Solar Tracking IoT System

Team 3

14/07/2025

1. Team members

Name	Matriculation Number	Email Address
Mofifoluwa Akinwande	2220059	Mofifoluwa-ipadeola.akinwande@stud.hshl.de
Rubayet Kamal	2219943	rubayet.kamal@stud.hshl.de
Moiz Zaheer Malik	2220074	Moiz-zaheer.malik@stud.hshl.de

Concept Description

This project is an IoT-based system that combines solar tracking and RGB LED control using two Arduino Uno boards and a Raspberry Pi.

- One Arduino is connected to two LDR sensors and a servo motor. It reads sunlight levels and rotates a solar panel to face the brighter side, helping capture more sunlight during the day. This process is called **solar tracking**.
- The second Arduino is connected to an RGB LED. It listens for color commands (like “red”, “green”, or “blue”) sent over WiFi using the **MQTT** protocol. When it receives a command, it changes the LED to the corresponding color.
- The Raspberry Pi acts as the **MQTT broker**, which means it receives and forwards messages between devices.
- A small Python script on the Raspberry Pi processes light data from the solar tracker and sends simple feedback like “bright” or “dark” to another MQTT topic.

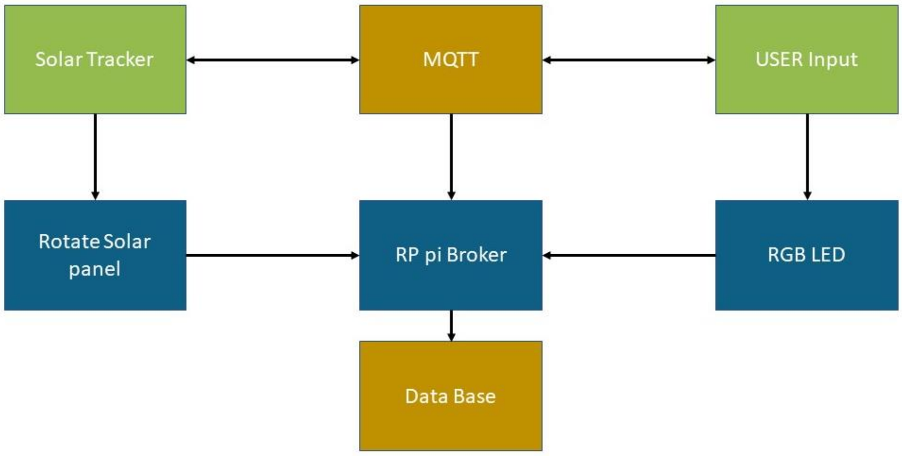

Figure 1: Concept Diagram

2. Project/Team management

2.1. Project Management Method

In this project, we ensured that tasks were evenly distributed among team members so that everyone feels involved not only with the project but with the learning process as well. To facilitate effective planning and version control throughout different cycles of the project, we used Github as our primary tool.

2.2 Task Division

Table 1: Task division among group mates

Name	Tasks
Mofifoluwa Akinwande	1) Communication between devices 2) Setting up the Cloud platform 3) Python script
Rubayet Kamal	1) Setting up the solar panel 2) Communication between devices 3) Assembly of the system
Moiz Zaheer Malik	1) Arduino logic for LDR and servo 2) Hardware design and setup 3) Mosquitto installation 4) Assembly of the system

Introduction

The project was developed as part of the Advanced Embedded Systems course and focused on combining mechanical systems with wireless communication. For the project we decided to design a Solar Tracking IoT system as it presented a difficult but fun challenge.
The goal for this system defined by us were:

1) Build a prototype for a device that effectively tracks sunlight for the solar panels using the appropriate sensors and actuators.

2) Implement effective communication between components of the system using an appropriate communication protocol.

3) Implement a Cloud platform for access of important data at any time.

4) Control of an actuator component remotely.

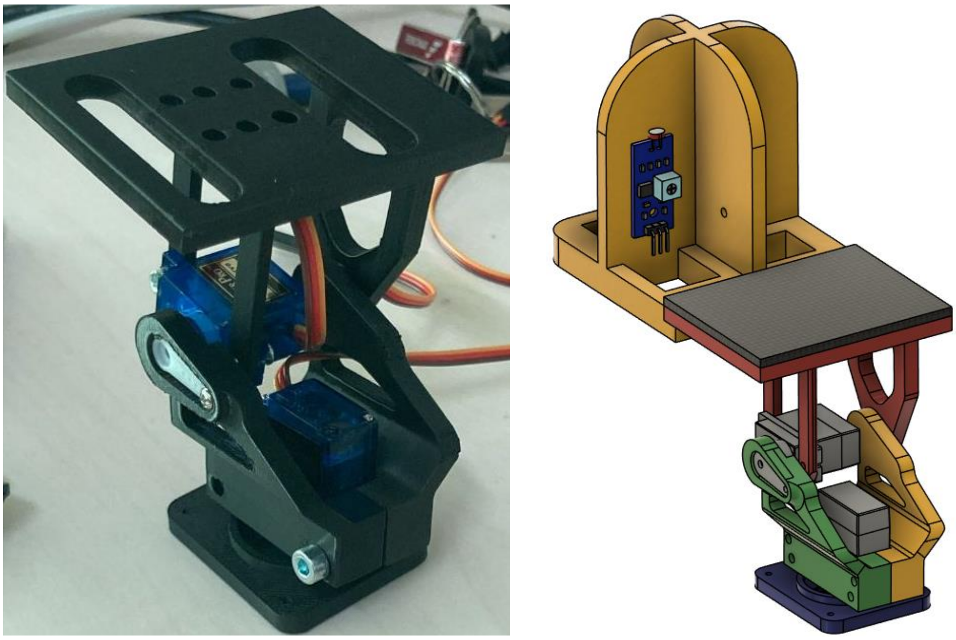

Figure 2: Designed Parts.

In order to achieve these goals, we defined what is important and what kind of components are needed (both hardware and software) for a successful implementation of our idea. The next section outlines the components used in our project.

3. Components Used

Hardware

Component	Function
Arduino Uno (x2)	One for solar tracking, one for LED control
Raspberry Pi	Acts as MQTT broker
Solar Panel (x2-3)	Powers the LED Arduino
Servo Motor	Moves the solar panel left and right
LDRs (2)	Detect sunlight direction
RGB LED	Shows color based on command
3D Printed Parts	Hold LDRs, servos, and solar panel

Software

Tool/Library	Purpose
Arduino IDE	Write and upload code to Arduino
Fusion 360	Design 3D printable parts
Mosquitto MQTT	MQTT broker on Raspberry Pi
WiFiNINA, PubSubClient	Connect Arduino to WiFi and MQTT
Python Script	Handle and process LDR readings
MQTT App/Terminal	Send commands to LED system

4. System Setup and Flow

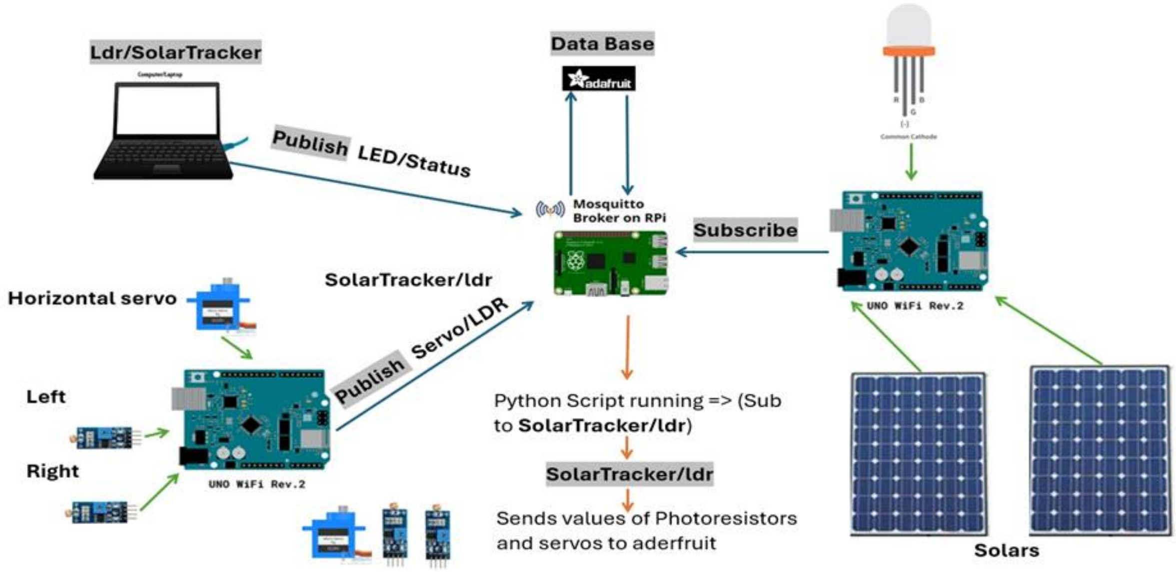

Figure 3: Overall communication flow.

B. How It Works

As seen from the diagram above, the Raspberry Pi acts as the central hub, controlling all the incoming ad outcoming messages. The first Arduino Wifi houses the sensors (LDR), actuator (servo motors) and publishes the topic “solartacker/ldr” whose payload is the values gotten from the photoresistors. The second Arduino WiFi houses the RGB led and the solar panels whose primary role is to generate current to power the device. It subscribes to the topic “led/status” which was published by a remote device which in this case was a laptop.

On the Raspberry Pi itself, a python script runs which also subscribes to the topic “solartracker/ldr” and implements a logic based on the payload received. The logic controls the payload of the topic “ldr/control/status” which can either be “dark” or “bright” depending on the current positions of the sensors.

To further improve the IoT capabilities of this system, a cloud platform Adafruit IO was used to hold the data readings gotten from the photoresistors. The script also houses the key of the Adafruit IO cloud platform and the feeds “leftLDR” and “rightLDR”, where the LDR data readings were sent to and can be retrieved from. From the cloud platform the data can be accessed at any time for further analysis and collation as long as the Raspberry Pi is operational.

To ensure effective communication between all components of the system the MQTT protocol was used. MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol perfect for IoT systems. We installed Mosquitto, a popular open-source MQTT broker, on the Raspberry Pi and it was configured to accept connections from devices on our WiFi network. For this project, the Raspberry Pi acted as the MQTT broker and the main MQTT clients were the photoresistors and the remote device (laptop).

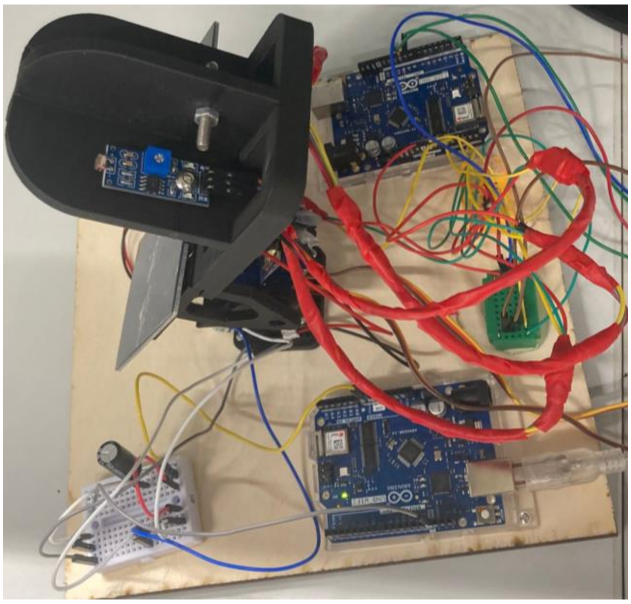

Figure 4: Final Setup.

5. Challenges

While we were able to implement our idea, the project wasn't without its challenges. This section briefly describes some of the problems we encountered.

Firstly, for our prototype the initial idea was to create a dual-axis system where there were two servo motors controlling the vertical and horizontal movement of a 3D structure that was meant to house the solar panel, however after assembling the prototype we discovered that the servo responsible for the vertical movement wasn't strong enough.

Another challenge was to power the second Arduino WiFi using the solar panels. The 5V solar panels were connected in parallel in order to provide double the current that each panel supplies. Despite the Arduino being powered on through the solar panel when exposed to the sunlight, it was not enough to drive the LEDs. Therefore, when the Arduino was to supposed to subscribe to the “led/status” topic published by the Laptop via MQTT, the LEDs did not turn on as expected. Using a power cable instead of the solar panels showed that Arduino worked as expected and therefore it was current supply problem from the solar panels.

6. Final Results

Despite the challenges faced, some of the goals which we set out to achieve for this project were successfully met.

The solar tracker reliably rotated the panel left and right throughout the day based on light levels.

The RGB LED changed colors instantly when a message was sent from a phone or laptop showing effective communication.

The MQTT communication between devices was fast, stable, and accurate.

The mechanical parts we designed fit perfectly, and the 3D printed components performed well in practice.

7. Future Improvements

As it this was an IoT based project, there are good opportunities to further scale the system and add and improve the overall functionality.

Some of these include using ESP32 boards instead of Arduino WiFi modules as they offer more reliable wireless connections and more peripherals.

Adding additional sensors to measure the temperature as a way to optimize the system is another way to further improve the system.

Another possible improvement would be to create a web dashboard to control and view data.

Finally, to further debug the problem with the solar panels, we could use some devices to see how much current and voltage the panels were able to generate and if needed add more solar panels.

8. Summary

This project combined electronics, 3D design, and networking into a working IoT system.

It’s a good example of how simple tools and parts can build a smart device. Anyone with basic knowledge of Arduino and Raspberry Pi can build and expand this system.

9. References

[1] GitHub, “GitHub,” [Online]. Available: https://github.com/boredfifo/SS2025_HWE_LAB_TEAM3_/tree/main/Advanced%20Embedded%20Engineering/Project.

[2] MQTT, “MQTT – a lightweight messaging protocol for small sensors and mobile devices,” [Online]. Available: https://mqtt.org.

[3] Fusion 360, Autodesk, “Fusion 360,” [Online]. Available: https://www.autodesk.com/products/fusion-360.

[4] Adafruit IO, “Adafruit IO,” [Online]. Available: https://io.adafruit.com.

[5] Arduino, “Arduino,” [Online]. Available: https://www.arduino.cc.

[6] Raspberry Pi, “Raspberry Pi,” [Online]. Available: https://www.raspberrypi.org.