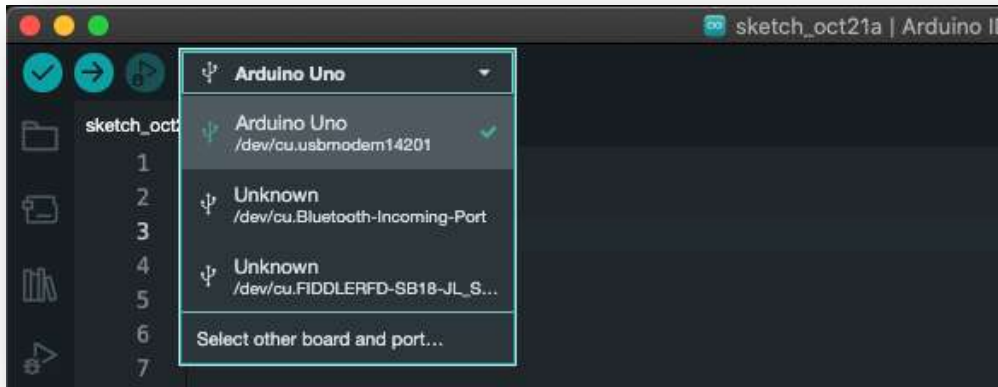


Arduino Hardware Connection

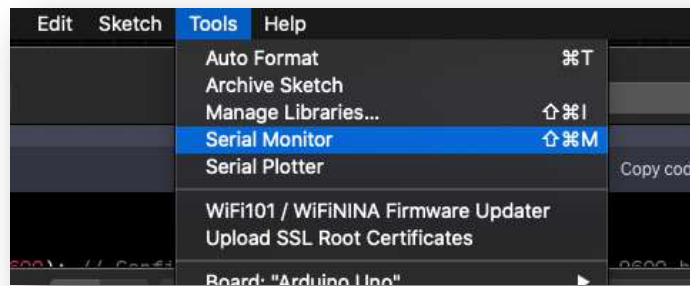
1. Preparing your Arduino to send data to the PCB Solution FBL software.
2. In the first example, we are going to send data automatically every 3 seconds. The goal is to see if PCB Solution FBL receives these values and adds them to the pins on the screen.
3. Connect your Arduino board and select your board from the list.



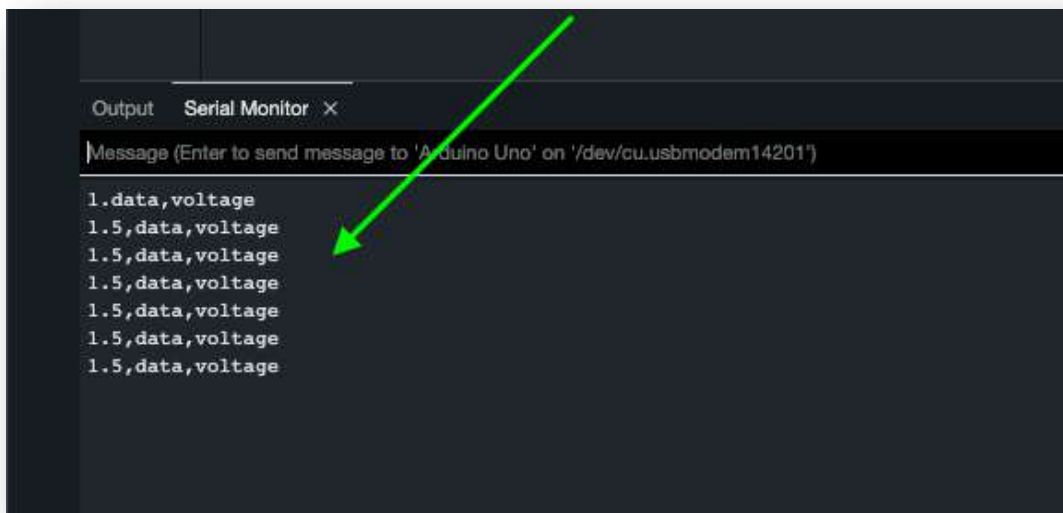
4. Upload the following code to your Arduino board by clicking the "Upload" button, which looks like a right-pointing arrow. (Do not connect anything additional to your Arduino board yet.)



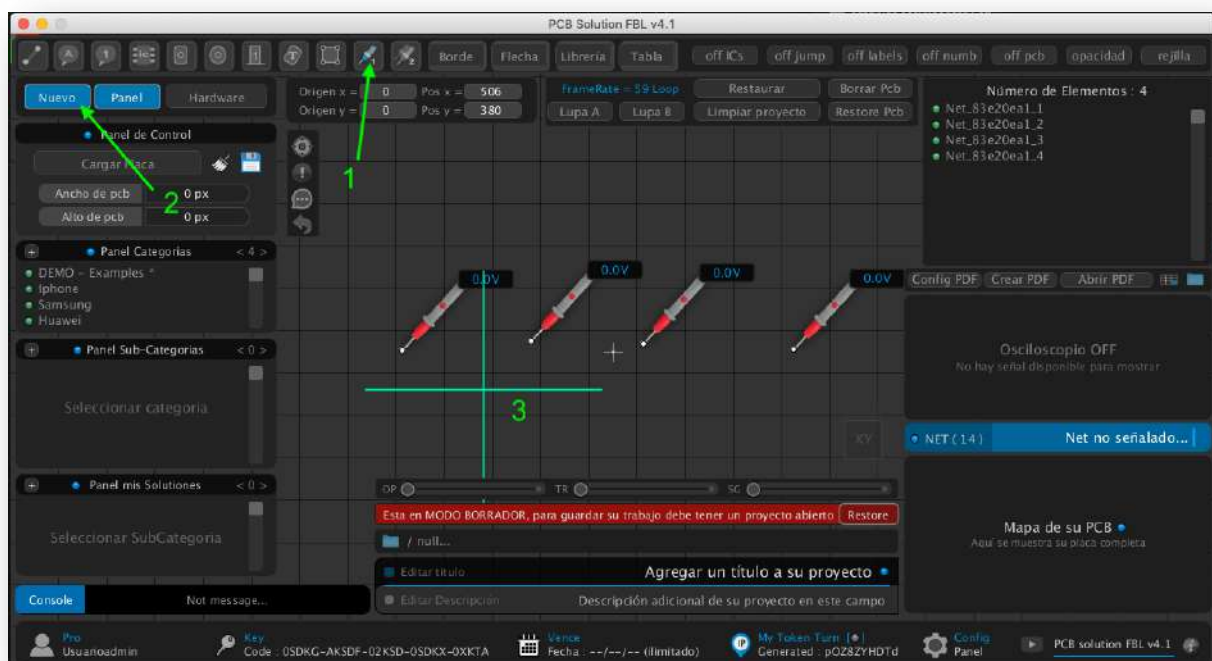
5. Now that your Arduino has the test code, run the Arduino Serial Monitor to see if the board sends data every 3 seconds. You should see something like this.



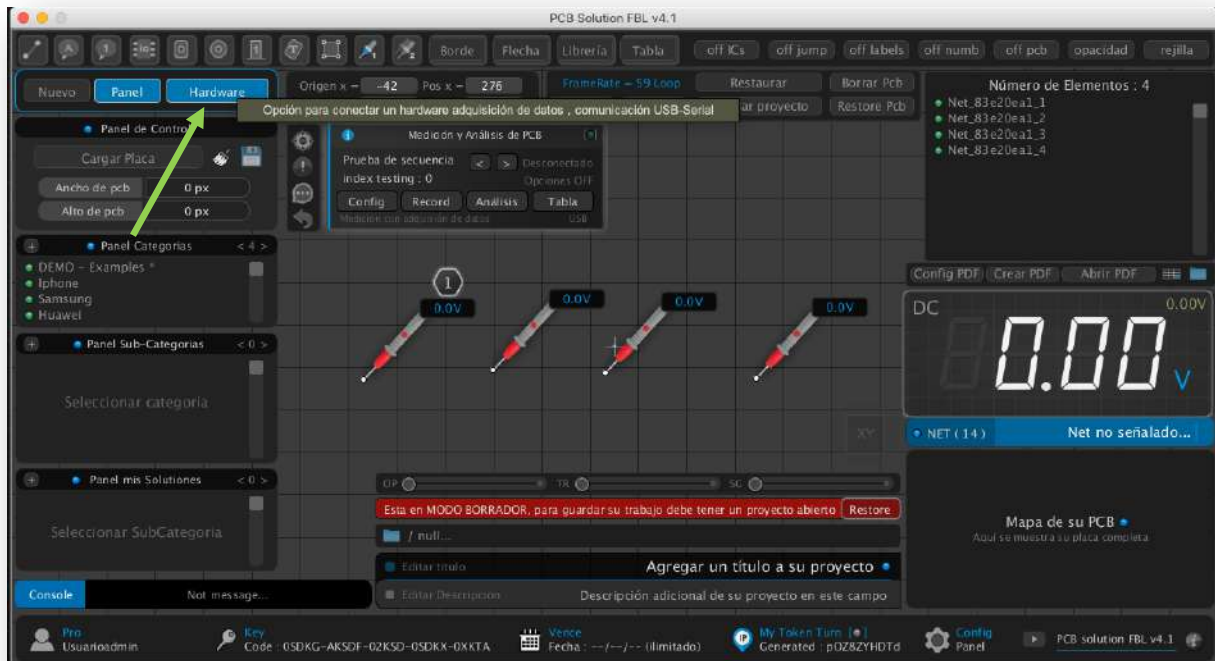
6. You should see this printed every 3 seconds. Your Arduino is ready! Close the Arduino program.



7. Now, start PCB Solution FBL. You need to create a few probes. As an example, you should create 4 voltmeter probes on the screen: **Probe > New > Click on the screen**.



8. Now, activate the "HARDWARE" button.



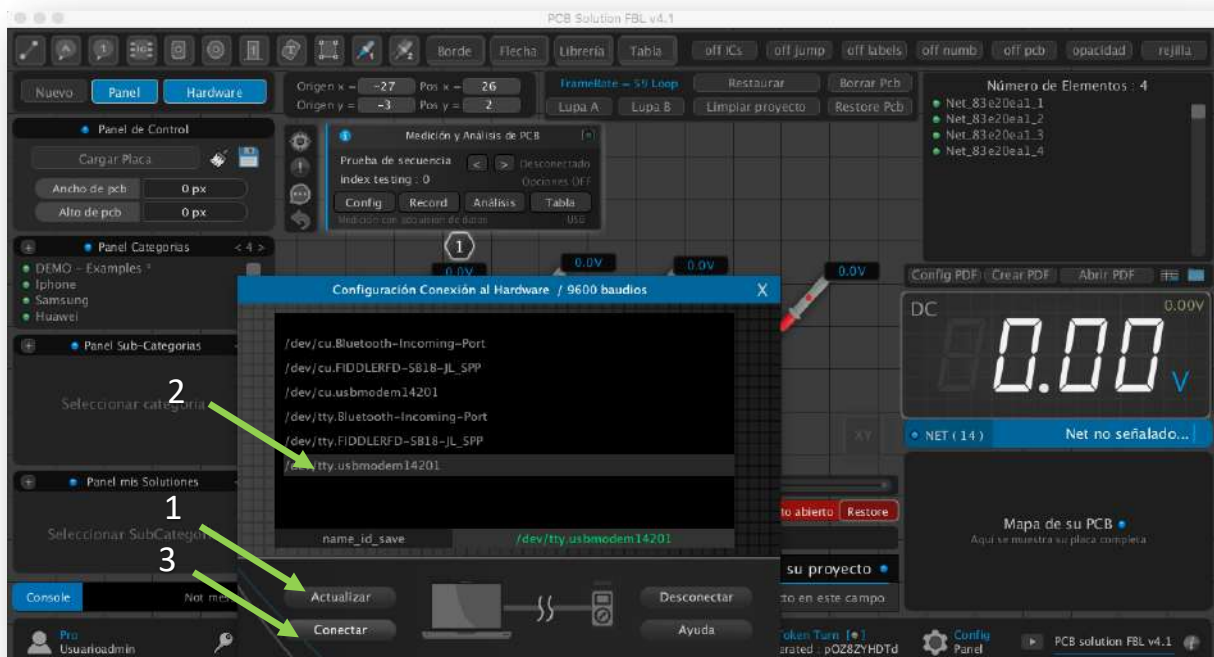
9. Connect your Arduino board and press the "Configuration" button on the hardware panel that appeared.



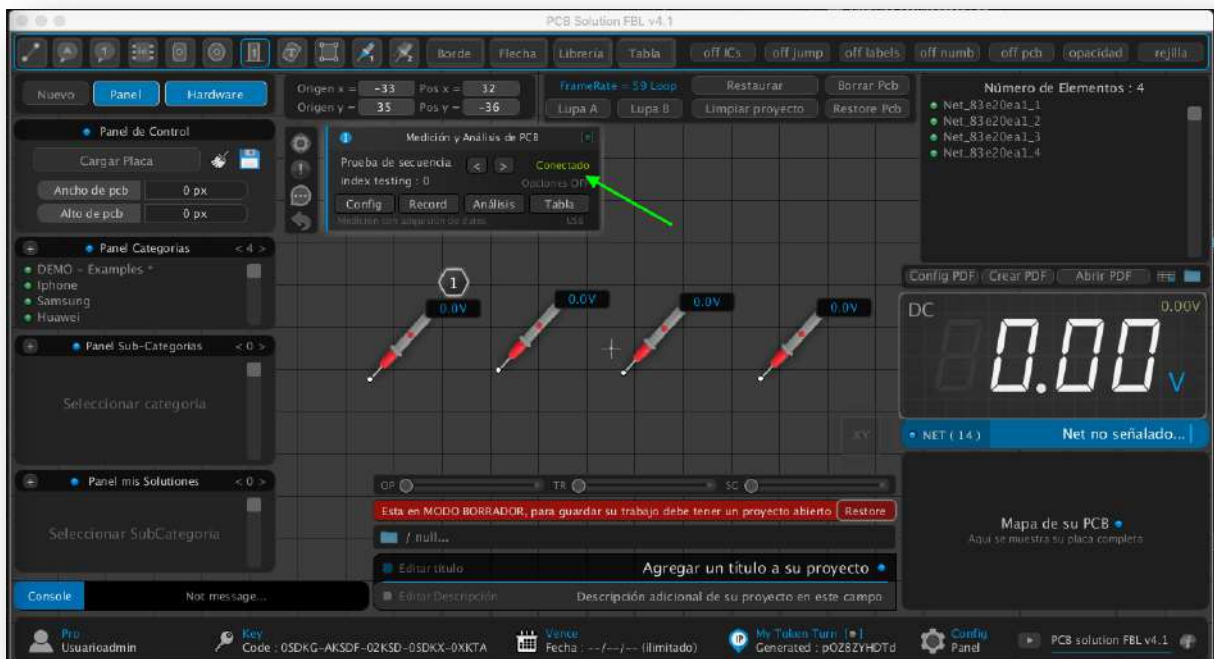
Any Arduino board should work for connecting to the program. We have tested with the following boards: (Arduino Uno, Arduino Mega, Arduino Nano...).

10. Press **"Refresh"** until your Arduino appears in the list and select your board with a left-click (if connected , the windows closes)

- On Windows, it appears as a name like **"COM."**
- On Mac, it appears as **"/dev/tty.usbmodem..."**



11. Now you will notice it says **"Connected"** on the control panel.



12. Everything is ready. You should now click the **"Record"** button to receive the voltages that Arduino will be sending. You'll notice that when it receives a value, the program automatically moves to the next available probe.

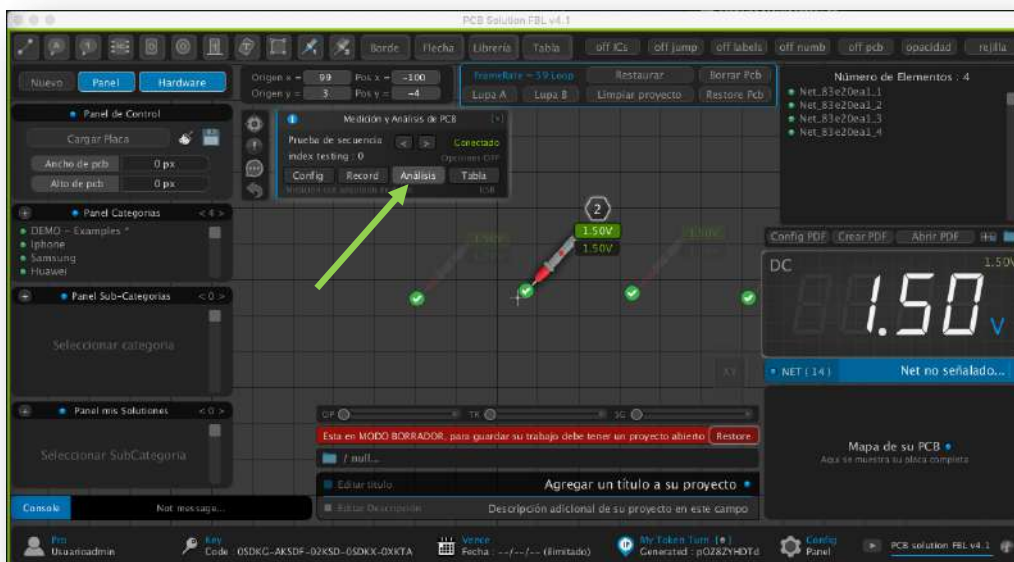


Sending ...

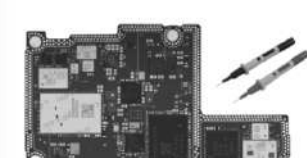


PCB good
(Taking samples)

13. Now, activate the **"Analysis"** button. You can see that new voltage captures are being taken, but they are saved in a second field. Here, we simulate the analysis on another faulty board using the good voltages as a reference. When it turns red, it's because the program detected that the voltage read on the second board is different, which may indicate a fault in the line or component.



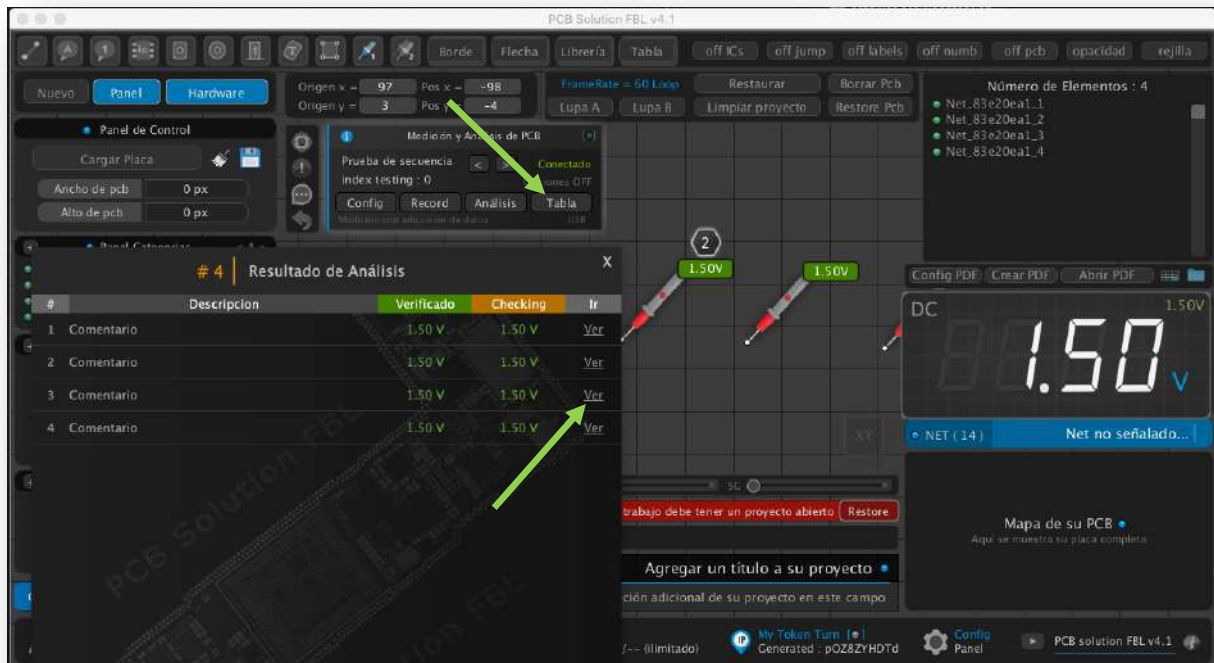
Sending ...



PCB checking
(Capture and compare)

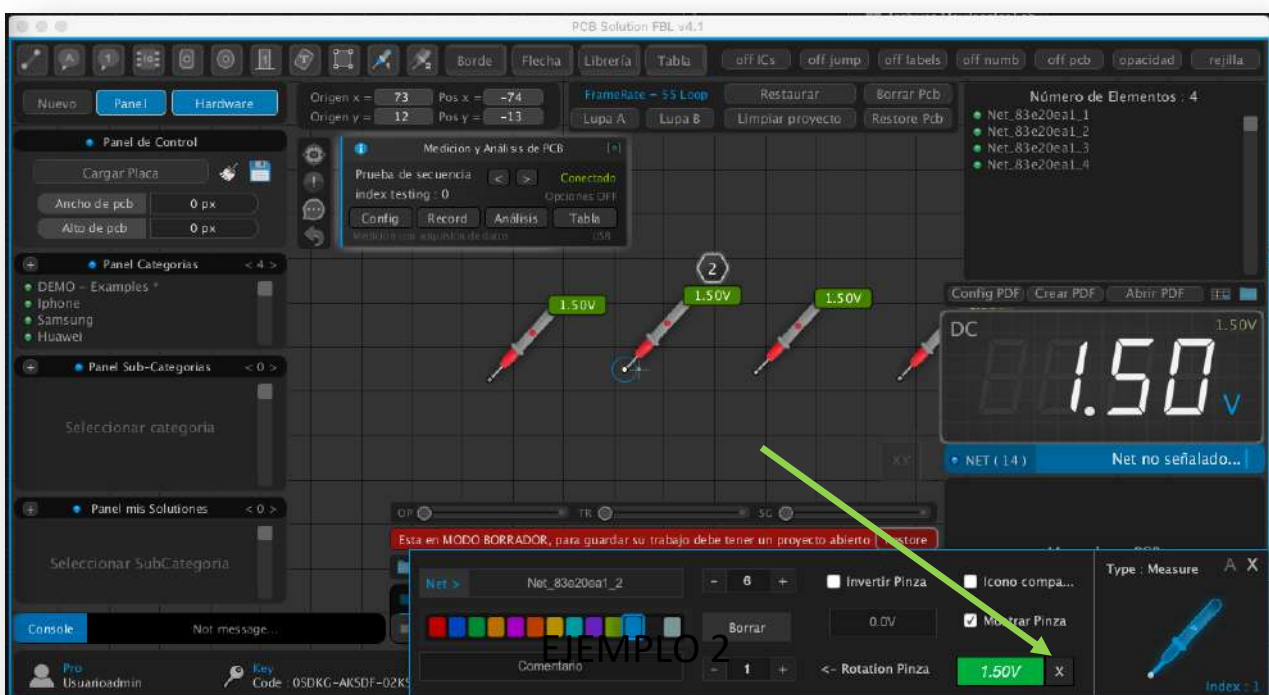
Wait until voltage has been captured on all probes, and then you can deactivate the **"Analysis"** button.

14. Press the **"Table"** button to view the list of comparative voltages. The first column (green) represents the voltages captured from a board in good condition, and the second column displays the voltages from a second board that you're trying to analyze for any anomalies.



NOTE: Within the table, you can click on "Ver" or "View" to scroll the screen to where the measuring probe is located.

ADDITIONAL NOTE: If you want to delete the voltage capture made by the hardware for the probe, you can do so by right-clicking on the measuring probe, and in properties, click on the "X" where the captured voltage is.



After testing the automatic data transmission, now you can test this second code. In this example, we will send a voltage from Arduino to PCB Solution FBL every time we press a mechanical switch.

```

cpp
Copy code

int switchPin = 2;
int analogPin = A0;
float voltage = 0.0;

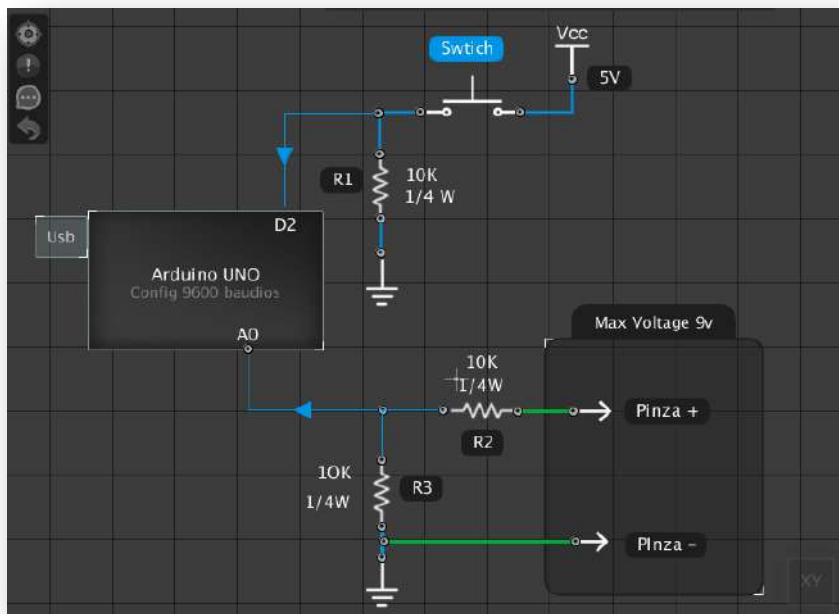
void setup() {
  pinMode(switchPin, INPUT); // Deja el pin 2 como entrada sin resistencia p
  Serial.begin(9600);
}

void loop() {
  if (digitalRead(switchPin) == LOW) { // Si el interruptor se presiona (LOW)
    int rawValue = analogRead(analogPin); // Lee el valor analógico en A0
    voltage = (rawValue * 5.0 / 1023.0) * 2; // Convierte el valor a voltaje

    // Envía la cadena por el puerto serie
    Serial.print(voltage, 2); // Muestra el valor con 2 decimales
    Serial.print(",data,");
    Serial.println("voltage");
    delay(1000); // Espera 1 segundo para evitar múltiples lecturas
  }
}

```

Build this basic test circuit for a very simple voltmeter.



Important: This circuit is designed to measure voltages from 0 to 9 volts. Ensure that you do not measure voltages higher than this because it lacks overvoltage protection (OVP), and do not measure negative voltages. The purpose is solely to test how you can send voltages using an Arduino board.

The design of a more robust circuit for a more comprehensive meter will depend on the developer who is trying to use this feature of PCB Solution FBL. It allows you to connect an Arduino as an interface to your own multimeter that you have developed.

That's great news! As you write this note, you are already developing an official voltmeter for the software that will include a safe design with all the necessary components like fuses, precision resistors, operational amplifiers, filtering capacitors, overvoltage protection, and more details.

You will soon release your own meter with other important innovations and ongoing improvements to your PCB Solution FBL software.



Sincerely,
Movilcenter Lab Team
Lima, Peru

Web oficial : www.movilcenter-lab.com