PHASE 3: Building and Developing with IoT

The project involves deploying IoT sensors near water bodies and flood-prone areas to monitor water levels and provide early flood warnings through a public platform. The objective is to enhance flood preparedness and response by issuing timely warnings to both the public and emergency response teams. This project includes defining objectives, designing the IoT sensor network, developing the warning platform, and integrating them using IoT technology and Python.

Design Thinking:

Project Objectives: Define objectives such as real-time flood monitoring, early warning issuance, public safety, and emergency response coordination.

IoT Sensor Network Design: Plan the deployment of IoT sensors to monitor water levels in flood-prone areas.

Early Warning Platform: Design a web-based platform to display real-time water level data and issue flood warnings.

Integration Approach: Determine how IoT sensors will send data to the early warning platform.

To work with the code in Wokwi, follow these steps:

1) **Access Wokwi:**
   Go to the Wokwi website(https://wokwi.com/) and log in or create an account if needed.

2) **Create a New Python Project:**
   a) Click on "New Project."
   b) Name your project and select "Python" as the programming language.
   c) Click "Create" or "Start Project" to create the project.

3) **Edit the Python Code:**
   a) In the project editor, you will see a Python code editor.
   b) Replace any default code with the Python code provided.

PYTHON CODE:

```
from wokwi.arduino import WokwiBoard, LiquidCrystal

import random


board = WokwiBoard()

lcd = LiquidCrystal(board, rs="D7", en="D6", d4="D5", d5="D4", d6="D8", d7="D9",
cols=16, rows=2)


water_level = 0


def simulate_rainfall_sensor():

  return random.uniform(0, 5)


def simulate_river_gauge():

  return random.uniform(0, 10)


def update_display():

  lcd.clear()

  lcd.print("Water Level:")

  lcd.setCursor(0, 1)

  lcd.print(f"{water_level:.2f} cm")


board.simulation.setup()


while True:

  board.simulation.loop()
```

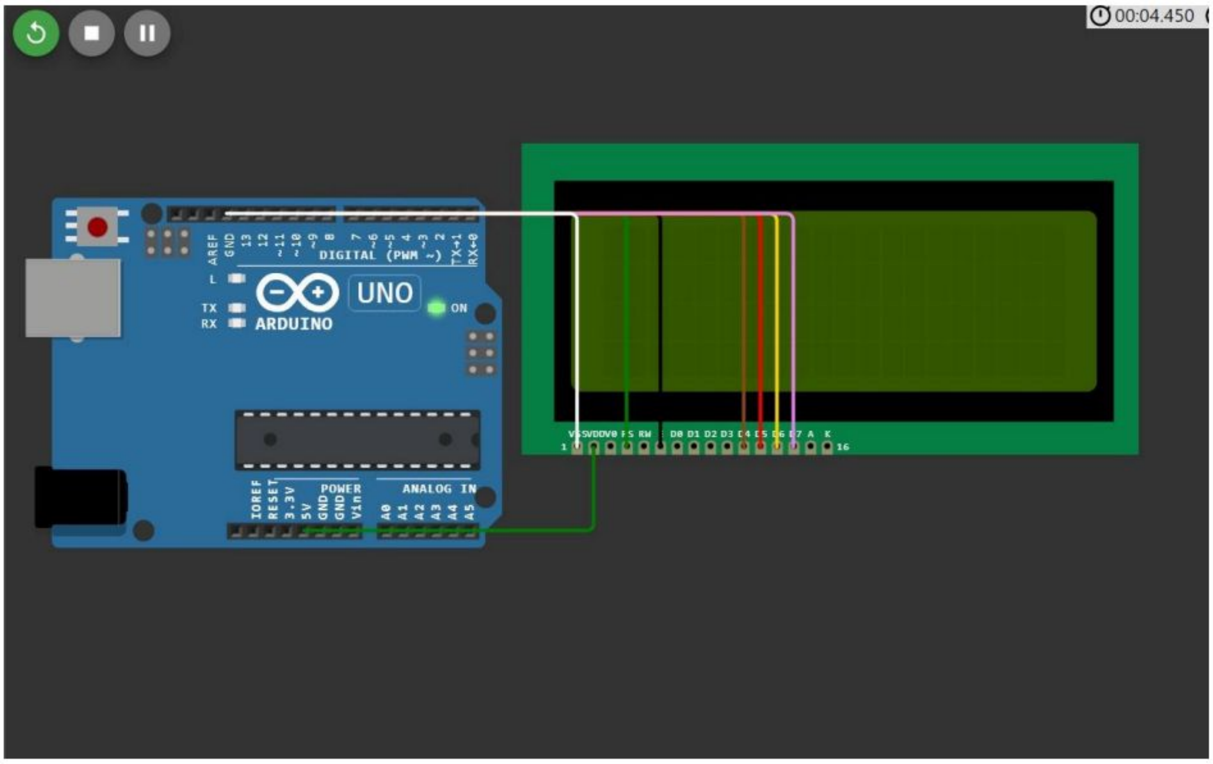
```
water_level += 0.1

if water_level > 100:

  water_level = 0


rainfall = simulate_rainfall_sensor()

river_discharge = simulate_river_gauge()


update_display()
```

4) **Add the Components:**
   a) Click on the "Components" tab to access various virtual components.
   b) Add the required components for the code, such as a virtual LiquidCrystal display.

5) **Start the Simulation:**
   a) Click on the "Simulation" tab.
   b) Click "Start Simulation" to run the code.

6) **Observe the Simulation:**
   a) The code will run within the simulation environment.
   b) You can observe the simulation output, including the virtual LiquidCrystal display.

7) **Interact and Test:**
   a) You can interact with the simulation and verify that the code behaves as expected.


00:04.450
DIGITAL (PWM ~)
UNO
ARDUINO
ON
POWER
ANALOG IN
VSS VDD V0 RS RW E D0 D1 D2 D3 D4 D5 D6 D7 A K
1
16