

COAL Lab Projects - Groups - Section A

Group #	Reg. No	Name	Project Title
G1	2022-CS-08	Noor Fatima	PIR-Based Human Motion Detection and Fan/Light Controlling
	2022-CS-12	Noor Fatima Anwar	
	2022-CS-29	Iqra Tariq	
	2022-CS-152	Saria	
G2	2022-CS-05	Maryam Mueen	Custom Text Display on 1602 LCD
	2022-CS-21	Abdullah Fayyaz	
	2022-CS-35	Hafiz M. Abdullah Mughal	
	2022-CS-42	Kanza tul Islam	
G3	2022-CS-14	Maria Imran	Environment Light Intensity Meter
	2022-CS-22	Ali Tariq	
	2022-CS-46	Syed M. Danish	
	2022-CS-52	Eeman Waheed	
G4	2022-CS-19	Bilal Iftikhar	Smart Watering System for Garden
	2022-CS-44	Syeda Malahat Batool	
G5	2022-CS-13	Adnan	Controlling Character LCD
	2022-CS-17	Ameer Hamza Naseer	
	2022-CS-20	Zubair	
	2022-CS-30	Awais	
G6	2022-CS-06	Abdullah Faseeh	Smart Color Mixer using RGB LED
	2022-CS-07	Harman Iftikhar	
	2022-CS-10	Saboor	
	2022-CS-37	M. Rauf	
G7	2022-CS-27	Shahbaz Ali	Smart Streetlights Controller using LDR
	2022-CS-43	Omer	
	2022-CS-49	Nouman	
	2022-CS-50	Bisma	
G8	2022-CS-11	Asad Ullah	Smart Trashcan using IR Obstacle Detection Sensor
	2022-CS-32	M. Mudassir	
	2022-CS-53	M. Kamran	
	2022-CS-54	Hamza Waseem	
G9	2022-CS-16	Touseef Haider	Flammable Gas Detection and Alarm System
	2022-CS-36	Ahsan Ullah	
	2022-CS-51	Zohaib khan	
G10	2022-CS-02	Junaid	IoT based Warehouse Sensor Network
	2022-CS-04	Ijlal	
	2022-CS-23	Kumail Haider	
	2022-CS-25	Danish	
G11	2022-CS-38	Muskan Awais	IoT Weather Station
	2022-CS-39	Abeer Fatima	
	2022-CS-45	Mahrukh	
G12	2022-CS-01	Saad Malik	Smart Incubator
	2021-CS-10	Usama Mahboob	
	2022-CS-41	Waleed Ahmad	
G13	2022-CS-47	Arsal Gujjar	IoT based Parking System
	2022-CS-48	Khuram Gil	
	2022-CS-248	Afaq	
G14	2022-CS-03	Abdullah Zahid	Home Automation System
	2022-CS-24	Husnain	
	2022-CS-31	Ahmad Khalid	
	2022-CS-40	Zohaib Afzal	

Important Notes:

- AVR Module will be programmed on Arduino UNO (Atmega328p) using AVR Assembly in Microchip Studio.
- IoT Module will be program on ESP32 using C/C++ language in Arduino IDE.
- Use “MQTT Dash” app ([Link](#)) on your smartphone to communicate with the ESP32 using MQTT Protocol.
- Data transferring between Arduino UNO and ESP32 will be done using UART serial communication (the *Tx*, *Rx* pins on Arduino and ESP32).

Project Report Template

Project marks are divided into these sections:

- Complete Working Project Submission = 40 marks
- Project Report both in soft and hard form = 20 marks

The Project Report will contain the following items:

1. Project Title
2. Project Photo (color photo, on a single page)
3. Description
4. Methodology Used
5. Data Flow Diagram (DFD) (draw DFD and Flowcharts using Draw.io [Link](#))
6. Detailed Flow-Chart
7. Circuit diagram (draw circuit diagram using EasyEDA [Link](#))
8. Detailed explanation about the working of each component
9. AVR module code (well formatted)
10. IoT module code (well formatted)
11. Code Documentation
12. MQTT Dash app's dashboard screenshots
13. Project video links of YouTube and LinkedIn (and other social media platforms if any).
14. GitHub link of your project codes.
15. References

Format of the report should be:

- **Font Style:** Times New Roman
- **Default Font Size:** 12
- **Headings Font Size:** 16
- **Pictures/Diagrams:** Align Center

Details and Guidelines for Each Group

Group: G1

Project Title: PIR-Based Human Motion Detection and Fan/Light Controlling

Description

AVR Module: Using PIR motion detection sensor to detect the movement of a person in the room and then turn on an alarm (buzzer) if it is nighttime. Detect the night and day by using LDR module.

IoT Module: In ESP32, get the status of the alarm from Arduino Uno and send it to the Smartphone using MQTT Protocol. Establish communication between the ESP32 and the Arduino using UART Serial communication. Create a button in MQTT Dash app to turn on/off the alarm manually. Also send the day or night status to your smartphone. Send all the values of all sensors to [ThingSpeak](#) cloud as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
Buzzer Module	1	Link
LDR Module	1	Link
PIR Motion Sensor	1	Link

Group: G2

Project Title: Custom Text Display on 1602 LCD

Description

AVR Module: Write assembly program to control a 16x2 LCD module to show count down from 9 to 0. Start displaying the countdown by pressing a push button attached to the Arduino UNO. Also write a function to display the received text from the IoT Module to the 16x2 LCD.

IoT Module: In ESP32, get the custom text from user from the smartphone using MQTT protocol using MQTT Dash app and send that custom text to Arduino UNO (using UART communication) to display that text to the 16x2 LCD. Send all the values of all sensors to firebase realtime database as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Pin Jumper Wires	20	Link
Push Button 12x12mm	2	Link
16x2 LCD	1	Link
Male Header Pins Strip for LCD	1	Link
3.9 Kilo-ohm resistance	2	Link

Group: G3

Project Title: Environment Light Intensity Meter

Description

AVR Module: Write a program to measure the amount of light in the room and then display the results on a 16x2 LCD. If the light intensity is high, then turn on a Green LED. If it is darkness, then turn off the Green LED and Turn on a Red LED.

IoT Module: Send the reading of the LDR module to the smartphone using MQTT protocol. Also write the values to the firebase realtime database. Send the values of LDR sensor to [ThingSpeak](#) cloud as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Pin Jumper Wires	30	Link
LDR Module	5	Link
Red and Green LEDs	2	Link
100-ohm resistance	2	Link
3.9 Kilo-ohm resistance	2	Link
16x2 LCD	1	Link
Male Header Pins Strip for LCD	1	Link

Group: G4

Project Title: Smart Watering System for Garden

Description

AVR Module: Develop a system for automatically watering the plants when their soil has low water. Write assembly program to detect the soil moisture levels using soil moisture sensor then when the level of water is less than a threshold value, then turn on an LED. When the water level in the soil is high then turn off the LED. Display the soil moisture level on the 16x2 LCD screen too.

IoT Module: Send the realtime values of the soil moisture level to the smartphone using MQTT protocol. Also write these values to the firebase realtime database. In the MQTT Dash app on smartphone, create a button to manually ON and OFF the LED. Send the values of soil moisture sensor to [ThingSpeak](#) cloud as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	30	Link
16x2 LCD	1	Link
3.9 Kilo-ohm resistance	2	Link
Male Header Pins Strip for LCD	1	Link
Soil Moisture Sensor	1	Link
RED LED (simulating water pump)	1	Link
100-ohm resistance	2	Link

Group: G5

Project Title: Controlling Character LCD using Smartphone

Description

AVR Module: Write a program to display 4 different messages on 16x2 character LCD. Use 4 push buttons on MQTT Dash app.

By pressing Button#1, LCD will start displaying 4 different messages one after other with 3 second delay between each message. For example, display “Hello World” then after 5 seconds, display “COAL LAB Project” and so on.

By pressing Button#2, LCD will stop displaying the messages and clear the screen.

By pressing Button#3, LCD will start display a blinking message “AVR Assembly” on line 1 and “by Sir. Tehseen” on line 2 of LCD.

Button#4 will toggle (ON/OFF) the backlight of LCD.

IoT Module: Create the 4 buttons in the MQTT Dash app and send 1 to Arduino when we press Button#1, send 2 for Button#2, send 3 for Button#3 and send 4 for Button#4 to Arduino UNO.

Create a 5th Button in the MQTT Dash app to override the default behavior of the LCD and by pressing that on/off button, the LCD will stop showing any text. Send the test to firebase realtime database as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Pin Jumper Wires	20	Link
16x2 LCD	1	Link
3.9 Kilo-ohm resistance	2	Link
Male Header Pins Strip for LCD	1	Link

Group: G6

Project Title: Smart Color Mixer using RGB LED

Description

AVR Module: Write an assembly program to control an RGB LED. It has three LEDs combined in a single LED. It has Red, Green and Blue colored LED. We can control individual LED. Using your smartphone, in the MQTT Dash app, create 5 buttons to control this RGB LED in the following way:

Mode 1: By pressing Button#1, produce yellow color.

Mode 2: By pressing Button#2, produce magenta color.

Mode 3: By pressing Button#3, produce white color.

Mode 4: By pressing Button#4, all the LED will turn ON and OFF in a loop. With delay of 0.5 second between each LED.

Mode 5: By pressing Button#5, all the LEDs will turn OFF.

On each power on, beep the buzzer for 3 seconds and turn on and off all the LED before going to the main logic.

IoT Module: Get the name of color from the user from MQTT Dash app and create the combination of the color in ESP32 then send that command to Arduino UNO to glow the RGB LED to produce the required color. Send feedback to the user in the MQTT Dash app to show the current working mode e.g. Mode 1, Mode 2 etc. Send the current values of all the LEDs to [ThingSpeak](#) cloud as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Buzzer Module	1	Link
Breadboard	1	Link
Pin-to-Pin Jumper Wires	30	Link
RGB LED Module	1	Link
100-ohm resistance	2	Link

Group: G7

Project Title: Smart Streetlights Controller using LDR

Description

AVR Module: Write the assembly program to control the Streetlights automatically based on the day and night. Use 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 some 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 should be an option to change the threshold value to detect the night. That threshold value will be provided by the user through MQTT using his smartphone.

IoT Module: Using your smartphone, in MQTT Dash app, get the threshold value from the user and 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 realtime database as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
White LED (as streetlights)	5	Link
Pin-to-Pin Jumper Wires	20	Link
LDR Module with Analog Output	1	Link
100-ohm resistance	5	Link

Group: G8

Project Title: Smart Trashcan using IR Obstacle Detection Sensor

Description

AVR Module: Write an assembly program to control the trash can lid. It opens the lid automatically when we approach to throw something in the trash can. Then automatically closes the lid. Use Servo motor to open and close the lid. Use IR obstacle detection module to detect the position of hand. Then based on that, open the lid. Use servo motor to open and close the lid of the trash can. Also attach a buzzer module to beep the sound whenever the lid is opened.

IoT Module: Use your smartphone to manually open and close the trash can lid. Also give feedback in the MQTT Dash app about the status of the trash can. Send the status of lid to [ThingSpeak](#) cloud as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
Servo Motor	1	Link
Buzzer Module	1	Link
IR Obstacle Detection Module	1	Link

Group: G9

Project Title: Flammable Gas Detection and Alarm System

Description

AVR Module: Write the assembly program to detect the LPG/Methane gas and then turn on the alarm and turn on a Red LED as the indication of danger. Use gas sensor in analog mode.

Then if the gas vanishes then the system will go to standby mode by turning off the alarm and LED. And turn on a Green LED as an indication of safe environment.

IoT Module: Use MQTT Dash app on your smartphone to manually turn on/off the alarm. Also get the analog values of the gas sensor and show in the app. Also send these values of the gas sensor to firebase realtime database.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
MQ2 Gas Sensor	1	Link
Red and Green LEDs	2	Link
100-ohm resistance	2	Link
Buzzer Module	1	Link

Group: G10

Project Title: IoT based Warehouse Sensor Network

Description

AVR Module: Considering the environment of a warehouse, use the temperature sensor and fire detection sensors to monitor the environment. If there is fire, then trigger the alarm. Turn on an LED if the temperature of the environment is high and turn on the alarm if the temperature crosses a certain threshold.

IoT Module: Using ESP32, get the values of temperature and fire sensor and send the status of the fire to the smartphone using MQTT protocol. Also send the value of current temperature. Also send these values to the firebase realtime database as well. Create a button in MQTT Dash app to manually turn off and on the alarm from your smartphone.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Flame Sensor Module	1	Link
Thermistor Sensor Module	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
Buzzer Module (for Alarm)	1	Link

Group: G11

Project Title: IoT Weather Station

Description

AVR Module: On Arduino UNO, use the LDR module to measure the light intensity of the environment, use Flame detection sensor, MQ2 gas detection sensor, and Water Level Sensor to detect the rain. Attach an LED with the Arduino and if there is rain then turn on the LED, attach another LED which will turn on if there is fire or gas in the environment.

IoT Module: In ESP32, use DHT11 temperature and humidity. Receive the values of all the sensors from the Arduino and send these values to your smartphone on MQTT Dash app using MQTT protocol. Send all the values of all sensors to [ThingSpeak](#) cloud as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
LEDs	3	Link
Flame Sensor Module	1	Link
LDR Module	1	Link
MQ2 Gas Sensor	1	Link
Water Level Sensor (to Detect Rain)	1	Link
DHT-11 Sensor	1	Link

Group: G12

Project Title: Smart Egg Incubator

Description

AVR Module: Use temperature sensor to detect the environmental temperature of an egg incubator and turn on an LED if the temperature is low. Turn off the LED if the temperature is high, this mechanism will maintain the temperature inside the incubator. Also turn on/off an exhaust fan based on the humidity value received from the ESP32. Beep the buzzer 2 times when the exhaust fan turns on. Beep the buzzer 3 times when the fan turns off.

IoT Module: In ESP32, get the value of DHT-11 humidity and temperature from the egg incubator. Send the values of temperature and humidity and the status of fan and LED to [ThingSpeak](#) cloud as well. Send all these values to your smartphone as well using MQTT protocol. Create a button in MQTT Dash app to manually turn on/off the LED and exhaust fan.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
5V Fan	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
DHT-11 Sensor	1	Link
Buzzer Module	1	Link

Group: G13

Project Title: IoT based Parking System

Description

AVR Module: Control traffic lights in such a way that red, yellow, and green lights turn on and off in the same way as the actual traffic lights do. When the traffic light is Green then allow the cars to enter the parking area where there are only 2 parking slots. Attach IR obstacle detection sensors to each slot to detect the presence of the cars. Use another IR obstacle detection sensor to detect the presence of the car on the gate of the parking lot. If all the 2 parking slots are occupied and a 3rd card tries to enter the parking lot, then beep the buzzer, and glow a RED LED to indicate that the parking lot is full otherwise allow the car to park and glow a Green LED while car passes from the parking lot gate. Also count the total number of cars parked in the parking lot since the system powered on.

IoT Module: In ESP32, receive the data about the number of currently parked cars from the Arduino and the counting the total number of cars parked since the system is powered on and send all the values to firebase realtime database and send them in the MQTT Dash app on your smartphone as well.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
Traffic Light LED Module	1	Link
IR Obstacle Detection Sensor	3	Link
Buzzer Module	1	Link
Red and Green LEDs	2	Link

Group: G14

Project Title: Home Automation System

Description

AVR Module: You have to automate a room of a house containing 2 fans and 3 lights. From those 2 fans, attach one fan on a PWM pin of Atmega328p to control the speed of that fan. Receive the on/off command from your smartphone from MQTT Dash app to control the lights and fans. Also detect the intensity of the light in the room using LDR sensor.

IoT Module: In ESP32, communicate with the MQTT Dash app from your smartphone where in the app create 4 buttons to turn on and off the 3 lights and 1 fan individually and create 1 range button to turn on and off the 4th fan and control its speed as well. Send the value of LDR and the status of all the lights and fans to the [ThingSpeak](#) cloud as well. Also send these values to your smartphone on MQTT Dash app using MQTT protocol.

Components Required:

Components	Quantity	Image Link
Arduino UNO with USB Cable	1	Link
ESP32 Development Board with Micro USB Cable	1	Link
Breadboard	1	Link
Pin-to-Hole Jumper Wires	10	Link
Pin-to-Pin Jumper Wires	20	Link
LEDs	5	Link
5V Fan	2	Link
LDR Module	1	Link