Smart Shoes Project

Integrating Sensors, IoT, and GPS for Enhanced Footwear

Team Members: **Meysam Jamali, Mohammadamin Rezaei Sepehr**

Course: **IoT**

Introduction

1 Objective

Design and implement smart shoes with integrated sensors to monitor health metrics and activity.

2 Platform

Developed using ESP32, Ubidots for data visualization, and an Android Application for user interface.

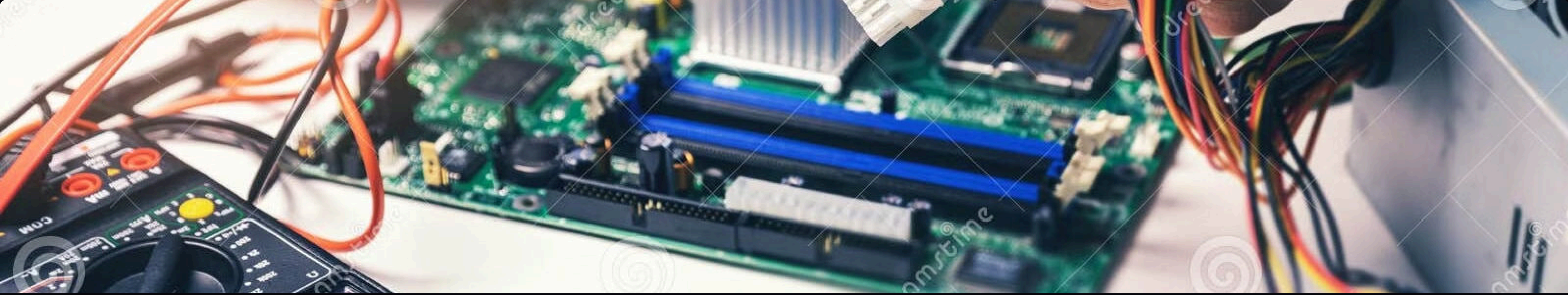
Project Overview

1 Components

- Sensors: DHT22, MPU6050, GPS module, Pressure sensor, Potentiometer
- Connectivity: WiFi (ESP32), MQTT (Ubidots)
- Power: Battery simulation and monitoring

2 Functionalities

- Real-time data collection (temperature, humidity, pressure, gait)
- GPS tracking and speed monitoring
- Health metrics (heart rate)
- Data transmission to Ubidots and Android app

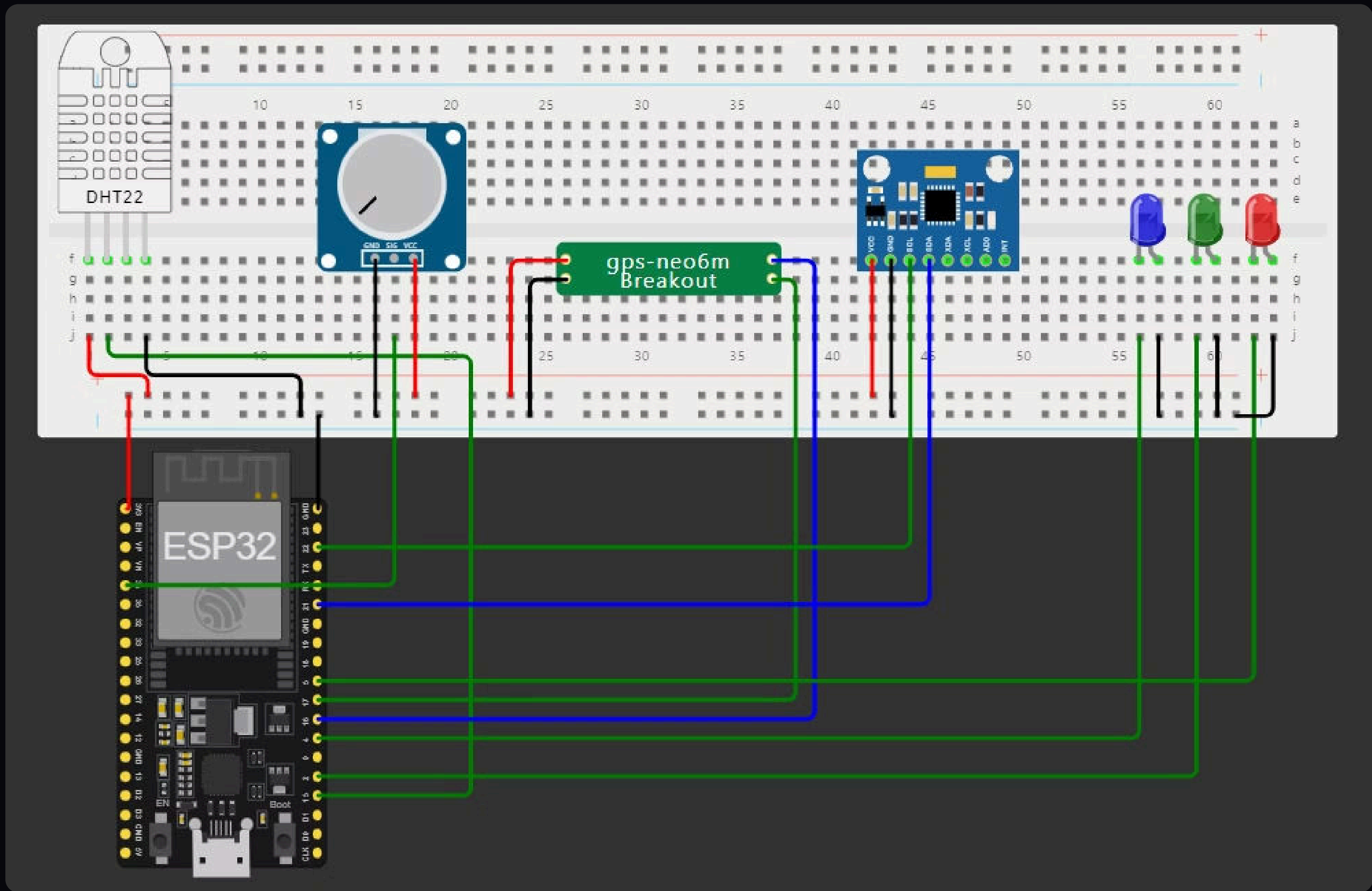


Hardware Components

ESP32 Microcontroller: Core controller for all sensors and communication.

- **Sensors:**
  - **DHT22:** Measures temperature and humidity.
  - **MPU6050:** Detects motion and orientation, providing acceleration and gyroscope data.
  - **GPS Module:** Tracks location and speed.
  - **Pressure Sensor & Potentiometer:** Simulates pressure data.

Hardware Components



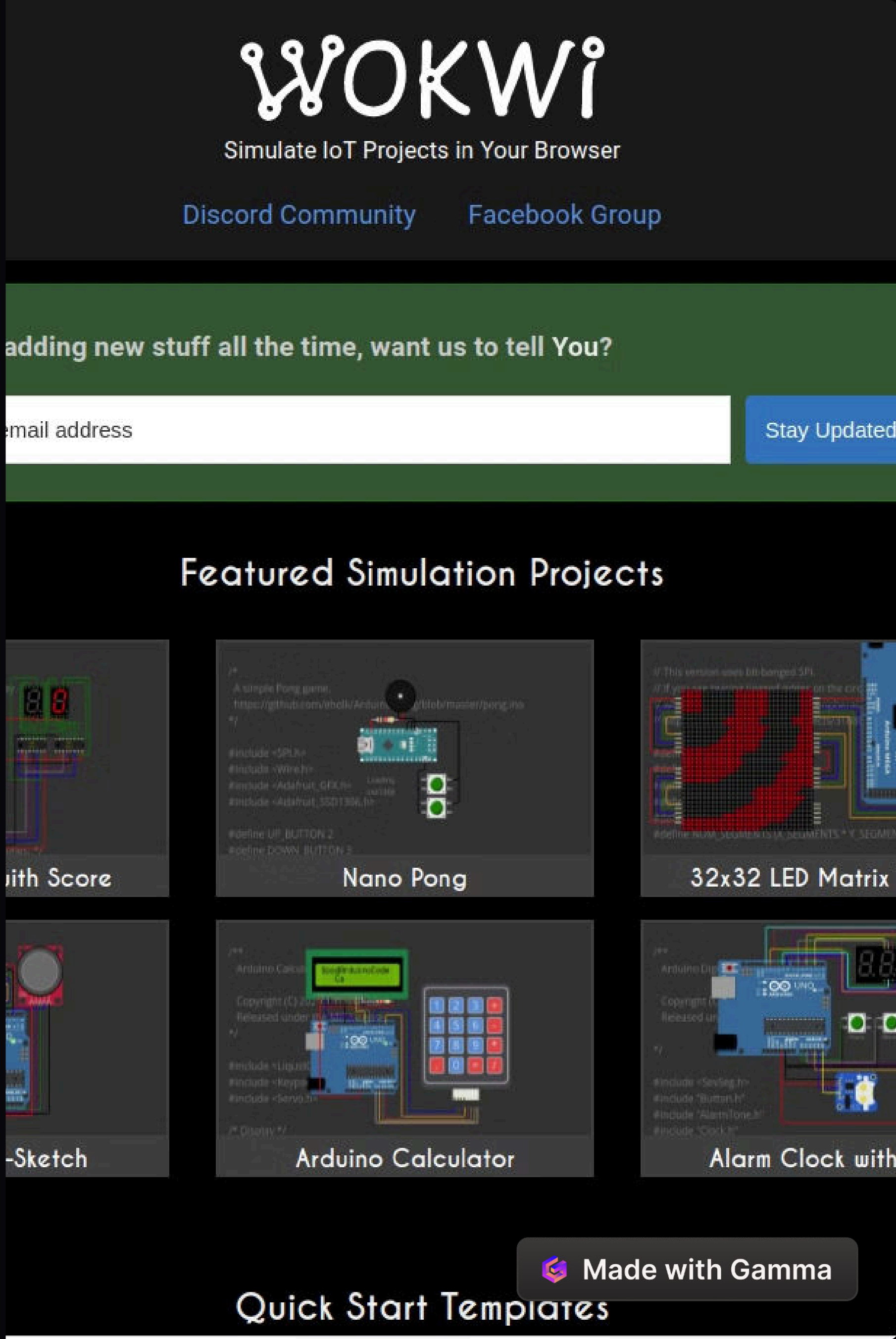
Software Components

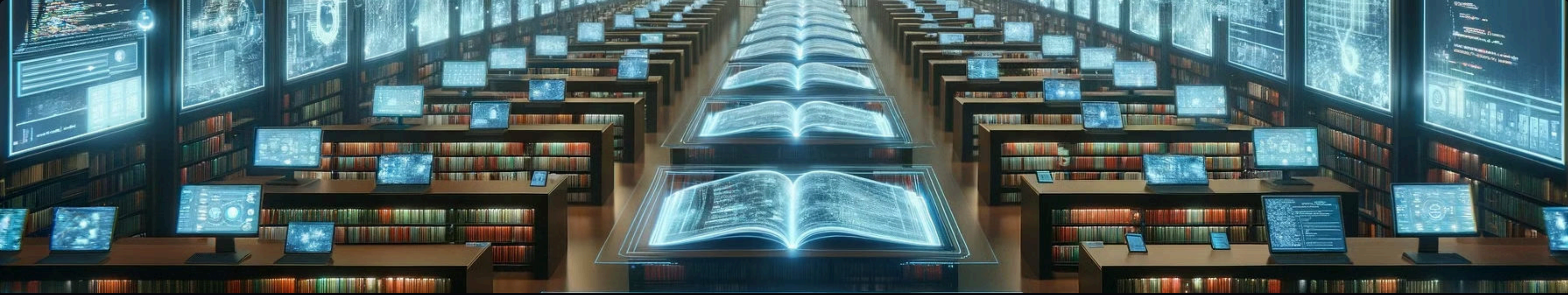
1 Libraries Used

For sensor interfacing, WiFi, and MQTT communication

2 Platforms

- **Wokwi:** Simulation environment
- **Ubidots:** Data storage and visualization
- **Android Application:** User interface





Libraries Used

- **DHT.h:** For interfacing with the DHT22 temperature and humidity sensor.
- **WiFi.h:** For connecting the ESP32 to a WiFi network.
- **PubSubClient.h:** For MQTT communication with Ubidots.
- **TinyGPS++.h:** For parsing GPS data.
- **Wire.h:** For I2C communication with MPU6050.
- **MPU6050.h:** For interfacing with the MPU6050 sensor.

DHT22 Sensor

1 Functionality

Measures environmental temperature and humidity.

2 Connection

Pin 15: Connected to the data pin of the DHT22 sensor.

3 Simulation

- Humidity is increased by 15% to simulate realistic measurements.
- Temperature is increased by 20°C for similar reasons.

4 Data Transmission

Real-time readings are transmitted via MQTT to Ubidots for monitoring.

MPU6050 Sensor

1 Functionality

- Detects motion and orientation.
- Provides acceleration and gyroscope data.

2 Connection

SDA Pin (21), SCL Pin (22): Connected to the I2C bus of the ESP32.

3 Usage

- Initialized to provide real-time motion data, which is used to determine the user's gait (walking, running, or stopped).

4 Data Transmission

Real-time readings are transmitted via MQTT to Ubidots for monitoring.

GPS Module

1 Functionality

- Tracks geographic location and speed.

2 Connection

TX Pin (16), RX Pin (17): Connected to the serial port of the ESP32.

3 Simulation

- Manually generated GPS data to simulate movement.

4 Haversine Algorithm

- Calculates the distance traveled based on GPS coordinates.
- This method computes the shortest path between two points on the Earth's surface, providing accurate distance measurements for the user's movement.

5 Data Transmission

Real-time readings are transmitted via MQTT to Ubidots for monitoring.

Pressure Sensor & Potentiometer

1 Functionality

- Simulates foot pressure data to monitor the user's gait.

2 Connection

Potentiometer (Pin 34): Connected to analog input pins of the ESP32.

3 Simulation

- Potentiometer values are mapped to realistic pressure ranges (950 to 1050).

4 Pressure and Gait Adjustments:

- Pressure changes based on gait status:
  - Walking: Pressure increased by 8 units.
  - Running: Pressure increased by 30 units.
  - Stopped: Pressure decreased by 4 units.

5 Data Transmission

Real-time readings are transmitted via MQTT to Ubidots for monitoring.

Heart Rate Simulation

1 Functionality

- Simulates heart rate data as no physical sensor is available.

2 Simulation

- Heart rate values are generated based on the current gait:
  - Walking: Random values between 80-100 BPM.
  - Running: Random values between 100-140 BPM.
  - Stopped: Random values between 60-80 BPM.

3 Data Transmission:

- Real-time heart rate data is sent to Ubidots via MQTT.

Battery Simulation

1 Functionality

- Simulates battery level changes.

2 Simulation

- Battery level decreases by 0.5% each loop iteration.
- When battery level reaches 0, it resets to 100%.

3 Data Transmission:

- Battery level data is sent to Ubidots via MQTT.

Gait Detection and LED Indicators

1 Functionality

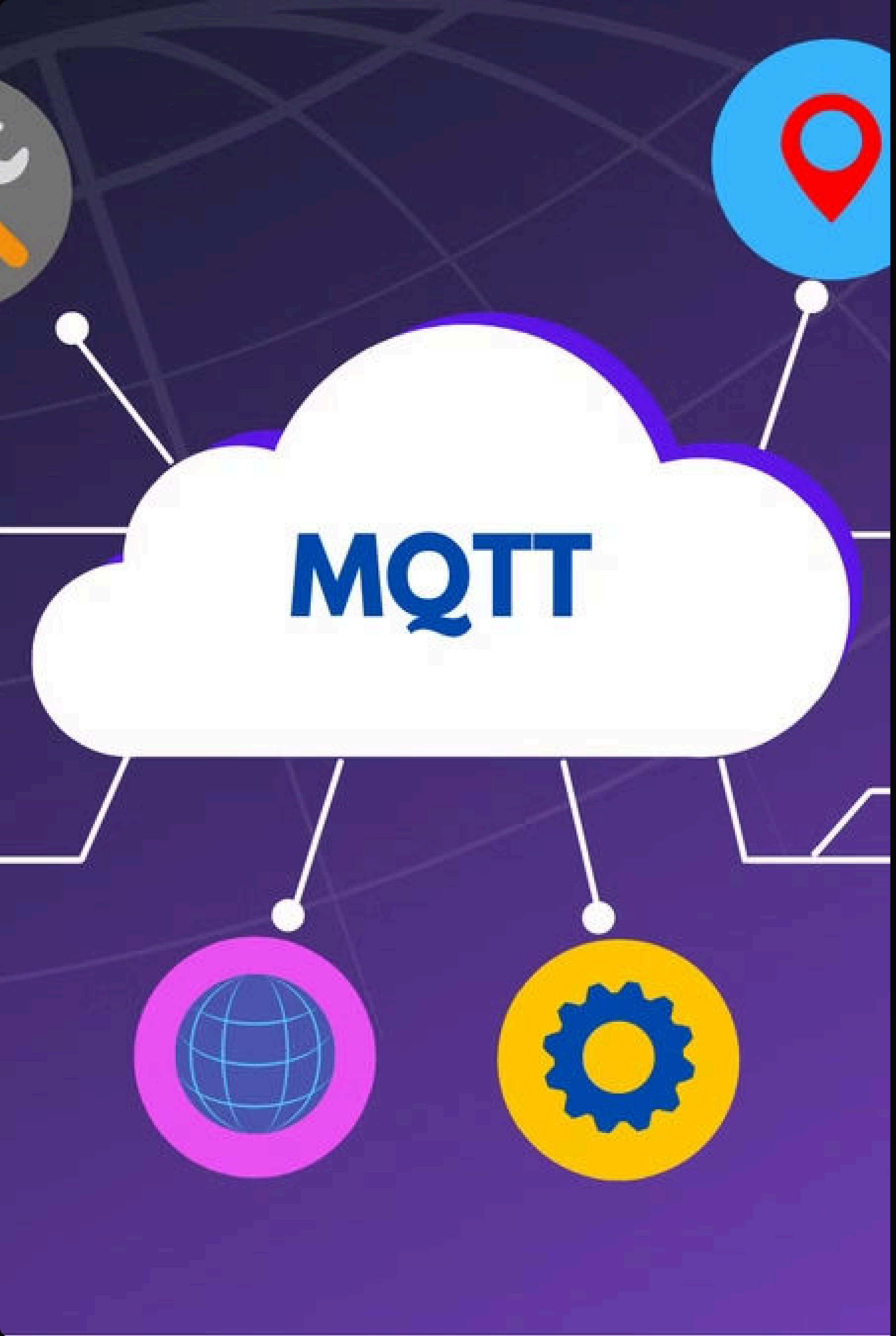
- Determines whether the user is walking, running, or stopped.

2 Method

- Based on time intervals and pressure changes:
  - Walking: Slight increase in pressure.
  - Running: Larger increase in pressure.
  - Stopped: Decrease in pressure.
- LEDs indicate the current gait status (Walking, Running, Stopped).

3 Indicators

- Walking: LED on pin 2.
- Running: LED on pin 4.
- Stopped: LED on pin 5.



WiFi and MQTT Setup

1. **WiFi Connection:** Connecting ESP32 to a WiFi network.
2. **MQTT Connection:** Setting up MQTT client to communicate with Ubidots.
    - Attempts to reconnect if the MQTT client is disconnected.

Data Transmission to Ubidots

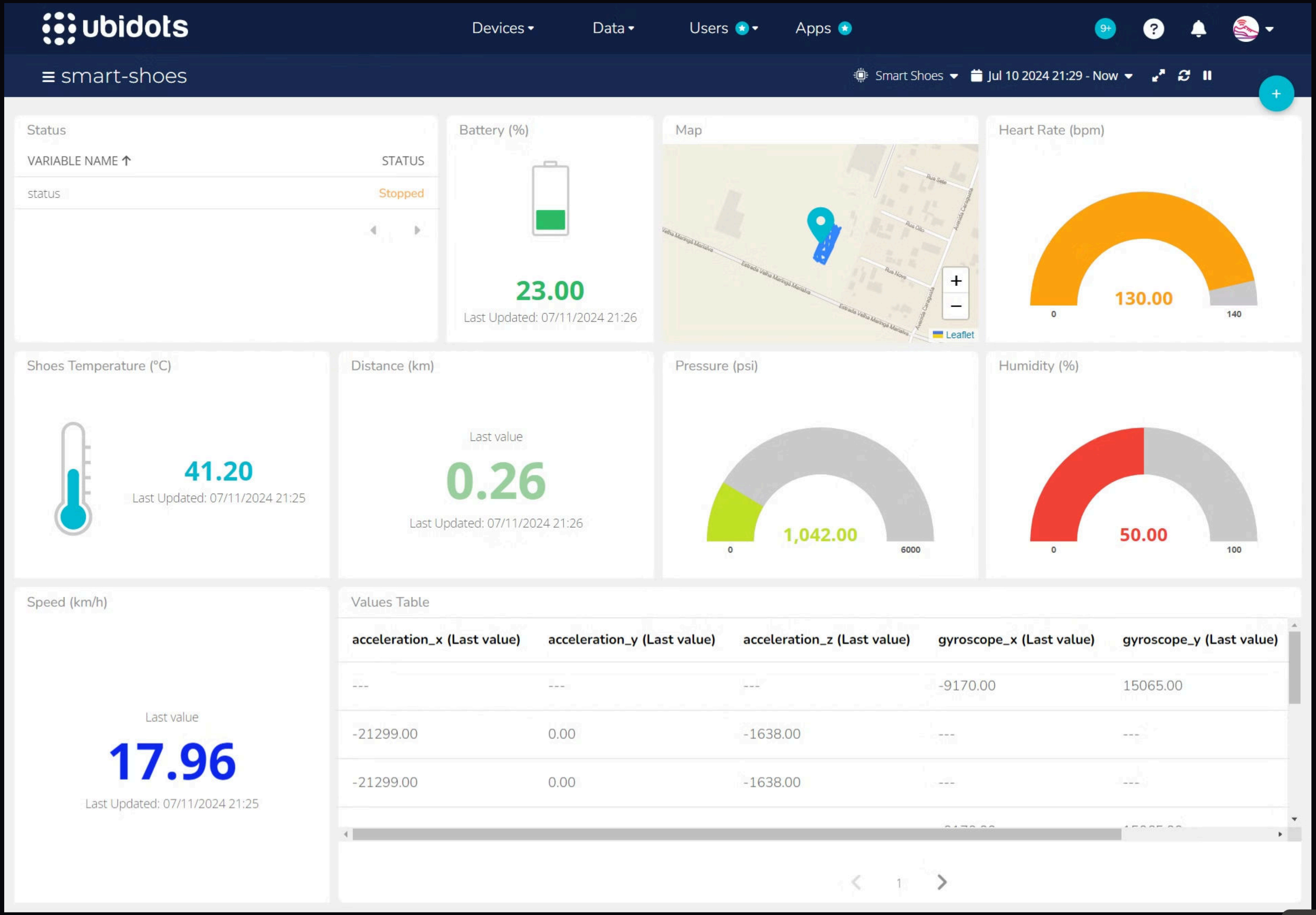
- Sensor data (temperature, humidity, pressure, gait, GPS) is formatted and sent to Ubidots.
- Data includes real-time readings and contextual information.

Results and Data Visualization

Ubidots and Android Application Dashboard:

- Graphs displaying temperature (Celsius unit), humidity (Percent unit), and pressure (Pascal unit) over time.
- GPS location tracking on a map (latitude and longitude).
- Gait status with corresponding time intervals.
- Heart rate monitoring graph (Beats per minute or BPM).
- Battery monitoring graph (percent).
- Speed monitoring graph (km/h).
- Distance monitoring (km).
- Gyroscope in 3 axis.
- Acceleration in 3 axis.
- Date and Time (no available widget).

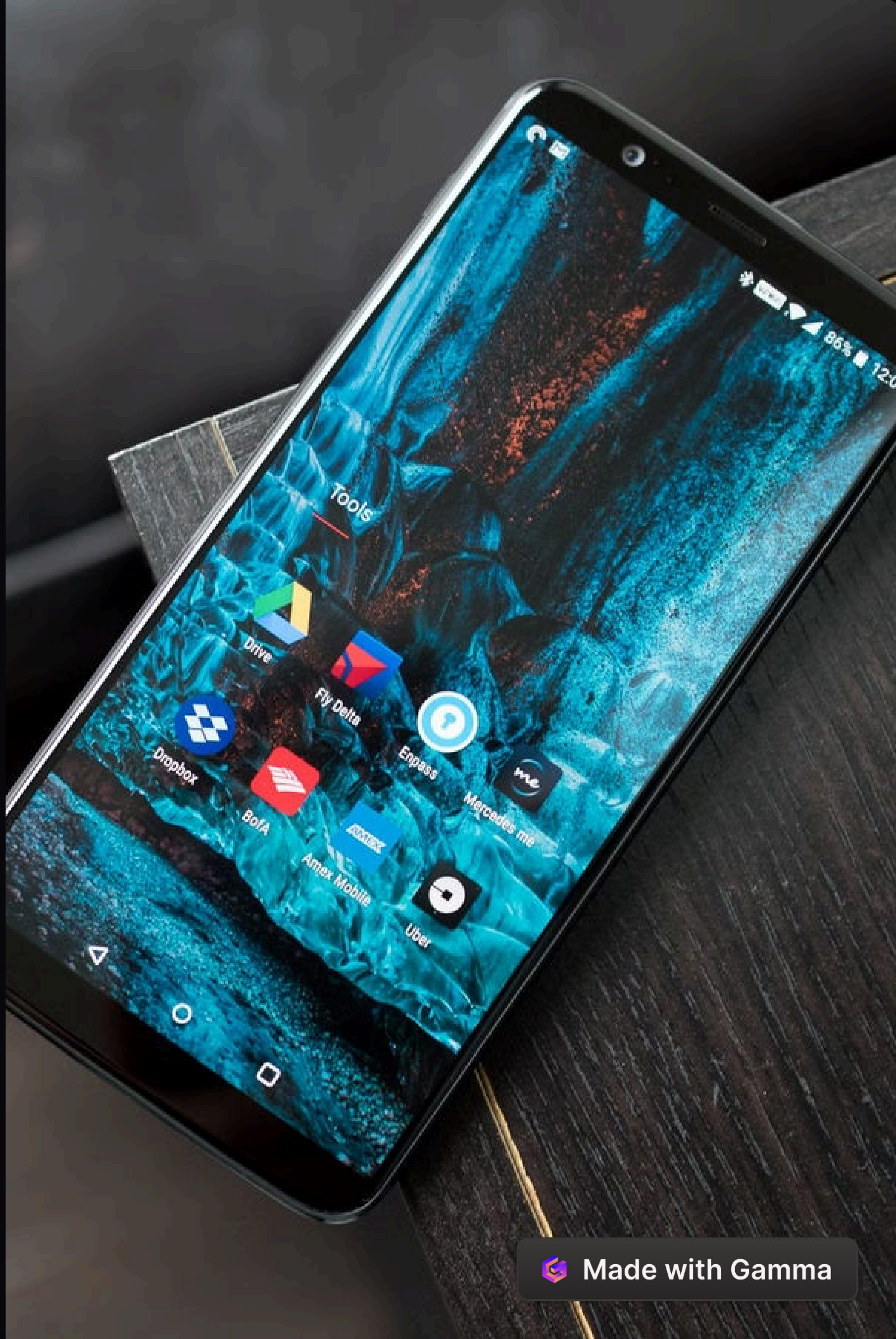
Ubidots Dashboard



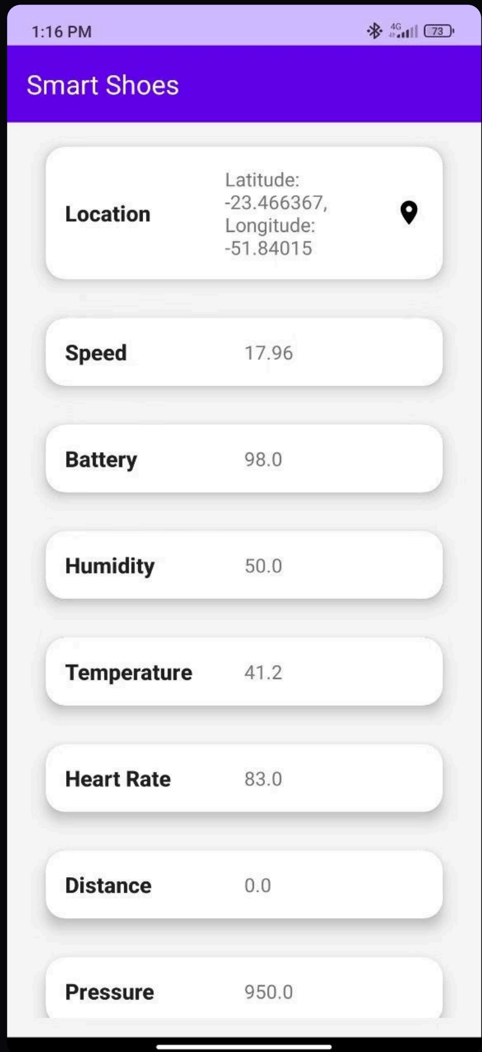
Made with Gamma

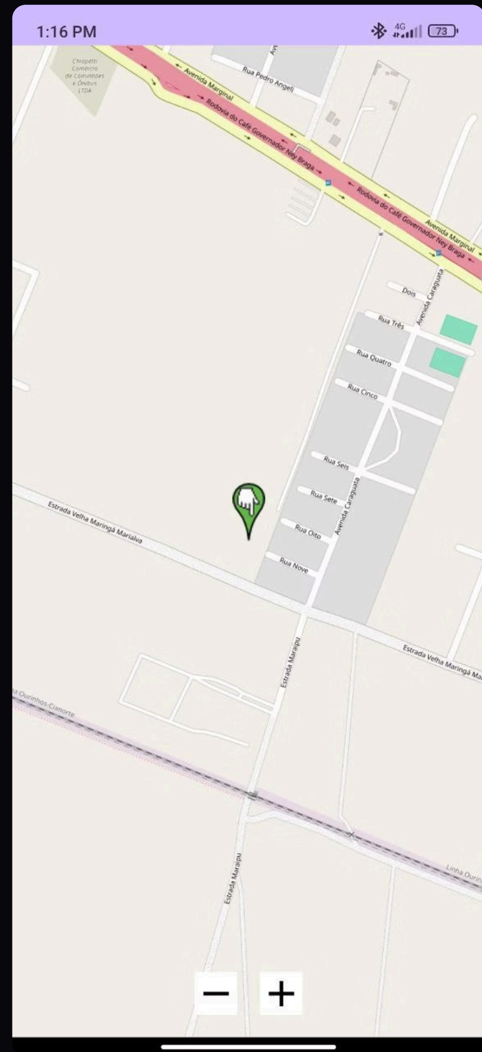
Android Application Overview

- **Features:**
  - Displays real-time sensor data (Temperature, Humidity, Pressure, Gait).
  - GPS tracking shows current location and speed.
  - Health monitoring including heart rate.



Android Application





Conclusion

1 Summary

- Successfully integrated multiple sensors to create smart shoes.
- Real-time data collection and transmission to Ubidots and Android app.

2 Q&A

Open the floor for questions.

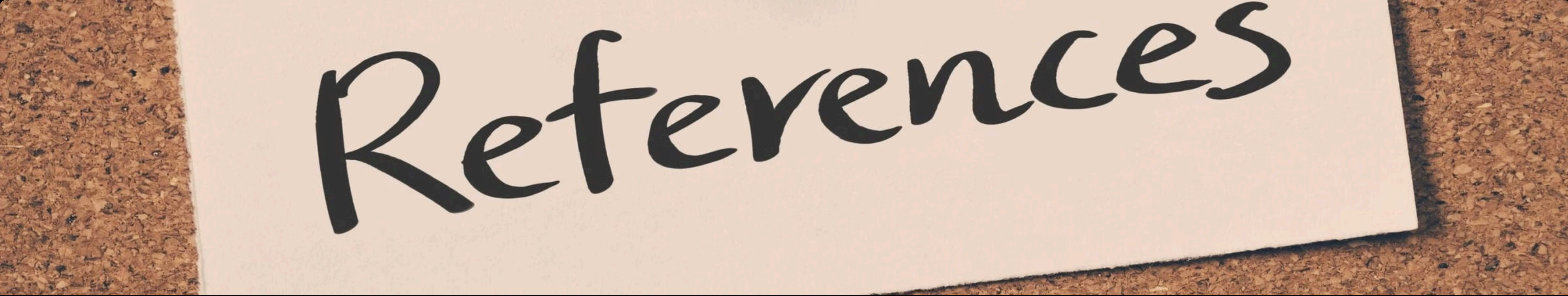
Project Links and Information

Wokwi: https://wokwi.com/projects/402494050827141121

Ubidots: https://stem.ubidots.com/app/dashboards/667b0c693563fe000e032f57?devices=668011d8f3fa49000bd01b74

Ubidots Username: smartshoes
Ubidots Password: mars23435676

Github:



References

- GPS_Simulation[wokwi] - Google Drive
- ESP32 Pinout Reference: Which GPIO pins should you use? | Random Nerd Tutorials
- Pressure Transducer - Wokwi ESP32, STM32, Arduino Simulator
- MPU6050 sensor with Arduino uno in wokwi website.
- ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE | Random Nerd Tutorials
- ESP32 Temperature/Humidity sensor stops updates after several days
- ESP32 WiFi Networking | Wokwi Docs
- and etc.

Thank you for your attention

Thank you to dear professors **Davide Ancona**, **Giorgio Delzanno**, and dear assistant instructor **Ali Varasteh Ranjbar**