AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: GR Team

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Logo:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Table of Contents **2**

1. Project Title 3

2. Project Category 3

3. Project Description 3

4. Keywords 4

5. Target Audience 4

6. Project Objectives 4

7. Project Timeline 4

8. Resources Provided 5

9. Tools and Technologies Used 5

10. Step-by-Step Implementation Plan 5

11. Evaluation Criteria 9

12. Languages Available 9

13. Contributors 9

14. Feedback and Continuous Improvement 10

Submission Notes **10**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

---

1. Project Title:

IoT Rain Gauge Weather Station with Raspberry Pi Pico

---

2. Project Category:

AI / IoT / Smart Environment Monitoring

---

3. Project Description:

This project demonstrates how to build a DIY IoT rain gauge weather station using a Raspberry Pi Pico. The system measures rainfall using a tipping bucket rain gauge sensor and sends the data to a cloud dashboard for visualization.
Students learn how environmental sensors, microcontrollers, and data platforms combine to create smart weather monitoring systems.
*An optional extension introduces simple AI-based rainfall prediction based on historical data***.**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Keywords:

IoT, Weather Station, Rain Gauge, Raspberry Pi Pico, Environmental Monitoring, AI Prediction

---

5. Target Audience:

VET students, secondary school students, STEM clubs, beginner IoT developers

---

6. Project Objectives:

- Understand how rainfall sensors work.
- Learn how to connect sensors to microcontrollers.
- Collect and transmit environmental data through IoT platforms.
- Visualize weather data on dashboards.
- Explore basic AI techniques for simple rainfall prediction.

---

7. Project Timeline:

Day 1 – Introduction and hardware assembly
Day 2 – Sensor calibration and microcontroller programming
Day 3 – Data transmission to cloud platform
Day 4 – Dashboard visualization and data analysis
Day 5 – Optional AI rainfall prediction activity

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. Resources Provided:

- Raspberry Pi Pico microcontroller
- Tipping bucket rain gauge sensor
- Jumper wires and breadboard
- WiFi module (ESP8266/ESP32 if needed)
- Computer with Python or MicroPython
- Internet connection

---

9. Tools and Technologies Used:

- Raspberry Pi Pico
- MicroPython or Python
- IoT cloud platform (ThingSpeak / Blynk / Node-RED)
- Rain gauge sensor
- Data visualization dashboard

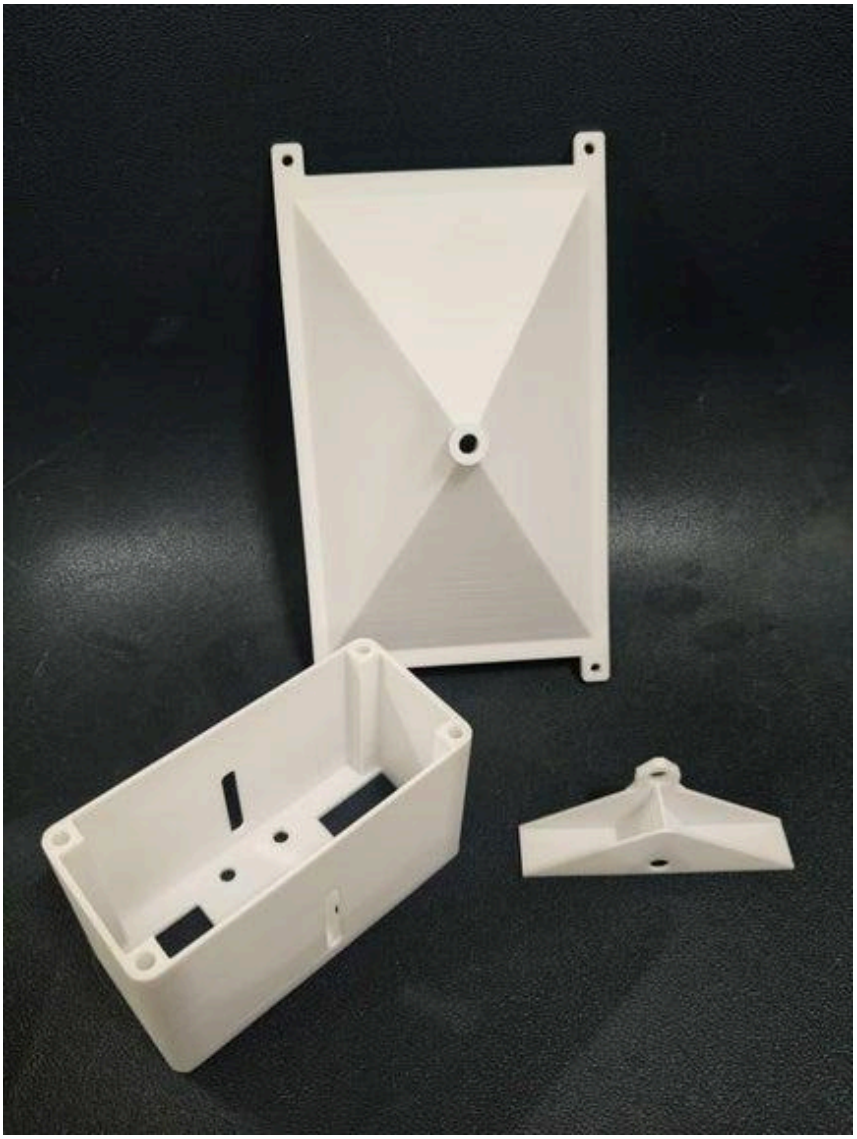
---

10. Step-by-Step Implementation Plan:

Step 1: The Tipping Mechanism (Heart of the Project)

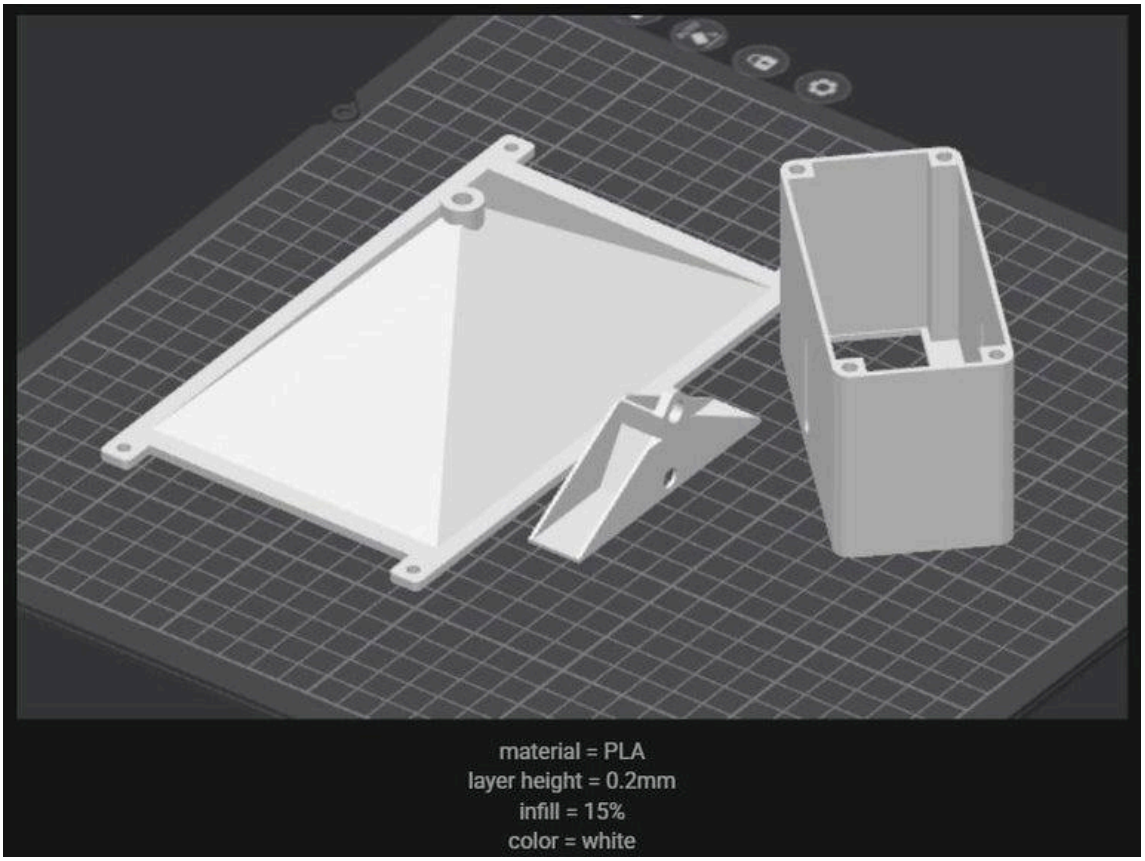
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



The heart of this project is the DIY Rain Gauge. We used a **Tipping Bucket** mechanism.
How it works: A funnel directs rain into a seesaw-like rocker. When one side fills up, it tips over using gravity. A magnet attached to the rocker swings past a sensor (we used a Hall Effect Sensor), sending a digital signal to the Pico W.
3D Printing the Parts: We used 3D-printed parts for the mechanism. We have attached the STL files below so you can replicate this project easily. Here is the file breakdown:

- **The Base (Base 2.stl):** Holds the sensor and electronics foundation.
- **The Tipping Bucket (Cucchiaio 2.stl):** The rocker mechanism itself.
- **The Funnel (Imbuto 2.stl):** Directs the rain into the bucket.

Print Settings: We recommend using PLA or PETG filament. An infill of **15% or higher** is suggested for durability against the weather.
ASSEMBLY INSTRUCTIONS:
1. The Axle: We used a **Bolt & Nut** to hold the tipping bucket in the center.

- *Critical Tip:* Do not tighten the nut too much! The bucket must swing freely. We used **Hot Glue** between the nut and bolt to lock it in place without clamping down on the bucket.

2. The Sensor Upgrade (Hall Effect vs. Reed Switch): Originally, we planned to use a Reed Switch. However, we found that the magnetic pull was too strong and caused the bucket to get stuck.

- **The Fix:** We upgraded to a **Hall Effect Sensor**. It detects the **Neodymium Magnet** glued to the bucket digitally.
- **The Result:** Zero friction and smooth tipping!

Base 2.stl

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



- Cucchiaio 2.stl

View in 3DDownload



- Imbuto 2.stl

View in 3DDownload

Step 2: The "Tupperware" Hack (DIY Stand)

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Co-funded by
the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.





View 2 more

Instead of building a complicated metal stand to hold the rain gauge, we hacked a plastic storage container.

1. **The Holder:** We cut the lid of the container to fit the 3D-printed rain gauge base perfectly. The rain gauge sits inside, protected but stable.
2. **Drainage:** We drilled 4 holes at the corners and cut square openings at the bottom.
3. **Protection:**

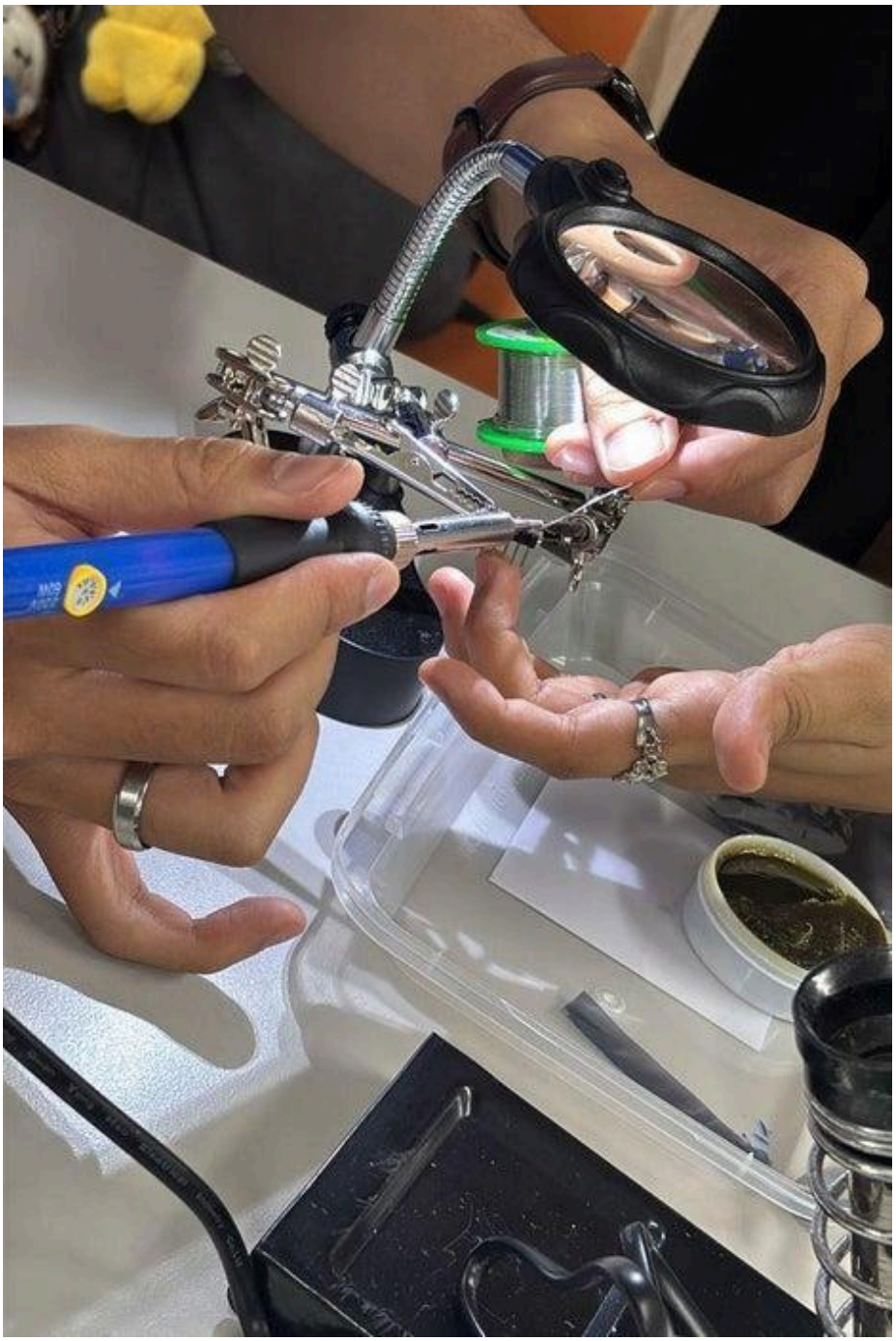
- **Mesh Filter:** We glued a **Mesh** over the funnel to stop insects or leaves from entering.
- **Sealing:** We used **Clay** and a **Hot Glue** around the edges where the rain gauge meets the container to prevent leaks.

Why drill holes? As you can see in the photos, the holes ensure rainwater flows out immediately. We want to measure the rain, not store it!

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 3: Electronics & Soldering



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.





Component Pin	Pico
Signal / Data	GP
VCC	3V3 (P
GND	GND (
Wire 1	GP
Wire 2	GND (Pin 1
USB Cable	Micro-U

Time to connect the brains. We used a **Raspberry Pi Pico W** because it's affordable, supports MicroPython, and has built-in WiFi.

1. Soldering We soldered header pins to the Pico W so it fits securely into the breadboard (or you can solder wires directly for a permanent fix).

2. The Wiring Guide Here is the detailed connection breakdown:

- **DHT11 (Temperature & Humidity)**
- **Signal:** Connect to **GP15**.
- **VCC (+):** Connect to **3V3 (Physical Pin 36)**.
- **GND (-):** Connect to any **GND** pin (**Physical Pin 38**).
- **Rain Gauge (Hall Effect Sensor)**
- **Signal:** Connect to **GP16**.
- **Ground:** Connect to any **GND** pin.

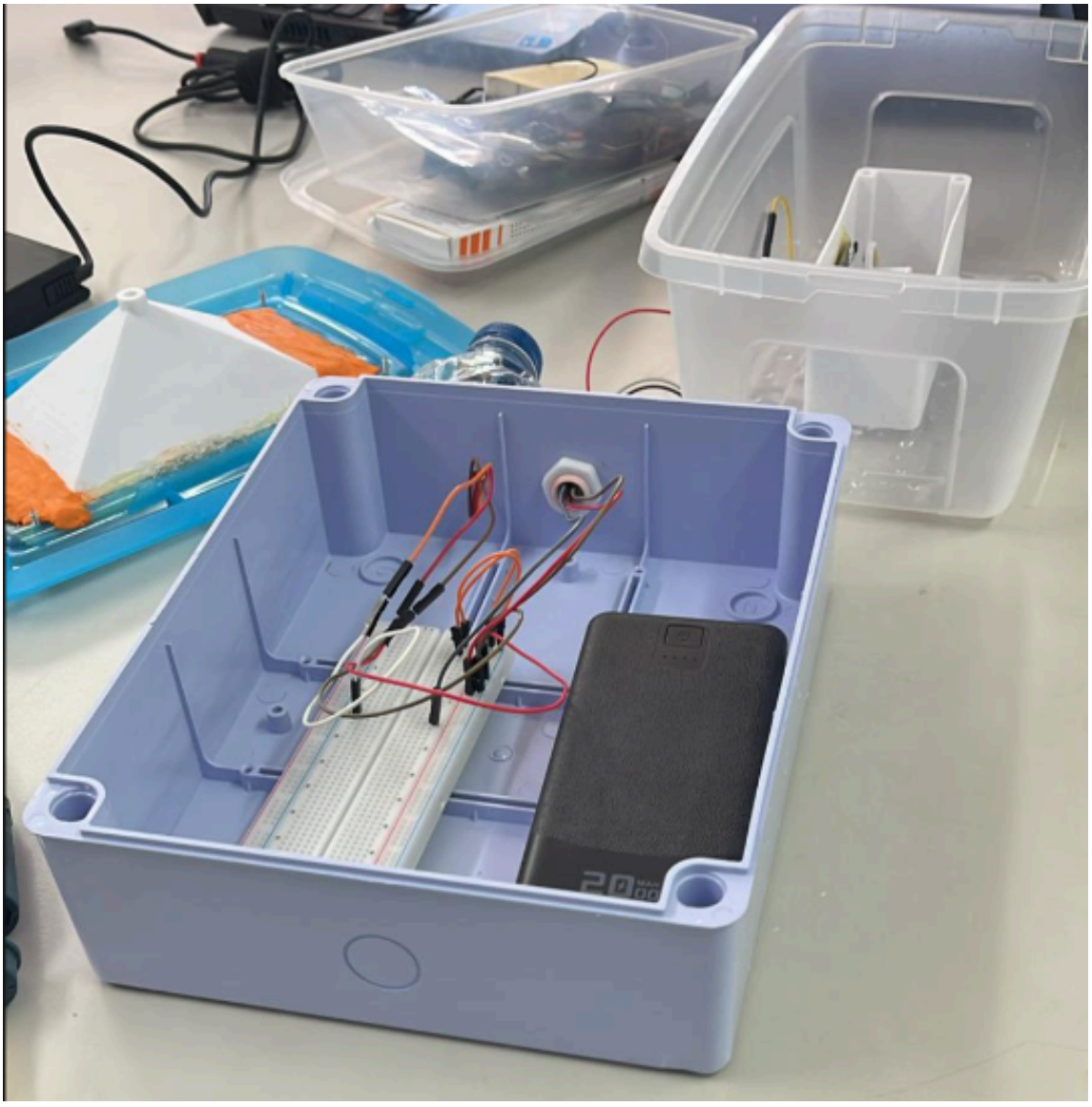
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

- *Note:* The code uses machine.Pin.PULL_UP. This means you **do not** need an external resistor. Simply connect one wire to GP16 and the other to Ground. When the bucket tips (magnet passes), it closes the circuit, triggering the count.
- **LED Indicator**
- **No external wiring required.** We utilize the Pico's **Onboard LED**. The code flashes this to show the "heartbeat" and successful data uploads.
- **Power Source**
- Connect your Power Bank to the **Micro-USB port**.
- *Tip:* Our code includes a "Keep-Alive" function that pulses the power draw to prevent modern power banks from auto-shutting off.

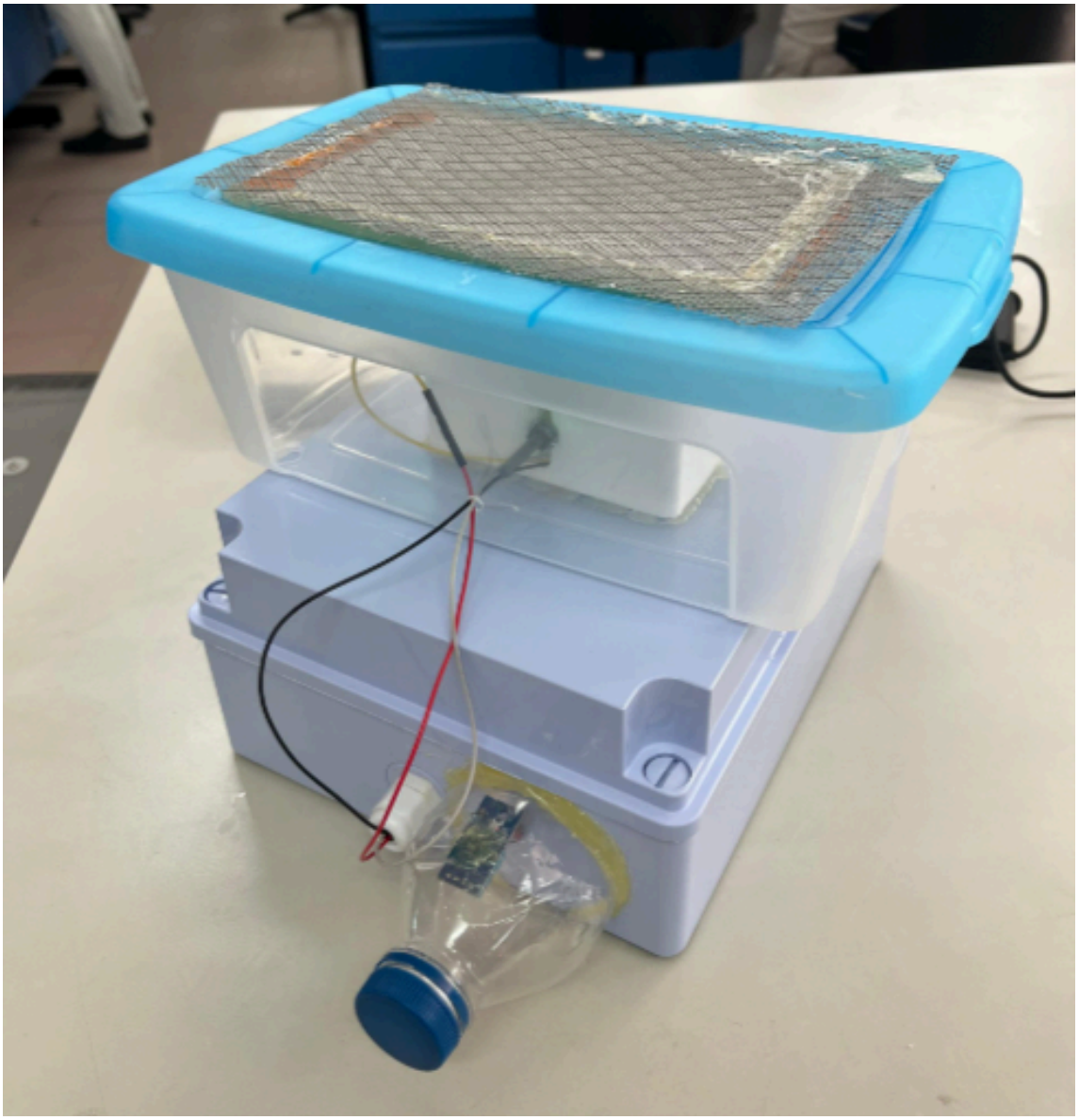
3. Waterproofing (Crucial!) Once the wiring was tested, we applied **Conformal Coating** to all exposed connections on the sensor side. This is a vital step to prevent humidity from corroding the circuit when deployed outdoors.

Step 4: Boxing It Up (Final Assembly)



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



We placed all the sensitive electronics into the **IP56 Waterproof Box**.

1. **Cable Management:** We drilled holes in the box and used **Cable Glands** to route the wires from the rain gauge (outside) to the Pico (inside). This keeps the box watertight.
2. **Power:** The **20,000mAh Powerbank** fits inside perfectly. After 7 days of operation, it still had plenty of charge left.
3. **The "Bottle Shield":** The DHT11 sensor needs to be outside to measure air temp, but it can't get wet.

- *Solution:* We cut a **Plastic Bottle**, glued it to the outside of the box, and hid the sensor inside it. It acts as a perfect umbrella!

Step 5: Calibration (Crucial Step!)

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



A rain gauge is useless if it's not accurate. To calibrate ours, we utilized a **beaker** to measure the water volume.

The Process:

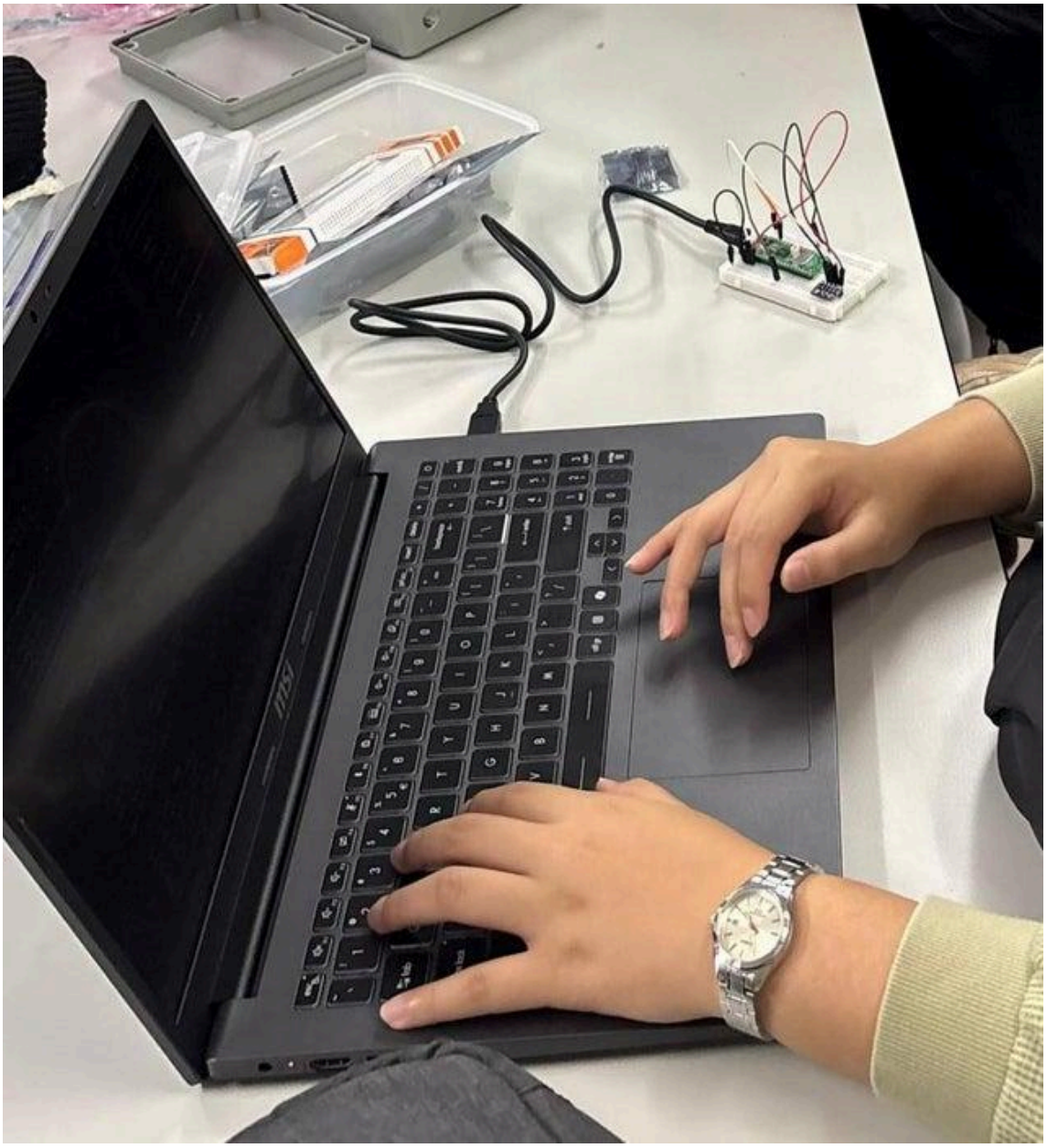
1. We filled a beaker with a known volume of water (100ml).
2. We slowly poured the water through the 3D-printed funnel to simulate rain.
3. We counted the total number of "tips" registered by the microcontroller.
4. By dividing the **Total Volume** by the **Number of Tips**, and factoring in the surface area of our funnel, we calculated the specific rainfall depth per tip.

The Result: We found that one tip equals exactly **0.173 mm** of rainfall.

Step 6: The Code (MicroPython)

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



With the electronics ready, we flashed the code. We used MicroPython on the Raspberry Pi Pico W. The code handles three main tasks:

1. **Connects to WiFi**.
2. **Counts Tips:** Uses an Interrupt (IRQ) to detect when the bucket tips.
3. **Uploads Data:** Sends the rainfall (mm), temperature, and humidity to ThingSpeak every 30 minutes.

Critical Configuration: In the code, you will see the variable MM_PER_TIP = 0.173. This value is **not random** but it is the specific result we obtained from the **Water Calibration (Step 5)** we did. You must update this variable with *your own* calibration result to ensure the rain data is accurate!
Code Snippet: Here are the critical parts you need to configure in ***main.py***:

```
# --- 1. CONFIGURATION (EDIT THIS) ---
SSID = 'YOUR_WIFI_SSID' # Replace with your WiFi SSID
PASSWORD = 'YOUR_WIFI_PASSWORD' # Replace with your WiFi password
THINGSPEAK_API_KEY = 'YOUR_API_KEY' # Copy from ThingSpeak
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

```
# CALIBRATION VALUE (From Step 5)
MM_PER_TIP = 0.173 # Update this value based on your syringe test results!

# --- 2. RAIN COUNTING LOGIC ---
# This function runs automatically whenever the bucket tips!
def hall_interrupt_handler(pin):
global tip_count, last_tip_time
current_time = time.ticks_ms()

# Debounce: Ignore signals that happen too fast (<0.5s)
if time.ticks_diff(current_time, last_tip_time) > 500:
tip_count += 1
last_tip_time = current_time
```

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

11. Evaluation Criteria:

- Correct assembly of hardware components
- Working rainfall data acquisition
- Successful transmission of data to cloud dashboard
- Ability to interpret rainfall data
- Optional: basic predictive analysis using collected data

---

12. Languages Available:

English

---

13. Contributors:

AI-SAVES Project Team

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Feedback and Continuous Improvement:

Participants are encouraged to document challenges, improvements, and additional sensor integrations such as temperature, humidity, or wind sensors.

Submission Notes:

Please ensure that all required documents and resources are attached to your submission. The AI-SAVES@ProjectHub review team will evaluate the submission based on the provided information. Once approved, the project will be published on the hub, and partners will be notified.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.