FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DAQ TECHNOLOGY AND ANALYSIS 2

BERL2214	SEMESTER 1	SESI 2024/2025

LAB 6: MONITORING AND CONTROLLING VIA ARDUINO CLOUD WEBPAGE

NO.	STUDENTS' NAME	MATRIC. NO.
1.	MUHAMMAD AZRUL BIN REDZUAN	B122310626
2.		
3.		
4.		

PROGRAMME	2 BERL
SECTION / GROUP	S11 / GROUP 10
DATE	25/6/2025
NAME OF INSTRUCTOR(S)	1. FAREES EZWAN BIN MOHD SANI @ ARIFFIN
	2.

EXAMINER'S COMMENT(S)	TOTAL MARKS

1.0 INTRODUCTION

This project is called **"Intelligent Light with Human Detector"**. It is a smart lighting system that turns on a light only when it's dark and someone is nearby. The system uses an **ESP8266 (NodeMCU)**, a **light sensor (LDR)** to detect day or night, and a **motion sensor (PIR)** to detect movement. The system can be controlled using a smartphone app called **Blynk**, and a **relay** is used to turn the whole system ON or OFF.

1.1 Project Purpose

The purpose of this project is to:

- Save electricity by turning on the light only when needed.
- Make lighting automatic and more convenient.
- Let users control the system remotely using a phone.
- Show sensor data and system status on the Blynk app.

1.2 Project Goals

- Detect day and night using a light sensor (LDR).
- Turn on the light at night only when motion is detected.
- Show all sensor data and status on the Blynk app.
- Use a Blynk button to turn the whole system ON or OFF using a relay.
- Show a countdown timer in the app when motion is detected.
- Make lighting safer and smarter for home use.

2.0 CIRCUIT DIAGRAM

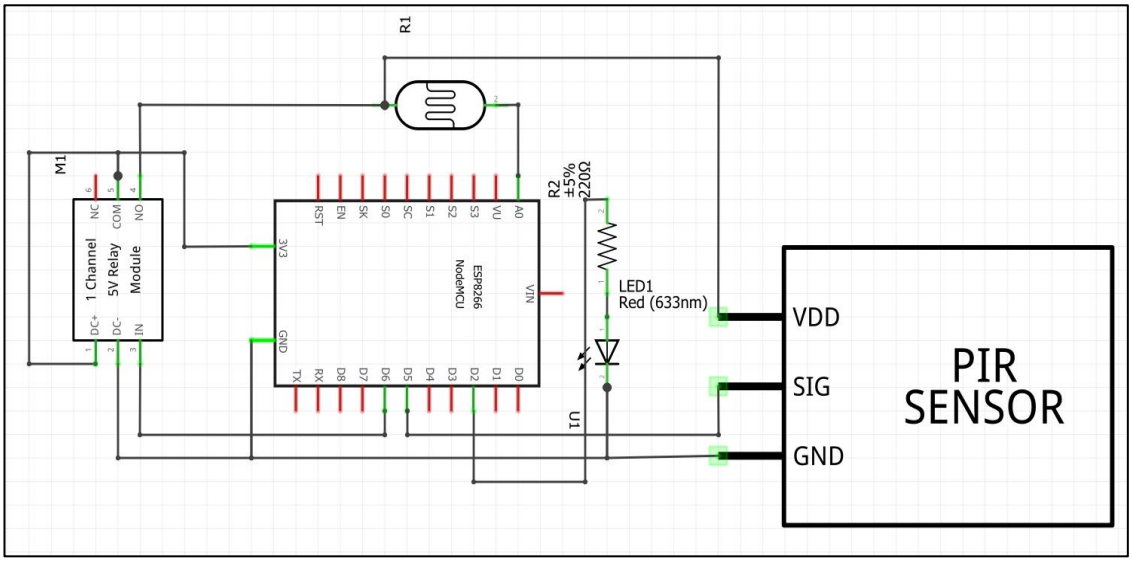


Figure 2.1: Schematic Circuit Diagram for Project

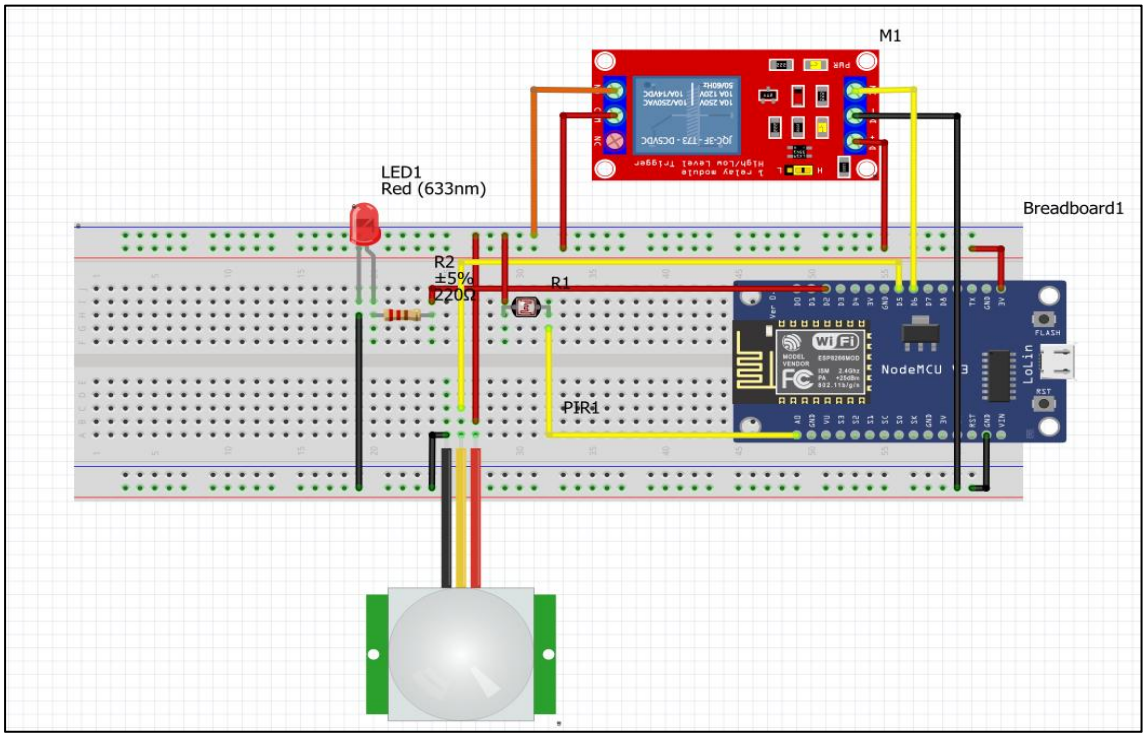
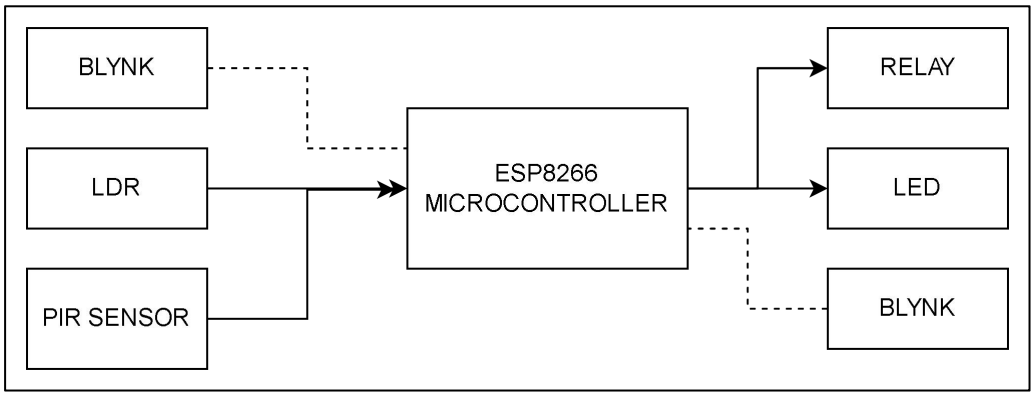


Figure 2.2: Hardware Circuit Diagram for Project

2.1 Block Diagram



2.2 Flow Chart

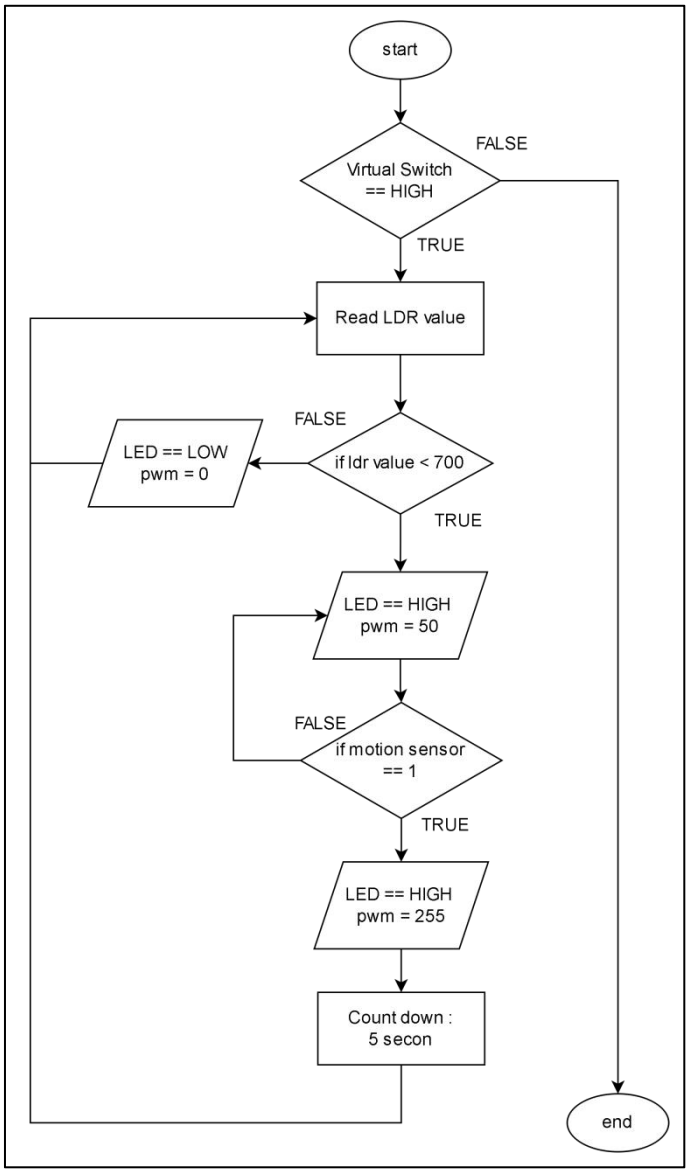


Table2.1: Component Pin Connection and Function List

No.	Component	ESP8266 Pin	Function
1	**LED**	D2	Output to control light (via PWM)
2	**PIR Sensor**	D5	Digital input to detect motion
3	**LDR (Light Sensor)**	A0	Analog input to detect light level
4	**Relay Module**	D6	Digital output to turn ON/OFF system
5	**Blynk App**	**Virtual Pins**	**Remote control and monitoring**
	LED Widget (V1)	V1	Show LED brightness
	Ldr Value Display (V2)	V2	Show LDR raw value
	Voltage Value Display (V3)	V3	Show LDR voltage
	Day Label (V4)	V4	Show day/night/system status
	Movement (V5)	V5	Show motion detection text
	Button (V6)	V6	ON/OFF control for the system
	Timer Display (V7)	V7	Show countdown timer
6	**Power Supply**	3.3V / 5V & GND	Power for ESP8266 and sensors

Description:

This project uses an ESP8266 to control a smart lighting system based on light level and human motion. An LDR detects day or night, while a PIR sensor detects movement to turn on an LED at night. A relay is used to turn the whole system ON or OFF through the Blynk app.

3.0 CODE AND MOBILE APP

Wi-Fi and Blynk Setup

```
#define BLYNK_TEMPLATE_ID "TMPL65mLBbL1u"
#define BLYNK_TEMPLATE_NAME "Project GP7"
#define BLYNK_AUTH_TOKEN "aShZubT87tIPdFHIB6JJhtPqsIEqxDKD"

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char ssid[] = "UTeM-IoT";
char pass[] = "123456789";
```

- This sets up the connection to the **WiFi network** and links to the **Blynk project** using your template ID and token.

Component Setup

```
int led = D2;
int pir = D5;
int ldr = A0;
int relay = D6;
```

- Defines the pin connections for:

  LED (D2), PIR sensor (D5), LDR sensor (A0 - analog), Relay (D6)

Blynk Button to Control System

```
BLYNK_WRITE(V6) {
  systemEnabled = param.asInt();
  digitalWrite(relay,systemEnabled);
}
```

- This function triggers when the **Blynk button (V6)** is pressed.
- Turns **relay ON or OFF** depending on the button state.
- Updates the systemEnabled variable so that the rest of the system knows if it should run.

Setup Function

```
27  void setup() {
28    Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
29    Serial.begin(115200);
30    pinMode(led, OUTPUT);
31    pinMode(pir, INPUT);
32    digitalWrite(relay,LOW);
33  }
```

- Initializes the serial monitor, Blynk, and component pin modes.

System Condition

```
35  void loop() {
36    Blynk.run();
37
38    if (!systemEnabled) {
39      analogWrite(led, 0);
40      Blynk.virtualWrite(V1, 0);
41      Blynk.virtualWrite(V4, "SYSTEM OFF");
42      Blynk.virtualWrite(V5, "SYSTEM OFF");
43      Blynk.virtualWrite(V7, "SYSTEM OFF");
44      return;  // Skip loop
45    }
```

- If the system is turned off from the Blynk app, everything else is **skipped**.
- All widgets are updated to reflect “SYSTEM OFF” status.

Sensor Readings and Logic

```
47    ldrValue = analogRead(ldr);
48    motion = digitalRead(pir);
49    voltage = ldrValue * (3.3 / 1024);
```

- Reads LDR value (light level), PIR sensor (motion), and calculates voltage from LDR.

Day/Night and Motion Control

```
56    if (ldrValue > Day) {
57      Serial.println(" - MORNING");
58      analogWrite(led, 0);
59      Blynk.virtualWrite(V1, 0);
60      Blynk.virtualWrite(V4, "MORNING");
61      Blynk.virtualWrite(V5, "...");
62    }
63    else {
64      Serial.println(" - NIGHT");
65      if (motion == HIGH) {
66        Serial.println("Movement Detected");
67        analogWrite(led, 255);
68        Blynk.virtualWrite(V1, 255);
69        Blynk.virtualWrite(V5, "Movement Detected");
70
71        for (int i = 5; i >= 0; i--) {
72        Blynk.virtualWrite(V7, i);
73        delay(1000);
74    }
```

```
76        else {
77          analogWrite(led, 50);
78          Blynk.virtualWrite(V1, 50);
79          Blynk.virtualWrite(V5, "");
80          Blynk.virtualWrite(V4, "NIGHT");
81        }
82      }
```

- If it's day (ldrValue > 700), the LED stays off.
- If it's night and motion is detected → LED fully on for 5 seconds.
- If no motion → LED is dimmed to 50 (PWM).
- **Countdown timer** (5 to 0) is shown in Blynk (V7).

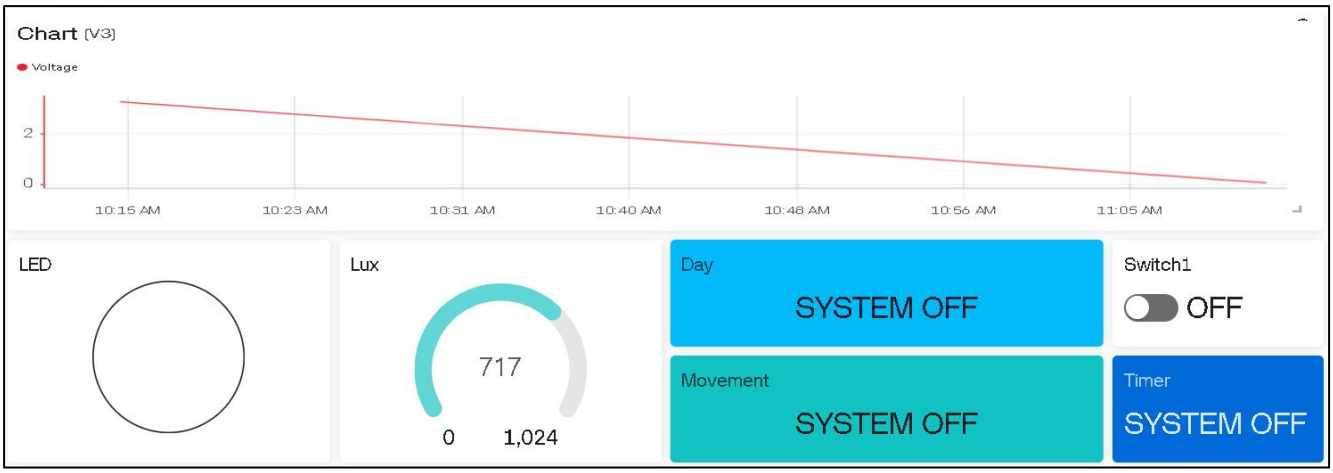
Display Values in Blynk

```
84    Blynk.virtualWrite(V2, ldrValue);
85    Blynk.virtualWrite(V3, voltage);
86    Blynk.virtualWrite(V7, "...");
87    delay(2000);
88  }
```

Sends live data to the Blynk app:

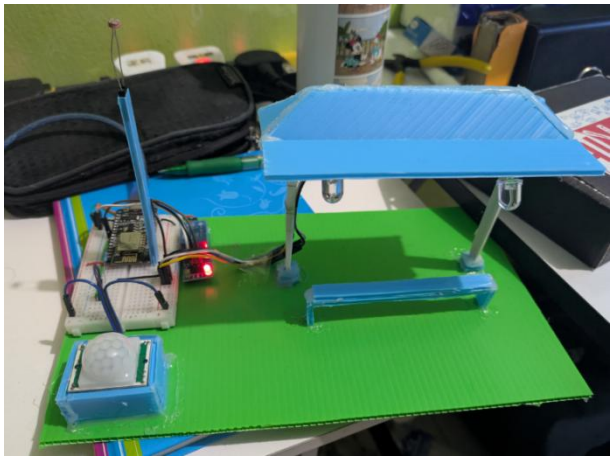
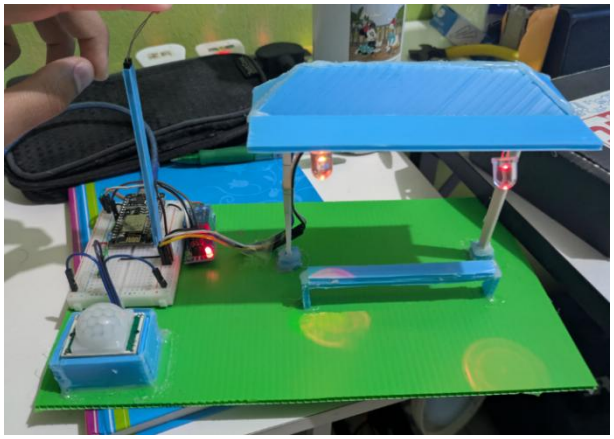
V2: LDR value, V3: Voltage from LDR, V7: Reset countdown

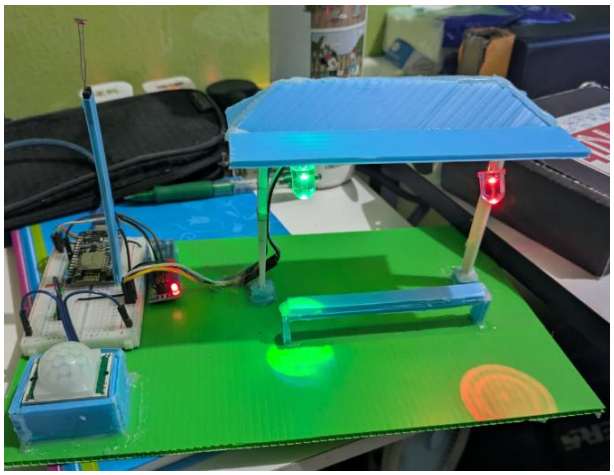
Table 3.1: Blynk Design for Monitoring system



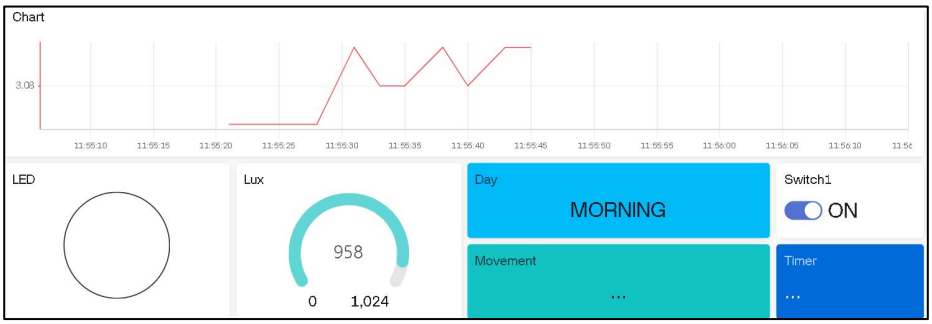
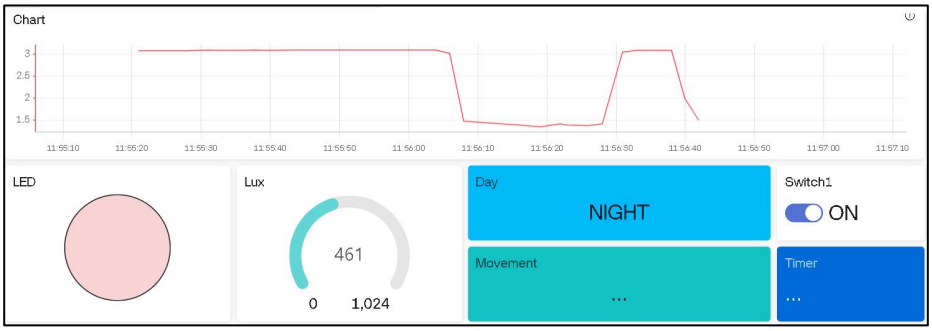
4.0 TESTING RESULTS

4.1 Hardware

MORNING	
NIGHT	

NIGHT (MOVEMENT)	

4.2 Software

MORNING	
NIGHT	

NIGHT (MOVEMENT)	

4.3 Challenged and Solution

Challenge	Cause	Solution
System didn't remember ON/OFF after restart	ESP8266 lost variable state on power cycle	Used Blynk.syncVirtual(V6) to sync button status
Relay module not activating with D6	Relay needs 5V logic, but ESP8266 gives 3.3V	Used relay module compatible with 3.3V or transistor buffer
PIR false triggers	Interference or warm environment	Waited 1-2 mins after powering PIR (stabilizing time)
Countdown blocks loop execution	Using delay(1000) inside for-loop	Acceptable for this project, but can be improved using millis()
LDR readings not accurate in certain light	Environmental lighting inconsistent	Adjusted threshold value (Day = 700) through testing

Summary:

- All main functions (LDR, PIR, LED, Blynk control) work as expected.
- Some minor hardware challenges (relay compatibility, PIR warm-up) were solved with simple adjustments.
- The system is **reliable**, **responsive**, and easy to control using the **Blynk app**.

5.0 CONCLUSION

The *"Intelligent Light with Human Sense"* project successfully achieved its goal of creating an automatic lighting system that responds to both ambient light and human presence. By using an ESP8266 microcontroller, LDR, PIR sensor, and relay module, the system is able to turn on lights only when necessary—specifically at night and when motion is detected—thus saving energy and improving convenience. The integration with the Blynk mobile app allows for remote control, real-time data display, and system status monitoring.

This project demonstrates the practical use of IoT in smart home automation and energy efficiency. All core functions, including light sensing, motion detection, relay control, LED response, and countdown display, worked effectively during testing.

5.1 Future Improvements

Here are some suggestions to improve the system further:

- Add a **real-time clock (RTC)** to schedule system operation based on time.
- Use **millis() instead of delay()** to avoid blocking the main loop during countdown.
- Add **notifications** in the Blynk app when motion is detected.
- Include **adjustable threshold** values (e.g. LDR sensitivity) via Blynk sliders.
- Expand the system to control **multiple lights** or rooms with different sensors.