Immersive Technology: Project Description/ Working Manual

Name: Daniel Cyril Obon

Student ID: C2650218

1. Project Title

EnviroView: AR-Based IoT Environmental Monitoring System

2. Introduction

This project aims to demonstrate a novel way to visualize a dynamic environment monitoring system created by implementing an IoT system with Augmented Reality (AR). An ESP32 microcontroller collects real-time temperature, humidity, and water level data using a DHT11 sensor and an HC-SR04 ultrasonic sensor, which is then sent to the Blynk IoT Cloud to be accessed by the AR Unity application later. A target image defined by an imaged stored in a Vuforia (2024) database, is to be detected in AR to display the live sensor data.

3. System Components

3.1 Hardware Components

- **ESP32 Microcontroller:**
  Microcontroller with Wi-Fi and Bluetooth connectivity, used to collect and transmit sensor data to the cloud.
- **DHT11 Temperature and Humidity Sensor:**
  Measures the temperature and humidity in an area, sending the data to the ESP32
- **HC-SR04 Ultrasonic Sensor:**
  Measures the water level (in percentage scale) by using ultrasonic waves and calculating the return time.
- **Jumper Wires:**
  - **Male-to-Male Jumper Wires:** Used to connect the DHT11 sensor to the ESP32.
  - **Male-to-Female Jumper Wires:** Used to connect the HC-SR04 sensor, which has male pins to the ESP32.
- **Micro-USB Cable:**
  Used for flashing the Arduino logic program onto the ESP32 and provides it with power.
- **Smartphone (Android):**

Runs the AR application built with unity, which enables the user to scan the target image (stored in Vuforia database) and view real-time sensor data.

3.2 Software Components

- **Arduino IDE:**
  Cloud service used to store sensor data from the ESP32 and allows data retrieval through API requests (get methods).
- **Unity Engine:**
  A cross-platform development engine used to create the AR applications that displays the sensor data in an engaging manner.
- **Vuforia Engine:**
  Provides image target recognition capabilities for AR by detecting a matching physical object following images in the Vuforia database.
- **Visual Studio Code:**
  Code editor used for editing the main C# script for the Unity application.
- **Internet Connection:**
  Essential for the ESP32 to connect to Blynk servers and for the Unity app to get updated sensor data over HTTP.

4. System Architecture

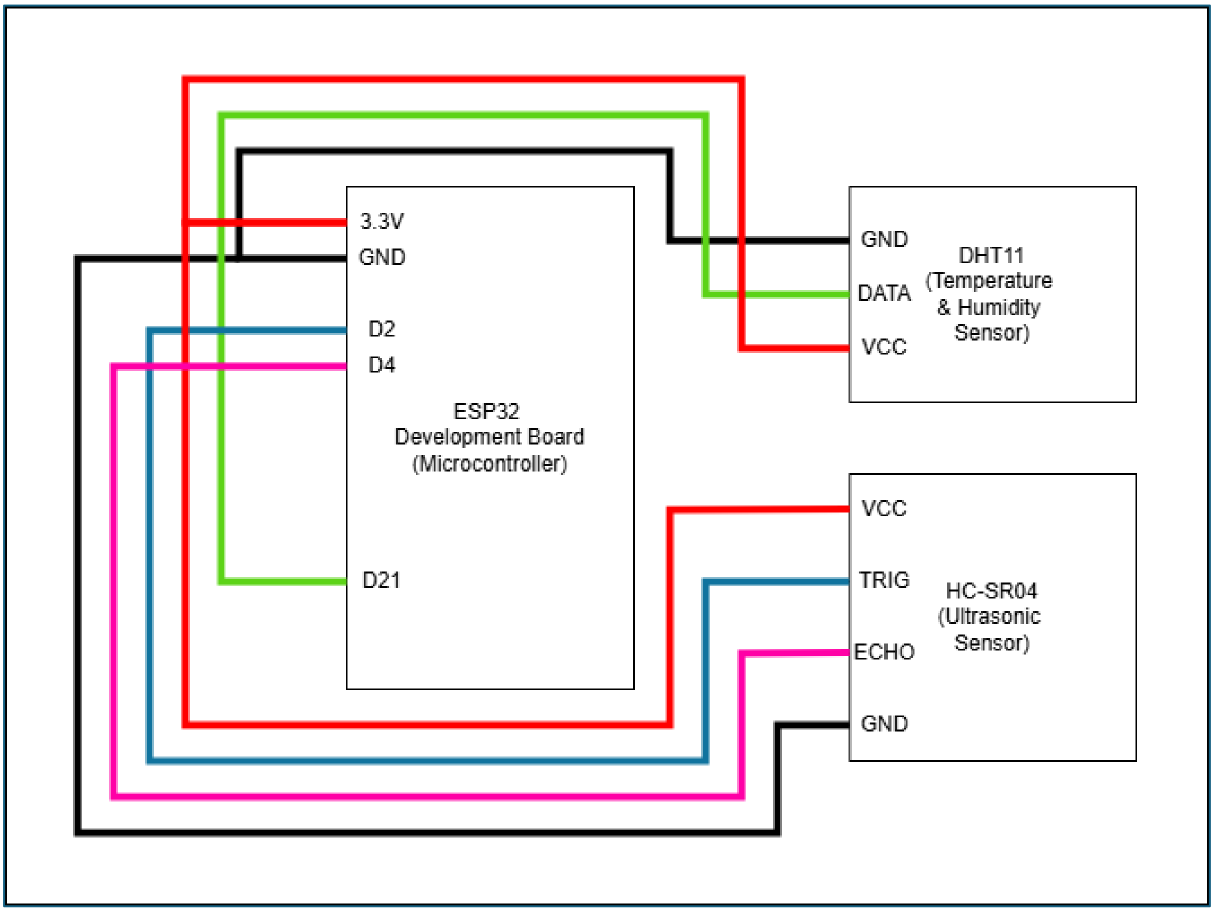

Figure 1: Circuit Diagram of the AR-Based IoT Environmental Monitoring System

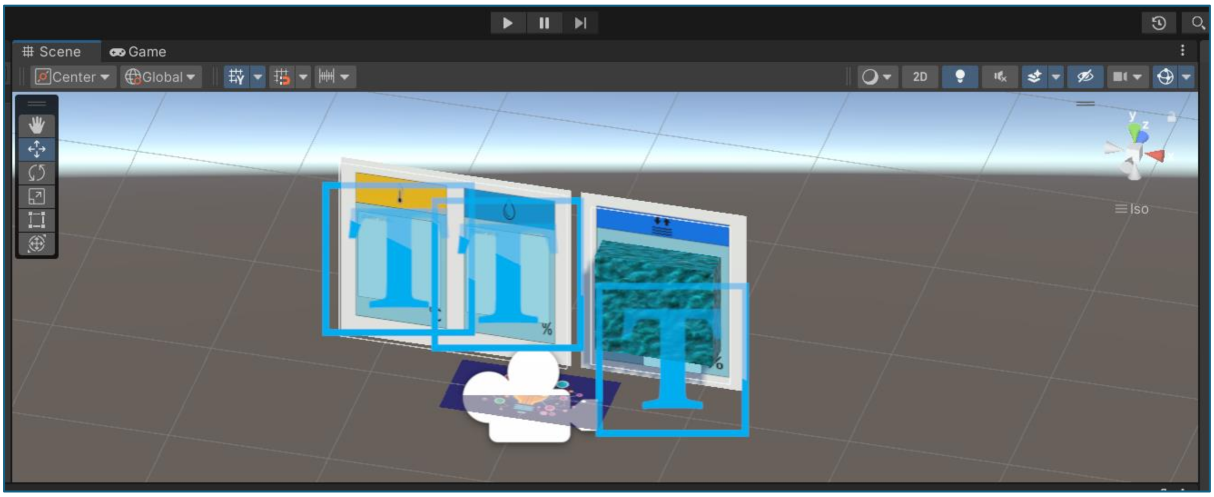

Figure 2: Unity AR Scene Design

The Unity scene design (Figure 2) for this project consists of a dynamic Augmented Realty (AR) environment where users can visualize real-time sensor data. The scene uses a target image that is stored in a database in Vuforia to trigger the AR content when detected. When activated, the app will display UI elements such as the appropriate backgrounds, text fields for the environmental variables (temperature, humidity and water level) fetched from Blynk Cloud from the monitoring system (Figure 1), and a resizable cube with water texture (LowlyPoly, 2019) representing the water level in a tank. The overall layout is simple and intuitive, with well-placed text fields and AR visuals, ensuring that the user can easily interact with and interpret the data in real-time.

5. Setup Instructions

5.1 Setting up the ESP32

1. **Install Arduino IDE:**
   - ➢ Download and install the Arduino IDE

2. **Add ESP32 Development Board to Arduino IDE:**
   - ➢ Open Arduino IDE and download the library "esp32" by Espressif Systems

3. **Connect ESP32 to PC:**
   - ➢ Use a Micro-USB cable to connect the ESP32 to your computer.
   - ➢ Select the correct Board (ESP32 Dev Module) and COM Port (COM 7) from Tools.

4. **Wire the Sensors:**
   - ➢ **DHT11 Sensor:**
     - ▪ VCC → 3.3V on ESP32
     - ▪ GND → GND
     - ▪ DATA → Digital Pin 21
   - ➢ **HC-SR04 Sensor:**
     - ▪ VCC → 3.3V on ESP32

- GND → GND
- Trig → Digital Pin 2
- Echo → Digital Pin 4

5. Upload Code:

- Upload the Arduino sketch that reads sensor values and sends them to Blynk using HTTP or Blynk API.

5.2 Preparing the Unity Project

1. **Install Unity and Create New Project:**

   - Install Unity Hub and Unity Editor (preferably Unity 2022.3.40f1 or later).

2. **Open AR-IoT Monitoring System in Unity:**

   - Open Unity Hub and select "Add > Add Project from Disk > AR-IoT Monitor".

3. **Build to Android Device:**

   - Connect an Android device via USB.
   - Change the minimal API Level to 6 by going to "File > Build Settings > Android > Player Settings > Other Settings > Minimal API Level > Android 6.0".
   - Build and run the project apk on the device by going to "File > Build Settings > Build > Your Android Device > Downloads".
   - Go to the downloads folder in your Android device and install the "AR-IoT Monitor" apk.

6. Features

1. **Real-time Temperature and Humidity Display**
   - **Potential in Health and Education:**

     The ability to monitor and visualize and temperature and humidity in real time can be used in areas such as science labs, classrooms and health facilities to ensure a safe environmental condition. Moreover, the need of a target image only also allows a portable experience and reduces culture in spaces from sensors.

2. **Real-time Water Level Visualization**
   - **Potential Industrial Use:**

     The water level monitoring has applications in agriculture, water treatment plants and liquid management. Due to its flexible scaling, which measures the initial depth distance to determine the lowest level, (0% initial depth) it is highly adaptable to various industrial applications.

3. **Dynamic AR Scene Updating Based on Sensor Readings**
   - **Visual Appeal and Engaging Interaction:**

     The real-time updating of the AR scene keeps users engaged by providing immediate visual feedback. Besides that, the cube (water level) has a water texture and dynamically adjusts its scale based on the real-time data, making the experience visually pleasing and not static.

4. **Portable and Expandable for Other Sensors and Target Images (Futureproof)**
   - **Creative Functionality:**

     The ability to integrate additional sensors offers creative possibilities for an endlessly customizable one-stop dashboard, building complex systems.
   - **Potential Business and Industrial Use:**

     The system can be easily expanded to include additional sensors in the future (e.g. pressure, pH, CO sensor) for use in numerous different purposes and industries. Thus, allows scalability for businesses to expand the functionality of the system to monitor other parameters

7. References

Blynk (n.d.) *Blynk: Internet of Things platform.* Available at: https://blynk.io/ (Accessed: 21 April 2025)

ELEGOO (2024) *ELEGOO Mega 2560 the Most Complete Starter Kit Tutorial.* Available at: https://www.elegoo.com/en-gb/blogs/arduino-projects/elegoo-mega-2560-the-most-complete-starter-kit-tutorial?srsltid=AfmBOoqi5TNNmHabMT6Pf5ucZwCktYBKg3AazW6EX0sUU8CxF47Ux7P5 (Accessed: 21 April 2025)

LowlyPoly (2019) *Stylize Water Texture.* Available at: https://assetstore.unity.com/packages/2d/textures-materials/water/stylize-water-texture-153577 (Accessed: 01 May 2025)

Vuforia (2024) *Vuforia SDK Downloads*. Available at: https://developer.vuforia.com/downloads/sdk (Accessed: 21 April 2025).