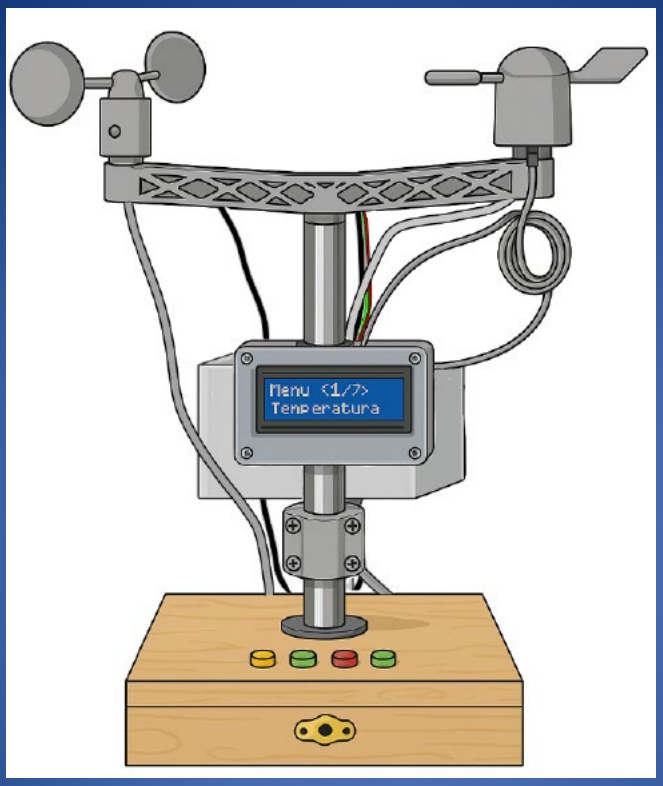
2025



Weather Station

USER MANUAL

ICE KUROBA|

Contenido

1. Introduction

This manual is designed to guide the user in the construction, setup, operation, and maintenance of a weather station developed using Raspberry Pi 5. The system collects real-time environmental data such as temperature, humidity, rainfall, wind speed and direction, air quality, and UV radiation. All data can be viewed interactively using a local LCD interface and physical buttons.

2. System Requirements

Hardware:
- Raspberry Pi 5 (with 5V 3A power supply)
- DHT11 sensor (temperature and humidity)
- MQ-135 sensor (air quality)
- GUVA-S12SD sensor (UV radiation)
- Mechanical rain gauge (tipping bucket)
- Anemometer (wind speed sensor)
- Analog wind vane (direction sensor)
- MCP3008 (ADC)
- 16x2 I2C LCD display
- 4 push buttons (up, down, select, back)
- Cooling fan (optional)
- Wooden enclosure (ventilated)
- Breadboard, jumper wires, resistors, connectors

Software:
- Raspbian OS (updated)
- Python 3
- Libraries: gpiozero, adafruit_dht, spidev, adafruit-circuitpython-mcp3xxx, I2C_LCD_driver

3. Assembly Instructions

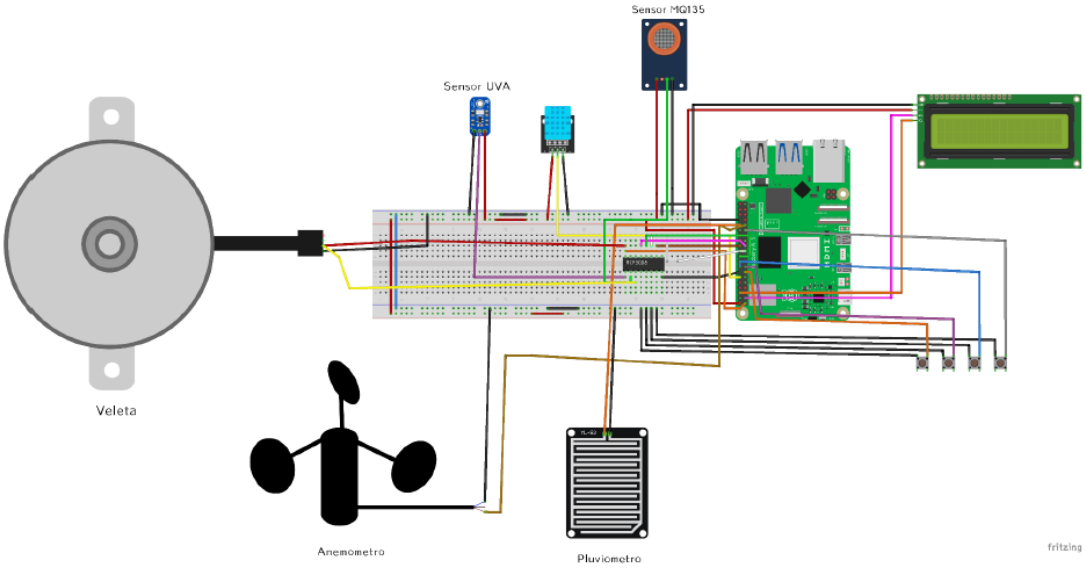
Step-by-step Guide:

1. **Prepare the enclosure**: Drill ventilation holes and ports for the sensors, LCD, and buttons.
2. **Mount components**: Secure the Raspberry Pi and breadboard inside. Attach sensors and fan in appropriate external locations.
3. **Wire sensors**: Follow wiring instructions based on GPIO, I2C, or SPI. Use MCP3008 for analog sensors.
4. **Install software**: Install Raspbian OS, Python, and all necessary libraries.
5. **Deploy code**: Place project files in /home/pi/Desktop/Proyecto. Ensure scripts like main.py and modules (sensores.py, botones.py, etc.) are present.
6. **Connect display and buttons**: Attach the 16x2 LCD via I2C and wire buttons to their GPIO pins.

7. **Test components**: Power the system and run the program to check all sensors and interactions.

4. Hardware Connections

Component	Type	GPIO/Physical Pin	Protocol
DHT11	Digital	GPIO 17 / Pin 11	GPIO
MQ-135	Analog	MCP3008 CH1	SPI
GUVA-S12SD	Analog	MCP3008 CH0	SPI
Wind Vane	Analog	MCP3008 CH2	SPI
Anemometer	Digital	GPIO 6 / Pin 31	GPIO
Rain Gauge	Digital	GPIO 12 / Pin 32	GPIO
LCD (16x2)	Digital	SDA GPIO 2 / SCL GPIO 3	I2C
Buttons	Digital	GPIO 27, 22, 23, 5	GPIO
Fan	Digital	GPIO (user-defined)	GPIO



5. Using the Weather Station

How to Start:

1. Plug in the power supply to the Raspberry Pi.
2. Wait until the system boots into Raspbian OS.
3. Open a terminal window.
4. Navigate to the project folder:

cd /home/pi/Desktop/Proyecto

5. Run the program:

python3 main.py

Interacting with the System: Once the system is running, the LCD screen will show:

Menu (1/7):
Temperature

Use the buttons as follows:

- **Up (GPIO 27):** Scroll up through menu items.
- **Down (GPIO 22):** Scroll down through menu items.
- **Select (GPIO 23):** Confirm the current selection and show the sensor reading.
- **Back (GPIO 5):** Return to the main menu or exit a reading.

Example Interactions:

- Selecting **Temperature**:

Temperature:
26.5 C

- Selecting **Air Quality**:

Air Quality:
Good

- Selecting **UV Radiation**:

Radiation UV:
0.85 V

Returning to Menu: After showing a sensor reading, the display will return to the menu in 3 seconds or immediately if the **Back** button is pressed.

Important Notes:

- The UV sensor reads photon intensity and converts it to voltage. Values closer to 1.0 V mean stronger radiation.
- The rain gauge and anemometer count pulses; values accumulate over time.
- If any sensor fails, “Error sensor” will be displayed.

Hidden Feature: Pressing the sequence **Back → Back → Up** triggers a hidden Easter egg that launches the game DOOM (requires chocolate-doom and WAD file).

6. Calibration and Maintenance

• DHT11: No regular calibration needed. Keep the sensor clean and dry.
• MQ-135: Should be calibrated in a clean air environment. Requires warm-up time.
• GUVA-S12SD: Clean the lens regularly to maintain accurate readings.
• Rain gauge and anemometer: Check that moving parts rotate/flex freely and are not blocked by debris.
• Wind vane: Ensure proper rotation and secure mounting.
• Fan: Verify it activates above 40ºC for internal cooling.

7. Customization and Expansion

• Add new sensors: Extend the Python code with new reading functions and menu entries.
• Change thresholds: Modify alert levels or behavior (e.g., activate fan at 35ºC instead of 40ºC).
• Cloud connection: Use platforms like Firebase, AWS IoT, or ThingSpeak to upload data remotely.
• User interface: Integrate a touchscreen or web-based dashboard using Flask or Node-RED.

8. Troubleshooting

• LCD shows nothing: Check power supply and I2C connection. Try rebooting the Pi.
• DHT11 returns 'None': Check wiring, and ensure the library is correctly installed.
• Buttons not responding: Test GPIO inputs and debounce logic.
• Analog sensors read 0 or 1023: Verify MCP3008 VDD and SPI connections.
• Program doesn’t start: Ensure main.py and all dependencies are in the correct folder.

9. Demo Video

Watch here: https://youtu.be/68Y0czJYqBI