STEM MODULE

INTERNET OF THINGS FOR A SUSTAINABLE FUTURE

IZANOORDINA AHMAD
ZARITH LIYANA ZAHARI
FAEZAH HARUN
SAFYZAN SALI
SITI MARWANGI MOHAMAD MAHARUM

STEM:
Internet of Things for A Sustainable Future

Izanoordina Ahmad
Zarith Liyana Zahari
Faezah Harun
Safyzan Salim
Siti Marwangi Mohamad Maharum

STEM:
Internet of Things for A Sustainable Future

Izanoordina Ahmad
Zarith Liyana Zahari
Faezah Harun
Safyzan Salim
Siti Marwangi Mohamad Maharum

First Publication 2025
©Copyright by Universiti Kuala Lumpur Publishing 2025

All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written consent of Universiti Kuala Lumpur Publishing Unit, including in any network or other electronic storage or transmission, or broadcast for distance learning

eISBN 9786297837253

Published by

UNIVERSITI KUALA LUMPUR PUBLISHING
1016, Jalan Sultan Ismail, 50250 Kuala Lumpur MALAYSIA
Email: uniklpress@unikl.edu.my

PREFACE

STEM Module: Internet of Things For A Sustainable Future

STEM Module: Internet of Things For A Sustainable Future is an initiative by the STEM (Science, Technology, Engineering, and Mathematics) Committee of UniKL British Malaysian Institute (UniKL BMI) under the Centre for Science, Engineering, and Technology (CSET), Universiti Kuala Lumpur (UniKL). It has been developed to enhance students' engagement in Science, Technology, Engineering, and Mathematics (STEM) by incorporating innovative learning approaches through Internet of Things.

This module aligns with and supports the government's direction in advancing Technical and Vocational Education and Training (TVET), promoting the adoption of Industry 4.0 (IR 4.0) technologies, and contributing to the achievement of the Sustainable Development Goals (SDG). By integrating these national and global priorities, it seeks to prepare students with the knowledge and skills necessary to thrive in a rapidly evolving industrial and technological landscape.

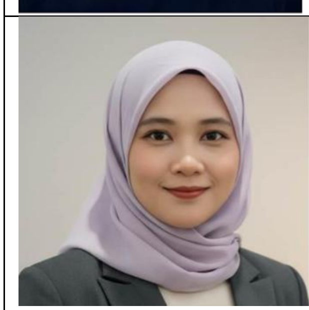
The convergence of technology and environmental imperatives in IoT has ushered in an era of unprecedented challenge and opportunity. As the global community grapples with the escalating crises of climate change, resource depletion, and ecological degradation, the need for innovative, scalable, and impactful solutions has never been more urgent. This ebook, *Internet of Things For A Sustainable Future*, is born from the conviction that the Internet of Things (IoT) holds a transformative key to addressing these systemic problems.

While the path to a sustainable future is complex and fraught with challenges—including issues of data privacy, security, and digital equity—the potential of IoT to drive positive change is undeniable. We hope this book will inspire a new generation of innovators to look at a sensor, a cloud platform, or a stream of data, and see not just technology, but the building blocks of a resilient, equitable, and sustainable world.

It is our belief that by connecting things, we can, in turn, better connect with the well-being of our planet. The future is connected. The future must be sustainable.

Thank you.

AUTHORS' BIOGRAPHY

	Assoc. Prof. Ts. Dr Izanoordina Ahmad Area: Aritificial Intelligent, Internet of Things, Wireless Sensor Networks, Electrical and Electronics. UniKL BMI izanoordina@unikl.edu.my	
	Ts. Dr. Zarith Liyana binti Zahari Area: Artificial Intelligence (AI), Machine Learning, Deep Learning, Signal Processing, Microprocessor Architecture, and Embedded Systems Programming zarithliyana@unikl.edu.my	
	Ts. Dr. Faezah Harun Area: III–V compound semiconductor materials and devices, IoT, machine Learning UniKL BMI faezah@unikl.edu.my	
	Mr. Safyzan Salim Area: Internet of Things (IoT), LoRa, and Machine Learning applications. UniKL BMI safyzan@unikl.edu.my	
	**Assoc. Prof. Ts. Dr.** **Siti Marwangi Mohamad Maharum** Area: 5G/mmWave networks, the Internet of Things (IoT), and AI-driven engineering applications. UniKL BMI sitimarwangi@unikl.edu.my	

FOREWORD

I am truly delighted to celebrate the successful completion of five remarkable STEM modules developed collaboratively by UniKL MICET, RCMP, MITEC, BMI, and MESTECH, with the invaluable support of UniKL Press and MESRA. This achievement reflects the passion, innovation, and teamwork that define UniKL's spirit of excellence in advancing STEM education across Malaysia.

Aligned with the Dokumen Standard Kurikulum dan Pentaksiran (DSKP) under the Ministry of Education (KPM), these modules offer practical, creative, and outcome-based learning experiences that bring science, technology, engineering, and mathematics to life. Their integration with STEM4U initiatives, in collaboration with MOSTI and industry partners such as Alibaba.com, demonstrates our strong synergy between academia and industry. Grounded in the Sustainable Development Goals (SDGs), these modules inspire forward-thinking and responsible learning.

I have high hopes that the modules will become a standard kit of STEM modules for UniKL in the long run. They ignite curiosity, creativity, and confidence in every learner—nurturing the next generation of innovators and changemakers.

PROF. DR. KUSHSAIRY ABDUL KADIR
DEPUTY PRESIDENT, ACADEMIC & TECHNOLOGY
UNIVERSITI KUALA LUMPUR (UniKL)

ACKNOWLEDGEMENT

From the Director of UniKL CSET

As the Director of the Centre for STEM Enhancement & TVET (CSET), I am honoured to acknowledge the completion of five STEM modules developed collaboratively by UniKL MICET, RCMP, MITEC, BMI, and MESTECH, with the support of UniKL Press. These modules reflect a year-long effort and strong commitment to educational excellence, developed under the close guidance of the Ministry of Education (KPM) and fully aligned with the *Dokumen Standard Kurikulum dan Pentaksiran* (DSKP). They are designed to support both teachers and students in navigating the STEM curriculum more effectively.

In addition, the modules incorporate input from MESRA to ensure relevance to current school needs—ranging from short 45-minute activities to two-day workshops—and have been successfully implemented through STEM4U initiatives alongside MOSTI and industry partners such as Alibaba.com. The content is also aligned with the Sustainable Development Goals (SDGs), promoting forward-thinking and responsible education. Together, we hope that these modules will be a standard practice of activities used by all at UniKL when engaging with schools and industry in the years to come.

I would like to extend my deepest appreciation to the President of Universiti Kuala Lumpur, Prof. Ir. Dr. Azman Senin, Deputy President (Academic & Technology) Prof. Dr. Kushsairy Abd Kadir, dedicated authors and UniKL Press, for their dedication, continuous support and guidance in improving the publication process. Most importantly, Alhamdulillah—for Allah's grace in easing this meaningful journey.

TS. DR. MUZAFAR ZULKIFLI
DIRECTOR, CENTER OF STEM ENHANCEMENT & TVET (CSET)
UNIVERSITI KUALA LUMPUR (UniKL)

MODULE INFORMATION

STEM TITLE	RELATED SUBJET	RELATED LEARING AREA	RELATED CONTENT STANDARD AND LEARNING STANDARD	ACTIVITY	*SDG	PAGE
STEM 1: Internet of Things for A Sustainable Future	Technology Applications (RBT)/ Control Structure (SK)	1.1 Mechatronics Design	Mechatronics Design 1.1.5 1.1.6 1.1.7 Control Structure (SK) 1.4.1 1.4.2 1.4.3 1.4.4	Mechatronics Design Control Structure (SK)	9	1 - 37

TABLE OF CONTENT

STEM MODULE

STEM 1:

INTERNET OF THINGS FOR A SUSTAINABLE FUTURE

STEM 1: Internet of Things For A Sustainable Future

Learning Area	Technology Applications (RBT) /1.4 Control Structure (SK)
Content Standard	1.1 Mechatronics Design
Learning Standard	Students could: 1.1 Mechatronics Design 1.1.5 Produce sketches of parts of the product design that will be modified based on the block diagrams involved. 1.1.6 Make improvements to the product based on the sketches that have been made. 1.1.7 Evaluate the functionality of the modified product. 1.4 Control Structure (SK) 1.4.1 Explain the flow of the optional control structure 1.4.2 Write a program using the optional control structure by combining relational operators and logical operators 1.4.3 Explain the flow of the iterative control structure 1.4.4 Write a program using the iterative control structure

Title: Internet of Things for A Sustainable Future

Introduction	IoT workshop or bootcamp modules provides a practical platform for students from secondary schools that could strengthen their skills in science, technology, engineering, and mathematics (STEM). The students will learn how to work with sensors, data, and networks, which are fundamental to these fields. They will work in groups of four to develop the mini project based on the given gudelines and procedures. To reinforce learning and ensure a fun and engaging experience, the workshop involve coding and programming using Python for IoT application, the understanding of basic architecture: sensors, actuators, connectivity, data processing, and user interfaces. Students can gain hands-on experience in these areas, which are highly valuable in the modern workforce. The workshop could be divided into 5 sub modules; **1. Introduction to IoT & The "Thing"** **2. Programming with Python: From Functionality input to Data Streams** **3. Sensing the World** **4. Connecting to the Cloud** **5. Making Sense of Data and Control** By the end of the workshop, students will have a comprehensive understanding of how to develop the application of IoT such as the green house for agriculture, understand the basic programming and be well-prepared to apply these skills in their future project competition.

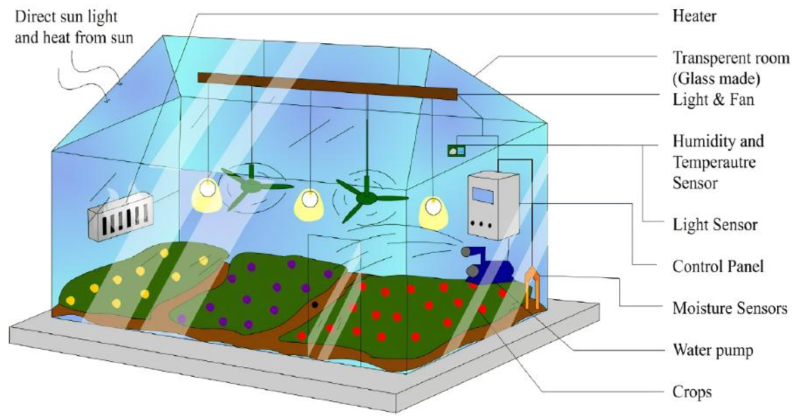
<table>
<tr><td></td><td>

Figure 1: Agriculture illustration</td></tr>
<tr><td>Problem Statement/ Key Issues</td><td>The implementation of IoT in Malaysia's such as its application in agriculture sector, often referred to as Smart Farming or Agriculture 4.0, faces the same fundamental challenges as other sectors, but with specific nuances due to the nature of farming. While the potential for IoT to boost productivity, improve food security, and reduce costs is immense, a number of barriers slow down its widespread adoption.

Include figure Problem-Based Learning Approach as below

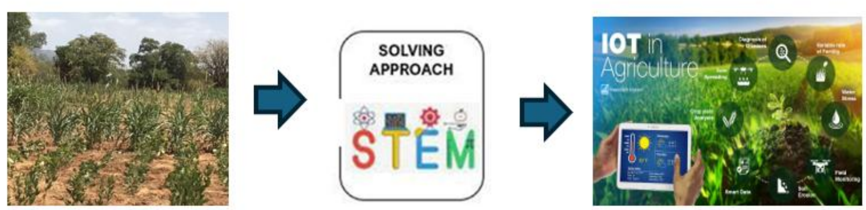


Figure 2: Problem Based Outcome</td></tr>
</table>

<table>
<tr><td>Objective</td><td>1. Create awareness on fundamental concepts of IoT in engineering disciplines.
2. Develop practical skills and proficiency in IoT and programming among students.
3. Provide hands-on experience by developing the prototype for project competition.
4. Foster engagement and interaction through fun activities.</td></tr>
</table>

<table>
<tr>
<td>STEM Elements</td>
<td>
<table>
<tr><td>Science</td><td>Technology</td><td>Engineering</td><td>Mathematics</td></tr>
<tr><td>√</td><td>√</td><td>√</td><td>√</td></tr>
<tr><td>Physics
Biology/Chemistry
Data Science</td><td>Technology application: hardware and software components that make the system work.</td><td>Manufacturing product. Practical application of scientific principles to design, build, and maintain the IoT system.</td><td>Data Analysis and Statistics:
Algorithm Development:
Computational Thinking:</td></tr>
</table>
Notes:

Technology and engineering element referring to the four domains and skills in design and technology subjects.
Mathematics element referring to the process standards.

<table>
<tr><th colspan="8">Codes of practices</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr>
</table>
Notes:
Module should contain at least two out of four of STEM elements (Science, Technology, Engineering and Mathematics)
Codes of Practices:
1. Problem statement
2. Model/ flow diagram development
3. Plan and conduct the investigation
4. Analysis and Interpretation
5. Apply mathematics/computational
6. Proposed Problem solving
7. Discuss and justify
8. Gather all the information (Communication/presentation).
</td>
</tr>
<tr><td>Skills</td><td>Observation, critical thinking, teamwork</td></tr>
<tr><td>Values</td><td>Scientific attitude, noble value</td></tr>
<tr><td>Apparatus</td><td>ESP32, temperature sensor DHT11 or BME, soil moisture, breadboard, jumper, OLED, cloud platform.</td></tr>
<tr><td>Materials</td><td>Soil and Plant</td></tr>
<tr><td>Suggested time</td><td>4 - 6 hours (Learning and Hand-on) for each module
45 minutes (Demonstration)
1 - 2-day workshop for all modules</td></tr>
</table>

CONFIDENTIAL

Strategy: 5E	
Engagement	Shows a few pictures related with IoT Implementattion in any application.
Exploration	Gather information related with IoT in agriculture such as its effect, source, mitigation action and treatment method • Conduct experiment or hands-on
Explanation	Conduct presentation on the findings
Elaboration	Guide students to do the programming, circuit construction and collect the data and send it to the cloud
Evaluation	Reflection of the learning process by completing the evaluation.

Procedure	**Step 1: Installation of Python and VSCode**

1. Install Python

- Download the latest version of Python from the official website:
  🔗 https://www.python.org/downloads/
- After installation, verify that Python is installed correctly:
  1. Open **Command Prompt**.
  2. Type:
     python --version
  3. Press **Enter** — the installed version should be displayed.

2. Install Visual Studio Code (VSCode)

- Download VSCode from the official website:
  🔗 https://code.visualstudio.com/download
- Install VSCode following the setup instructions.

3. Install Required Extensions in VSCode

1. Open VSCode.
2. Go to the **Extensions** panel.
3. Install and enable the following extensions:
   - **Python**
   - **Code Runner**

4. Restart VSCode

- Close VSCode completely and reopen it to apply changes.

5. Configure Code Runner Settings

1. Click the **gear icon** (⚙) at the bottom-left corner of VSCode to open **Settings**.
2. Search for:
   Code Runner: Run In Terminal
3. Enable (✓) this option.

6. Disable Automatic Updates

1. In **Settings**, search for:
   Update
2. Scroll down and disable (untick) the following:
   - **Extensions: Auto Check Updates** — to prevent automatic extension updates that may disrupt the programming process.

- **Update: Enable Windows Background Updates** — to prevent automatic background updates.

STEP 2: CIRCUIT CONSTRUCTION USING WOKWI.

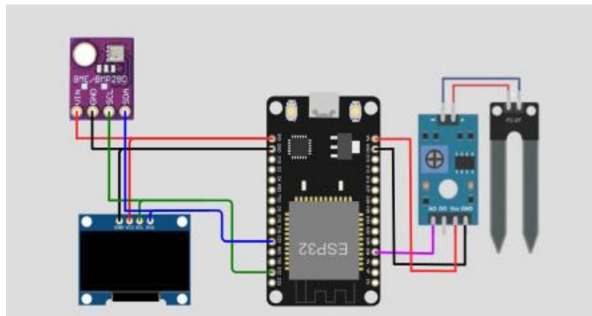


Figure 3 : Illustration of Circuit Diagram for Smart Plant Monitor for Green House

1. Components Required

- **ESP32** microcontroller
- **OLED Display** (I2C)
- **BME280 Sensor** (I2C)
- **Soil Moisture Sensor** (Analog output)

2. Connecting the OLED Display (I2C)

1. Connect **VIN** (OLED) → **3V3** (ESP32)
2. Connect **GND** (OLED) → **GND** (ESP32)
3. Connect **SDA** (OLED) → **GPIO 21** (ESP32)
4. Connect **SCL** (OLED) → **GPIO 22** (ESP32)

3. Connecting the BME280 Sensor (I2C)

The BME280 shares the same I2C pins as the OLED display.

1. Connect **VIN** (BME280) → **3V3** (ESP32)
2. Connect **GND** (BME280) → **GND** (ESP32)
3. Connect **SDA** (BME280) → **GPIO 21** (ESP32)
4. Connect **SCL** (BME280) → **GPIO 22** (ESP32)

4. Connecting the Soil Moisture Sensor (Analog Output)

1. Connect **VCC** (Sensor Module) → **3V3** (ESP32)
2. Connect **GND** (Sensor Module) → **GND** (ESP32)
3. Connect **A0** (Analog Output) → **GPIO 34** (ESP32)

5. Final Checks

- Ensure all connections are **firm** and **correct** according to the diagram.
- Avoid loose wires that could cause short circuits.
- Double-check the pin assignments before powering the ESP32.

Step 3: Setting WiFi Connection and Programming Codes

1. Prepare the Workspace

- Create a single **workspace folder** to store all required files.
- Save the following four files in this folder:
    1. boot.py
    2. main.py
    3. BME280_library.py
    4. OLED_library.py

2. Configure WiFi Hotspot

- Set up a WiFi hotspot operating on the **2.4 GHz** band.
- Ensure the SSID and password match the details to be used in your code.

3. Write the Main Program

- Open **main.py** in your code editor (e.g., VSCode).
- Write or paste the required program code for the project.
- Ensure that:
    - WiFi connection credentials are correctly entered.
    - Necessary libraries (BME280_library.py and OLED_library.py) are properly imported.
- Save the file after editing.
-

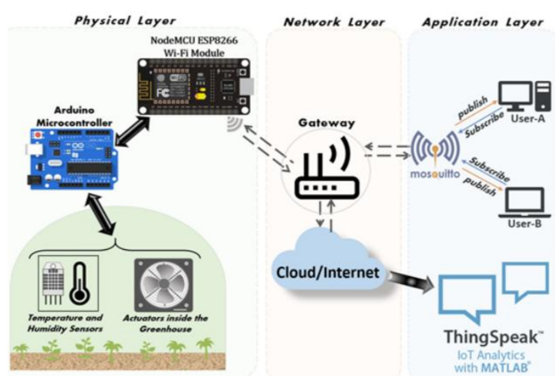


Figure 3 : Illustration of Project Process

Safety precaution

1. Preparation

1. Gather all required components according to the circuit diagram.
2. Ensure you have a suitable power source (battery, adapter, or bench power supply).
3. Confirm that voltage and current levels are safe for handling.
4. Keep the workbench organized — messy wires can cause accidental shorts.

2. Component Handling

1. Handle components carefully — resistors, capacitors, and ICs can be damaged by heat or static.
2. Discharge capacitors before touching them, as they can store dangerous charges even when the power is off.

	3. Avoid overheating components during soldering; use proper soldering techniques. **3. Safety During Wiring** 1. Always switch off power before making or changing connections. 2. Use insulated tools (screwdrivers, pliers) when handling live circuits. 3. Double-check wiring for correct polarity and connections before turning on power. 4. Never touch live wires or exposed conductive parts when power is on
Table of Data	1. Collect the data for temperature, humidity, pressure and the soil moisture by testing the circuit in the real environment. 2. Observe the data in ThingSpeak

Data/ hours	**Temperature (°C)**	**Humidity (%)**	**Pressure (Pa)**	**Soil moisture (%)**
8am				
10am				
12pm				

Example of the data compilation

Discussion	1. State the process involved. 2. Discuss the graph of the parameters 3. Describe the; a) Benefits: • Increased Efficiency: Reduces manual labor and optimizes resource usage. • Improved Outcomes: Ensures ideal growing conditions, leading to healthier plants and higher yields. • Educational Value: Provides a practical, real-world application of scientific and engineering principles. b) Potential Enhancements: • Adding more sensors: Integrate light sensors, pH sensors, or even cameras for pest detection. • Introducing AI/ML: Use machine learning algorithms to predict plant needs and automate systems more intelligently, going beyond simple threshold-based rules. • Expanding the ecosystem: Connect the greenhouse to a local weather station or a smart home system for a more integrated solution. *general framework;* ***guiding questions*** *that work across subjects* *Prepare 2 versions:* *Version 1 – for teacher/instructor: questions and answers* *Version 2 – for student: questions only*

Conclusion	Based on the analysis of the greenhouse IoT project, a comprehensive conclusion can be drawn that addresses its practical application, educational value, and future potential. The student could conclude the impact of the parameters toward the plant growth. Relate the results obtained with the objective of the experiment
Reference	Batra, N. (2024). *IoT Fundamentals with a practical approach*. CRC Press.
Vocabulary	1.Internet of Things 2.Agriculture 3.Cloud 4.Sensors 5.Sustainable 6.Python

REFERENCES

Haider, M. A., & Alberd, J. (2024). Design a Learning Model to Integrate IoT Applications into the Engineering Curriculum.

Krasniewski, A. (2025). Integrating Project-Based Learning into innovative studies in IoT Engineering.

Yen, S. T., & Chen, J. C. (2025). STEM Education in Natural Science Teaching to Secondary School Students: Case Study of Making a pH Measuring Pen in Soil Application of IoT Technology.

Emon, Kaiser & Kai, Elson & Md Saad, Mohamad Hanif & Hussain, Aini & Ayob, Afida & Ahmad, Ammar. (2021). Internet of Things (IoT) Enhanced Educational Toolkit for Teaching & Learning of Science, Technology, Engineering and Mathematics (STEM). 194-199. 10.1109/ICSET53708.2021.9612579.

SUMMARY

STEM Module:
Internet of Things for A Sustainable Future

The STEM Module: Internet of Things For A Sustainable Future is an initiative by the UniKL BMI STEM Committee that uses IoT technology as an innovative learning approach to enhance student engagement in Science, Technology, Engineering, and Mathematics (STEM). This program aligns with national priorities such as advancing TVET and adopting Industry 4.0 (IR 4.0), while also supporting the global Sustainable Development Goals (SDG). By focusing on IoT's transformative potential to address crises like climate change and resource depletion, the module prepares students with crucial skills, directly impacting each STEM element: applying Science to environmental problems, building expertise in Technology like sensor networks and cloud platforms, using Engineering for designing sustainable systems, and enhancing Mathematics skills for data analysis and informed decision-making. The overarching goal is to inspire a new generation of innovators to use technology as the building blocks for a resilient and sustainable future.

GLOSSARY

STEM Module: Internet of Things For A Sustainable Future

Module Overview:

This module introduces the foundational concepts of the Internet of Things (IoT). It will provide a clear understanding of what IoT is, how everyday objects can be connected to the internet, and the essential components that make these devices "smart." The goal is to make the concept of IoT relatable using everyday examples like smartwatches and smart homes, followed by hands-on learning using simple hardware.

There will be five main topics as below.

1. Introduction to IoT & The "Thing"
2. Programming with Python: From Functionality input to Data Streams
3. Sensing the World
4. Connecting to the Cloud
5. Making Sense of Data and Control

Learning Objectives:

By the end of this module, students will:

1. Understand the basic concept of IoT.
2. Learn what makes a device "smart" and how it connects to the internet.
3. Explore the key components involved in IoT devices, focusing on sensors, actuators, and microcontrollers.
4. Complete a hands-on project to understand the input-output process without internet connectivity.

Next, the focus and the details for each topic explain as follows.

Focus:

Understanding the concept of IoT and delving into its fundamental components and how they interact to create intelligent, interconnected systems.

Contents:

Topic 1: Introduction to IoT & The "Thing"

1. Introduction to IoT & The "Things"

What is IoT?

The Internet of Things (IoT) refers to a network of physical objects, or "things," that are embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet. These devices can range from everyday objects like your smartwatch, refrigerator, or home lighting system.

Key Questions:

- What is an IoT device?
- How do these devices collect data from their environment?
- What makes a device "smart"?

Real-World Examples:

- **Smartwatches**: Track health data and sync with your phone to provide notifications and data insights.
- **Smart Homes**: Devices like thermostats, lights, and security cameras can be controlled remotely through mobile apps or voice assistants.
- **Fitness Trackers**: Devices that monitor your physical activity and send data to your smartphone for analysis.

2. Key Concepts

How IoT Devices Work:

1. **Sensors**: Devices that collect data from the environment (e.g., temperature, motion, light).
2. **Connectivity**: The data from sensors is transmitted over a network (Wi-Fi, Bluetooth, etc.) to a central system.
3. **Processing**: The data is analyzed or processed either on the device or via a cloud service.
4. **Action**: Based on the processed data, the IoT device takes action (e.g., turning on a light, sending an alert).

What Makes a Device "Smart"?

- A device is considered "smart" when it can interact with its environment by receiving and sending data and making autonomous decisions based on that data. For example, a smart thermostat learns your temperature preferences and adjusts itself accordingly.

3. Hands-On Activity: LED Control with a Button

Objective:
Students will build a simple circuit using an Arduino Uno (or NodeMCU) to turn an LED on or off by pressing a button. This project demonstrates the basic principles of input and output (I/O) operations.

Materials Needed:

- Arduino Uno or NodeMCU (ESP32)
- Breadboard
- LED light
- Push button
- Jumper wires
- Resistor (220 ohms)

Instructions:

1. **Set Up the Circuit**:

- Connect the **LED** to one of the digital pins on the Arduino (e.g., Pin 13).
- Connect the **push button** to another digital pin (e.g., Pin 2).
- Use a resistor to protect the LED from excess current.
- Connect the ground (GND) to the negative rail on the breadboard.

2. **Write the Code**:

- Start with a simple Arduino script to detect when the button is pressed and turn the LED on or off.
- Use the digitalRead() function to check the button state and digitalWrite() to control the LED.

Sample Code:

```
int ledPin = 13;

int buttonPin = 2;

int buttonState = 0;


void setup() {

  pinMode(ledPin, OUTPUT);

  pinMode(buttonPin, INPUT);

}


void loop() {

  buttonState = digitalRead(buttonPin);


  if (buttonState == HIGH) {

    digitalWrite(ledPin, HIGH);

  } else {

    digitalWrite(ledPin, LOW);

  }

}
```

3. **Test the Project**:

- Upload the code to the Arduino board.
- Press the button to see the LED turn on and off based on the button press.

Learning Outcomes from the Activity:

- Students will understand how inputs (button presses) can trigger outputs (LED actions).
- They will experience basic programming for IoT devices and see how physical objects interact with simple commands.

4. Discussion & Wrap-up

- **Reflection**:
  Discuss how this basic concept of input (button) and output (LED) can be applied to more complex IoT devices that interact with multiple sensors and devices over the internet.
- **Extension**:
  Encourage students to think about how this project can be extended, such as adding more LEDs or creating a similar setup with sensors like motion or temperature.

Assessment:

- **Practical Evaluation**: Students will be assessed on their ability to build the circuit and implement the code correctly.
- **Quiz**: A short quiz on the concepts of IoT, including the components of IoT systems (sensors, actuators, connectivity, etc.).

End of Module

This module lays the foundation for understanding IoT devices and introduces the basic principles of connecting physical objects to the internet. It provides both theoretical knowledge and practical skills necessary for building IoT projects in the future.

Practical Evaluation Questions:

1. **Circuit Building:**

   - Describe the steps to connect a push button and an LED to an Arduino Uno board. Include the components you would use and how they are connected.

2. **Code Implementation:**

   - Write a simple Arduino code to turn on an LED when a push button is pressed. Explain how the code works, focusing on the digitalRead() and digitalWrite() functions.

3. **Testing and Debugging:**
   - After setting up your circuit and uploading the code, you notice that the LED does not turn on when the button is pressed. What steps would you take to troubleshoot and fix this issue?
4. **Extension Task:**
   - Modify the circuit and code so that the LED blinks on and off each time the button is pressed (toggle effect). Explain how you would approach this task.

Quiz Questions:

1. **Multiple Choice:**
   - What is the main purpose of a sensor in an IoT device?
     - A) To send data over the internet
     - B) To collect data from the environment
     - C) To process data and make decisions
     - D) To control other devices in the system
     -

       Answer: B) To collect data from the environment

2. **Fill in the Blanks:**
   - An IoT device is made up of several key components. These include sensors, actuators, and ____________ (Answer: connectivity).
   - The ____________ (Answer: actuator) is the part of an IoT system that performs actions based on processed data.

3. **True or False:**
   - An actuator is responsible for collecting data from the environment. **Answer: False** (Actuators perform actions, not collect data.)

4. **Short Answer:**
   - How do IoT devices connect to each other and share data? **Answer: IoT devices connect through networks like Wi-Fi, Bluetooth, or cellular data to transmit data to other devices or cloud services.**

5. **Multiple Choice:**

- Which of the following is NOT an example of an IoT device?
  - A) Smart thermostat
  - B) Refrigerator with internet access
  - C) Microwave oven without internet capabilities
  - D) Fitness tracker

Answer: C) Microwave oven without internet capabilities

6. **Short Answer:**

- What makes a device "smart" in the context of IoT? **Answer: A device is considered "smart" when it can collect data, process it, and take actions or communicate with other devices through the internet.**

7. **Matching:**

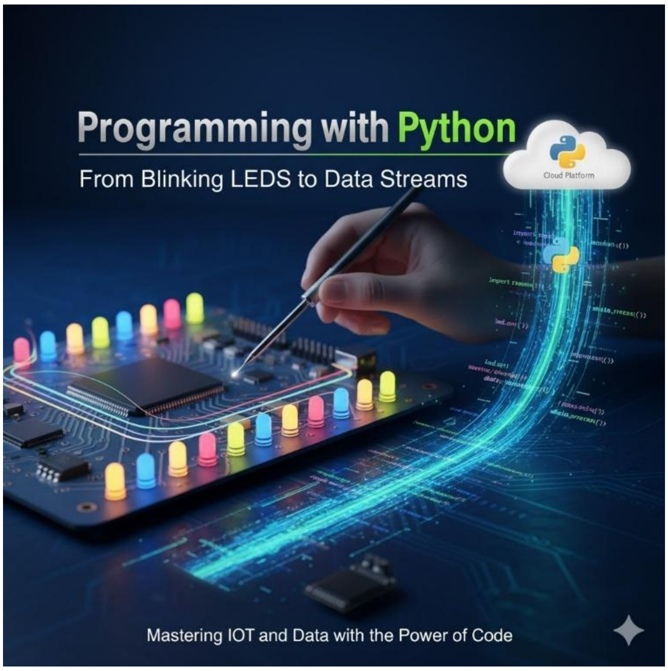
- Match the following components with their descriptions:

1. **Sensor** - A) Receives data from other IoT devices
2. **Actuator** - B) Collects data from the environment
3. **Connectivity** - C) Takes action based on data received

Answer:

- B) Collects data from the environment
- C) Takes action based on data received
- A) Receives data from other IoT devices

Topic 2: Programming with Python: From Blinking LEDs to Data Streams



Module Overview:

This module introduces Python as a powerful and user-friendly language for programming IoT devices. It focuses on the syntax, logic, and libraries necessary to control hardware and handle data.

Learning Objectives:

By the end of this module, students will:

1. Understand the Basics of Python Programming
2. Familiarize with MicroPython or CircuitPython Environments
3. Develop Hands-on Skills in Sensor Integration and Data Processing
4. Apply IoT Connectivity Using Python for Real-World Applications

1. Introduction to Phyton"

What Phyton?

Python is a high-level, versatile, and easy-to-learn programming language known for its readability and simplicity. It is widely used for a variety of applications, including web development, data analysis, machine learning, and IoT programming. Python's syntax is designed to be straightforward, which makes it a great choice for both beginners and experienced developers.

Key Questions:

- Why is Python considered a beginner-friendly language?
- How does Python compare to other programming languages like Java or C++ in terms of ease of use and versatility?
- What are the advantages of using Python for programming IoT devices?
- How do libraries in Python help simplify complex tasks like sensor integration or data handling?
- What are the differences between Python, MicroPython, and CircuitPython, and when should you use each?

Real-World Examples:

- **Web Development**: Python is used in frameworks like Django and Flask to build dynamic websites.
- **Data Science and AI**: Python's libraries such as Pandas, NumPy, and TensorFlow are widely used for data manipulation and machine learning.
- **IoT Programming**: Python, especially MicroPython, is used to program microcontrollers like ESP32 to control sensors, process data, and interact with cloud platforms.
- **Automation**: Python is commonly used for automating repetitive tasks, like file management or data entry, in both small and large-scale applications.

2. Key Concepts

How Python Works:

Python operates through an **interpreter** that executes code line by line, which makes it easier for developers to test and debug their programs. Here's how it works:

- **Interpreter:** Python code is processed by an interpreter, meaning the source code is converted into machine code at runtime, allowing for immediate feedback during execution.
- **Dynamic Typing:** Python doesn't require variable types to be declared. It infers the type of a variable at runtime, which adds flexibility in coding.
- **Garbage Collection:** Python has an automatic memory management system, which helps developers by automatically handling memory allocation and deallocation, reducing memory leaks.

Python is popular in the programming world because it is:

- **Readable:** Python's syntax is simple, which makes the code easy to understand and maintain.

- **Portable:** It can run on various operating systems (Windows, MacOS, Linux) without requiring changes to the code.
- **Extensive Libraries:** Python has a wide range of libraries and frameworks that allow for quick development of software and applications.

What Makes Python "Smart" in Applications?

Python is considered "smart" in applications for the following reasons:

- **Versatility in Tasks:** Python is highly adaptable. From web development to artificial intelligence, Python can handle various tasks efficiently, making it suitable for many types of projects.
- **Powerful Libraries and Frameworks:** Python's ecosystem is rich in libraries and frameworks (like TensorFlow for AI, Flask/Django for web development, and Pandas for data analysis), which make implementing advanced features straightforward without reinventing the wheel.
- **Support for Automation:** Python excels in automation tasks such as data processing, file handling, and system administration. It's used to create intelligent systems that can automate complex tasks, like monitoring sensor data or managing databases.
- **Integration with IoT Devices:** Python, especially MicroPython and CircuitPython, allows easy communication between devices like microcontrollers (ESP32, Raspberry Pi) and various sensors, making it an excellent choice for building smart applications that interact with the physical world.
- **Machine Learning and AI:** Python is the go-to language for AI and machine learning due to powerful libraries like scikit-learn, Keras, and PyTorch, which allow for easy creation of smart systems that can learn and improve over time.

3. Hands-On Activity: Smart Plant Monitor for Greenhouse

Objective:
The primary objective is to gain practical experience integrating hardware, programming, and cloud connectivity by developing a **real-time environmental monitoring system** using MicroPython on an ESP32.

Materials Needed:

1. ESP32 Development Board (must support MicroPython)
2. Environmental Sensor BME280 Sensor (for Temperature, Humidity, Pressure)
3. Soil Sensor
4. Analog Soil Moisture Sensor
5. SSD1306 OLED Display
6. Micro-USB Cable,
7. Breadboard,
8. Jumper Wires

Instruction (Step-by-Step Coding & Setup):

Step 1: ThingSpeak Cloud Setup

1. **Create Channel:** Sign up for **ThingSpeak** and create a new channel.
2. **Define Fields:** Name at least **four fields** for the data you'll send: Temperature, Humidity, Pressure, and SoilMoisture.
3. **Get API Key:** Navigate to the **API Keys** tab and copy your **Write API Key** and the channel's **Write URI** (the full HTTP URL for posting data). You will plug these into your Python code.

Step 2: Hardware Wiring

1. **BME280 & OLED (I2C):** Connect the BME280 and the OLED display to the ESP32's **I2C pins** (typically SDA and SCL). The code provided uses SCL_Pin = Pin(22, Pin.OUT) and SDA_Pin = Pin(21, Pin.OUT).
2. **Soil Moisture Sensor (ADC):** Connect the analog output (A0/AO) of the Soil Moisture Sensor to an **ADC pin** on the ESP32. The code uses SoilMoisture_pin = ADC(Pin(34)).

Step 3: MicroPython Code Implementation

1. **Upload Libraries:** Ensure the required libraries (bme280_lib, oled_lib, urequests) are installed or uploaded onto the ESP32 filesystem.
2. **Adapt Code:**

- **Update Pins:** Verify and adjust the pin definitions in the code to match your actual wiring.
- **Update ThingSpeak Details:** Paste your ThingSpeak Channel ID and Write URI into the appropriate variables in the code.

3. **Main Loop Logic:** The code should execute the following loop every 15 seconds:

- Read suhu (temperature), kelembapan (humidity), and tekanan (pressure) from the BME280.
- Read the raw moisture_value from the ADC and convert it to a moisture_percentage.
- Print all four readings to the local terminal using print().
- Update the **OLED display** (skrin.text(), skrin.show()) with the current readings.
- Construct the **JSON payload** (data_from_sensors) mapping the values to your ThingSpeak field names.
- Send the payload using urequests.request('POST', WRITE_URI, json=data_from_sensors, headers=http_header).
- Pause for 15 seconds (sleep(15)) to comply with ThingSpeak's rate limit.

Test the Project:

Test	Action	Expected Result
Local Terminal & Display Check	Run the MicroPython code on the ESP32 board.	1. The OLED Display should immediately show the sensor values. 2. The IDE Terminal/Shell should continuously print the readings (Temperature, Humidity, Pressure, Moisture %) every 15 seconds.
Sensor Data Integrity Check	Briefly warm the BME280 sensor (e.g., hold it) or move the soil sensor from dry air into water.	The printed values in the terminal and on the OLED display must reflect the change in real-time. This verifies the sensor reading and data processing logic.
Cloud Transmission Check	Wait for at least two cycles (30 seconds) of data transmission.	Open your ThingSpeak Channel's Private View in a web browser. The charts for all four fields should begin updating with the new data points, confirming successful Wi-Fi connection, correct JSON formatting, and API submission.

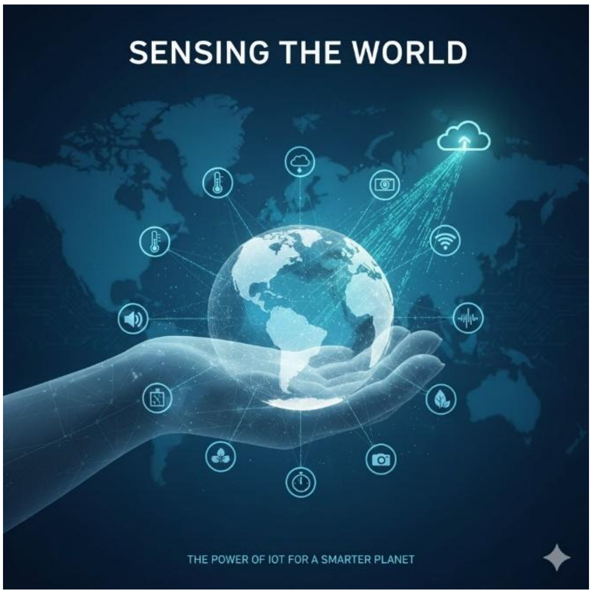
Learning Outcomes from the Activity:

1. **Interface Sensors:** Successfully read and process data from I2C-based digital sensors (BME280) and analog sensors (Soil Moisture Sensor).
2. **Control Output:** Display real-time sensor data on a local output device (OLED screen) and the terminal.
3. **Implement Connectivity:** Configure the ESP32 to connect to a Wi-Fi network and transmit structured data (JSON payload) to a public IoT cloud platform (ThingSpeak) via HTTP POST requests.

4. Discussion & Wrap-up

- **Reflection**:
  Discuss how this basic concept of input (button) and output (LED) can be applied to more complex IoT devices that interact with multiple sensors and devices over the internet.
- **Extension**:
  Encourage students to think about how this project can be extended, such as adding more LEDs or creating a similar setup with sensors like motion or temperature.

Topic 3: Sensing the World



Module Overview:

This module introduces students to the integration of sensors, microcontrollers, and actuators in developing smart systems that support sustainability, particularly under SDG 11: Sustainable Cities and Communities. Through the theme "Sensing the World," students explore how electronic components interact to collect, process, and respond to real-world environmental data. Using hands-on learning activities, learners apply scientific, technological, and engineering principles to address issues like efficient water usage and environmental monitoring. The module emphasizes critical thinking, problem-solving, and data analysis, bridging classroom theory with practical, industry-relevant IoT applications that prepare students for the future of smart technology and sustainable innovation.

Learning Objectives:

By the end of this module, students will:

1. Identify and explain the function of sensors, actuators and microcontrollers.
2. Integrate sensors and actuators into simple applications.
3. Demonstrate correct wiring and pin configurations.
4. Design, build and test a functional Smart System
5. Present and communicate project outcome effectively.

1. "Sensing the World"

What is Sensing and Actuation Systems?

This module focuses on understanding how sensors, microcontrollers, and actuators work together to create smart, automated systems that interact with their environment. Sensors detect physical changes (such as light, motion, or temperature), microcontrollers process the data, and actuators perform actions in response—like turning on lights, sounding alarms, or moving mechanical parts. This forms the foundation of many real-world automation and IoT applications, enabling machines and systems to operate intelligently with minimal human intervention.

Key Questions:

- How do sensors, microcontrollers, and actuators communicate within an automated system?
- What happens to data once it is collected by a sensor?
- How do these systems contribute to efficiency, safety, and sustainability?

Real-World Examples:

- **Smart Traffic Systems:** Use motion and infrared sensors to detect vehicle flow and adjust traffic lights to reduce congestion.
- **Industrial Automation:** Employ sensors and robotic arms controlled by microcontrollers to streamline manufacturing and ensure precision.
- **Smart Buildings:** Integrate temperature, motion, and light sensors to automatically control air conditioning, lighting, and energy usage.
- **Healthcare Monitoring:** Use wearable sensors to track patient vital signs and send real-time data to healthcare systems.
- **Environmental Monitoring:** Deploy IoT stations to measure air quality, noise levels, or water pollution for urban planning.
- **Smart Parking Systems:** Utilize ultrasonic sensors to detect available parking spaces and guide drivers through mobile apps.
- **Home Security Systems:** Combine motion sensors, cameras, and alarms for automated safety and surveillance.
- **Agriculture Automation:** Use soil and humidity sensors with motorized irrigation systems to optimize water usage and crop growth.

2. Key Concepts

How the Devices Work:

The system operates through the interaction of three main components:

1. **Sensors** detect physical changes: temperature, light, sound, motion, or humidity and convert them into signals that a microcontroller can process. For example, a temperature sensor measures heat and outputs a signal representing the temperature level. Other common sensors include light-dependent resistors (LDR) for brightness, PIR sensors for motion, and soil moisture sensors for detecting water content in soil.

2. **Microcontrollers** (such as Arduino, ESP32 or Raspberry Pi Pico) act as the "brain" of the system, processing signals from the sensors and making decisions based on input data. For instance, if a soil moisture sensor detects that the soil is dry, the microcontroller can decide to turn on an LED or trigger an alert.

3. **Actuators** perform real actions based on the microcontroller's commands. These include LEDs (light indication), buzzers (sound alerts), motors (movement), and relays (switching high-power devices).

What Makes a Device "Smart"?

A device becomes smart when it can sense, process, and respond automatically without direct human control. This intelligence comes from the integration of sensors that gather data, microcontrollers that analyze and decide based on that data, and actuators that perform corresponding actions. In other words, a smart device combines real-time data acquisition, automated decision-making, and responsive actions, making it capable of adapting to changing conditions — for example, turning on an alert when the soil is dry or adjusting temperature settings in a controlled environment.

3. Hands-On Activity: Smart Plant Monitor

1. **Objective:**
   To design and build a Smart Plant Monitoring System using sensors, a microcontroller (ESP32), and actuators.
2. To understand the interaction between input (sensors), process (microcontroller), and output (actuators) in an IoT-based automation system.
3. To collect and analyze environmental data such as temperature, humidity, pressure, and soil moisture, relating them to plant growth and sustainability.

Materials Needed:

Core Components:

- ESP32 microcontroller
- Breadboard
- Jumper wires
- USB cable

Sensors:

- DHT11 or BME280 (temperature, humidity, and pressure sensor)
- Soil moisture sensor
- Light sensor (optional)

Actuators:

- OLED or LCD display
- LED or buzzer
- Small DC motor (optional for automation)

Power Source:

- 5V adapter / power bank / battery

Supporting Tools:

- Resistors, potentiometer, multimeter, and laptop

Additional Materials:

- Plant pot and soil
- Plastic case or cardboard for enclosure

Instructions:

Set Up the Circuit:

1. Connect the OLED Display (I2C)

- VIN → 3V3 (ESP32)
- GND → GND (ESP32)
- SDA → GPIO 21 (ESP32)
- SCL → GPIO 22 (ESP32)

2. Connect the BME280 Sensor (I2C)

- VIN → 3V3 (ESP32)
- GND → GND (ESP32)

- SDA → GPIO 21 (ESP32)
- SCL → GPIO 22 (ESP32)

3. Connect the Soil Moisture Sensor (Analog Output)

- VCC → 3V3 (ESP32)
- GND → GND (ESP32)
- A0 → GPIO 34 (ESP32)

4. Check all wiring connections before powering up the ESP32.

Write the Code:

Use VSCode or the Arduino IDE to upload the program.

Students may use the provided sample code or modify it for additional functionality.

The code should:

- Read sensor data (temperature, humidity, pressure, and soil moisture).
- Display values on the OLED/LCD screen.
- Trigger an LED or buzzer alert if the soil moisture is below a set threshold.

Sample Code:

```
// Smart Plant Monitor using ESP32 + BME280 + Soil Sensor + OLED
#include <Wire.h>
#include <Adafruit_BME280.h>
#include <Adafruit_SSD1306.h>
// --- OLED Display Settings ---
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
// --- BME280 Sensor ---
Adafruit_BME280 bme;  // I2C interface
// --- Soil Sensor ---
const int soilPin = 34;  // Analog input for soil moisture sensor
int soilValue;
// --- Actuator (LED or Buzzer) ---
const int alertPin = 25;  // GPIO for LED/Buzzer
```

```
// --- Moisture Threshold ---
int dryThreshold = 1500;  // Adjust based on your sensor calibration
void setup() {
  Serial.begin(115200);
  pinMode(alertPin, OUTPUT);
  digitalWrite(alertPin, LOW);
  // Initialize OLED
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED not found!");
    while (1);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);

  // Initialize BME280
  if (!bme.begin(0x76)) {
    Serial.println("BME280 not found!");
    while (1);
  }

  displayMessage("Smart Plant Monitor", "Initializing...");
  delay(2000);
}

void loop() {
  // --- Read BME280 Data ---
  float temperature = bme.readTemperature();
  float humidity = bme.readHumidity();
  float pressure = bme.readPressure() / 100.0F;  // Convert Pa to hPa

  // --- Read Soil Moisture ---
```

```
  soilValue = analogRead(soilPin);

  // --- Display Data on OLED ---
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println("Smart Plant Monitor");
  display.println("--------------------");
  display.print("Temp: "); display.print(temperature); display.println(" C");
  display.print("Hum : "); display.print(humidity); display.println(" %");
  display.print("Pres: "); display.print(pressure); display.println(" hPa");
  display.print("Soil: "); display.print(soilValue); display.println(" ADC");
  display.display();

  // --- Control LED/Buzzer based on Soil Moisture ---
  if (soilValue > dryThreshold) {
    digitalWrite(alertPin, HIGH);  // Soil is dry → alert ON
  } else {
    digitalWrite(alertPin, LOW);   // Soil is wet → alert OFF
  }

  delay(2000);
}

void displayMessage(String line1, String line2) {
  display.clearDisplay();
  display.setCursor(0, 10);
  display.println(line1);
  display.setCursor(0, 30);
  display.println(line2);
  display.display();
}
```

Test the Project:

1. Power the circuit using the USB cable or external supply.
2. Observe the readings displayed on the OLED screen.
3. Test system response by changing soil moisture levels (e.g., dry vs. wet soil).
4. Record the measured data for analysis in the provided data table.

Learning Outcomes from the Activity:

By completing this activity, students will be able to:

1. Identify and connect sensors and actuators correctly to a microcontroller.
2. Write and upload a simple IoT program for environmental data monitoring.
3. Analyze sensor data to understand its impact on environmental and plant conditions.
4. Demonstrate critical thinking, problem-solving, and communication skills through project presentation.
5. Relate classroom learning to real-world applications in smart agriculture and sustainable technology.

4. Discussion & Wrap-up

Reflection:

- Discuss how the Smart Plant Monitor system uses sensors to collect real-time environmental data and how the microcontroller processes that data to make decisions.
- Compare how similar sensing and automation concepts are applied in industry, such as in smart homes, automated irrigation, or climate control systems.
- Talk about the importance of accurate data collection and how sensor calibration or environmental factors might affect readings.
- Reflect on the role of technology in sustainability, especially how IoT devices can help reduce waste, conserve resources, and improve efficiency.

Extension:

- Extend the project by adding more sensors such as light, pH, or rain sensors to create a complete smart greenhouse system.
- Enhance the system with wireless connectivity (WiFi/Blynk/Google Sheets) to upload sensor data to the cloud for real-time monitoring. (IoT element)

Assessment:

- **Practical Evaluation**: Students will be assessed on their ability to correctly assemble the circuit, connect sensors and actuators to the microcontroller, and upload a functional IoT program. Evaluation will focus on accuracy, functionality, and adherence to safety practices.
- **Presentation**: Students will deliver a brief presentation explaining their project, including the purpose, design, and environmental impact. This will be followed by a short quiz assessing their understanding of IoT concepts—such as the role of sensors, actuators, connectivity, and data processing in smart agriculture systems. Communication skills and the ability to relate their work to real-world applications will also be evaluated.

End of Module

This module serves as an essential introduction to the world of microcontrollers. Through hands-on activities and guided learning, students have gained foundational skills in connecting sensors and actuators, programming basic functions, and interpreting environmental data. The experience not only builds technical competence but also encourages critical thinking and real-world problem-solving, key components in developing smart, sustainable technologies. This foundation will support further exploration and innovation in future embedded systems projects.

Practical Evaluation Questions:

Component Identification:

- Can you identify the sensors and actuators used in your circuit?
- What is the purpose of each component?

Circuit Design:

- Can you explain why you connected each sensor or actuator to specific pins on the microcontroller?
- What precautions did you take to ensure correct wiring and prevent damage?

Code Implementation:

- What is the function of your code, and how does it interact with the connected components?
- Can you walk through how sensor data is read and displayed or transmitted?

Debugging:

- If the circuit is not working as expected, how would you troubleshoot the issue?
- What tools or methods can you use to test if a sensor is working properly?

Real-World Application:

- How could this project be scaled or modified for real-life smart agriculture use?

Presentation marking scheme:

Criteria	Excellent (4)	Good (3)	Satisfactory (2)	Needs Improvement (1)
Content Accuracy	Clear, in-depth explanation of IoT concepts and project.	Mostly accurate; minor errors.	Some misunderstandings; key concepts missing.	Major errors; lacks understanding of key concepts.
Technical Understanding	Demonstrates strong grasp of system components and code.	Understands most components and logic.	Basic understanding; explanations somewhat unclear.	Lacks technical understanding.
Real-World Connection	Effectively relates project to smart agriculture or sustainability.	Some connection to real-world context.	Vague or minimal real-world relevance mentioned.	No clear connection to real-world applications.
Presentation Skills	Clear, confident, well-organized delivery.	Mostly clear and organized; minor issues.	Somewhat unclear or disorganized.	Difficult to follow; lacks clarity.
Teamwork & Communication	Excellent collaboration and shared speaking roles.	Good collaboration; some participation imbalance.	Limited collaboration; one or two dominant speakers.	Poor teamwork; lacks coordination.
Visual Aids (if applicable)	Effective use of visuals or demonstrations.	Some visuals used; support understanding.	Minimal or poorly integrated visuals.	No visuals or irrelevant ones used.

Topic 4: Connecting to the Cloud

Module Overview:

The primary focus of learning about the cloud using any cloud platform such as ThingSpeak is on Internet of Things (IoT) data analytics and application development.

Learning Objectives:

1. Understand Cloud Data Management and Storage
2. Develop the Real-Time Data Visualization

Content:

1. **Data Collection (Connect Devices to the Cloud)**

This involves learning how to configure hardware and use APIs to send data streams to the cloud.

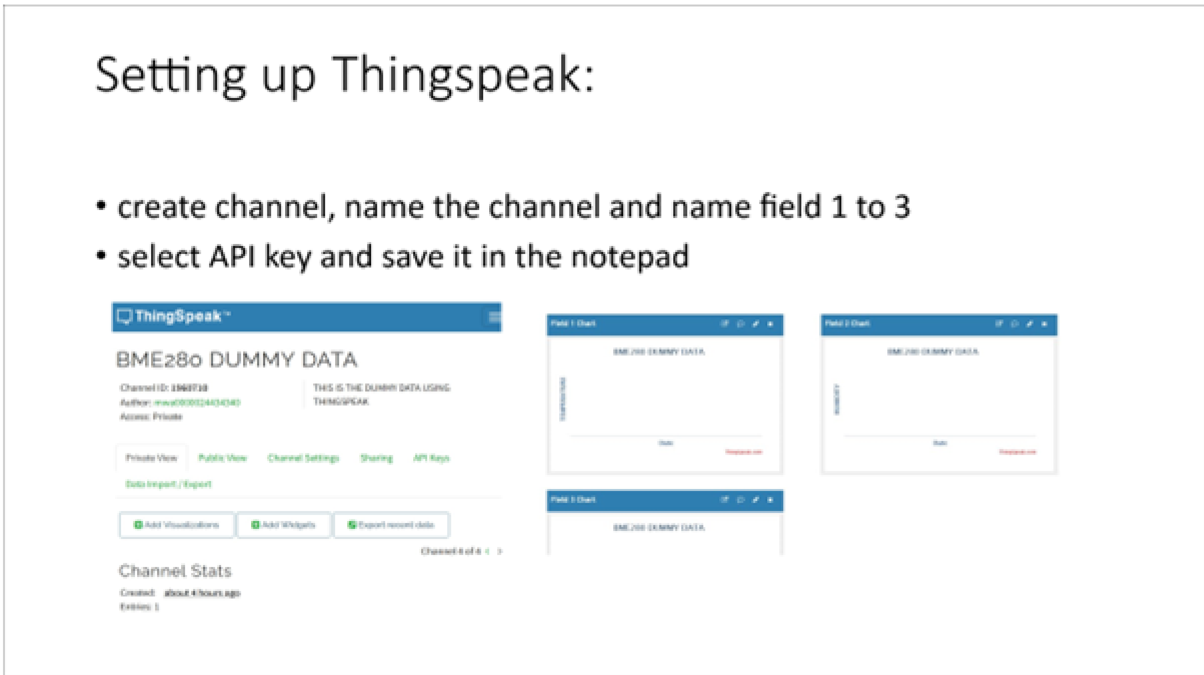
- IoT Device Connectivity: Learning to connect various hardware, such as Arduino, Raspberry Pi, ESP8266, and ESP32 devices, to the cloud.
- Data Channel Configuration: Understanding how to create and manage ThingSpeak Channels, which act as data storage repositories in the cloud (each channel stores up to eight data fields).
- IoT Communication Protocols: Using REST and MQTT APIs to write (publish) data from your devices to the ThingSpeak channels.
- Security: Learning about API keys for securing your private data channels.

2. **Data Analysis and Visualization (Cloud Analytics)**

This section focuses on processing the collected data within the cloud environment.

- Real-time Visualization: Generating instant charts and plots of live data streams directly in the cloud.
- Build custom visualizations beyond the default charts.

3. **Practical session**



Topic 5: Making Sense of Data and Control

Module overview:

The main objectives of learning IoT using cloud platforms are to enable students or developers to master the skills needed to design, implement, and manage a complete, real-world IoT solution. The students need to implement their project into the real scenario such as in the greenhouse.

Learning Objectives:

1. Develop Device-to-Cloud Integration in real scenario.
2. Implement the prototype as a hands on exercise.

Content:

1. **Data Collection (Connect Devices to the Cloud)**

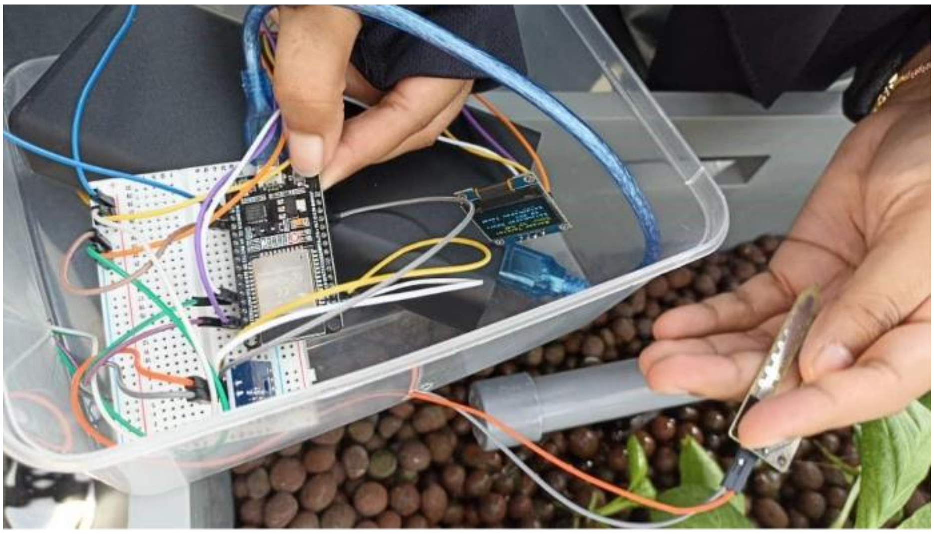
To understand and implement the secure transmission of data from various hardware devices (sensors, microcontrollers) to a cloud platform using standard IoT protocols (MQTT, HTTP/REST).

2. **Implement the prototype as a hands-on exercise.**

- To physically connect and configure hardware (e.g., Arduino, ESP32) and verify that data is successfully transmitted, stored, and displayed in the cloud environment.

- To test the end-to-end functionality of the IoT system, ensuring that data processing, visualization, and any defined actions (like sending alerts) work reliably.

Implementation of soil moisture sensors for greenhouse project.



Connecting the ESP32 and sensors for Greenhouse

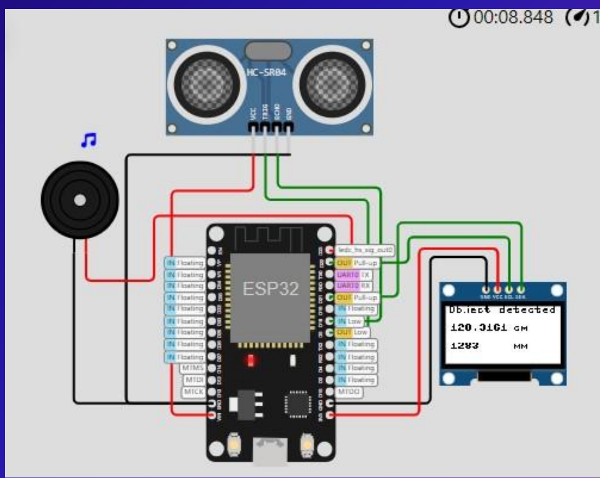
a) Exercise for Wokwi

EXERCISE 1: CAR REVERSE SENSOR SYSTEM

- Create a reverse car system that integrate between ESP32 and ultrasonic sensor, OLED and buzzer.
- Components: ESP32, Ultrasonic sensor, OLED and buzzer.
- #STEP1: IMPORT LIBRARY-ULTRASONIC, OLED,PWM,TIME
- #STEP2: DECLARE CONNECTION PIN USING OOP
    - #STEP2.1 FOR ULTRASONIC SENSOR
    - #STEP2.2 FOR OLED USING OOP
    - #STEP2.3 FOR BUZZER USING OOP
- #STEP3 : THE PROCESS- use while loop
    - #for ultrasonic sensor on terminal
    - #to display the distance on OLED
    - # check the distance/range for <50

ANSWER:

https://wokwi.com/projects/363572947733572609



b) Activities for students

ASSESSMENTS/ ACTIVITIES FOR THE STUDENTS:

NAME:
STUDENT ID:
CLASS:

WORD SEARCH PUZZLE:

I	N	T	E	R	N	E	T	O	F
T	H	I	N	G	S	S	E	N	S
O	R	S	D	A	T	A	C	O	L
L	E	C	T	I	O	N	B	L	U
E	T	O	O	T	H	W	I	F	I
C	O	N	N	E	C	T	E	D	D
T	E	C	H	N	O	L	O	G	Y
H	A	C	K	I	N	G	P	R	I
V	A	C	Y	S	M	A	R	T	C
I	T	Y	N	E	T	W	O	R	K

c) Exercise for Python programming.

LETS DO GROUP EXERCISES!

1. DECIDING THE BOOK THICKNESS

```
print (“\nThis program will decide your book thickness\n”)
# obtain input from the user:
pages = int(input(“Please enter your book pages: ”)
if pages < 20:
        print (“This book is thin.”)
elif pages > 99:
        print (“This book is thick.”)
else:
        print (“This book is perfect for your reading.”)
```

2. GUESSING A NUMBER

```
print (This program will ask you to guess a correct number)
target = 66
# obtain input from the user:
value = int(input("Enter an integer between 1 and 100: "))
if value > target:
        print("WRONG!", value, "is high/too high")
elif int(value) < target:
        print("WRONG!", value, "is low/too low")
else:
        print("Perfect! Kudos!")
```

d) Example of code for Green House Project

```
Program code for Real-Time Environmental Monitoring for Greenhouse
This program was taken from various of resources and being editted for
learning purposes.
This program code is written, edited and credited to:
  Ts. Dr. Siti Marwangi Mohamad Maharum
  UniKL British Malaysian Institute

'''

#Import all related modules and libraries
from machine import Pin, SoftI2C, ADC
from utime import sleep
import bme280_lib
import oled_lib
import urequests as requests

#Pin declaration for all connected sensors/modules
SCL_Pin = Pin(22, Pin.OUT)
SDA_Pin = Pin(21, Pin.OUT)
I2C_Pin = SoftI2C(scl=SCL_Pin, sda=SDA_Pin)
SoilMoisture_pin = ADC(Pin(34))  # Change GPIO number if needed
SoilMoisture_pin.atten(ADC.ATTN_11DB)  # Set attenuation for full range (0-
3.6V)

#Object (i.e. Sensor/Module with library) Declaration
skrin = oled_lib.SSD1306_I2C(width = 128, height = 64, i2c = I2C_Pin)
bme_sensor = bme280_lib.BME280(i2c = I2C_Pin)

#Our main program starts here!
while True:
  # Read the input parameter value from BME280 sensor using library
  suhu = bme_sensor.temperature
  kelembapan = bme_sensor.humidity
  tekanan = bme_sensor.pressure

  # Read the analog value from the sensor
  moisture_value = SoilMoisture_pin.read()
  # Convert the raw ADC value to percentage
  moisture_percentage = 100-((moisture_value / 4095) * 100)

  # Display the readings to VS Code terminal
  print("====== BACAAN TERKINI ==========")
  print('Temperature: ', suhu)
  print('Humidity: ', kelembapan)
  print('Pressure: ', tekanan)
  print('Soil Moisture Sensor: ', str(moisture_percentage)+'%')
```

```
  # Display selected input readings on OLED screen
  skrin.fill(0)
  skrin.text("Bacaan Terkini", 10, 0, 1)
  skrin.text("Suhu: "+suhu, 3, 10, 1)
  skrin.text("Kelembapan Udara:  ", 3, 20, 1)
  skrin.text(kelembapan, 15, 30, 1)
  skrin.text("Kelembapan Tanah: ", 3, 40, 1)
  skrin.text(str(moisture_percentage)+"%", 15, 50, 1)
  skrin.show()

  # Step 1 for data submission to cloud (ThingSpeak)
  # Arrange data from sensor/sensors in JSON format
  # JSON format --> {'name':'value'}
  data_from_sensors = {
                      "channel_id": COPY_YOUR_CHANNELID_FROM_TSPEAK,
                      "field1": suhu,
                      "field2": kelembapan,
                      "field3": tekanan,
                      "field4": moisture_percentage
                      }

  # Step 2. Define http header
  http_header = {'Content-Type':'application/json'}

  # Step 3. Submit them!
  data_submission =
requests.request("POST",json=data_from_sensors,headers=http_header,url='Copy
your WRITE url from your TSpeak account')
  data_submission.close()

  # System delay (ThingSpeak limit 15 sec for its dashboard update)
  sleep(15)
```

TThe STEM Module: Internet of Things For A Sustainable Future is an initiative by the UniKL BMI STEM Committee that uses IoT technology as an innovative learning approach to enhance student engagement in Science, Technology, Engineering, and Mathematics (STEM). This program aligns with national priorities such as advancing TVET and adopting Industry 4.0 (IR 4.0), while also supporting the global Sustainable Development Goals (SDG). By focusing on IoT's transformative potential to address crises like climate change and resource depletion, the module prepares students with crucial skills, directly impacting each STEM element: applying Science to environmental problems, building expertise in Technology like sensor networks and cloud platforms, using Engineering for designing sustainable systems, and enhancing Mathematics skills for data analysis and informed decision-making. The overarching goal is to inspire a new generation of innovators to use technology as the building blocks for a resilient and sustainable future.