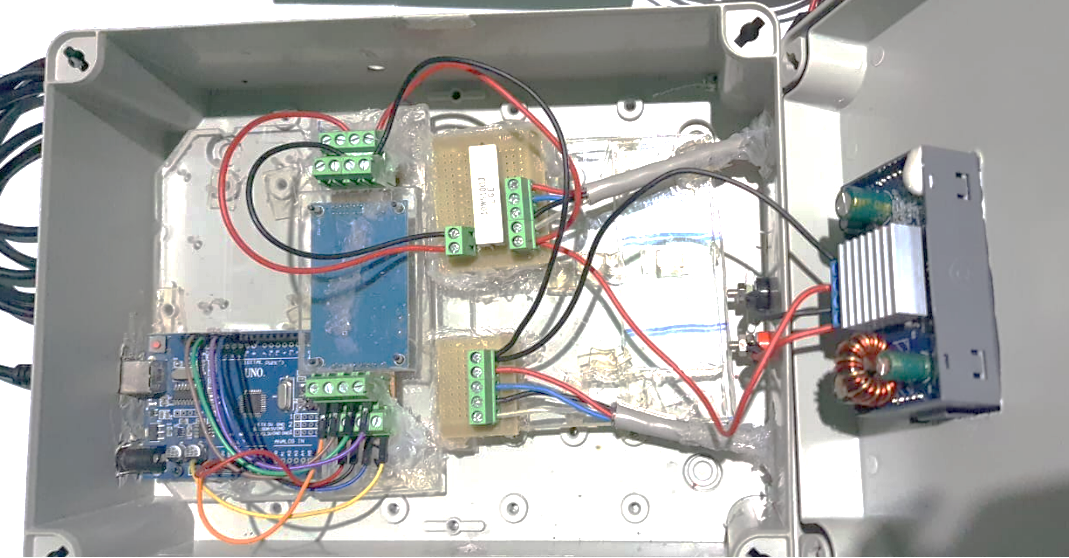
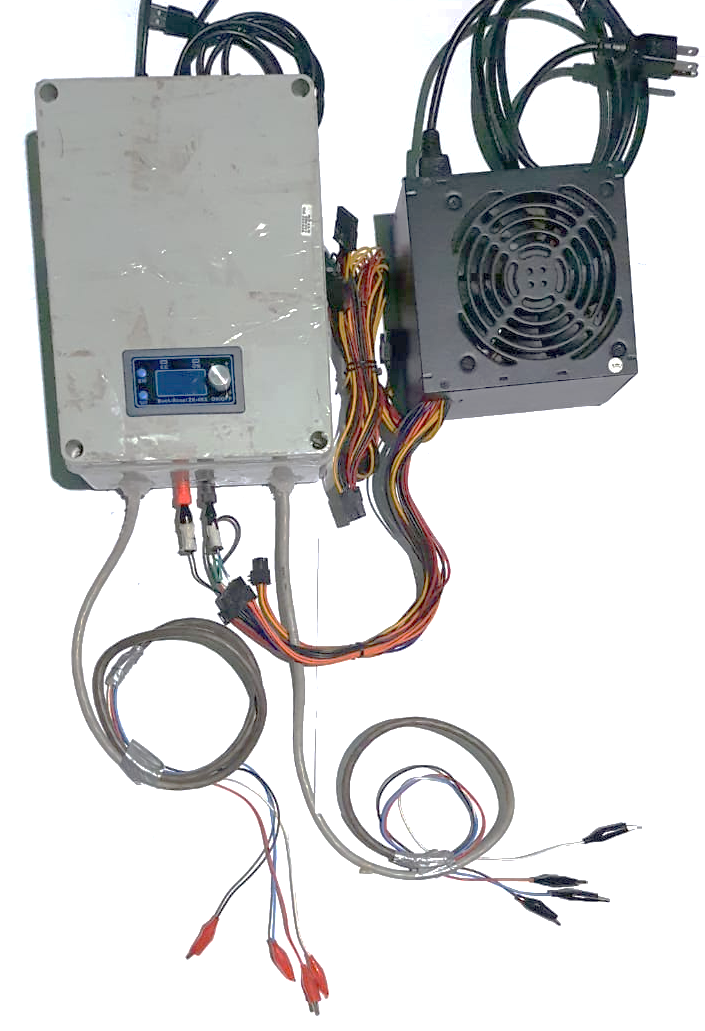
**PSLer**

**Supplementary Figure 1.** Complete PSLer apparatus with all modules. (A) Sensor & Logger; (B) DC power supply unit; (C) Input sensing wires; (D) Adjustable power supply unit; (E) Voltage sensor circuit; (F) ADS1256; and (G) Arduino Uno.



**A**

**B**

**C**

**D**

**E.1**

**E.2**

**F**

**G**

**For electrical signal from metabolic reactions**

**For external input voltage monitoring**

## **Materials**

* 2 custom circuits (voltage sensor circuit) [E1, E2]
* 1 Arduino Uno [G]
* 1 ADS1256 [F]
* 1 ZK-4KX Buck-boost voltage converter [D]
* 1 Power supply unit [B]
* 4 Terminal block connector 2 pins
* 7 Terminal block connector 2 pins
* 2 Terminal block connector 3 pins
* 1 Ceramic Cement Resistor 10W 10 ohm [in E1]
* 20 breadboard wire
* 2 units of 2 m of Thermostat Flexible 4-wire electrical copper cable isolated
* 8 Alligator Clips
* 2 Banana plugs
* Acrylic sheet
* Steck Junction Box Grey 234 X 174 X 90 mm

## **Prerequisites**

1. Wire Preparation (20 units of 20 cm breadboard wire)
2. Strip the Wires: Using wire strippers or a suitable tool, carefully strip approximately 0.5 cm of insulation from each end of the wires to expose the metal wire for connection.
3. Apply Solder:

* Heat a soldering iron to the appropriate temperature (typically around 350°C or 662°F).
* Apply a small amount of solder to the exposed wire ends. This helps to ensure a reliable and solid connection when attaching the wires to terminals or connectors.
* Allow the solder to cool and solidify, forming a firm and stable connection.

1. Inspect Conductivity:

* Power on the multimeter and set it to continuity testing mode
* place the multimeter probes at each end of the wire segment.
* If the wire has good conductivity, the multimeter will emit a sound, indicating continuity.

1. Wire Preparation (2 units of 2 meter of flexible 4-wire electrical copper cable):
2. Securely attach alligator clips to one end of each wire. These clips will facilitate easy connection and disconnection from the reactor's electrodes.
3. Carefully strip approximately 1-2 cm of insulation from the opposite end of each wire to expose the copper conductors. Apply a small amount of solder to the exposed wire ends to ensure a reliable and solid connection when attaching the wires to the nodes on terminals E.1 and E.2. Finally, inspect conductivity.
4. Circuit-Voltage sensor preparation
5. Anode and Cathode circuit

* Familiarize with the breadboard's layout for E.1 e E.2, identifying the power rails and terminal strips where components will be inserted according to Supplementary Figure 2.

1. 10-ohm resistor:

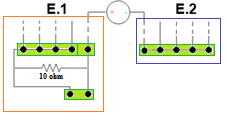
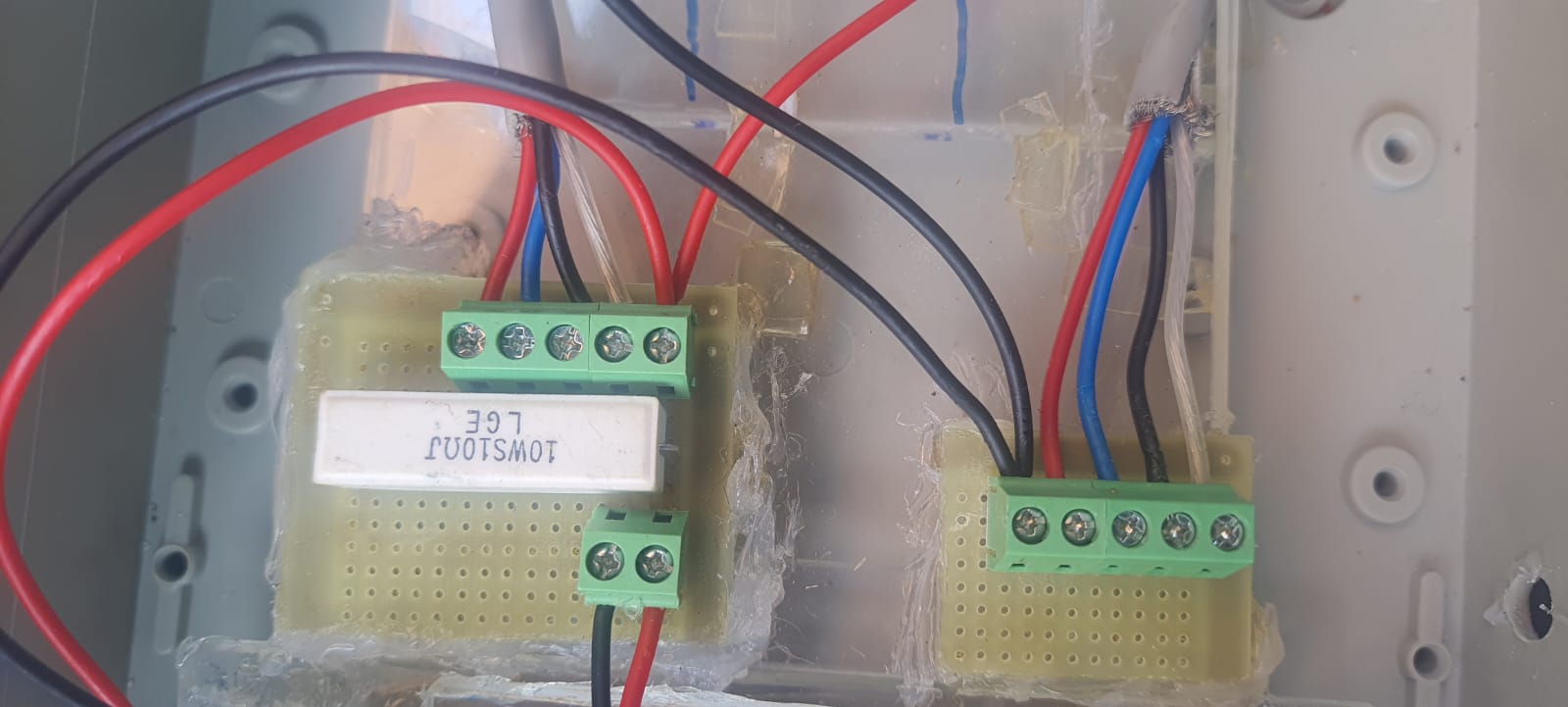
* Take the 10-ohm resistor and bend the leads to fit into the breadboard.
* Insert the leads and ensure the resistor is firmly seated on the breadboard.

1. Terminal blocks (green):

* Insert the terminal blocks into the breadboard

1. Solder connections: Solder all the connection points.
2. Check conductivity

**Supplementary Figure 2.** Anode and Cathode breadboard layout

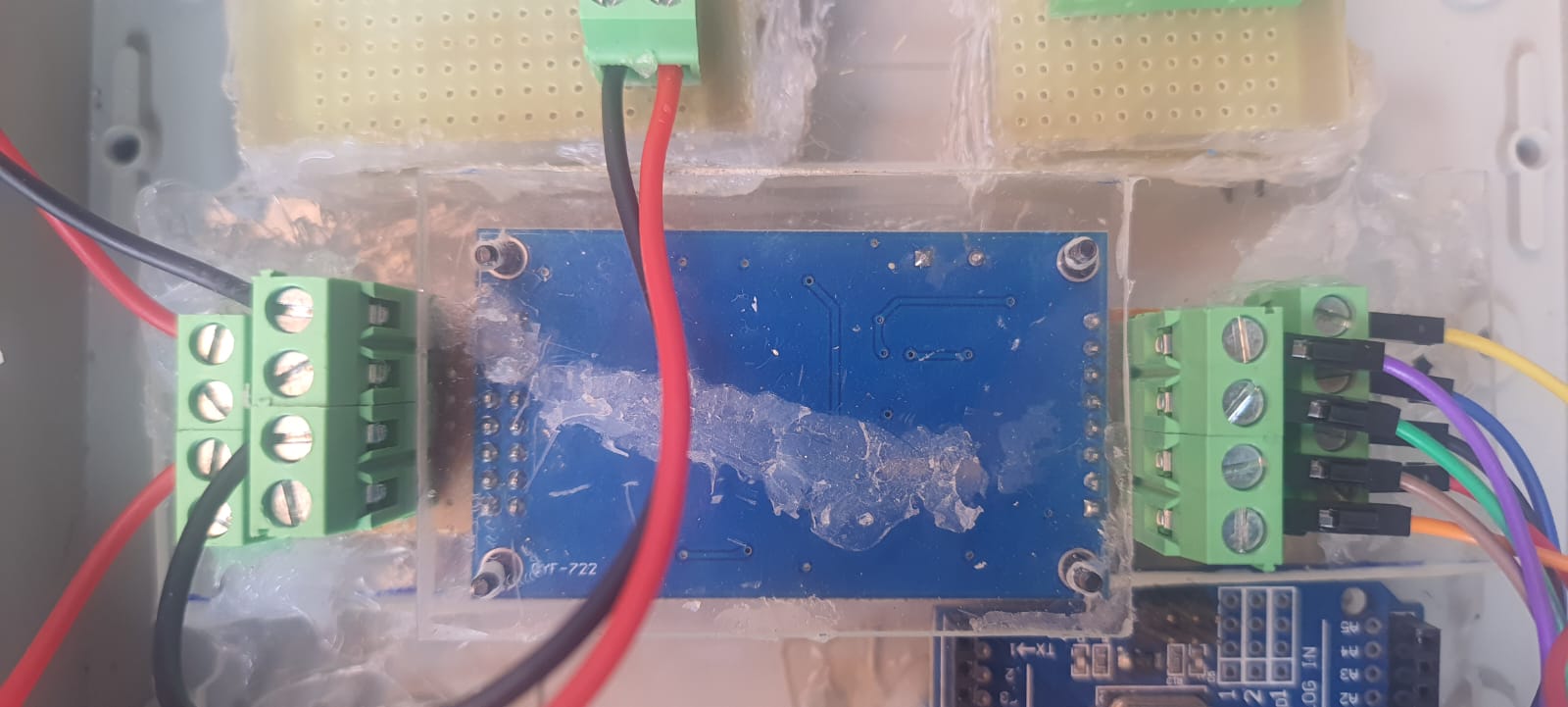


d. Adjusting breadboards and terminal blocks in ADS1256

1. Identify the ADS1256 Pins: Locate and identify the specific pins on the ADS1256 board that will be used for connections according to Supplementary Figure 3.
2. Adjust the Breadboard: Position the ADS1256 on the breadboard so that the pins align with the corresponding slots.
3. Insert Terminal Block Connections: Place the terminal block connections into the breadboard holes, ensuring they match up and align with the pins of the ADS1256.
4. Solder the Connections: Secure the terminal block connections by soldering them to the ADS1256 pins for a firm electrical connection.

|  |  |  |  |
| --- | --- | --- | --- |
| ADS1256 – Arduino UNO |  | Voltage circuit sensor-ADS1256 | |
| 5v | **SENSOR 1** | **AIN0** |
| GND | **AIN1** |
| SCLK | **SENSOR 2** | **AIN2** |
| DIN | **AIN3** |
| DOUT | **SENSOR 3** | **AIN4** |
| DRDY | **AIN5** |
| CS | **SENSOR 4** | **AIN6** |
| PDWN | **AIN7** |

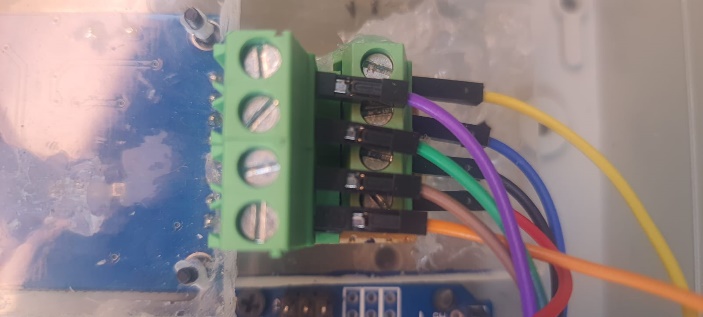
**Supplementary Figure 3.** ADS1256 pins identification.



**ADS1256**

**Terminals for sensors**

**Terminals for ADS1256-ArduinoUno connections**



**Breadboard**

**Block terminals**

**ADS1256 pins**

**Supplementary Figure 4.** Breadboards and terminal block configurations on the ADS1256 board.

e. For board enclosure

* The enclosure box (Steck Junction Box, Grey, 234 x 174 x 90 mm) was used to house and protect the circuit.
* Cuts and holes were made in the plastic box manually or with a Dremel tool, utilizing rotary and polishing attachments.
* To shield the electrical boards, acrylic sheets (200 x 150 x 5 mm) were cut with a cutter.
* Screws were used to mount the electrical boards onto the acrylic, insulating and protecting the conductive parts of the voltage sensor circuit, Arduino UNO board, and ADS1256 board.
* Banana plugs and cables were used as the interface between the ZK-4KX Buck Boost voltage converter and the power supply unit



**Assembly**

**Supplementary Figure 5.** Cuts and holes in the circuit box.

## **Step 1:** Connect wires to the circuit voltage sensor

1. Identification of Output Terminals on the Converter: Identify the positive (+) and negative (−) output terminals on the ZK-4KX Buck boost voltage converter. These terminals will supply the regulated voltage to the voltage sensor circuit.
2. Connection to the Voltage Sensor Circuit:

* Positive Output Connection: Connect the positive output terminal of the converter to the first node of terminal E.1 (Anode circuit) on the voltage sensor circuit.
* Negative Output Connection: Connect the negative output terminal of the converter to the first node of terminal E.2 (Cathode circuit) on the voltage sensor circuit.

1. Attachment of Anode Wires: Identify the flexible 4-wire electrical copper cable, which is insulated, and attach each of the wires to the remaining four nodes on terminal E.1 (Anode circuit).
2. Attachment of Cathode Wires: Similarly, identify the flexible, insulated 4-wire electrical copper cable and attach each wire to the remaining four nodes on terminal E.2 (Cathode circuit).
3. Important Considerations:

* Verification: After all connections are made, adjust the voltage output of the ZK-4KX Buck-boost voltage converter within the range of 0-1.25V. Use an external multimeter to measure and verify that the voltage across the nodes corresponds accurately to the set value.
* Voltage should be set up to 1.25 V. This specific voltage level is crucial as it matches the input voltage range implemented by the code running on the Arduino Uno.

**Step 2:** Connect ADS1256 to Arduino Uno

1. Identify the Connection Points: Refer to the pinout table to match the corresponding pins between the Arduino Uno and the ADS1256 module.
2. Make the Connections accordingly to the Table 1:

**Tabla 1**. Pinout connections.

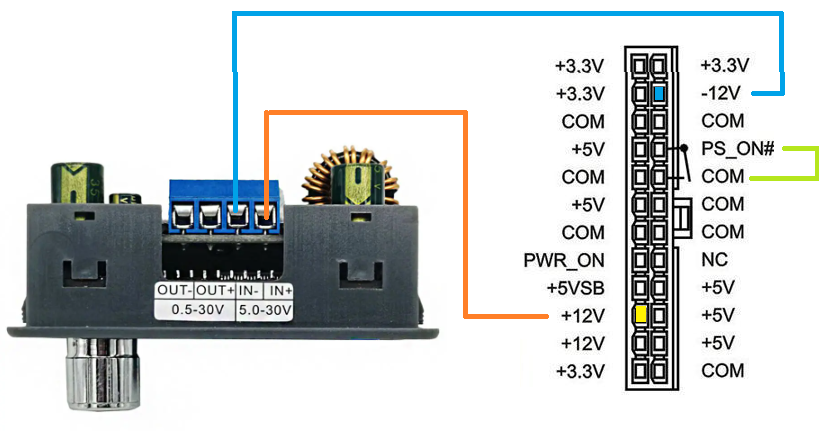
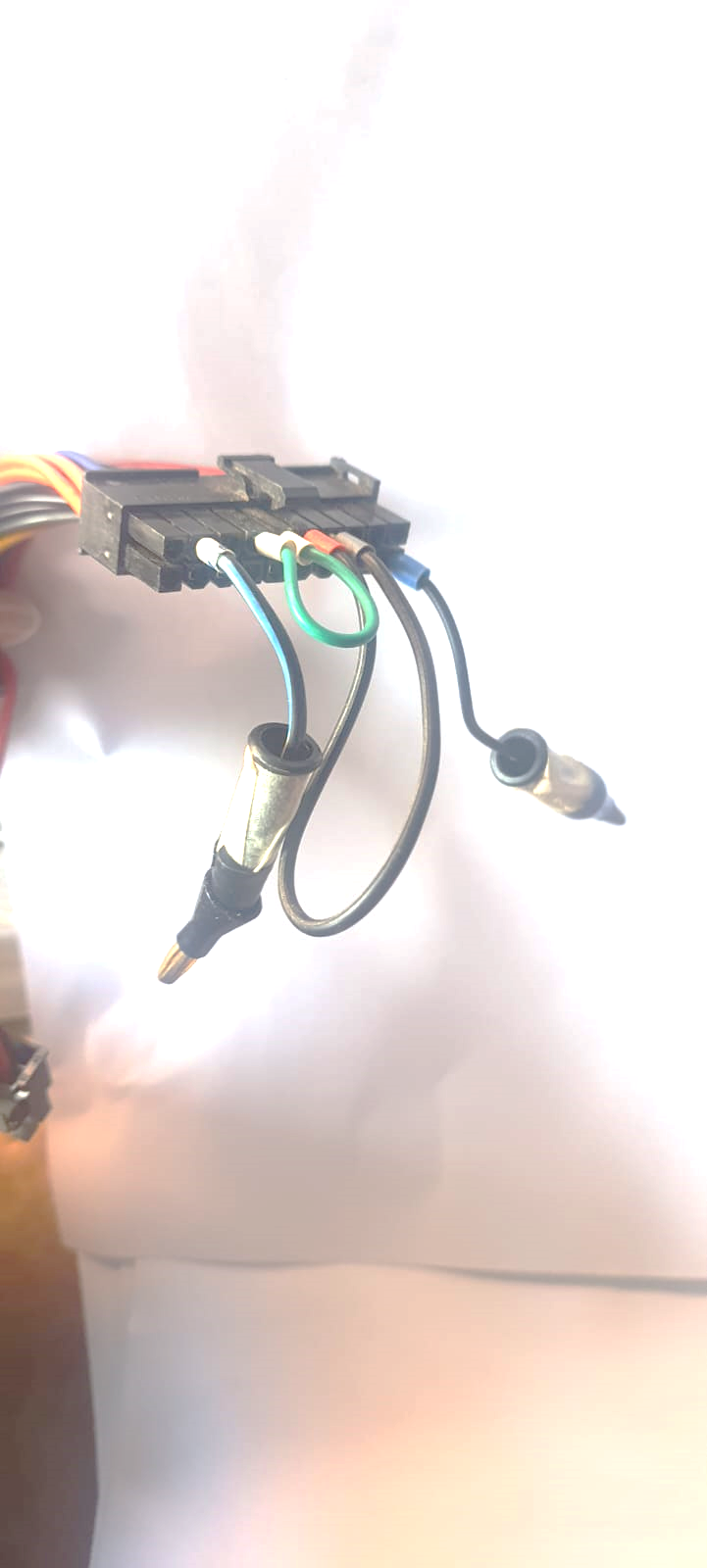
|  |  |
| --- | --- |
| ADS1256 Module  (F) | Arduino UNO  (G) |
| 5v | 5v |
| GND | GND |
| SCLK | pin 13 (SCK) |
| DIN | pin 11 (MOSI) |
| DOUT | pin 12 (MISO) |
| DRDY | pin 9 |
| CS | pin 10 |
| PDWN | 5v |

1. Verification: Double-check that no wires are loose or improperly connected, which could lead to communication errors between the Arduino Uno and the ADS1256 module.

## **Step 3:** Power Supply Integration for Sensor Module

1. Locate Power Terminals: Identify the +12V and -12V output terminals and the PS\_ON and GND terminals on the power supply unit (PSU). Also, identify the + and – input terminals on the ZK-4KX Buck-boost voltage converter.
2. Activate Power Supply: Use a jumper wire to bridge the PS\_ON terminal with the GND terminal on the PSU. This action triggers the power supply to turn on, enabling the +12V power rail.
3. Connect Power Lines: Attach the +12V output from the PSU to the corresponding input terminal on the ZK-4KX Buck boost voltage converter.
4. Important Considerations:

* Make sure all wiring is adequately insulated and securely connected to avoid short circuits or loose connections.
* Before powering up the system, carefully review all connections using the pinout diagram to confirm they are correct.



**Supplementary Figure 6.** Power supply Integration.

## **Step 4:** Plug sensor into the ADS1256

1. Identify Differential Inputs: Locate the ADC differential inputs on the ADS1256 board: AIN0-AIN1, AIN2-AIN3, AIN4-AIN5, and AIN6-AIN7.
2. Connect Sensors:

• Sensor Pairing: Connect each 2-wire sensor to one of the ADC differential input pairs.

* For Sensor 1, connect the wires to AIN0 and AIN1.
* For Sensor 2, connect the wires to AIN2 and AIN3.
* For Sensor 3, connect the wires to AIN4 and AIN5.
* For Sensor 4, connect the wires to AIN6 and AIN7.

1. Important considerations:

• Ensure all connections are secure and properly insulated to prevent interference and signal loss.

• Connect the ADS1256 board to the power supply and verify it receives the correct voltage.

## **Step 5:** Connect PSLer to computer

1. Locate the USB Port: Identify an available USB 2.0 port on the computer. These ports are typically black.
2. Connect the Arduino: Use the Arduino USB cable to connect the Arduino board to the identified USB 2.0 port on the computer.

------------------------**THE ELECTRONICS ARE NOW READY FOR TESTING**------------------------------------------------------------------------------------------------------------------------------------

## **Step 6:** Burn Arduino Uno code and start

1. Open Arduino IDE: Launch the Arduino Integrated Development Environment (IDE) on the computer.
2. Select the Arduino Board and Port

* Go to Tools > Board and select "Arduino Uno" from the list.
* Go to Tools > Port and select the appropriate COM port to which the Arduino is connected. This is typically labeled as "COMx" or "/dev/ttyACMx" depending on your operating system.

1. Load the code: Open the Arduino code by going to File > Open and selecting your .ino file (Table 2).
2. Verify the Code: In the Arduino IDE, click the "Verify" button (checkmark icon) to compile the code and check for errors.
3. Upload the Code: Click the "Upload" button (right arrow icon) in the Arduino IDE to burn the code to the Arduino Uno. Wait for the upload process to complete. The IDE will indicate when the upload is successful.
4. Start the System: Once the code is successfully uploaded, the Arduino Uno will begin executing the programmed tasks.

## **Step 7:** Run Python code in Spyder-Anaconda navigator

1. In Anaconda Navigator, locate and click on the "Spyder" icon to open the Spyder IDE.
2. In Spyder, go to File > Open and navigate to the location of your Python script (.py file) (Table 2).
3. Click the "Run" button (green play icon) in the toolbar, or press F5 on your keyboard to execute the Python script.

**Table 2:** Design file summary

|  |  |  |
| --- | --- | --- |
| Design file | File type | Location of the file |
| PSLer\_AR | ARDUINO IDE | Code (<https://doi.org/10.5281/zenodo.13351141>; https://github.com/Carla-ifr/PowerSupply-DataLoggerMES.git)  Modified from: <https://github.com/adienakhmad/ADS1256.git> |
| PSLer\_PY | PYTHON | Code (<https://doi.org/10.5281/zenodo.13351141>; https://github.com/Carla-ifr/PowerSupply-DataLoggerMES.git)  Modified from <https://doi.org/10.5281/zenodo.12808300> |

* **PSLer AR code 1:** Arduino IDE code that sense the electrical signals.
* **PSLer PY code 2:** PYTHON code that records and saves the signals in CSV file.

### **References**

1. Fernando K. Dá para criar uma empresa disso! [Internet]. Tutoriais Tecnologia Tendencias. 2020. Available from: https://www.fernandok.com/2020/10/da-para-criar-uma-empresa-disso.html
2. Akhmad A, Sepúlveda A, Axel. ADS1256 [Internet]. 2019. Available from: https://github.com/adienakhmad/ADS1256.git
3. TI. Very Low Noise , 24-Bit Analog to Digital Converter. Texas Instruments [Internet]. 2013; Available from: https://www.alldatasheet.com/datasheet-pdf/pdf/336384/TI/ADS1256.html
4. Flores-Rodriguez C. The AGDAC: An Innovative Market-Oriented, Open-Source Multiplexing System for Advanced Biogas Volume Measurements in Research Experiments. Zenodo. <https://doi.org/10.5281/zenodo.12808300>