**Smart Streetlight Controller using LDR**



Session: 2022 – 2026

**Submitted by:**

Saba Shahdin 2022-CS-112

Ariba Shafaqat 2022-CS-113

Wajeeha Javed 2022-CS-136

Afeera Fatima 2022-CS-151

**Submitted to:**

Sir Tehseen

**Course:**

CS-271L Computer Organization and Assembly Language

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Contents

[1 Project Photo: 3](#_Toc155234725)

[2 Description: 3](#_Toc155234726)

[2.1 **AVR Module:** 3](#_Toc155234727)

[2.2 **IOT Module:** 4](#_Toc155234728)

[3 Methodology: 4](#_Toc155234729)

[3.1 Mqtt Broker: 4](#_Toc155234730)

[3.2 AVR Module: 4](#_Toc155234731)

[3.3 Firebase Real-time: 4](#_Toc155234732)

[4 Detailed Explanation: 4](#_Toc155234733)

[4.1 Arduino UNO: 4](#_Toc155234734)

[4.2 LDR (Light Dependent Resistor): 4](#_Toc155234735)

[4.3 ESP32: 5](#_Toc155234736)

[4.4 Mqtt Broker: 5](#_Toc155234737)

[4.5 Firebase: 5](#_Toc155234738)

[5 AVR Module Code: 5](#_Toc155234739)

[6 ESP32 Module Code: 6](#_Toc155234740)

[7 Code Documentation 8](#_Toc155234741)

[7.1 AVR Module Code: 8](#_Toc155234742)

[7.1.1 Macros: 8](#_Toc155234743)

[7.1.2 Analog to Digital Conversion: 8](#_Toc155234744)

[7.1.3 I/O pins Configurations: 8](#_Toc155234745)

[7.1.4 Data from UART: 9](#_Toc155234746)

[7.1.5 Output Logic: 9](#_Toc155234747)

[7.2 IOT Module Code: 9](#_Toc155234748)

[7.2.1 Libraries: 9](#_Toc155234749)

[7.2.2 Connection to Wifi: 9](#_Toc155234753)

[7.2.3 Connection to Mqtt Broker: 9](#_Toc155234754)

[7.2.4 Receiving Publish Data from Mqtt Broker: 9](#_Toc155234755)

[7.2.5 Sending Data to Smart Phone from ESP32: 9](#_Toc155234756)

[7.2.6 Sending Data to Firebase: 9](#_Toc155234757)

[8 Mqtt Dashboard Screenshots: 10](#_Toc155234758)

[9 Firebase Real-time Database Screenshot 10](#_Toc155234759)

[10 GitHub Profile Link: 11](#_Toc155234760)

[11. YouTube link: 11](#_Toc155234761)

[12 References: 12](#_Toc155234762)

[12.1 Introduction to ESP32 12](#_Toc155234763)

[12.2 Introduction to LDR 12](#_Toc155234764)

[12.3 Firebase Working 12](#_Toc155234765)

[12.4 Mqtt Publish and Subscribe 12](#_Toc155234766)

[12.5 UART Communication 12](#_Toc155234767)

[12.6 Arduino Serial Communication 12](#_Toc155234768)

# Project Photo:

Complete Working Setup of Smart StreetLight using LDR

# Description:

## **AVR Module:**

The AVR Module is handled to control the Streetlights automatically based on the day and night. Using LDR module to detect the intensity of the light in the environment in analog form. When the light intensity is very low (it means it is evening) then turn on LED in a row working as streetlights. And when the intensity of the light in the environment is high (it means it is day) then the system should turn off the LEDs. There is also a feature of changing the threshold value to detect the night. That threshold value can be provided by the user through MQTT using his smartphone.

## **IOT Module:**

Using your smartphone, in MQTT Dash app, user publish the threshold value and ESP32 send that value to the Arduino UNO to set the threshold value for night detection. Also get the current value of the LDR sensor from the Arduino and show it in the smartphone. Also send the state of streetlights to your smartphone. Send the values of LDR sensor to firebase real-time database as well.

# 2 Methodology:

## Mqtt Broker:

First we use the Mqtt Broker “Broker.hivemq.com” we connect our smartphone to the mqtt Broker using Mqtt Dashboard. The user publishes the data on the topic and ESP32 receives that data as a threshold from cell phone by subscribing that topic using same Mqtt Broker.

## AVR Module:

The Arduino UNO coded in AVR assembly receives the threshold value form esp32 and when the LDR value is less than the threshold it turns off the light and vice versa.

The AVR code read the analogue the value of LDR and sends it the ESP32 using TX and Rx pins which send data using serial communication using UART protocol.

The ESP32 receives the LDR value and then publish it to the Mqtt Broker and the value is received on the smartphone using Mqtt Broker.

## Firebase Real-time:

The firebase cloud also reads the LDR value and display it in a real time database

# Detailed Explanation:

## Arduino UNO:

The Arduino UNO receives the value of threshold form esp32 and sends the value of LDR sensor. to the esp32 using UART protocol via TX and Rx of Arduino and Esp32.The TX pin of Arduino is connected to RX of ESP32 and the RX pin of Arduino is connected to TX of ESP32.

## LDR (Light Dependent Resistor):

An LDR or light dependent resistor is also known as photo resistor, photocell, photoconductor. It is a one type of resistor whose resistance varies depending on the amount of light falling on its surface. When the light falls on the resistor, then the resistance changes. When the value of LDR is less than the threshold it turns off the light and when it is high it turns on the light.

## ESP32:

**ESP32** is a series of low-cost, low-power system on a chip microcontroller with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 connects to the mqtt Broker and creates connection between smartphones and ESP32 using Mqtt Broker.

## Mqtt Broker:

The Mqtt Broker serves as a server between both the ESP32 and smartphone. The ESP32 subscribe to the topic on Mqtt and receive data published on smartphone as a threshold. The ESP32 sends the LDR value to the smart phone using Mqtt Broker.

## Firebase:

The firebase is a google cloud real-time database that shows the LDR value on the firebase.

# AVR Module Code:

.include "m328pdef.inc"

.include "delay\_Macro.inc"

.include "UART\_Macros.inc"

.include "div\_Macro.inc"

.def A = r16

.def AH = r17

.cseg

.org 0x0000

Serial\_begin ;Begin the Serial Communication

; I/O Pins Configuration

SBI DDRB,5 ; Set PB5 pin for Output to LED

CBI PORTB,5 ; LED OFF

; ADC Configuration

LDI A,0b11000111 ; [ADEN ADSC ADATE ADIF ADIE ADIE ADPS2 ADPS1 ADPS0]

STS ADCSRA,A

LDI A,0b01100000 ; [REFS1 REFS0 ADLAR – MUX3 MUX2 MUX1 MUX0]

STS ADMUX,A ; Select ADC0 (PC0) pin

SBI PORTC,PC0 ; Enable Pull-up Resistor

LDI r18 ,200;

loop:

Serial\_read

CPI r16, 0 ; Check if data received

BREQ skip\_update ; If no data received, skip updating threshold

Mov r18, r16 ; Store the value in register r16 at the address 0x20 in data space

Mov r21 , r16

skip\_update:

LDI r18 , 200

mov r21 , r18

LDS A,ADCSRA ; Start Analog to Digital Conversion

ORI A,(1<<ADSC)

STS ADCSRA,A

wait:

LDS A,ADCSRA ; wait for conversion to complete

sbrc A,ADSC

rjmp wait

LDS A,ADCL ; Must Read ADCL before ADCH

LDS AH,ADCH

delay 100 ; delay 100ms

Serial\_writeReg AH ; Write Data to UART

CP AH, r18; Compare AH (LDR value) with the threshold

BRLO LED\_OFF ; Jump if AH < threshold

; LED ON condition

SBI PORTB, 5 ; Set PB5 (LED) ON

RJMP loop ; Continue looping

LED\_OFF:

; LED OFF condition

CBI PORTB, 5 ; Set PB5 (LED) OFF

RJMP loop ; Continue looping

# ESP32 Module Code:

//Libraries Used

#include <WiFi.h> //Wifi Connection

#include <PubSubClient.h> //Mqtt Publish and Subscribe

#include <ESP32Firebase.h> // Firebase Connection

const char\* ssid = "SSID"; // Wifi Connection

const char\* password = "password";

const char\* mqtt\_server = "Broker.hivemq.com"; // MQTT broker address

const int mqtt\_port = 1883; // MQTT default port

const char\* inTopic = "ldr"; // Topic to subscribe to

const char\* outTopic = "threshold"; //Topic to publish to

#define REFERENCE\_URL "https://ldr-value-fd7db-default-rtdb.asia-southeast1.firebasedatabase.app/" // Your Firebase project reference url // Firebase Project to receive data

Firebase firebase(REFERENCE\_URL); //FireBase Class Object

WiFiClient espClient; //WifiClient Object

PubSubClient client(espClient); // Pubsub Client Object

long currentTime , lastTime;

int count = 0 ;

char messages[50];

int receivedData = 0; //Data to send to Mqtt and ESP32

void connectToWiFi() { //Connection to Wifi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

}}

void reconnect() { // Connect to Mqtt Broker

while (!client.connected()) {

if (client.connect(mqtt\_server )) {

client.subscribe(inTopic);

}

}}

void callback(char\* topic, byte\* payload, unsigned int length) { // Receiving Publish data from Mqtt

String receivedValue = "";

String value = "";

for (int i = 0; i < length; i++) {

value += (char)payload[i];

}}

void setup() { // Main Function

Serial.begin(9600);

connectToWiFi();

client.setServer(mqtt\_server , 1883);

client.setCallback(callback);

}

void loop() { // Loop for functions to run

if (!client.connected()) {

reconnect();

}

client.loop();

if (Serial.available() > 0) { // Continuously sending data to topic

receivedData = Serial.read();

}

currentTime = millis();

if(currentTime - lastTime > 2000)

{

snprintf(messages , 75 , "%ld" , receivedData);

client.publish(outTopic , messages);

firebase.setInt("LDR/ldr", receivedData); // Sendng data to firebase

lastTime = millis();

}}

# Code Documentation

## 6.1 AVR Module Code:

### Macros:

#### UART Macros

To read string from UART protocol using **Serial\_Read** and to send string to UART to Esp32 using **Serial\_Write\_reg** and the **Serial. Begin** to initiate the serial Communication

#### Div Macro

Macros is used in the UART serial communication macros.

#### Delay Macro

To provide delay in the different instruction

### Analog to Digital Conversion:

These lines of code read the analogue value of value and convert them into digital to perform different operations on it.

; ADC Configuration

LDI A,0b11000111; [ADEN ADSC ADATE ADIF ADIE ADIE ADPS2 ADPS1 ADPS0]

STS ADCSRA, A

LDI A,0b01100000 ; [REFS1 REFS0 ADLAR – MUX3 MUX2 MUX1 MUX0]

STS ADMUX, A; Select ADC0 (PC0) pin

SBI PORTC, PC0; Enable Pull-up Resistor

### I/O pins Configurations:

We have the set the pin13 of Arduino UNO to be detected as output. The LED is connected to the Pin13 written as PORTB, 5 in AVR which is initially on and then off and on based on the input it gets.

; I/O Pins Configuration

SBI DDRB,5 ; Set PB5 pin for Output to LED

CBI PORTB,5 ; LED OFF

### Data from UART:

We use the serial read to read the received data from ESP32 and then store it in the register and also write the LDR value to UART.

### Output Logic:

We compare the threshold receive by user and then on the light if it is higher or equal to LDR value and vice versa. The LED is connected to pin 13 of Arduino UNO detected as PORTB, 5 IN AVR code.

## IOT Module Code:

### Libraries:

#### #include <WiFi.h>

To connect ESP32 to the Wi-Fi

#### #include <PubSubClient.h>

To publish and receive data to the Mqtt Broker

#### #include <ESP32Firebase.h>

To connect the ESP32 to the Firebase Real-time Database

### Connection to Wifi:

This function gets the wifi ssid and connect the wifi to the networks to provide the internet facility.

### Connection to Mqtt Broker:

This function gets the mqtt broker id and connect the esp32 to the Mqtt to provide the facility of sending and receiving of data from smartphone.

### Receiving Publish Data from Mqtt Broker:

A call back function to subscribe the topic of Mqtt and then receive that value on the Esp32.

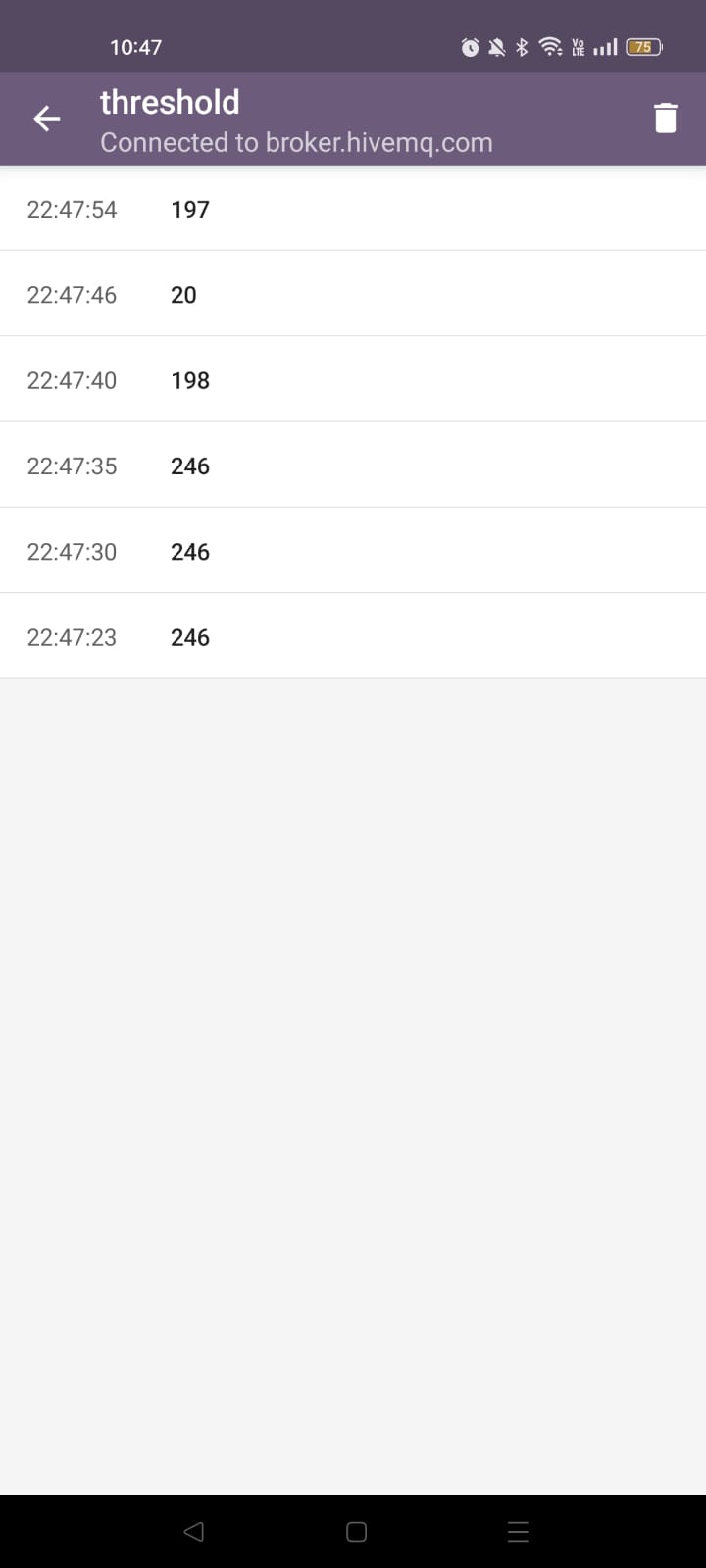
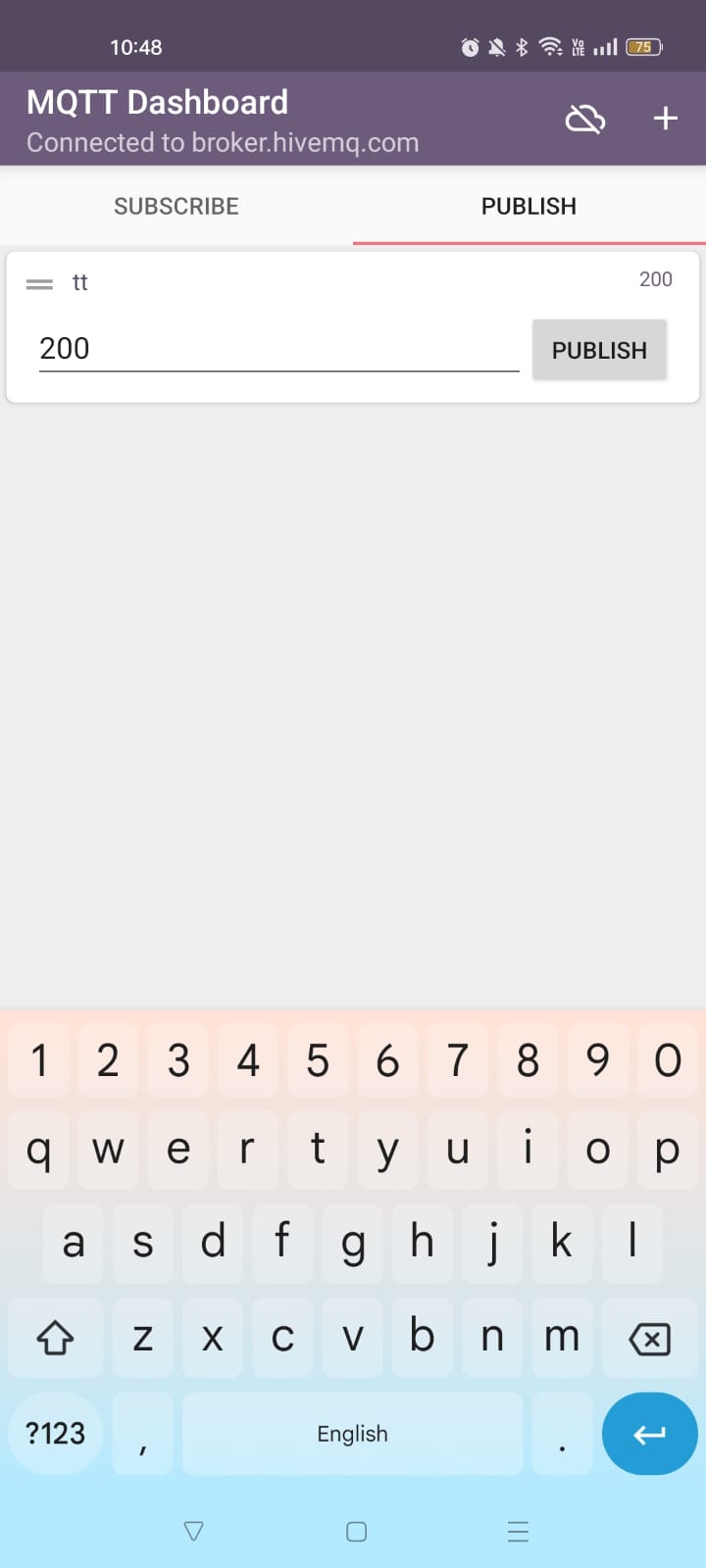
### Sending Data to Smart Phone from ESP32:

A function in loop continuously reads the LDR sensor value and send sit to the Mqtt and the Topic which subscribe it receives that value.

### Sending Data to Firebase:

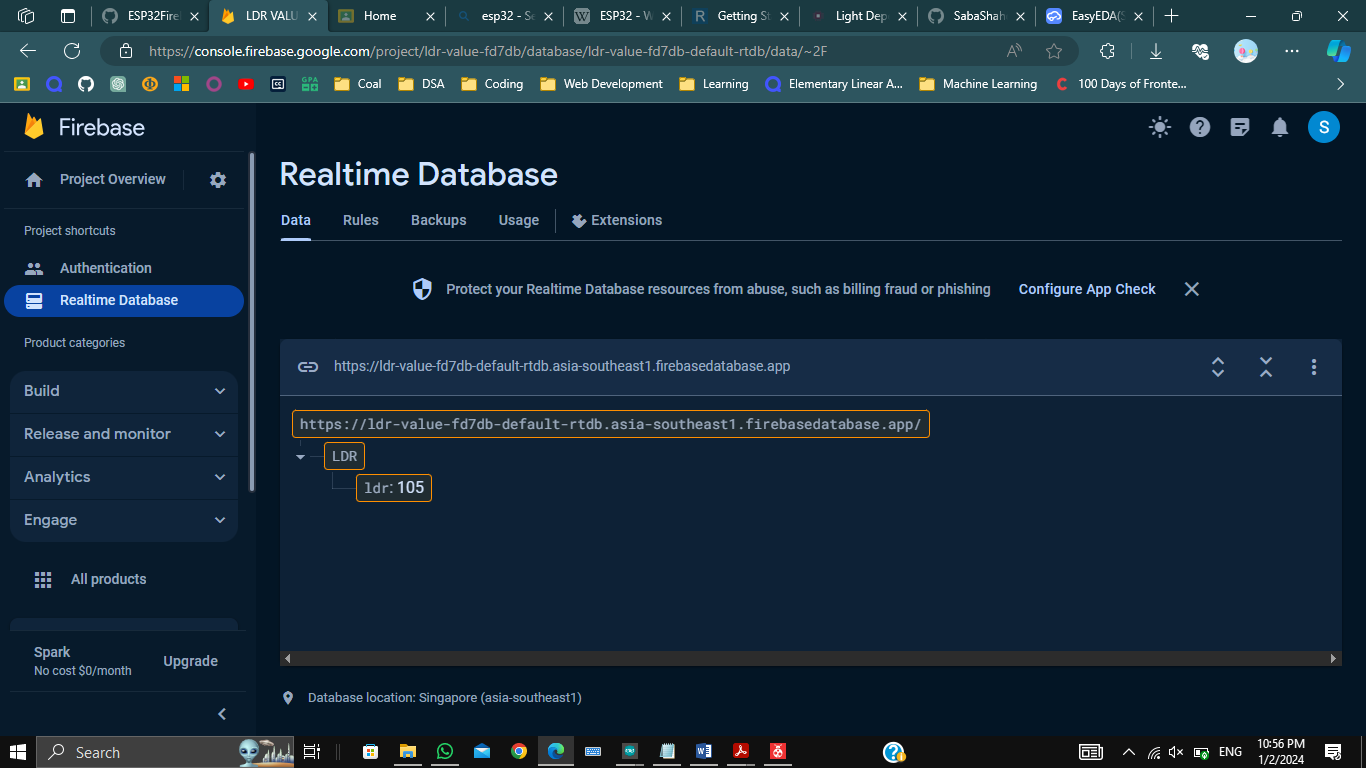
This function sends the LDR value to the real-time database.

# Mqtt Dashboard Screenshots:



Sending data to Esp32 Received Value of LDR

# Firebase Real-time Database Screenshot



# GitHub Profile Link:

[SabaShahdin/COAL (github.com)](https://github.com/SabaShahdin/COAL)

<https://github.com/afferafatima/COAL>

# 11. YouTube link:

<https://youtu.be/vzoEeP14uTc?si=B_nNSwNmAmZkvdiC>

# References:

## Introduction to ESP32

[ESP32 - Wikipedia](https://en.wikipedia.org/wiki/ESP32)

## Introduction to LDR

[Light Dependent Resistor (LDR) - Working Principle and Its Applications (watelectronics.com)](https://www.watelectronics.com/light-dependent-resistor-ldr-with-applications/#google_vignette)

## Firebase Working

<https://www.youtube.com/watch?v=cm-Qe2HMJGk&t=353s>

## Mqtt Publish and Subscribe

<https://www.youtube.com/watch?v=5tG3JXFYrUo>

## UART Communication

<https://youtu.be/TGusjcKSNIU?si=OensL2x-ufn_fFvm>

## Arduino Serial Communication

<https://youtu.be/LubYc87S9tQ?si=Ujh8en-hraFu9pBI>