Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

Good Practice Collection: Microcontroller Weather Stations (English)

This collection provides practical guidelines for students and teachers in VET (Vocational Education and Training) contexts, focusing on the field of IoT (Internet of Things) and microcontroller-based data systems. The projects cover skills related to data acquisition, network transmission, database management, and web visualization.

1. Good Practice: ESP32 – Complex IoT Weather Station with Wi-Fi

This good practice leverages the Wi-Fi capabilities of the ESP32 microcontroller to implement a completely self-contained, networked data acquisition system.

Lesson Plan (English)

Title	**ESP32 – Complex IoT Weather Station with Wi-Fi**
STE(A)M Subjects	Technology, Engineering, Data Science, IT
Target Age	16+ (VET, Higher Education)
Duration	6 hours (3 x 2-hour sessions)
Key Question	How can we create a fully digital, energy-efficient system for collecting and displaying environmental data?

Short Presentation

The project uses an **ESP32 microcontroller** to read data from the DHT11 temperature and humidity sensor. The ESP32 sends the measured data to a remote web server via a Wi-Fi connection using **HTTP POST** requests. On the server side, **PHP** code receives the data, stores it in an SQL database, and displays it on a webpage (HTML/CSS). This practice covers the entire IoT system chain, from **data acquisition to web display**.

Objectives/Learning Outcomes 🎯

Category	**Learning Outcomes (LOs)**
For Students	Students will be able to outline the general research design (A1) for a complete IoT system. They will be able to connect a microcontroller to a Wi-Fi network and transmit data using the HTTP protocol (B2, C2). They will be able to adapt the engineering design to the available infrastructure (A3). They will understand the

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

principles of receiving (POST) and querying (GET) data in web communication.

For Teachers Teachers will be able to design a work-based learning scenario focused on digital energy efficiency (A2). They will be able to motivate students toward STE(A)M subjects by visualizing practical project data.

Tools and Data 🛠

Category	Description
Hardware	ESP32 microcontroller, DHT11 temperature/humidity sensor, jumper wires.
Software/Infrastructure	Arduino IDE (for C++ code), PHP web server with SQL database (e.g., MySQL), Wi-Fi network.
Data	Temperature and humidity data (measured values), HTTP codes and server response messages (for error analysis).

Guidelines (Lesson Plan) 💡

Part	Activity	Pedagogical Focus (Based on Sources)
Part 1: Introduction and Hardware (2 hours)	Introduction to the DHT11 sensor and ESP32 basics, local data acquisition (via serial monitor). Theoretical fundamentals of network connectivity: SSID, password, `connectWiFi()` function.	*Flipped Classroom* approach: Students acquire basic concepts independently (e.g., DHT11 operation), focusing classroom time on the practical implementation of code and connectivity.
Part 2: Data Transmission and Server Communication (2 hours)	Understanding the ESP32 **HTTP POST** code (`http.POST(postData)`) and setting the necessary HTTP headers. Reviewing the PHP/SQL code for receiving data and writing to the database (transaction management/table truncation after 10 records).	Research Methodology: Focus on the **infrastructure for obtaining and managing data** (B2). *Problem-Solving (RBT):* Analyzing error codes (e.g., `httpCode`) to understand the reasons for failed requests.

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

Part 3: Web Visualization and Conclusion (2 hours)	Understanding the structure of the webpage (HTML/PHP) that uses the **GET** method to query data and display it in a table. Reviewing the data collection design (A2): What happens when 10 records reach the limit?. Project refinement and presentation of results.	Collaborative Learning: Students share their work and discuss how data visualization (e.g., graphs) could be improved (A4). Encouraging the creation of Open Educational Resources (OER) based on knowledge gained.

---

2. Good Practice: Arduino UNO – Cost-Effective Wi-Fi Solution (ESP-01S)

This practice demonstrates how the weather station can be implemented with a less powerful but cheaper microcontroller (Arduino UNO), utilizing the ESP-01S module as a Wi-Fi modem.

Lesson Plan (English)

Title	**Arduino UNO + ESP-01S: Wi-Fi Adaptation Under Constraints**
STE(A)M Subjects	Technology, Engineering, Physics, Problem-Solving
Target Age	14–18 (VET)
Duration	6 hours (3 x 2-hour sessions)
Key Question	How can we use devices with limited resources (memory, speed) for network communication?

Short Presentation

This project uses the Arduino UNO to read sensor data (DHT11) but employs the **ESP-01S module** as a Wi-Fi adapter. Communication between the two units occurs via a software serial port (`SoftwareSerial`) using **AT commands** to control the Wi-Fi functionality. This highlights the importance of managing limited resources, adapting programming techniques (e.g., incorporating **delays**), and using the existing web infrastructure (PHP/SQL) without changes.

Objectives/Learning Outcomes 🎯

Category Learning Outcomes (LOs)

Students will be able to recognize hardware constraints (e.g., memory/speed).
They will be able to adapt a research design (A3) to the available infrastructure.

Discover the Green Life with Robots
2023-1-HU01-KA210-VET-000156243

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

For Students	They will be able to set up software serial communication and use AT commands. They will be able to send HTTP POST requests to an existing web interface via the microcontroller.
For Teachers	Teachers will be able to demonstrate the challenges of engineering design and programming through a low-budget, yet functional, solution.

Tools and Data 🛠

Category	Description
Hardware	Arduino UNO, ESP-01S Wi-Fi module, DHT11 sensor, jumper wires (especially TX/RX connections).
Software/Infrastructure	Arduino IDE (SoftwareSerial library), PHP web server with SQL database (used without change).
Data	Temperature and humidity data, AT commands via serial communication.

Guidelines (Lesson Plan) 💡

Part	Activity	Pedagogical Focus (Based on Sources)
Part 1: Hardware Constraints and Serial Communication (2 hours)	Review of wiring the Arduino and ESP-01S (VCC, GND, RX, TX). Setting up and testing serial communication (`SoftwareSerial`). Theoretical review of performance differences compared to the ESP32.	*Critical Thinking:* Discussing how hardware selection impacts project complexity and reliability.
Part 2: Wi-Fi Connection with AT Commands (2 hours)	Implementing the `connectWiFi()` function using AT commands (e.g., `AT+CWMODE=1`, `AT+CWJAP`). Analyzing the necessity of the `delay()` function while waiting for AT command responses to ensure reliable operation.	Research-Based Teaching (RBT): Students learn to document workflows, such as precise delay times, and justify design decisions (A4).
Part 3: HTTP POST with AT Commands (2 hours)	Creating the `sendToServer()` function that manually builds the HTTP request (with AT commands), including the TCP connection and the Content-Length header. Testing	Problem-Solving Skills: Students learn how to implement tasks manually (low-level) that a more powerful device handles automatically (high-level).

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

data transmission, comparing it to the ESP32 solution (performance discrepancies).

---

3. Good Practice: Arduino UNO – Wired Ethernet Solution

This practice introduces a third approach, focusing on network stability using the wired Ethernet Shield W5100, circumventing Wi-Fi limitations.

Lesson Plan (English)

Title	**Arduino UNO – Wired Stability with Ethernet Shield W5100**
STE(A)M Subjects	Technology, Networking, Engineering, Physics
Target Age	14–18 (VET)
Duration	6 hours (3 x 2-hour sessions)
Key Question	When is a wired connection necessary for data collection, and how do we configure low-level network settings (e.g., IP addresses)?

Short Presentation

This project connects the Arduino UNO and the DHT11 sensor to the network using the **Ethernet Shield W5100**. Since the Arduino UNO does not have built-in Wi-Fi, it accesses the internet via a wired connection. The code uses the `Ethernet.h` library and demonstrates how to configure network parameters such as the **static IP address** and MAC address.

Objectives/Learning Outcomes 🎯

Category Learning Outcomes (LOs)

For Students	Students will be able to select and use libraries dependent on the hardware infrastructure (e.g., `Ethernet.h` instead of `WiFi.h`). They will be able to initialize a local network interface (MAC, IP address setup). They will be able to discuss the pros/cons of wired versus wireless data transmission solutions (C2). They will be able to manage network client objects (`client.connect()`, `client.println()`, `client.stop()`).
For Teachers	Teachers will be able to provide a deeper understanding of networking technologies (IP addressing, DHCP vs. static configuration).

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.

Tools and Data 🛠

Category	**Description**
Hardware	Arduino UNO, Ethernet Shield W5100, DHT11 sensor, Ethernet cable.
Software/Infrastructure	Arduino IDE (Ethernet.h and SPI.h libraries), PHP web server with SQL database (the code uses `example.com` as a placeholder server).
Data	MAC address, static IP address, temperature and humidity data.

Guidelines (Lesson Plan) 💡

Part	**Activity**	**Pedagogical Focus (Based on Sources)**
		message).
Part 1: Network Initialization (2 hours)	Wiring and initializing the Ethernet Shield. Introducing the concepts of MAC address and static IP address, and understanding their significance in network communication.	Research Methodology (RBT): Students must understand the *variables* (e.g., IP address, port, server) and conditions (e.g., logic for switching to static IP upon DHCP failure) as part of the research design (A2.1).
Part 2: Wired Data Submission (2 hours)	Reading DHT11 data. Constructing the HTTP POST request using `client.println()` commands (manual implementation of the HTTP/1.1 protocol). Emphasizing the importance of starting and stopping TCP connections (`client.connect()`, `client.stop()`). Handling the server response (`client.readString()`).	Engineering: Analyzing the stability and reliability of wired protocols versus the potential errors associated with Wi-Fi.
Part 3: Error Analysis and Portability (2 hours)	Portability analysis of the wired solution: the server-side PHP/SQL code remains unchanged. Comparing wired/wireless solutions, project finalization.	Transferable Skills: Developing analytical and problem-solving skills during the diagnosis of network failures (e.g., analyzing the "Connection failed"

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authoriry can be held responsible for them.