جامعة الحسين التقنية
Al Hussein Technical University

ASSIGNMENT BRIEF

HTU Course No: 30202452	**HTU Course Name:** Internet of Things
BTEC Unit Code:	**BTEC UNIT Name:**

Student Name/ID Number/Section	
HTU Course Number and Title	30202452 Internet of Things
BTEC Unit Code and Title	
Academic Year	2024-2025 2
Assignment Author	Marwa Altahaineh
Course Tutor	Marwa Altahaineh
Assignment Title	Smart EcoPod – Self-Regulating Micro-Living Capsule
Assignment Ref No	1
Issue Date	17/05/2025
Formative Assessment dates	From 18/05/2025 to 29/05/2025
Submission Date	19/06/2025
IV Name & Date	Malek Allouzi 16/05/2025

Submission Format

The assignment is divided into parts, and each part should be submitted on the specific date that is shown in the assignment brief and guidance section.

Part 1:

In class closed book and open notes examination.

Part 2:

The submission of this part consists of Four parts:

- An individual written report that has a solution for tasks that ends with the word (**Report**).
- A full working one Python code for tasks that require code implementation, these tasks will end with the word (**Code**).
- **Discussion** with your instructor (and any other witness) about the submitted work.
- Fully functional Hardware that represents the tasks, you should bring it with you to your oral discussion.

You need to follow the following guidelines, failing to follow them may result an ‘unclassified’ grade.

Assignment Guidelines

- You are required to submit a well-formatted **Word version** report that provides a complete answer for all required tasks.
- The Hardware and code are group work, the group consists of two students, but the report is an individual report.
- The submission deadline of this part will be on Thursday 19/6/2025 at 11:59 PM.
- Full and clear answers for all required tasks, mention the task number and the subtask number before each answer**.**
- Don’t compress your work, you should submit your report and your code as separate files.
- Soft-copy submissions are only allowed, you are required to upload your submission files to the university’s eLearning platform through (https://elearning.htu.edu.jo/) within the submission date and time stated above. **NO SUBMISSION by EMAIL and NO LATE SUBMISSIONS WILL BE ACCEPTED.**

- If you commit any kind of plagiarism, HTU policies and regulations will be applied.
- The Discussion will be one to one oral discussion between you and your instructor, which includes debugging, analyzing, and evaluating tasks and the code.
- Attendance of the oral discussion is mandatory in the date and time determined by your instructor; the exact discussion schedule will be announced after your submission.

Unit Learning Outcomes

LO1 Hardware Assembly and Configuration

LO2 Setting Up Raspberry Pi and Python Programming for IoT

LO3 Understand and Apply Communication Protocols in IoT Systems

Assignment Brief and Guidance

Part 1:

In class closed book and open notes examination, you are allowed to bring one A4 page only contains your necessary notes. This part will be held on Saturday 17/5/2025.

Your answer should be clear, and providing a final answer without showing detailed steps is not acceptable.

Part 2:

Scenario

You are part of an innovative and forward-thinking project that aims to create smart solutions for small living spaces. Your team is responsible for designing a Smart EcoPod – a micro-living capsule supplied with sensors, intelligent controls, and remote monitoring to support modern eco-friendly metropolitan lifestyles or space missions.

Your job as an IoT developer is to create and deploy an operational model of the EcoPod's automated environmental control system. The system must control lighting and air circulation, temperature, and humidity, provide keypad interaction locally, and use cloud and MQTT protocols for communication remotely with other pods.

Task 1:

You will use the following peripheral components along with Raspberry Pi to design your basic smart control system for an indoor pod.

- DHT11 sensor: is used to monitor the pod's internal temperature and humidity.
- LED: simulates smart indoor lighting, which can be controlled by the user.
- Keypad: is a user interface for commands on the system.
- Motor: to mimic the ventilation (air circulation) system.

1) Design a schematic diagram of how each of the aforementioned components will be connected to your Raspberry Pi, and label the GPIO pins clearly for all components**.(Report)**

2) After designing the circuit and attaching all components to the Raspberry Pi, you must write the relevant Python code to run the system. **(Code)**

Your code should have the following functionality:

A) Reading DHT11 Sensor Data:

- Create a Python program that reads temperature and humidity values using the DHT11 sensor.
- Display the readings on the terminal, along with a timestamp to observe and track the environmental changes within the pod.

B) Smart Lighting Control (LED):

Update your code you will use the keypad to provide manual control of the lighting system (switching LED on & off) as follows:

- When the user presses the key 'A' on the keypad, the LED (simulated light) the light turns on.
- When the user presses the key 'B' on the keypad, the LED (simulated light) should turn off

C) Automated ventilation control (servo motors):

The ventilation system (represented by a servo motor) should respond either to keypad input or temperature conditions.

- Pressing key '0' on the keypad OR, if the temperature increases above 28°C, spin the servo 90° to open the ventilation flap.
- Pressing key '1' on the keypad OR, if the temperature dips below 22°C, spin the servo back to 0° to close the ventilation.

Task 2:

As a member of the smart automation team, you are in charge of allowing remote data gathering and visualization to enhance the comfort and customization of the smart pod experience, this company is introducing a remote environmental synchronization service. Accordingly, this service will enable the pod's circumstances to be tracked and analyzed remotely to change lighting and ventilation profiles for specific users. Consequently, you will modify your Python code to continuously gather temperature and humidity data from the DHT11 sensor and send it to a ThingSpeak channel. **(Code)**

1) Data should be uploaded every 20 seconds using the HTTP protocol.

2) Examine the existing IoT communication protocols and evaluate them based on their data sharing methodologies, efficiency, and usage.**(Report)**

Task 3:

Your EcoPod is part of a wider, interconnected network of smart environmental pods spread over several locations. These pods are meant to share environmental data and receive centralized orders via the MQTT communication protocol, ensuring uniform environmental performance across all units.

Above and over, to develop your system you are responsible for adding the following feature regarding inter-pod communication and command handling:

1) A secure shutdown procedure has been defined for all EcoPods.Your system must recognize the universal shutdown code *358#. When this code is entered locally using the keypad or received remotely from another pod over MQTT, the pod must:

- Deactivate all active systems.
- Turn off the light source (LED).
- Stop the ventilation system (motor).

This code is anticipated to propagate and take effect throughout the network. **(Code)**

2) Our EcoPod must continue operating even when internet connectivity is lost. Document the MQTT system's expected behavior when the pod loses internet access. Describe how to maintain communication between pods in the absence of an internet **(Report)**

3) Now, when EcoPods get environmental data via MQTT channels, they have to react dynamically.

The network's humidity information should be used to modify your pod's lighting system:

- Turn on the lights if the received humidity level rises above a predefined threshold.
- Turn off the lights if the humidity drops below a predefined threshold.
- Document the implementation of this feature.
- Demonstrate in detail how your pod interprets incoming MQTT messages and how decisions can be taken using the information. **(Report + Code)**

Learning Outcomes and Assessment Criteria			
Learning Outcome	**Pass**	**Merit**	**Distinction**
LO1 Hardware Assembly and Configuration	**P1** Understand how to connect basic input components (e.g., sensors, switches) using GPIO pins. **P2** Understand how to connect output components (e.g., LEDs, buzzers, motors) to GPIO pins.	**M1** Design a schematic diagram for a functional hardware system integrating multiple sensors and actuators.	**D1** Develop a Python program that varies PWM signals dynamically to control multiple PWM devices (e.g., LEDs or motors). **D2** Demonstrate the Ability to Develop and Implement a Fully Functional IoT Project Using MQTT Protocol for data sharing between devices and IoT cloud platform.
LO2 Setting Up Raspberry Pi and Python Programming for IoT	**P3** Write Python scripts to interact with basic input and output devices (e.g., read sensor data, keypad, control LEDs). **P4** Configure network settings and enable remote access (e.g., SSH, VNC) for headless operation. **P5** Identify and describe the function of Raspberry Pi GPIO pins.	**M2** Develop event-driven programs for GPIO operations and differentiate them over polling programs.	
LO3 Understand and Apply Communication Protocols in IoT Systems	**P6** Identify and understand at least one key IoT communication protocol (e.g., MQTT, HTTP, CoAP, WebSocket).	**M3** Design and implement a custom IoT project utilizing one IoT communication protocol for data sharing between an IoT device and an IoT cloud platform (e.g., AWS IoT, Google Cloud IoT, or ThingSpeak).	