SHEng
SMART HEALTHCARE ENGINEERING
2023-1-RO01-KA220-HED-000159985

Co-funded by
the European Union

SMART HVAC & PSYCHROMETRICS LAB

Class content material

This document is part of a training materials collection developed within the Erasmus+ project “Smart healthcare engineering”, SHEng, project No. 2023-1-RO01-KA220-HED-000159985.

The collection is intended for engineering students and professionals working in the healthcare sector. It covers technical subjects relevant to hospital infrastructure, equipment, materials and support services, helping readers understand how engineering contributes to safe patient care and the reliable operation of healthcare facilities.

These materials support classroom teaching and independent study. They should be used alongside the project handbook and the presentations, courses and other learning resources available on the project website. Each document addresses a specific subject, while the collection provides a broader understanding of the responsibilities and practical challenges of healthcare engineering.

The content is educational. Its application in practice requires consideration of the conditions of each healthcare facility, the applicable standards and the instructions provided by equipment manufacturers.

Project website: www.smarthealtheng.com

SHEng
SMART HEALTHCARE ENGINEERING

Transilvania University of Brasov
ULPGC Universidad de Las Palmas de Gran Canaria
MANİSA CELAL BAYAR ÜNİVERSİTESİ
Servicio Canario de Salud
SPITALUL MUNICIPAL SĂCELE
Co-funded by the European Union

SMART HVAC & PSYCHROMETRICS LAB

Hands-on Air Handling Unit Automation with Arduino & Processing
— Theoretical, Hardware, & Pedagogical Course Manual —

Carlos J. Sánchez Morales, University Las Palmas de Gran Canaria, Spain

Lab Workshop Module:	HVAC Automation & Real-Time Psychrometric Embedded Systems
Hardware Platforms:	Arduino Nano (ATmega328P), DFRobot CHT832X, PH2216A0 Relay Module
Software Platforms:	Arduino IDE 2.x, Processing 4.x GUI, CH340 USB Driver

1. Course Introduction & Executive Summary

Heating, Ventilation, and Air Conditioning (HVAC) systems constitute the core environmental regulation engine in modern civil, industrial, and healthcare infrastructure. Beyond simple thermal comfort, precise atmospheric control plays a vital operational role in specialized healthcare environments (e.g., surgical suites, isolation wards, cleanrooms), where humidity, air exchange rates, and thermal pressure gradients directly influence pathogen viability, airborne infection vectors, and bio-burden control.

The 'SMART HVAC & PSYCHROMETRICS LAB: Hands-on Air Handling Unit Automation with Arduino & Processing' workshop was designed to bridge the gap between abstract thermodynamic theory and physical embedded automation. Through the assembly, calibration, and algorithmic control of a benchtop scale Air Handling Unit (AHU), participants engage directly with closed-loop sensor-actuator loops, real-time psychrometric charting, and digital human-machine interfaces (HMI).

This course manual serves as the comprehensive foundational theoretical text supporting the laboratory experience. It covers physical hardware architectures, physical signal transduction, software ecosystem paradigms, open-source democratization, AI-augmented engineering workflows, thermodynamic principles of moist air, serial protocol communications, graphical interface design, control logic algorithms, and specialized healthcare engineering applications.

2. Open-Source Ecosystems & Educational Paradigms in System Engineering

2.1 The Open-Source Revolution: Arduino Architecture & How It Works

Arduino is an open-source electronics platform based on flexible, easy-to-use physical hardware and software. At its core, an Arduino board acts as the physical microcontroller—the embedded 'brain' that interacts deterministically with the physical world.

The architecture of an Arduino board (such as the Nano powered by the ATmega168/328 8-bit RISC microcontroller) revolves around a centralized processing core, hardware timers, Analog-to-Digital Converters (ADC), digital General-Purpose Input/Output (GPIO) pins, and hardware communication buses (UART, I2C, SPI). Unlike general-purpose operating systems, microcontroller firmware executes in a single-threaded execution loop (the setup() and loop() structure), guaranteeing microsecond-level timing determinism required for real-time sensing and switching.

System Operational Mechanism: How Arduino Works:

1. Physical Sensing: The microcontroller samples physical parameters (e.g., temperature, humidity) through digital data lines (I2C) or analog voltages.
2. Algorithmic Computation: The CPU evaluates incoming data against programmed control thresholds or mathematical equations.
3. Hardware Actuation: The microcontroller changes output pin states (0V LOW or 5V HIGH), driving transistors or relay driver coils to switch high-power loads.
4. Telemetry Stream: Concurrently, the board streams ASCII-encoded data packets across USB Serial (via CH340 UART bridge) to host computer software.

2.2 High-Level Visual Computing: Processing Architecture & HMI Role

Processing is an open-source visual software sketchbook and programming language built on top of Java. Originally designed for creative coding and computer graphics, Processing has become a premiere tool for rapid HMI (Human-Machine Interface) and SCADA (Supervisory Control and Data Acquisition) dashboard engineering.

Architecture & Operational Logic: Processing operates on an event-driven graphical architecture divided into setup() (canvas initialization, serial port binding) and draw() (a continuous loop running typically at 60 FPS). In an engineering automation context, Processing acts as the graphical supervisor: it continuously reads serial buffer streams sent by Arduino, parses incoming CSV strings, renders mathematical vector visuals (such as dynamic psychrometric charts), and captures graphical user events (button clicks, slider toggles) to send back control bytes to the hardware.

2.3 Low-Cost Cyber-Physical Systems for Complex Simulation

Simulating complex physical systems (such as thermodynamic air handling processes) traditionally required expensive industrial Programmable Logic Controllers (PLCs), proprietary SCADA licenses, and complex wiring. The open-source combination of Arduino and Processing democratizes physical simulation through low-cost, accessible technology:

1. Economic Accessibility & Scalability: High-cost industrial automation setups create entry barriers. Low-cost open-source platforms allow students to assemble individual physical benches, enabling personalized, error-tolerant exploration.

2. Hardware-in-the-Loop (HIL) Realism: Instead of relying purely on mathematical simulation software (e.g., MATLAB/Simulink), students experience real physical dynamics—including sensor drift, heat transfer delays, noise, and relay switching arcs.

3. Massive Community & Open Code Ecosystem: Open-source programming languages feature expansive global developer communities. Thousands of tested libraries (e.g., DFRobot_CHT832X) and open repositories allow students to integrate complex hardware within minutes.

2.4 Synergy with Artificial Intelligence (AI) & Future Engineering Capabilities

The current technological landscape is defined by the rapid convergence of low-cost open-source hardware and generative AI coding assistants (such as LLMs). This combination creates a transformative advantage for engineering education:

AI-Augmented Hardware Prototyping: AI tools act as 24/7 technical mentors, helping students debug syntax errors, optimize microcontroller memory usage, and generate custom Processing vector drawing routines in real time.

Development of Transversal Skills: By lowering technical barriers to entry in C++ and Java syntax, students can focus their cognitive effort on high-level system engineering concepts: control theory, thermodynamic state trajectories, closed-loop stability, and user experience design.

> ■ **PEDAGOGICAL CORE: TRANSVERSAL CAPABILITY DEVELOPMENT**
> *The combination of low-cost hardware, open-source libraries, and generative AI shifts technical learning from 'memorizing syntax' to 'systems architecture and critical problem-solving.' Students develop key transversal competencies: computational thinking, hardware-software troubleshooting, physical data interpretation, and interdisciplinary engineering logic—essential capabilities for modern Industry 4.0 and Healthcare 4.0 environments.*

3. Physical Hardware Transduction: Sensors, Relays & Actuators

To automate an Air Handling Unit, an engineer must master the physical and electrical translation layers that connect digital microprocessors with atmospheric thermodynamics. This section establishes the fundamental concepts of sensors, relays, and actuators.

3.1 The Concept of a Sensor (Physical-to-Digital Transduction)

A sensor is an electrical input device that measures a physical property from the environment and converts it into a quantitative electrical signal readable by a microprocessor.

In our benchtop AHU, the DFRobot CHT832X serves as a dual capacitive and semiconductor sensor. Atmospheric temperature alters semiconductor bandgap voltage, while relative humidity modifies the dielectric permittivity of a polymer film capacitor. An integrated Analog-to-Digital Converter (ADC) translates these continuous physical changes into 16-bit digital values, transmitted over the I2C bus (SDA/SCL lines) to Arduino.

3.2 The Concept of an Actuator (Digital-to-Physical Action)

An actuator is an output device that converts an electrical control signal from a microcontroller into physical energy (mechanical movement, heat, or ultrasonic vibration) to modify the state of the physical system.

Actuators in the AHU Lab: Our AHU utilizes two distinct physical actuation mechanisms:

1. DC Ventilation Fan (Blower): Converts electrical energy into rotary mechanical torque using a brushless DC motor, creating forced convective airflow through the duct.

2. Micro ZJ03 Ultrasonic Atomizer: Converts electrical signals into high-frequency mechanical vibrations using a piezoelectric ceramic disc (~1.7 MHz), atomizing liquid water into micro-droplets to perform adiabatic humidification.

3.3 The Concept of a Relay (Galvanic Isolation & Power Switching)

A relay is an electrically operated switch that allows a low-power digital signal (such as a 5V, 20mA Arduino GPIO pin) to safely control a high-power electrical circuit (such as a 12V DC fan, an ultrasonic driver, or a 230V AC heating coil).

Galvanic Isolation & Switching Logic: Operating Principle: When Arduino sets pin D2 or D3 to HIGH, current flows through an internal optocoupler and energizes an electromagnetic coil inside the relay. The magnetic field pulls a flexible mechanical armature, switching the connection from Normally Closed (NC) to Normally Open (NO). This completes the power circuit for the fan or atomizer without exposing the delicate microcontroller to high currents or electrical voltage spikes.

Domain Role	Device Component	Physical Parameter	Electrical / Signal Domain
SENSOR (Input)	DFRobot CHT832X	Temperature & Relative Humidity	Physical State → Digital I2C Telemetry
RELAY (Intermediary)	PH2216A0 Dual Module	Electromagnetic Switching	5V Logic Signal → High-Power Circuit Closure
ACTUATOR (Output)	DC Fan / ZJ03 Atomizer	Air Movement & Water Vapor Mist	Electrical Energy → Mechanical Airflow & Ultrasonic Mist

4. Theoretical Fundamentals of Psychrometrics & Air Conditioning

Psychrometrics is the thermodynamic study of physical and thermal properties of gas-vapor mixtures, specifically atmospheric air combined with water vapor. Understanding these properties is essential for designing and controlling Air Handling Units (AHUs).

4.1 Key Thermodynamic Parameters

Dry-Bulb Temperature (T_{db} °C): The true ambient air temperature measured by a standard thermal sensor unaffected by moisture evaporation or radiation.

Relative Humidity (RH (%)**):** The ratio of the partial vapor pressure of water in air (p_v) to the saturation vapor pressure (p_{sat}) at the same dry-bulb temperature: $RH = (p_v / p_{sat}) \times 100\%$.

Humidity Ratio / Moisture Content (W (g_{H2O}/kg_{dry_air})**):** The actual mass proportion of water vapor per unit mass of dry air.

Dew Point Temperature (T_{dp} °C): The temperature at which a given sample of moist air reaches 100% relative humidity and water vapor begins condensing into liquid phase.

Wet-Bulb Temperature (T_{wb} °C): The lowest temperature achievable solely by evaporative cooling under adiabatic conditions.

Enthalpy (h (kJ/kg)**):** The total heat content of moist air, comprising both sensible heat (temperature dependent) and latent heat (phase-change dependent).

4.2 Governing Mathematical Formulations

In automated psychrometric monitoring, exact mathematical formulas are used to calculate thermodynamic parameters dynamically. For atmospheric pressure P_{atm} = 101.325 kPa, saturation vapor pressure $p_{sat}(T)$ is determined using Magnus-Tetens approximations:

■ **GOVERNING PSYCHROMETRIC EQUATIONS**

$p_{sat}(T) = 0.61078 \times \exp[\,(17.27 \times T) / (T + 237.3)\,]$ [in kPa]

Actual Vapor Pressure: $p_v = (RH / 100) \times p_{sat}(T)$

Humidity Ratio $W = 621.98 \times [\, p_v / (P_{atm} - p_v)\,]$ [in g/kg_{dry_air}]

Dew Point Temperature $T_{dp} = (237.3 \times \ln(p_v / 0.61078)) / (17.27 - \ln(p_v / 0.61078))$

4.3 Core Air Conditioning Processes

AHU Process	Active Component	Thermodynamic Effect	Psychrometric Vector
Sensible Heating	Electric Heater Coil	T_db ↑, W constant, RH ↓	Horizontal Right Vector
Sensible Cooling	Chilled Water Coil	T_db ↓, W constant, RH ↑	Horizontal Left Vector
Isothermal Humidification	Ultrasonic Atomizer	T_db constant, W ↑, RH ↑	Vertical Upward Vector
Dehumidification	Cooling Coil (below T_dp)	T_db ↓, W ↓, RH ≈ 100%	Diagonal Down-Left Vector

5. Benchtop AHU Hardware Specifications & Wiring Matrix

The physical benchtop unit incorporates industrial-grade sensing and actuation scaled for laboratory safety and rapid prototyping:

Arduino Nano Microcontroller: Based on the ATmega168 chip operating at 16 MHz with 5V logic. Communicates with laptop host via CH340 USB-to-UART bridge.

DFRobot CHT832X Sensor: High-precision digital temperature and relative humidity sensor. Communicates via I2C interface (SDA/SCL). Operating voltage: 3.3V strict DC.

Dual Relay Module (PH2216A0): Electromechanical 2-channel relay board with optically isolated inputs (IN1, IN2) driven by 5V DC coils, capable of switching up to 250VAC/10A loads.

DC Ventilation Fan: Compact brushless 5V DC fan simulating the AHU supply/exhaust air blower.

Micro ZJ03 Ultrasonic Atomizer: Piezoelectric ceramic disc ultrasonic driver generating high-frequency vibrational mist (~1.7-2.4 MHz) to simulate adiabatic humidification.

5.1 Master Hardware Wiring & Interconnect Matrix

Component	Component Pin	Arduino Pin / Destination	Wire Color	Functional Description
CHT832X Sensor	VCC	3.3V Pin	Red	Power supply (STRICT 3.3V Max)
CHT832X Sensor	GND	GND Rail	Black	Common Ground
CHT832X Sensor	SDA	Analog Pin A4	Blue	I2C Serial Data Line
CHT832X Sensor	SCL	Analog Pin A5	Green	I2C Serial Clock Line

Dual Relay Module	VCC	5V Pin	Red	Relay Module Coil Power
Dual Relay Module	GND	GND Rail	Black	Relay Module Ground
Dual Relay Module	IN1	Digital Pin D2	Yellow	Fan Relay Control Signal
Dual Relay Module	IN2	Digital Pin D3	Yellow	Humidifier Relay Control Signal
Micro ZJ03 Atomizer	VCC / V+	Relay 2 NO Terminal	Red	Switched Power Line
Micro ZJ03 Atomizer	GND / V-	GND Common Rail	Black	Humidifier Power Ground

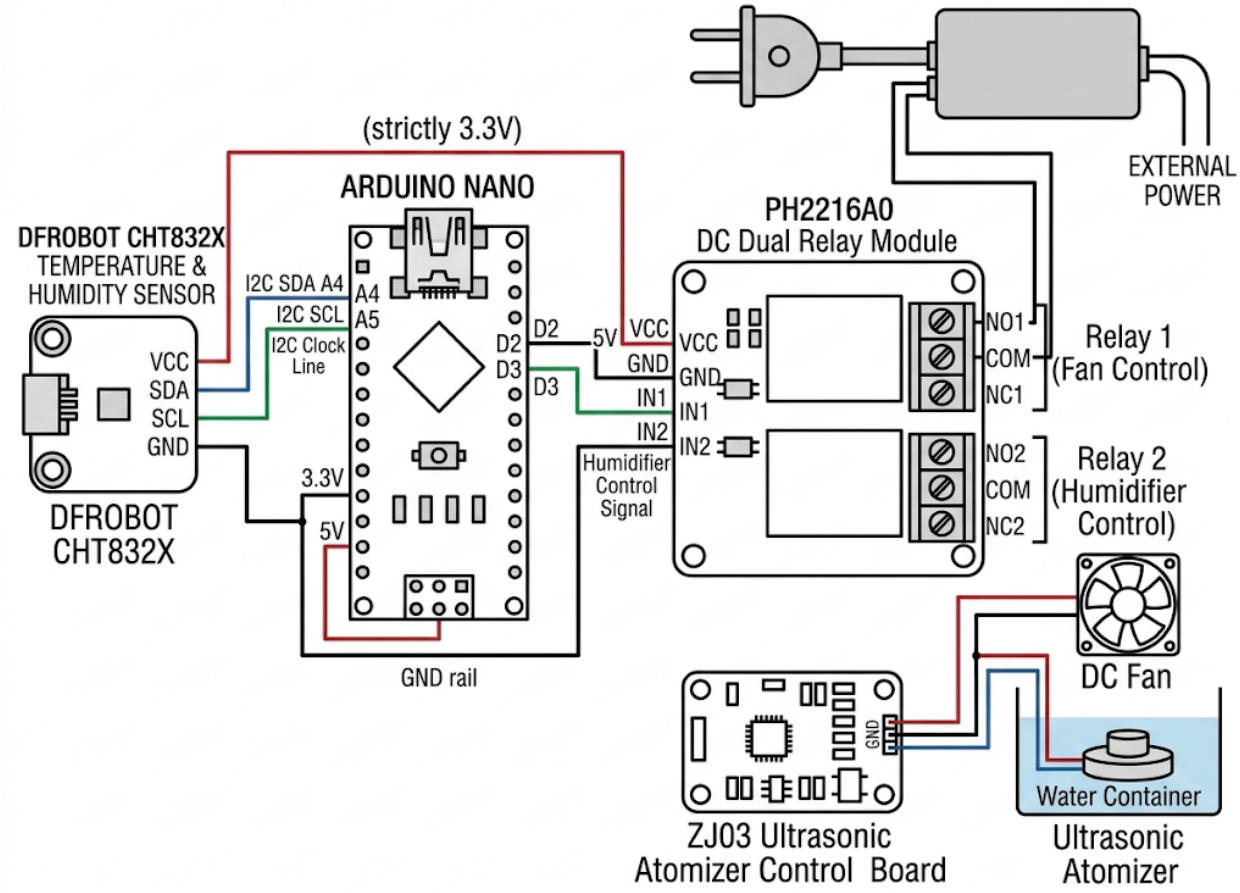

Fig 1. Hardware Electrical Interconnect & Component Connection Schematic

6. Embedded Firmware, Serial Protocol & GUI Integration

6.1 Serial Communication Standard & Decoding

The low-level firmware executing on Arduino acquires sensor data via I2C and transmits serialized ASCII CSV packets over USB-UART at 9600 baud. It continuously scans its RX buffer for command characters sent by Processing ('F', 'H', 'A', 'M').

■ SERIAL COMMUNICATION PROTOCOL SPECIFICATION

Arduino Outbound Packet Format:

[Temperature_degC] , [RelativeHumidity_pct] , [FanStatus_01] , [HumidifierStatus_01] \n

Example Serial String: "23.45,58.20,1,0\n"

Parsing Rule in Processing:

```
String inString = myPort.readStringUntil('\n');
float[] data = float(split(trim(inString), ','));
```

7. Control Logic Algorithms & Closed-Loop Automation

7.1 On-Off Hysteresis Control Strategy

To regulate relative humidity without causing rapid relay chattering (which damages mechanical relay contacts), an On-Off controller with adjustable hysteresis deadband ΔRH is implemented:

■ HYSTERESIS CONTROL ALGORITHM

IF RH_meas < (RH_setpoint - ΔRH) THEN:
Turn Humidifier Relay ON (Pin D3 = HIGH)
Ensure Fan Relay is ON (Pin D2 = HIGH) [Airflow required for mist dispersal]

ELSE IF RH_meas > (RH_setpoint + ΔRH) THEN:
Turn Humidifier Relay OFF (Pin D3 = LOW)

ELSE:
Maintain current relay state [Within Hysteresis Deadband]

7.2 Microcontroller Pseudocode

■ ARDUINO CLOSED-LOOP HYSTERESIS CONTROL CODE

```
// Global Constants & State Variables
float RH_setpoint = 55.0; // Target Relative Humidity (%)
float deadband = 2.5;    // Hysteresis Band (±2.5%)
bool fanState = false;
bool humState = false;

void loop() {
  float temp = cht832x.getTemperature();
```

```
  float humidity = cht832x.getHumidity();

  // Closed-loop Hysteresis Evaluation
  if (humidity < (RH_setpoint - deadband)) {
    humState = true;
    fanState = true;
  } else if (humidity > (RH_setpoint + deadband)) {
    humState = false;
  }

  digitalWrite(PIN_FAN, fanState ? HIGH : LOW);
  digitalWrite(PIN_HUMIDIFIER, humState ? HIGH : LOW);

  // Transmit telemetry string to Processing
  Serial.print(temp); Serial.print(",");
  Serial.print(humidity); Serial.print(",");
  Serial.print(fanState); Serial.print(",");
  Serial.println(humState);
  delay(500);
}
```

8. Laboratory Step-by-Step Practical Exercises

8.1 Exercise 1: Hardware Bench Assembly & Open-Loop Verification

Goal: Complete circuit wiring with USB power disconnected. Verify CH340 driver recognition, serial telemetry streams in Arduino IDE, and manual relay toggling ('F', 'H').

8.2 Exercise 2: Open-Loop Psychrometric Trajectory Mapping

Goal: Establish serial link between Processing GUI and Arduino. Manually activate fan and atomizer to plot isothermal humidification vectors in real time on the psychrometric chart.

8.3 Exercise 3: Closed-Loop Hysteresis Controller Tuning

Goal: Configure automated RH setpoint control at 60.0% with deadband ±2.0%. Introduce environmental humidity disturbances and analyze overshoot, undershoot, and stability cycles.

9. Smart Healthcare Engineering & Regulatory Standards

Within the SHEng and UNILAB initiatives, precise psychrometric control is essential for infection containment and cleanroom air quality.

High Humidity Risks (> 60% RH): Promotes fungal spore proliferation (Aspergillus spp.) and bio-film formation within HVAC ductwork.

Low Humidity Risks (< 40% RH): Accelerates evaporation of respiratory droplets into infectious droplet nuclei, increasing viral persistence (Influenza, SARS-CoV-2) and drying patient mucosal membranes.

Optimal Clinical Zone (40% - 60% RH): International hygiene standard (ASHRAE Standard 170) to minimize overall pathogen survival rates.

Healthcare Space	Target Temperature	Target RH Range	Pressure Gradient
Operating Rooms (OR)	20°C - 24°C	20% - 60% RH	Positive (+15 Pa min)
Airborne Infection Isolation (AIIR)	21°C - 24°C	30% - 60% RH	Negative (-2.5 Pa min)
Protective Environment (PE)	21°C - 24°C	30% - 60% RH	Positive (+8 Pa min)

10. Troubleshooting, Safety Protocols & References

> **■ CRITICAL CLASSROOM SAFETY RULES**
> *1. COLD-WIRING ONLY: Always disconnect USB power prior to wire manipulation on the breadboard.*
>
> *2. SENSOR VOLTAGE WARNING: The DFRobot CHT832X sensor MUST strictly be powered from the 3.3V pin. 5V will permanently destroy the IC.*
>
> *3. PORT CONFLICT MANAGEMENT: Close Arduino IDE Serial Monitor before running Processing to avoid COM port locks.*

10.1 Diagnostic Guide

Symptom / Error	Probable Root Cause	Corrective Action
Arduino upload fails (stk500_getsync)	Incorrect bootloader selected in IDE	In Arduino IDE, go to Tools > Processor and switch to 'ATmega168 (Old Bootloader)'
Sensor reads 0.0°C / 0.0% RH continuously	I2C wiring fault or incorrect library	Verify SDA -> A4 and SCL -> A5 connections. Ensure DFRobot_CHT832X library is correctly installed
Relays click repeatedly every second	Hysteresis deadband set too narrow	Increase deadband value (e.g., from ±0.5% to ±2.5% RH) in control code
Atomizer board gets warm but no mist	Insufficient water level or piezo disc dry	Ensure piezoelectric ceramic disc is in direct contact with water reservoir surface

10.2 Academic & Regulatory References

1. ASHRAE. (2021). Handbook - Fundamentals (SI Edition). American Society of Heating, Refrigerating and Air-Conditioning Engineers.

2. ASHRAE Standard 170. (2021). Ventilation of Health Care Facilities. ASHRAE ANSI Standard.

3. Banzi, M., & Shiloh, M. (2014). Getting Started with Arduino: The Open Source Electronics Prototyping Platform. Maker Media.

4. Reas, C., & Fry, B. (2014). Processing: A Programming Handbook for Visual Designers and Second Edition. MIT Press.

5. World Health Organization (WHO). (2009). Natural Ventilation for Infection Control in Health-Care Settings.

Acknowledgments

The content was supported by the project “Smart Healthcare Engineering (SHEng)”, Grant Agreement No. 2023-1-RO01-KA220-HED-000159985, co-funded by the Erasmus+ programme of the European Union. Results, views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor the granting authority can be held responsible for them.