implementation requires an input voltage of 12V. As with the water pump diagram, an NPN transistor is used as a switch to close the circuit between the LED strip and the 12V input. The LED strip is controlled by the Arduino through the D5 pin. The following illustration shows the circuit diagram for the LED strip implementation in the light box.





HEATER DIAGRAM

The heater circuit schematic can be observed in the following illustration. It should be noted that the circuit has both AC as well as DC voltages. Thus, it was necessary to isolate both tensions using optocouplers. The optocouplers used are the MOC3020 as well as the PC817. That's to say, all previous diagrams are not shown in this one. As such, the final implementation would have all diagrams in tandem.



The heater diagram consists of the Arduino microcontroller (bottom left) as well as an AC zero detection circuit (bottom right) and a heater controller (top). The Arduino is responsible for taking the thermocouple's measurements through the MAX6675 module and detecting the zero intercept of the AC voltage through the AC zero detection circuit's ZERO_OUT. The Arduino microcontroller is also responsible for triggering the heater controller circuit.

The AC zero detection circuit uses the rectified AC current as an input for the optocoupler PC817. This optocoupler then allows current to flow through its phototransistor grounding the 5V voltage. This grounding of V+ allows the microcontroller to detect the zero intercept of the AC voltage.

Lastly, once the Arduino detects the zero intercept, it can trigger the heater controller using the PID implemented through the PID_v1 library. The heater controller also makes use of a triac, which allows for the proper flow of AC current through the circuit, heating the thermal resistor. The triac is needed in this implementation, as the optocoupler is not able to properly handle AC current through the phototransistor; the phototransistor would be damaged which would render the signal from the Arduino microcontroller to the thermal resistor inaccurate

CODE

ARDUINO

The Arduino code implemented makes use of the PID_v1, MAX6675 and DallasTemperature libraries. These libraries allow for the correct operation for the thermal bath, as they make it easier to read the temperature information from the thermocouple as well as calculate the temperature response that the thermal bath should have, through the PID controller implemented in the Arduino microcontroller. The Arduino code can be observed in Annex 1. It should be noted that the user communication is done through Serial communication, as such if the Arduino microcontroller is not connected to the Raspberry PI, the default will be the Arduino's IDE.

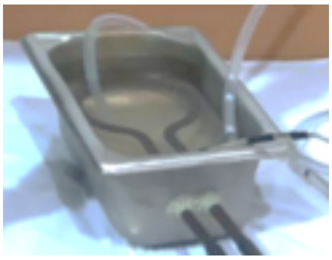
RASPBERRY PI

The Raspberry PI code consists of a serial communication log between the Raspberry PI and the device located in the /dev/ttyACM0 address, which for this implementation, is the Arduino Microcontroller. The code runs a loop that asks the user for commands, which are sent to the Arduino to control the thermal bath, as well as the light box. The Raspberry PI code can be observed in Annex 2. It should be noted that since the Arduino microcontroller reads the commands through serial communication, the same implementation can be controlled just using the IDE's monitor.

IMPLEMENTATION

CASINGS

The reservoir casing is made from stainless steel, which holds the heated water and the electric heater itself. The water pump's hoses are connected to the reservoir. The following illustration shows the reservoir's implementation.



The 3mm acrylic thermal bath casing is placed on top of the MDF light box, to properly light the microfluidic system held within. The following illustration shows the implementation. The light box template can be observed in Annex 3 at 1:3 ratio; and the thermal bed can be observed in Annex 4 at 1:2 ratio

CALIBRATION

To observe how water circulation affects the thermal bed's temperature, four different tests are conducted. The tests consist of powering the water pump when the reservoir's desired temperature has been reached. This is done using the desired temperature as scale: for the first test the desired temperature is as is. Next, is the temperature minus 0.25°C; then 0.5°C and finally 1°C. Using this information, and analyzing the standard deviation, as well as the EST variable, the test with the best behavior can be identified. Using the information from the identified test, a delta between the average temperature of the reservoir as well as the average temperature of the thermal bed with water circulation can be obtained.

The standard experimental uncertainty is obtained by observing the device's operation under repeatability conditions. For the analysis, the following expression was employed [9].

$$s(q) = \sqrt{\frac{1}{n-1} \cdot \sum_{j=1}^{n} (q_j - \bar{q})^2}$$

Where n is the number of samples, q es the sample and is the average of the samples.

The EST variable is the contribution of the measured uncertainty by the temperature sensor in the thermal bath. It is represented by the following expression [10].

$$\frac{(Lsup - Linf)}{\sqrt{12}}$$

Where Lsup is the average of the maximum reservoir temperature and Linf is the average of the temperature in the thermal bed with water circulation. Using this methodology, the following data is obtained.

Maximum Reservoir Temperature	Water Pump HIGH	Standard Experimental Uncertainty (Thermal Bed)	*Est*
25 ºC	Setpoint – 0.25 ºC	±1.58 ºC	0.685ºC
25 ºC	Setpoint – 0.5 ºC	±1.65 ºC	0.71 ºC
25 ºC	Setpoint – 1 ºC	±0.98 ºC	0.6 ºC
25 ºC	Setpoint ºC	±0.643 ºC	0.596 ºC

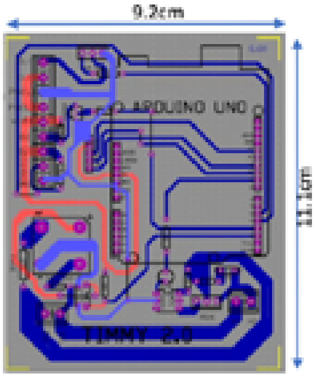
30 ºC	Setpoint – 0.25 ºC	±1.3 ºC	0.87ºC
30 ºC	Setpoint – 0.5 ºC	±2.23ºC	1.42ºC
30 ºC	Setpoint – 1 ºC	±1.76ºC	1.17ºC
30 ºC	Setpoint ºC	±1.2168ºC	1.2063ºC
35 ºC	Setpoint – 0.5 ºC	±0.683 ºC	0.62ºC
35 ºC	Setpoint – 0.25 ºC	±3.5ºC	1.44ºC
35 ºC	Setpoint – 1 ºC	±1.45ºC	1.13ºC
40 ºC	Setpoint – 0.25 ºC	±4.5 ºC	1.6ºC
40 ºC	Setpoint – 0.5 ºC	±2.3ºC	1.44ºC
40 ºC	Setpoint – 1 ºC	±3.46ºC	1.96ºC
40 ºC	Setpoint	±4.5ºC	2.37ºC
45 ºC	Setpoint – 0.25 ºC	±6.85 ºC	3.03ºC
45 ºC	Setpoint – 0.5 ºC	±0.5ºC	1.46ºC
45 ºC	Setpoint – 1 ºC	±5ºC	2.1ºC
45 ºC	Setpoint	±2.76ºC	2.3ºC
50 ºC	Setpoint – 0.25 ºC	±4.94 ºC	3.3ºC
50 ºC	Setpoint – 1 ºC	±4.6ºC	2.6ºC
55 ºC	Setpoint – 0.5 ºC	±6 ºC	3.9ºC
55 ºC	Setpoint – 1 ºC	±5.4ºC	3.4ºC

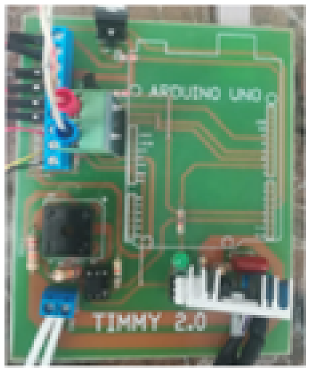
Using the information from the table, the best data set for each reservoir temperature is identified to be implemented in the Arduino code. This is done to obtain the best response from the thermal bed once the user sends a temperature command to the thermal bed.

RESULTS

PCB

The PCB implemented as well as the implementation is shown in the following illustration





A higher resolution image can be observed in Annex 5.

REGRESSION

For the regression, the data was taken into account from the time the water starts circulating; the heating process is then analyzed. The analysis is done using Sigmoid functions, since the behavior of these closely resembles the behavior of the heating process of the thermal bath. In the beginning, the water heats up rapidly; the heating decreases over time until it stabilizes at the desired temperature. The Sigmoid function constants are obtained using the python scipy.optimize library. The constants obtained are from the following expression.

$$\frac{a}{b + e^{-tc}}$$

Temperature Range	Water Pump HIGH	Regression *	ΔT *
Temp objective <= 25 ºC	Setpoint - 1 ºC	$y(t) = \frac{1.243e+02}{5.386+e^{-t*7.051e-03}}$	2 ºC
25 ºC < Temp obj <= 30ºC	Setpoint – 0.25 ºC	$y(t) = \frac{8.903e+01}{3.205+e^{-t*1.0915e-02}}$	3 ºC
30 ºC < Temp obj <= 35ºC	Setpoint – 0.5 ºC	$y(t) = \frac{5.491e+01}{1.701+e^{-t*6.376e-03}}$	2,8 ºC
35 ºC < Temp obj <= 40ºC	Setpoint – 0.5 ºC	$y(t) = \frac{3.773e+01}{1.032+e^{-t*6.625e-03}}$	5 ºC
40 ºC < Temp obj <= 45ºC	Setpoint – 0.5 ºC	$y(t) = \frac{6.547e+01}{1.634+e^{-t*6.608e-03}}$	5 ºC
45 ºC < Temp obj <= 50ºC	Setpoint – 1 ºC	$y(t) = \frac{3.567e+01}{7.825e-01+e^{3.262e-03}}$	5 ºC
50 ºC < Temp obj <= 60ºC	Setpoint – 1 ºC	$y(t) = \frac{4.0856e+01}{8.681e-01+e^{2.775e-03}}$	5 ºC

The regressions are implemented in the Arduino code, after which the device characterization is done to evaluate the data obtained and make sure that the Sigmoid equations closely match the temperatures of the thermal bed.

CHARACTERIZATION

The characterization is performed once the Sigmoid equations are implemented to observe the how they affect the performance of the thermal bath. The data can be observed below.

Desired Temperature	Water Pump HIGH	Sample Time	Circulation Time	Maximum Reservoir Temperature	Average Thermal Bath Temperature
23.6 ºC	Setpoint – 1 ºC	13.5 min	8.3 min	25.8 ºC	23.07 ºC
27.5 ºC	Setpoint – 0.25 ºC	20 min	15 min	29.4 ºC	27.4 ºC
46.5 ºC	Setpoint – 1 ºC	39 min	22 min	52.2 ºC	46.7 ºC
32.7 ºC	Setpoint – 0.5 ºC	22 min	15min	35.4 ºC	32.4 ºC
38.3 ºC	Setpoint – 0.5 ºC	32 min	19min	43.4C	38.9 ºC
43 ºC	Setpoint – 0.5 ºC	39 min	22min	46.9 ºC	43.45 ºC
57.5ºC	Setpoint – 1 ºC	65min	42min	61.43 ºC	57.54 ºC

The graphs obtained by the thermal bath's characterization can be observed in Annex 6-12.

OBSERVATIONS

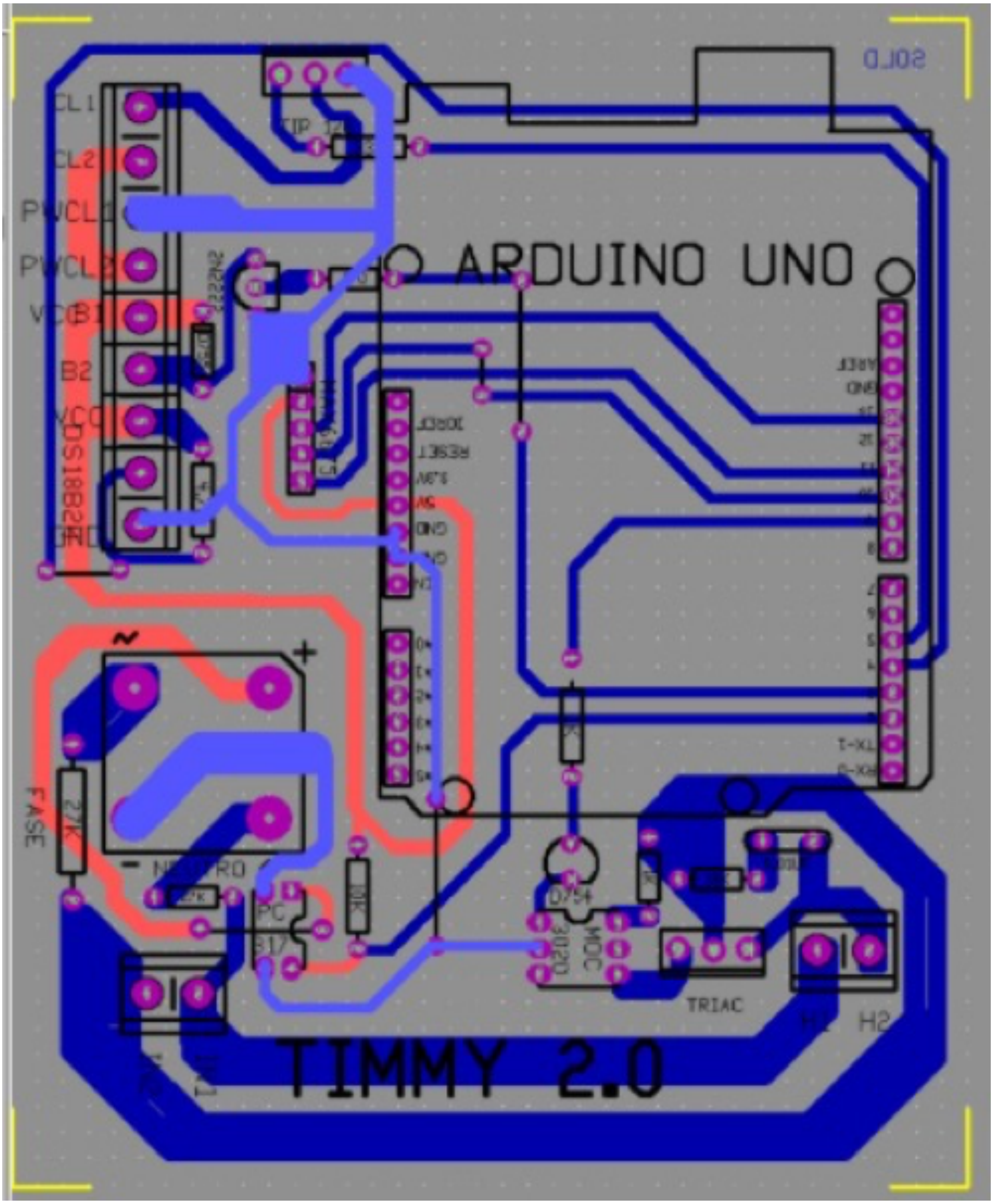
It can be observed that the average temperature of the thermal bath matches the desired temperature controlled by the Arduino. It should be noted that for temperatures in the range below 35°C the testing time is about 25 minutes, while for temperatures above 35°C, the testing time can reach up to 1 hour. For replicability, it should also be noted that the thermal bed is located at a height of 24.7cm above the heated reservoir.

It should also be noted that the water pump should only be used for about 10 minutes at a time. Intervals of operation of more than 10 minutes can cause damage within the water pump and as such render it unusable. Another restriction that this implementation poses is that it is not thermically isolated from the environment. This is because the user should be able to take both photo and video captures of the microfluidic system. As such, the heating process takes longer than if it were thermically isolated. Once the water starts circulating from the heated reservoir to the thermal bath, there is a temperature difference between that in the reservoir and the water which arrives at the reservoir after circulating. For a range of temperature between 20°C to 35°C there is a delta of 2.6°C. While for temperatures in the range of 36°C to 60°C, the temperature difference can be as much as 10°C.

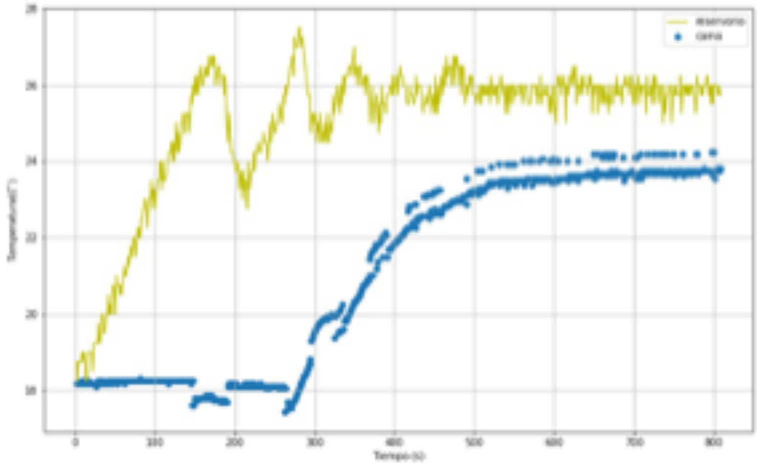
REFERENCES

1. Fricaval89. (n.d.). Baños Termostaticos para Laboratorio. FRICAVAL. Retrieved September 9, 2021, from http://fricaval89.com/productos/aparatos-instrumentos-materiales-equipos/banos/banos-termostaticos-laboratorio.html.
2. Vistronica. (n.d.). Módulo para Termocupla MAX6675. VISTRONICA S.A.S. Retrieved September 9, 2021, from https://www.vistronica.com/sensores/temperatura/modulo-para-termocupla-max6675-detail.html.
3. Optoacoplador MOC3020. AV Electronics. (2021, February 8). Retrieved September 9, 2021, from https://avelectronics.cc/producto/optoacoplador-moc3020/.
4. What is Optocoupler and how it works? Components101. (n.d.). Retrieved September 9, 2021, from https://components101.com/articles/what-is-optocoupler-and-how-it-works.
5. SEN DS18B20 2M. Sigma Electrónica. (2021, September 9). Retrieved September 9, 2021, from https://www.sigmaelectronica.net/producto/sen-ds18b20-2m/.
6. ¿Qué es Arduino? Arduino.cl - Compra tu Arduino en Línea. (2021, July 12). Retrieved September 9, 2021, from https://arduino.cl/que-es-arduino/.
7. Rodríguez de Luis, E. (2018, September 18). De cero a maker: todo lo necesario para empezar con Raspberry Pi. Raspberry Pi: todo lo necesario para iniciarse como maker desde cero. Retrieved September 9, 2021, from https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi.
8. Motors 1. Myzen. (n.d.). Retrieved September 9, 2021, from http://www.thebox.myzen.co.uk/Workshop/Motors_1.html.
9. Schmid, W. A., & Lazos Martinez, R. J. (2004, April). GUÍA PARA ESTIMAR LA INCERTIDUMBRE DE LA MEDICIÓN. Universidad Nacional Autónoma de México. Retrieved September 9, 2021, from http://depa.fquim.unam.mx/amyd/archivero/EstimaciondelaIncertidumbre_32954.pdf.
10. Guía técnica sobre trazabilidad e incertidumbre de las mediciones en la caracterización térmica de baños y hornos de temperatura controlada, CENAM, 2008.

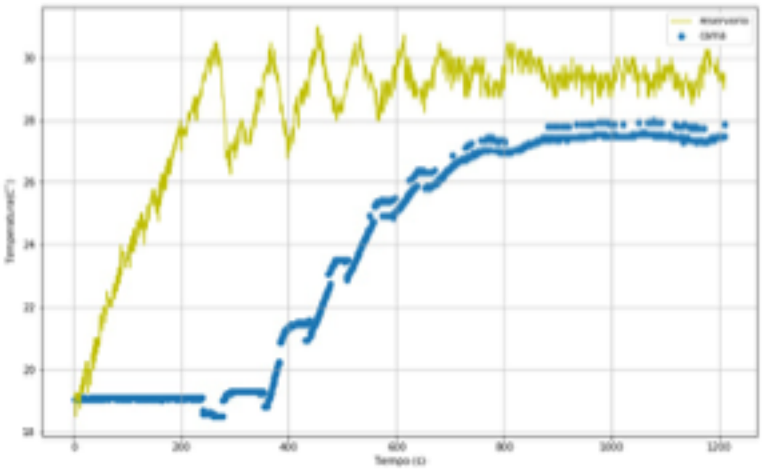
ANEXXES


CL1
CL2
PWCL1
PWCL3
VCCB1
B2
VCC
DS18B20
GND
TIP 120
ARDUINO UNO
27K
FASE
NEUTRO
PC
817
10K
MOC
3020
TRIAC
H1
H2
TIMMY 2.0

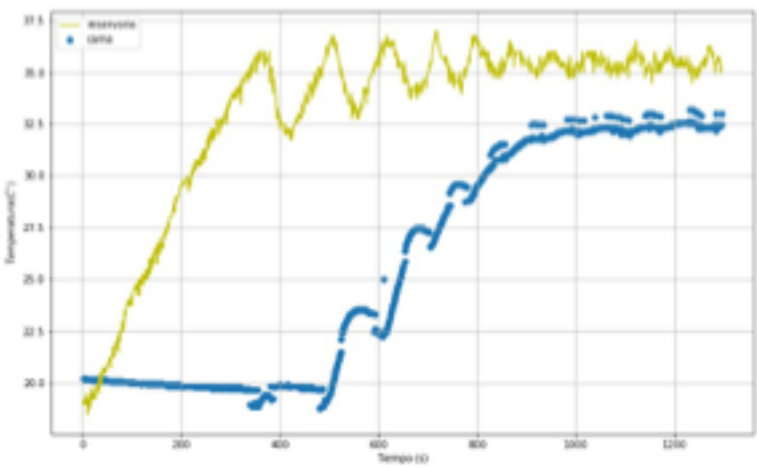
DEVICE CHARACTERIZATION AT 27.5 °C



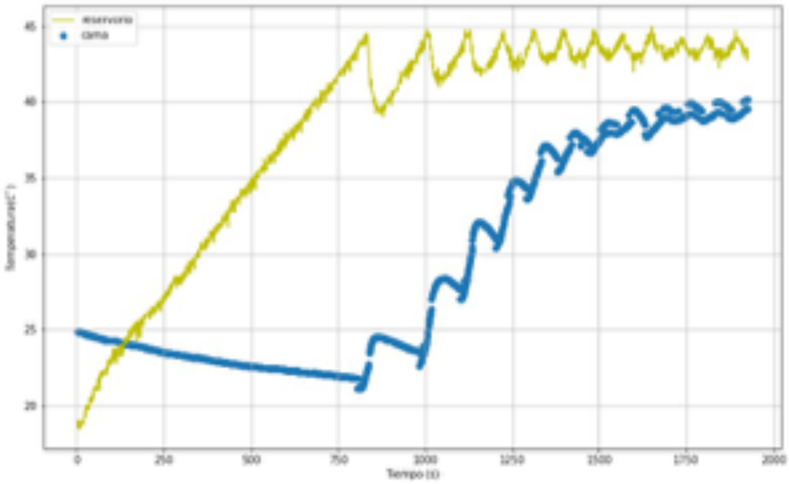
DEVICE CHARACTERIZATION AT 32.7 °C



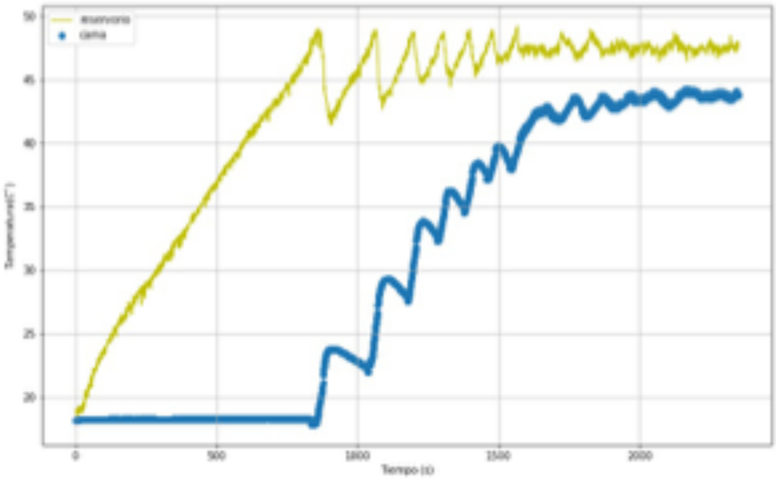
DEVICE CHARACTERIZATION AT 38.3 °C



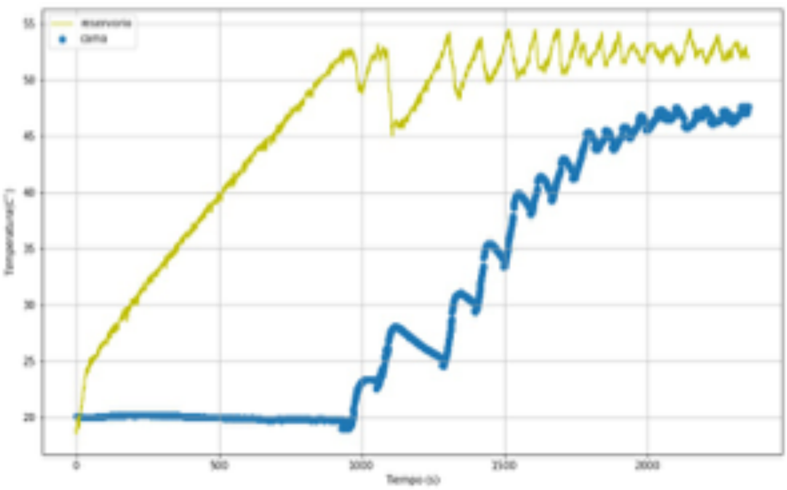
DEVICE CHARACTERIZATION AT 43.0 °C



DEVICE CHARACTERIZATION AT 46.5 °C



DEVICE CHARACTERIZATION AT 57.5 °C



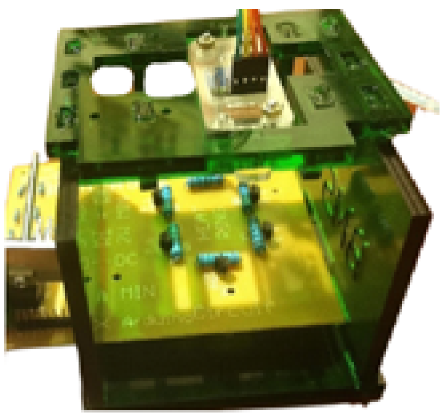
THERMAL CAMERA

Thermal cameras are electronic devices which detect infrared wavelengths and allow to detect heat signatures [1]. One such use, is in infrared spectrophotometry, where information about exothermic or endothermic chemical reactions can prove important in determining the molecules present within compounds.

The implementation consists of a device that facilitates the observation of thermal reactions in microfluidic systems, using an AMG8833 thermal camera which uses the near-infrared spectrum (NIR: 780nm – 3000nm). The implementation uses a Raspberry PI to process the information from the thermal camera, and to display the information through a monitor connected through HDMI. The Raspberry PI also serves as a hub which not only runs the software, but serves the purpose of a server, which allows the user to export the data obtained by the thermal camera through the thermal imaging process. It also allows the user to interact with an Arduino microcontroller through a serial monitor interface, which serves the same purpose as the Arduino IDE monitor.

The calibration employed as well as the software implemented are also shown. The calibration is done through linear regression, using the temperature average of the 4 center pixels of the AMG8833 thermal camera. A calibrator circuit which is controlled by the Arduino microcontroller is proposed as to allow for temperature adjustments without having to resort to heating plates or thermal baths as a temperature reference. The calibrator circuit, uses current mirrors to heat resistors to different temperatures, creating a temperature gradient ring, which the Raspberry PI can interpret as temperature references for the linear regression curve.

https://github.com/Biomicrosystems/Documentation/tree/main/DIY%20Book/Thermal%20Bath



MATERIALS

1x AMG8833 Thermal Camera
1x 5V Relay
1x ULN2003
1x LM7805
1x 100pF Capacitor
10x 1.5 Ohm Resistor 1/4W
1x 470 Ohm resistor 2W
1x 270 Ohm resistor 2W
1x 200 Ohm resistor 2W
1x 150 Ohm resistor 2W
1x 120 Ohm resistor 2W
1x 100 Ohm resistor 2W
6x LM35
1x AC-DC 12V 1.5A Adapter

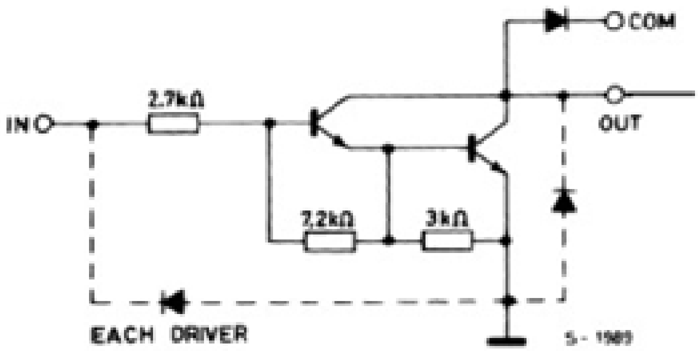
COMPONENT EXPLANATION

AMG 8833

The AMG833 is a thermal sensor which measures temperatures from 0°C to 80°C at up to 7 meters. The sensor which is developed by Panasonic features an 8x8 grid which allows to sense temperatures at about 10 fps. The sensor is able to communicate through I2C protocol and is compatible with both the Raspberry PI as well as an Arduino microcontroller. It has a sensing uncertainty of 0.25°C, and is able to operate in the range of 3.3-5V. It should be noted that the sensor has a 15 second delay after start up to stabilize the output temperatures being measured by the thermistors [2]. It should also be noted that for the sensor to function properly, the sensor has to be calibration with respect to the distance from the sensed object.

ULN 2003

The ULN2003 is an integrated circuit which is composed by 7 identical and independent drivers, which can drive up to 500mA; each driver is composed by a Darlington Pair. The circuit has a maximum Vce of 50V, while to get the drivers into saturation it only requires between 1V and 1.3V [3]. For this implementation, the ULN2003 integrated circuit is operated as a current mirror. The implementation was opted using this circuit, due to the fact that for the current mirror to function properly, the transistors used need to have the same characteristics and the same gain (°) values. It should be noted that transistor arrays such as the CA3046 NPN array as well as the CA3081 in emitter configuration would work just as well.
The following illustration shows the Darlington pair of one of the seven drivers in the ULN2003 [4].



ARDUINO

An Arduino is a microcontroller board based on the ATmega328P. It has an unregulated power source in the VIN pin which supports 7-12V, and a regulated 5V power source [5]. For this implantation the Arduino is the device which controls the temperature gradient ring which serves as the AMG8833's calibration circuit. The Arduino microcontroller operates the relay which closes the current mirror circuit and allows for the gradient ring to heat up, as well as take the measurements from the LM35 temperature sensors in each of the ring's 2W resistors to correlate the data. The Arduino is this implementation communicates with the Raspberry PI through the serial's write() and read() functions.

RASPBERRY PI

The Raspberry Pi is a low-cost, small-size computer which allows the user to run code [6]. For this implementation, the software as well as the serial interface with the Arduino microcontroller are implemented through python code. In this implementation, the job of the Raspberry PI is that of displaying the 8x8 temperature matrix obtained through the AMG8833, as well as the Arduino's IDE monitor. The Raspberry PI uses the Serial communication library to emulate the Arduino's monitor. The software also allows capture images and record video of the temperature matrix. The option to connect and control the software through a local network is also proposed.

CALIBRATION CIRCUIT

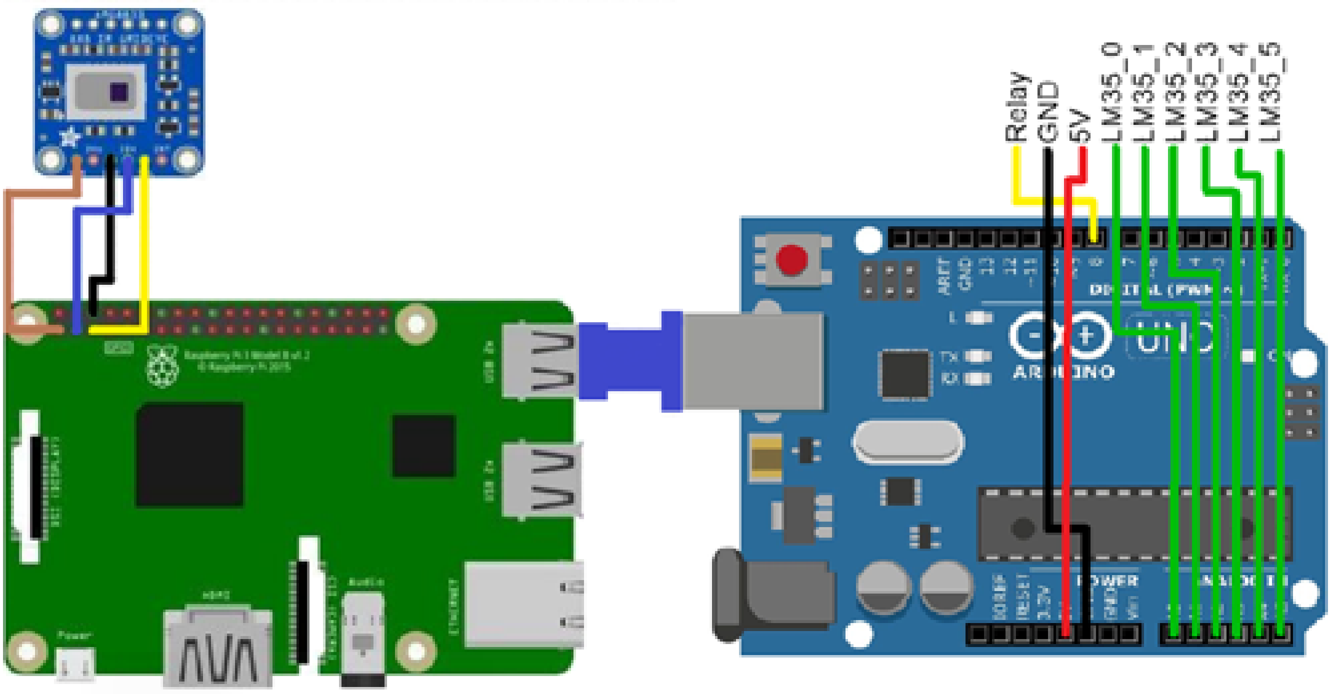
The calibration circuit implemented consists of a series of current mirrors which are connected to a series of resistors, which heat up as the current passes through them. The circuit has pins to interact with the Arduino microcontroller as well as a terminal block to connect the 12V 1.5 AC-DC adapter, which provides the current for the temperature gradient ring.

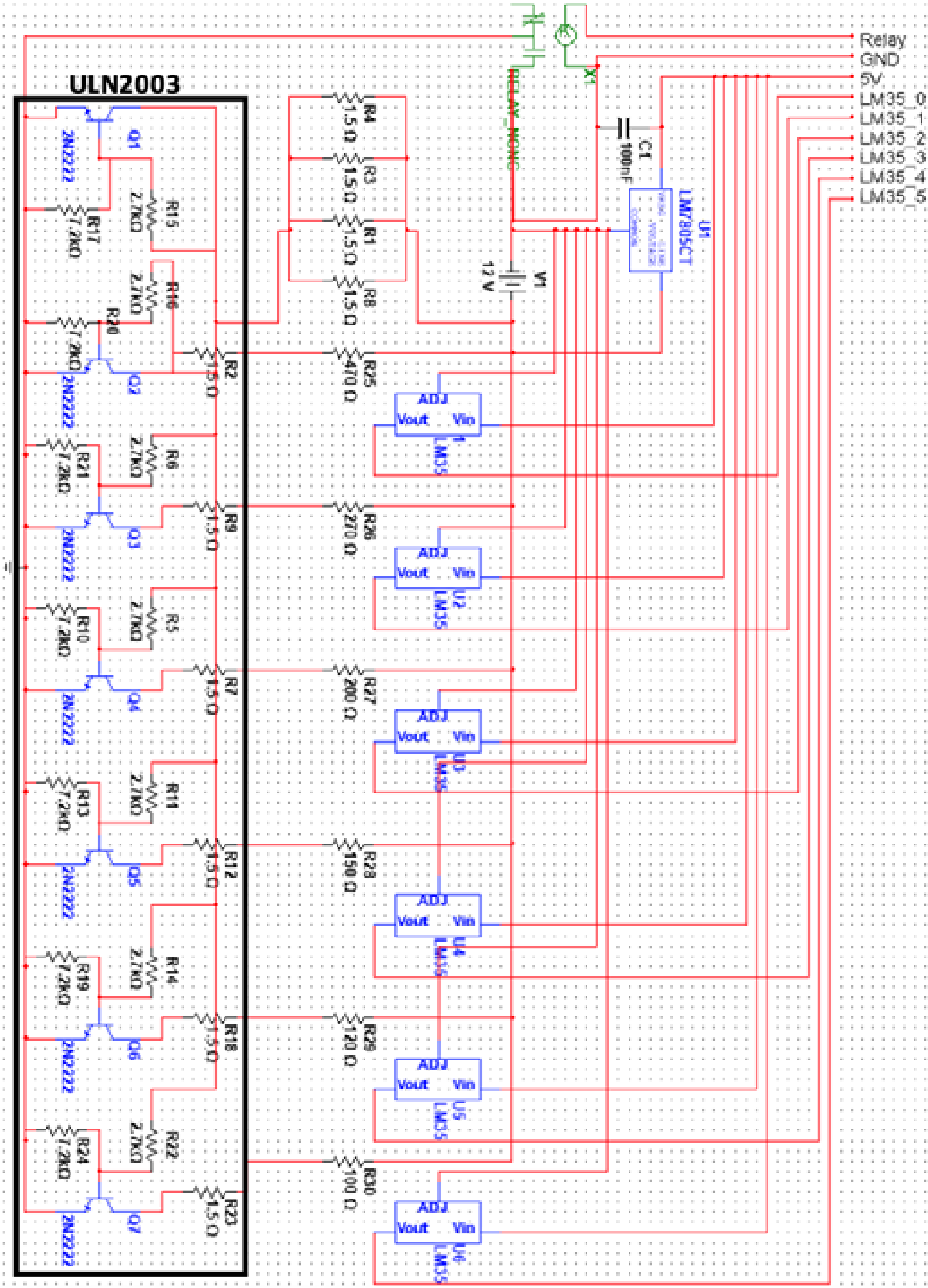
The Arduino has a pin connected to the circuit's relay which grounds the current mirror and allows for the current to flow through the 2W resistors. It should be noted that the relay grounds all of the drivers in the ULN2003 at the same time, since the drivers are independent. The time the current flows through the resistors can be modified in the Arduino code to alter the temperature reached by the gradient ring. It should also be noted that the ULN2003 may get hot, as each of the drivers will experience approximately 250mA for the duration that the relay remains active. As such, a heat sink may need to me employed.

The temperature gradient ring has LM35 sensors in close proximity (1mm) to each of the 2W resistors, to get the resistor temperature and allow for a correlation between the LM35 values and the values measured by the AMG8833 thermal camera. It should be noted that the temperature gradient ring shape can be modified, as long as all of the resistors are visible in the 8x8 matrix and are not too close together as to create temperature overlapping.
The circuit diagram can be observed in illustration 4.
The left section which incorporates the transistors, is an emulation of the ULN2003 as such, the ULN2003 would replace that section of the circuit.

OVERALL DIAGRAM

The overall circuit diagram with both the Arduino microcontroller as well as the Raspberry PI and the calibration circuit are shown below. It should be noted that for the proper operation of the device, a mouse, a keyboard as well as a screen monitor should be connected to the Raspberry PI to allow the user to interact with the thermal camera software. The overall implementation is to use two AC-DC adapters, since one powers the Raspberry PI and the second one is to power the calibration circuit, which is shown as a 12V power source in the llustration.

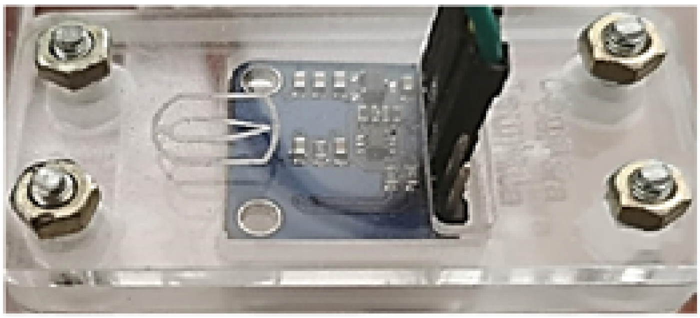



ULN2003
Relay
GND
5V
LM35_0
LM35_1
LM35_2
LM35_3
LM35_4
LM35_5
U1
LM7805CT
C1
100nF
V1
12 V
X1
RELAY_NONC
Q1
Q2
Q3
Q4
Q5
Q6
Q7
2N2222
ADJ
Vout
Vin
LM35
R25
470 Ω
R26
270 Ω
R27
200 Ω
R28
150 Ω
R29
120 Ω
R30
100 Ω

IMPLEMENTATION

CASING

The casing for the thermal camera is done using acrylic. The devices casing is implemented using 5mm acrylic, while the AMG8833 has its own separate casing as to protect the thermal camera using 2mm acrylic. The devices casing holds the AMG8833 casing as well as the calibration circuit. The 5mm acrylic casing can also hold a microfluidic system at a specified distance from the thermal camera's lens.



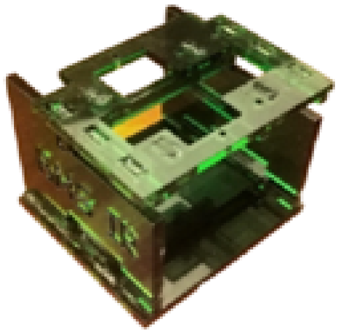
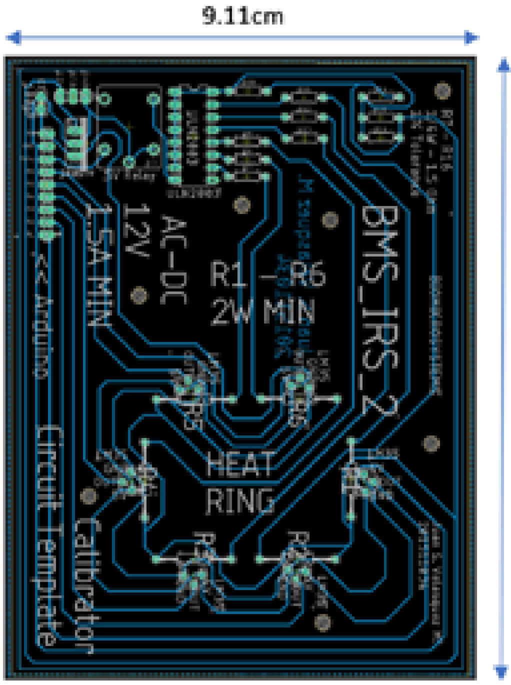
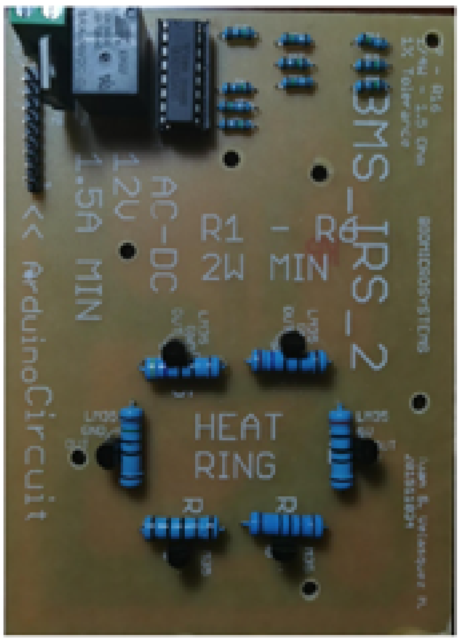


Illustration on the left shows the AMG8833's 2mm casing; the illustration on the right shows the devices 5mm acrylic casing. The 2mm and 5mm casing designs can be observed in Annex 1 and 3, respectively. The microfluidic system design can also be observed in Annex 2. It should be noted that acrylic is transparent for a portion of the wavelength in the NIR spectrum, as such, using the device in an environment with thermal noise, could create uncertainties in the measuring process.

PCB

The calibration circuit's PCB is shown below. The PCB has anchor points which allow the circuit to be secured to the 5mm acrylic casing. It should be noted that the heat ring's position is absolute in the Raspberry PI's code, thus, if the circuit's 2W resistor positions are modified, the code should also be modified to account for the changes. A higher resolution diagram can be observed in Annex 4.



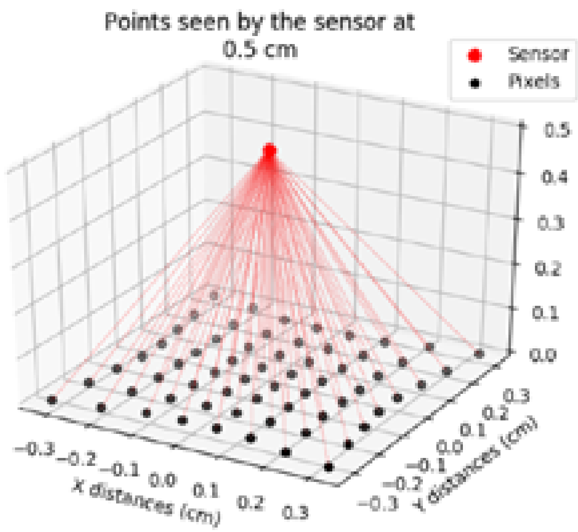


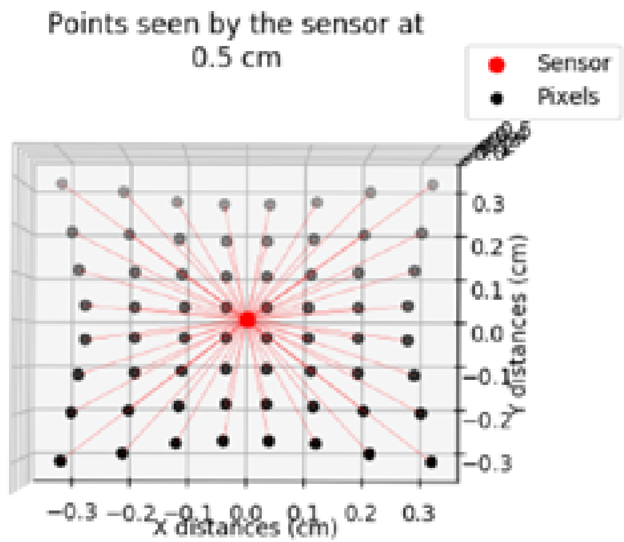
CALIBRATION

AMG8833 GEOMETRY

To properly calibrate the thermal camera, it is necessary to observe the laser distribution of the AMG8833 sensor. [2] shows the laser distribution as a function of their angle. A python code is implemented to transform the laser distribution as a function of distance. The python code can be observed in Annex 5. This tool allows to observe what the AMG8833 sees at a certain distance; this information allows to know which pixel readings accurately represent the thermal images obtained from measuring the microfluidic system. This implementation is needed as the area covered by the thermal camera is not a square, but a shape with elongated corners.

The python code also allows to observe which pixels of the AMG8833 read the 2W resistors in the calibration circuit. This allows the device to correlate the information obtained form the LM35 sensors in the calibration circuit, with the temperature measured by the thermal camera.
The following illustrations show the laser distribution of the AMG8833 sensor at a distance of 0.5cm using the python code implemented.
It should be noted that the distance between the thermal camera and the microfluidic system is optimized to allow the lasers to contact the microfluidic system's canal.





AMG8833 DISTANCE VS. TEMPERATURE

Since the thermal camera uses laser refraction to measure the temperature of the desired surface, the distance from the lens to the surface creates uncertainties in the measured value. Thus, it is necessary to create a calibration curve to account for the laser refraction variations due to distance.

This process can be recreated by placing the camera at a known distance from a heated surface. The calibration to account for the refraction variations was done at both 0.2cm as well as at 4.75cm. The former is the distance between the thermal camera and the microfluidic system and the latter is the distance between the thermal camera and the calibration circuit's gradient ring. The calibration process was conducted using the same temperatures at both distances to create a calibration curve which accounts for the refraction variations.

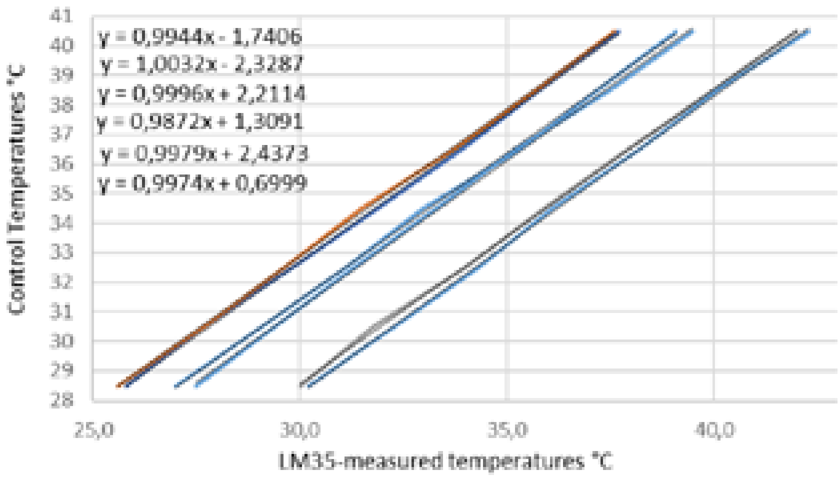
This calibration curve would also allow to correlate the data obtained by measuring the temperature gradient ring at 4.75cm and infer how the sensor should compensate to properly measure the temperatures at 0.2cm and corroborate the calibration process. This calibration was conducted at both 37°C and 52°C as follows.
These correction values allow for an estimate at any other temperature at 0.2cm. However, it should be noted that more points would provide for a more accurate prediction, using linear regression. Using this information, the correction for a temperature of 35°C at 0.2cm would approximately 5.1°C.

Distance	Temperature	Equation	Correction
0.2 cm	52°C	$Y = \left(54.295 * (0.2^{-0.077})\right) - X$	9.4583°C
0.2 cm	37°C	$Y = \left(38.931 * (0.2^{-0.056})\right) - X$	5.6028°C

CALIBRATION CIRCUIT LM35 SENSORS

For the device to correctly correlate the data form the calibration circuit, the temperature sensors in the circuit need to be calibrated. The calibration is conducted using a heated surface as well as a commercially calibrated thermal camera or thermometer. Each of the LM35 sensors are placed in contact with the heated surface and a linear regression is conducted in the range of 20°C to 60°C. The temperature range can be defined by allowing current to flow through the heat ring for a specific amount of time and measuring the heat produced by each for the resistors using the commercial thermal camera or high-resolution thermometer.

The linear regression equations obtained for each of the temperature sensors, can then be implemented into the Arduino microcontroller to compensate for measurement uncertainties in the heat ring. It should be noted that the higher the resolution of the device used for this calibration, the better the correlation between the calibration circuit and the thermal camera.



Although the LM35 sensors show a linear response, some have an offset. The data obtained show a maximum offset response of approximately 2.4°C. To further improve the measurements of the gradient heat ring by the LM35 sensors, thermal paste can be applied to increase the thermal conductivity to the temperature sensor. The thermal paste decreases the variations of the measurements as can be observed as follows. An illustration of the HY510 thermal paste used is also shown.

The gradient heat ring's resistor temperatures after 50 seconds of operation are observed above, using the HY510 thermal paste. Four measurements were taken to make sure that the resistors reach the same temperature after a certain operation time. The calibration circuit had a maximum variation of 9.2%, however, it should be noted that the variation is also seen by the AMG8833 thermal camera, thus, making the correlation possible.

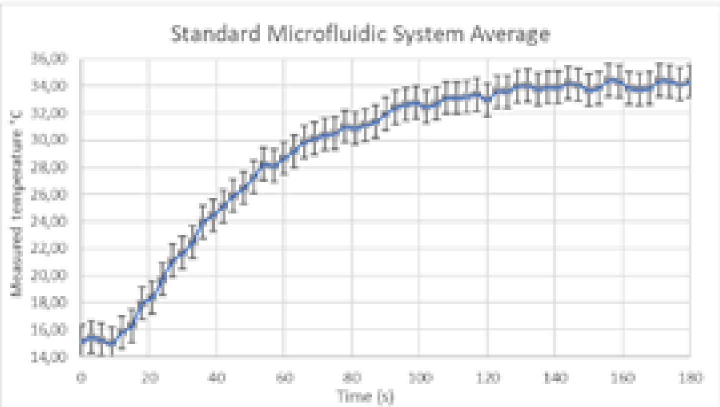
LM35 measurements with thermal paste °C				
X1	X2	X3	X4	Average
38,48	37,67	38,75	37,82	38,18
32,26	31,84	31,76	33,51	32,3425
49,85	49,09	50,71	51,74	50,3475
44,18	41,74	40,3	42,47	42,1725
50,07	52,35	52,81	51,48	51,6775
35,64	37,87	38,37	36,47	37,0875

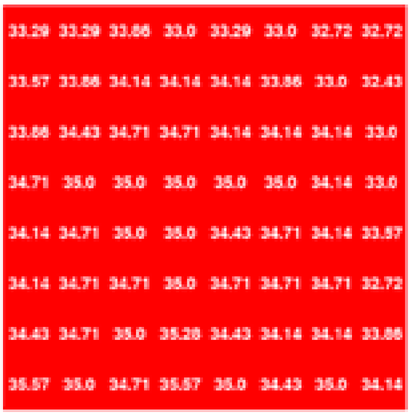
LINEAR REGRESSION

MANUAL REGRESSION

The linear regression process implemented in the software for the AMG8833 thermal camera consists of a .txt file which holds the constant of the Y = mX + b, where x is the average temperature of the four center pixels of the AMG8833 thermal camera. The calibration using this method takes a snapshot of the temperature array measured by the sensor and using the linregress function from the SciPy python library.

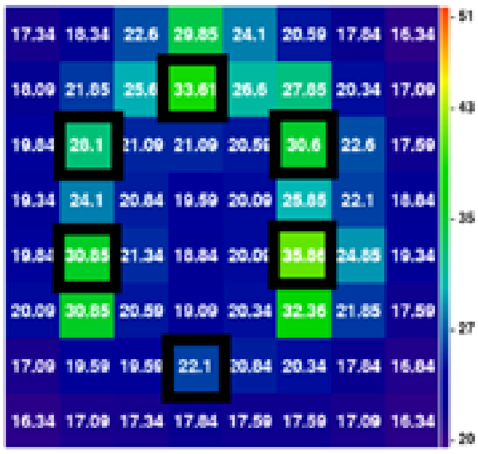
As such, the linear regression equation if no calibration is done to the device, is Y = 0.25X. Using the manual regression to calibrate the device, the equation obtained is Y = 0.2531X – 4.7765, which is then applied to a surface at a distance of 0.2cm. Using these calibration constants, the device is then used to measure the temperature of a microfluidic system at 35°C to observe is the calibration process is reliable. It should be noted that the 'b' constant already has the distance vs. temperature calibration correction of 5.1°C for 0.2cm at 35°C, resulting in a value of -4.7765.
Although there are some variations, the maximum relative error present in the temperature array is approximately 5.7%. Thus, the calibration process is proven reliable. This calibration process is then scaled for the temperature gradient ring.

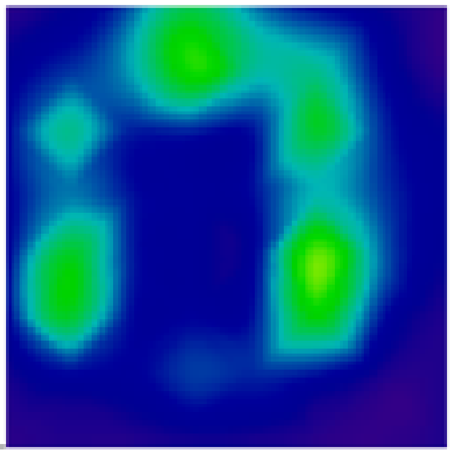




TEMPERATURE GRADIENT RING

This process is then automated using the calibration circuit to create the linear regression using the linregress function of the SciPy library. The calibration circuit and the AMG8833 thermal camera are placed in the casing and the pixels which correspond to the resistors are identified. Annex 6 shows the pixel number as well as the pin number which corresponds to the Arduino microcontroller. The pixel number is important, as the python code which measures the temperature array needs to correlate the resistor's pixel with the temperature measured to properly create a calibration curve using the data obtained by the calibration circuit.





It should be noted, that the illustration shown does not represent the temperature of the gradient heat ring after the 50 seconds of operation, but rather it shows the gradient heat ring as it heats up. Both illustrations are obtained using the software's .png capture feature.

Using this implementation, the calibration process is done four times to corroborate if it is reliable. The calibration equation obtained is Y = 0.274X – 4.955, which shows that there is about an 8% variation when compared with the manual linear regression calibration process performed.

CODE

ARDUINO

The Arduino code makes use of Serial communication to interpret the commands which the user sends through the Raspberry PI to control the calibration circuit. It records the data obtained by the calibration circuit and stores it using the EEPROM library. The device allows for various commands such as clc (clear), hlp (help), get and cal (calibrate).

It should be noted that if the command sent indicates for the Arduino to start the calibration process using the calibration circuit, the Arduino will be unresponsive for the duration of the calibration. Meaning, while the resistor's heat up, the Arduino can receive commands, but will not process them until the relay which controls the current flow to the calibration circuit's resistors is turned off. After which, the user can ask the Arduino for the data using the get command.

It should be noted that all of the LM35 calibration equations are implemented at the moment the Arduino measures the resistor's temperature. The Code can be observed in Annex 7.

INTERFACE

The device's interface consists of a window created using PyGame which shows the heat map seen by the thermal camera. It also shows, a list menu which allows to capture both images and video, a slider which allows to modify the pixelated heat map into a more detailed image by using interpolation, as well as a settings menu, which allows to modify the parameters of the device.

The interpolation is implemented using the SciPy interpolation library which allows to modify the pixelated heat which shows an 8x8 matrix to a 64x64 image; as seen in illustration 16. It should also be noted that the illustrations 15 and 16 are not the same image, as the device and interface work in real time. Thus, it is only possible to observe either the pixelated heat map or the interpolated image. The photo and video capture are implemented using a subprocess, thus the device takes snapshots of the interface to create the different files.
It should be noted that if an implementation is done using a different library which allows to capture both photo and video directly using the information form the AMG8833 thermal camera, it'll be possible to capture both types of images. The device allows to choose between different extension types when capturing video or saving data, whether it be in a .png format or in text such as .txt and .csv. The video capture allows for both .mp4 as well as .mkv and either full screen or just the heat map. The device also features a file explorer using a tKinter file directory, which allows to select the folder which will store the files created by the device.

The interface settings screen allows to modify the temperature limits, which determine the color range observed by the user. Using the same file which holds the constants for the Y = mX +b calibration equation, the temperature limits are saved for when the device is power off. The settings screen also allows to access the manual linear regression calibration, which creates a linear regression using the linregress function of the SciPy library.

SERVER

The python code allows to start a socket connection which allows a user in a local network to control the device's interface and collect data using the AMG8833 thermal camera. The server also allows to export files to the client using a file explorer implemented through the client's command line. To show the user running the server, the interface shows if a client has connected; it shows both the status of the client as well as the local IP which connects to it. The server, however, only allows for a client to connect, once the server function is called by the server interface.

The server implementation recognizes commands which allow to modify the temperature limits as well as the data and video extension. When the user calls for data collection or video capture through the client, the server automatically sends the file created through the TCP connection. However, the server allows to export files of the extension supported by the device by using the file explorer through the command line.

For the client to successfully disconnect from the server, a command is also sent, which either disconnects the current user from the current socket connection, or shuts off the server. At which point if another client is to connect, the server needs to be restarted from the server interface.

SERIAL COMMUNICATION

The serial communication with the Arduino microcontroller is implemented using the Serial library. A textbox is implemented using PyGame which sends the communication to the microcontroller, which then answers with the command sent and operates the calibration circuit. It should be noted that for the serial communication to work, the microcontroller's address is needed for the Raspberry PI to know to which port the information is headed. For this implementation, the Arduino's port address is directly linked to the text box. This would allow to communicate with multiple microcontroller's by implementing more instances of the text box class.

When the command to get the calibration, information is sent to the Arduino, the Raspberry PI overwrites the calibration file, once the data is received. This calibration is done using the SciPy library.

CALIBRATION

The manual calibration process implemented in the interface, allows to take multiple readings (at least two) of heated surfaces to correct or adjust the measured temperature to create a linear regression equation. The calibration screen features a save button to create or modify the file with the calibration constants as well as arrows to adjust the measured temperature.

In the case of the calibration circuit data, once the Arduino sends the data, the software loads the data in vectors to create the calibration constants using the linregress function.

CLIENT CODE

The client implementation to control the device is done through the command line. The code allows to control the main implementation through a local network as well as modify the temperature constants for the heat map.

It features a help command which shows the user all of the commands supported by the code. The code can be observed in Annex 9.

RECOMMENDATIONS

Regarding the calibration process, the procedure should be performed din a place with little to no thermal noise apart from the heated surface. Since the AMG8833 behaves like an infrared thermometer, any thermal fluctuations in the air can throw off the reading and create variations which in turn create uncertainties in the collection of data with the device.

It should also be noted that the warm air which is a consequence of the heated surface can also create thermal fluctuations which create uncertainty in the data collected. As the temperature of the heated surface gets higher, it is more likely that the calibration process would have errors. Thus, it is recommended that for temperatures above 50°C, the calibration process should be performed three times to make sure that the calibration constants obtained are reliable and accurately represent the data. This also has to be considered when using the calibration circuit, which depending on the time of operation can reach temperatures of up to 80°C.

OBSERVATIONS

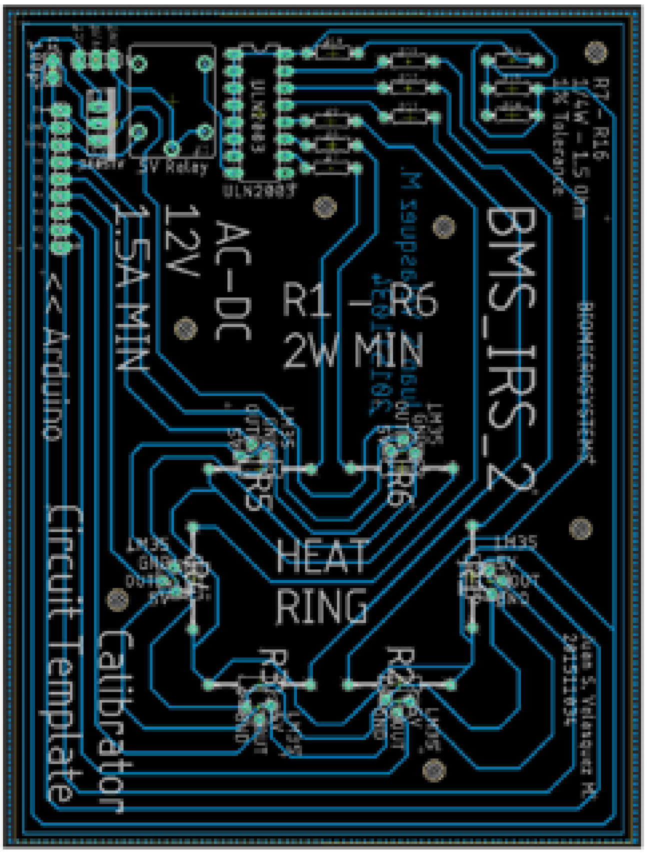
Regarding the implementation's code, it should be noted that the use of threads allows for the parallel operation the AMG8833 as well as the server which allows the interface to be controlled through a local network. Although this implementation allows for a very flexible data collection system, if the processor running the code has to run other processes, the program may become unresponsive. This software was used and tested as the only program running in the Raspberry PI at the time, thus allowing for very smooth data collection.

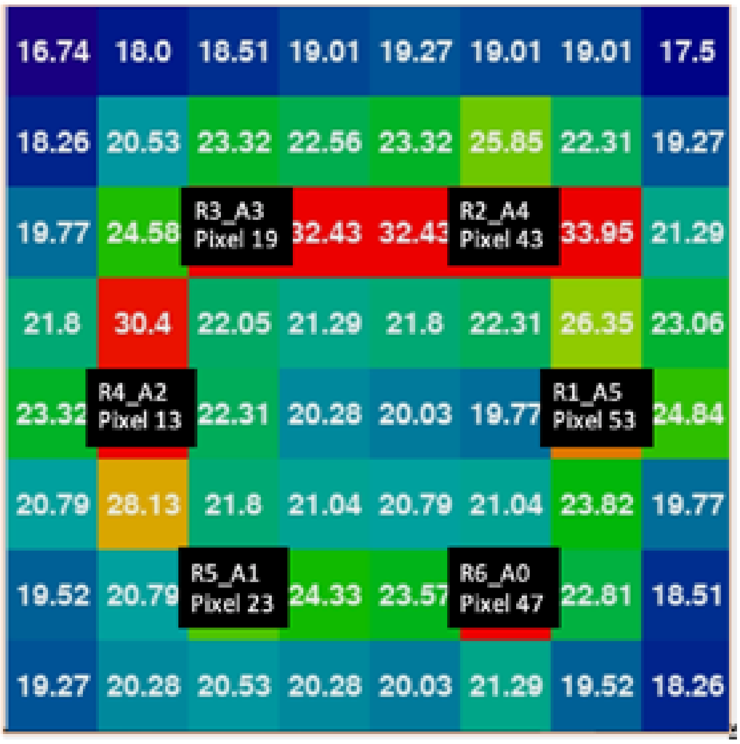
It should also be noted that the casing implemented allows for an Adafruit AS7262 6-channel visible light sensor next to the AMG8833 thermal camera. Using this implementation, would allow to make a simple spectrophotometer.

REFERENCES

1. Everything You Need To Know About Thermal Imaging Cameras. Everything You Need To Know About Thermal Imaging Cameras | RS Components. (n.d.). Retrieved September 9, 2021, from https://uk.rs-online.com/web/generalDisplay.html?id=ideas-and-advice%2Fthermal-imaging-cameras-guide.
2. Panasonic Corporation. (2011, August 30). SPECIFICATIONS FOR Infrared Array Sensor AMG8833. Google Drive. Retrieved September 9, 2021, from https://drive.google.com/file/d/1wcXSsOnkCn4Oi7sN5cG3tklMu7beDR9c/view.
3. Gabriel. (2018, February 23). EL ULN2003: DRIVER DE SALIDA PARA MICROCONTROLADORES. Inventable. Retrieved September 9, 2021, from https://www.inventable.eu/2018/02/09/uln2003-driver-salida-microcontroladores.
4. ULN2003A Darlington Array 7 NPN 500mA. Hobby Electronics. (n.d.). Retrieved September 9, 2021, from https://www.hobbytronics.co.uk/uln2003a-darlington-array.
5. ¿Qué es Arduino? Arduino. (2017, July 12). Retrieved September 9, 2021, from https://arduino.cl/que-es-arduino/.
6. Rodríguez de Luis, E. (2018, September 18). De cero a maker: todo lo necesario para empezar con Raspberry Pi. Raspberry Pi: todo lo necesario para iniciarse como maker desde cero. Retrieved September 9, 2021, from https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi.

ANEXXES


R7 - R16
1/4W - 1.5 Ohm
1% Tolerance
5V Relay
ULN2003
BMS_IRS_2
R1 - R6
2W MIN
AC-DC
12V
1.5A MIN
<< Arduino
HEAT
RING
Calibrator
Circuit Template


16.74 18.0 18.51 19.01 19.27 19.01 19.01 17.5
18.26 20.53 23.32 22.56 23.32 25.85 22.31 19.27
19.77 24.58
R3_A3
Pixel 19
32.43
R2_A4
Pixel 43
33.95 21.29
21.8 30.4 22.05 21.29 21.8 22.31 26.35 23.06
R4_A2
Pixel 13
22.31 20.28 20.03 19.77
R1_A5
Pixel 53
24.84
20.79 28.13 21.8 21.04 20.79 21.04 23.82 19.77
19.52 20.79
R5_A1
Pixel 23
24.33
R6_A0
Pixel 47
22.81 18.51
19.27 20.28 20.53 20.28 20.03 21.29 19.52 18.26

DIY LABS

GLOSARY

A

Astable oscilator
Oscillator producing a continuous square wave

C

Catalysis
Process through which the speed of a chemical reaction is increased.

Chemical reaction
Process in which one or more substances (reactants) become one or more different substances (products).

D

Dead volume
Volume in a syringe or microsystem that remains inside after being emptied. This may be due to the presence of air bubbles.

Duty cycle
Relationship between the time that a signal is active and that it is inactive. At 100% duty cycle the signal is always active.

E

Endothermic reaction
Chemical reaction characterized by increasing its enthalpy. These reactions need to take energy from your environment

Exothermic reaction
Chemical reaction characterized by releasing energy to the environment.

F

Fluidic microsystem
Set of microchannels through which fluids flow in small quantities. The shape of the channels determines the operation of the microsystem.

H

Heat sink
Device used to transfer heat to the environment, thus cooling electronic devices.

I

Infrared radiation
Electromagnetic wave whose wavelength is between 700 nm and 1,000,000 nm

M

Microchannel
Channel with a reduced diameter, generally present in a fluidic microsystem

Microfluidic
Science that studies the behavior of fluids at the micro / nanoliter scale, which circulate in artificial microsystems

Microliter
One thousandth of a milliliter

Milliliter
Amount of fluid that fits in a cube with a side 1 millimeter

N

Nanocomposites
Materials that incorporate nano-sized particles into a standard material matrix

Nanoliter
Millionth of a milliliter

Near infrared (NIR)
Infrared portion located between 780 nm and 3000 nm

O

Oscillator
Device capable of periodically and precisely generating delays or oscillations

P

Polymethylmethacrylate (PMMA)
Known commercially as acrylic, it is a transparent thermoplastic. It is a polymer of methyl methacrylate

R

Relay
Magnetic switch that changes state when it receives an electrical signal

T

Thermistor
Transistor whose resistance varies depending on the temperature

Thermostat
Device that measures temperature and performs actions to keep it at a desired level

Start being a Lab Maker