

University of Bahrain

College of Information Technology

Department of Computer Engineering

Embedded Systems (ITCE331)

Academic Year: 2022-2023

Design and Implementation of a Traffic Light Control System Using Arduino Uno and FreeRTOS

**Prepared by:**

Student name: Maged Hussain Masaad Ali Aljashoobi ID: 202004484

Student name: Waleed Saleh Ali Saleh ID: 202006448

Student name: Najim Abdulkarem Musaed Alfutini ID: 202003215

Section: 1

Date: 18-5-2023

# Contents:

|  |  |
| --- | --- |
| Topic | Page |
| Cover page | 1 |
| Contents, objectives, questions, and research plan | 2 |
| Introduction and design layout | 3 |
| Table of component | 4 |
| Design | 5 |
| Implementation and code explanation. | 7 |
| Simulation and physical connection | 9 |
| Answering the questions | 10 |
| Conclusion | 11 |

# Objectives:

1. Describe the traffic light system and List components that will be used in the project.
2. Design and implement a functional traffic light using Arduino Uno & FreeRTOS.
3. Simulate the traffic light control system using Summation And implement using physical components.

# Questions:

1. What are the advantages of using Arduino Uno and FreeRTOS in the design of a traffic light control system?
2. What are the challenges and limitations of implementing a traffic light control system using Arduino Uno?
3. What potential future enhancements could be made to the traffic light control system, and how could they be implemented?

# Research plan:

|  |  |  |
| --- | --- | --- |
| **Task** | **Done by** | **Timeline** |
| Define and discuss the project idea. | Waleed | 1-5-2023 |
| Make a list of the project components | Najim | 2-5-2023 |
| Writing the code for the system using FreeRTOS and implementing the design | Waleed Najim Maged | 10-5-2023  To 15-5-2023 |
| Simulate and test the system | Maged | 18-5-2023 |
| Writing the project report | Waleed Maged Najim | 20-5-2023 |

# Introduction:

Traffic light systems are one of the common systems used in our world in these generations. and it is overly critical because it manages society’s roads and prevents any traffic and accidents that can lead to people’s lives in most cases. So, in this project, we designed a simple embedded system of traffic light management that has Three one-way-only roads and three traffic lights (see Figure 1.1). This report presents the design and implementation of a sample traffic lights control system using the Arduino Uno microcontroller and the FreeRTOS operating system. And the report discusses the hardware and software components used in the system and the testing of the system.

We designed it in Arduino uno deploying FreeRTOS. The inputs: an ISR pushbutton. Outputs: LCD that explains the system’s status, a buzzer (to notify that the green light is about to turn off), and LED lights that represent the traffic lights.

# The design layout:

## The design layout (for a better understanding of its function):

* (Road 1) that has one direction to the right side. And it cannot go anywhere else.
* (Road 2) that has one direction to the left side. And it cannot go anywhere else.
* (Road 3) that has one direction to the upside. And cannot go anywhere else.

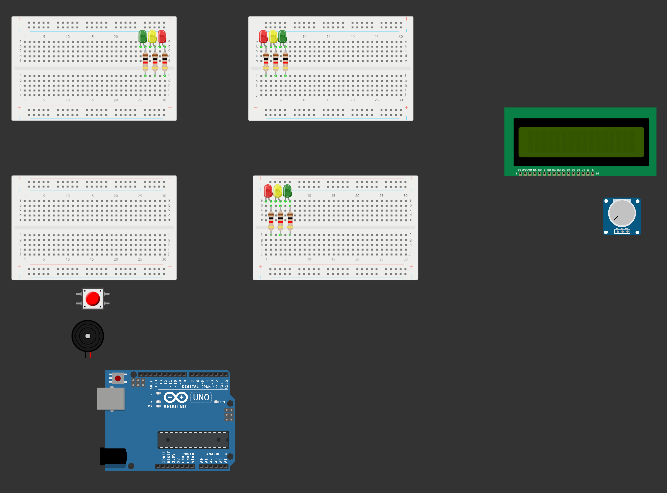
When road 3 traffic light is green, traffic lights 1 & 2 should be red so road 3 cars can drive the other way.

# Table of components:

|  |  |  |
| --- | --- | --- |
| Type | Description | Picture |
| 9 LEDs (3 Red, 3 Yellow, and 3 Green) | representing the traffic light, we have 3 traffic lights. |  |
| 9 Resistor 220 ohm | to limit the current through the LED and to prevent excess current that can burn out the LED |  |
| 1 Push button | When pressed suspending/stopping the traffic light system and pressing it again, it will resume/start the traffic system from where it stopped last time. |  |
| 1 16\*2 LCD | It will show the status of each traffic light and will count down how many seconds are left till the traffic switches red. |  |
| 1 Active buzzer | Will output sound multiple times along with the green light blinking. |  |
| 1 potentiometer | A potentiometer is an electronic component that allows you to vary the resistance in a circuit and the LCD is used to adjust the contrast of the display |  |
| 3 Breadboard | A breadboard is a reusable solderless device used to build and test electronic circuits. |  |
| Arduino Uno | Arduino Uno is a microcontroller board based on the ATmega328P microcontroller |  |

# Design:

* The Circuit Diagram before connections:



* Table of pins connection for the Arduino:

|  |  |
| --- | --- |
| Pin 2 | To the pushbutton pin 1 in Side A |
| Pin 4 | To Red LED of traffic 1 & 2 |
| Pin 5 | To Yellow LED of traffic 1 & 2 |
| Pin 6 | To Green LED of traffic 1 & 2 |
| Pin 7 | To Red LED of traffic 3 |
| Pin 8 | To Yellow LED of traffic 3 |
| Pin 9 | To Green LED of traffic 3 |
| Pin 10 | To the buzzer |
| A5 | To D7 in the LCD |
| A4 | To D6 in the LCD |
| A3 | To D5 in the LCD |
| A2 | To D4 in the LCD |
| A1 | To E in the LCD |
| A0 | To RS in the LCD |

* For Lcd:

Connect K, RW, and VCC to the ground.

Connect VDD and A to the power.

Connect V0 to the potentiometer.

* For push button:

Connect pin 2 in side B to the ground.

* For buzzer:

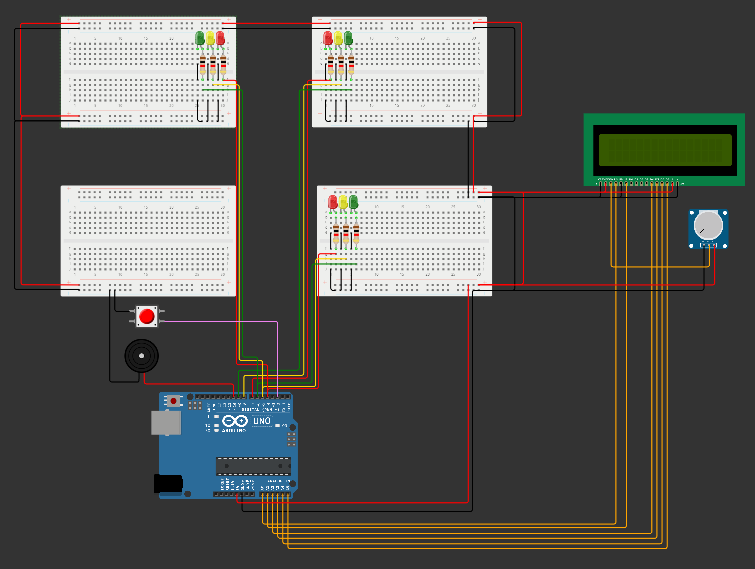
Connect the G pin to the ground.

* For potentiometer:

Connect the pin Vcc to the power.

Connect the GND to the ground.

* To power the Arduino we can use a USB connection or external power source like 9V Battery.
* The Circuit Diagram After the connections:



# Implementation and code explanation :

* The purpose of this code is to control a traffic light system using FreeRTOS and Arduino. The system consists of two traffic lights (1 and 2) and a third traffic light (3) that alternate between different states (red, yellow, and green). The system also includes a push button interrupt that can be used to stop and continue the traffic light tasks, and an LCD to show the status of the traffic lights.
* The code defines two task functions for traffic lights, traffic1\_2Task() and traffic3Task(), which alternate between different states to control the traffic lights. Each task uses semaphores to synchronize with the other task and to ensure that only one task is executed at a time. The tasks also use delay functions to control the timing of the traffic light states.
* The traffic1\_2Task() function controls traffic lights 1 and 2, and alternates between the following states:
  1. Red light for 1 second
  2. Yellow light for 1 second
  3. Green light (by default 5 seconds, we can change this by WaitingTime global variable)
  4. Blink green light 3 times each 400ms
  5. Yellow light for 1 second
  6. Red light for 0.5 second
  7. Give semaphore to traffic3Task()
* The traffic3Task() function controls traffic light 3, and alternates between the following states:

1. Red light for 1 second
2. Yellow light for 1 second
3. Green light (by default 5 seconds, we can change this by WaitingTime global variable)
4. Blink green light 3 times each 400ms
5. Yellow light for 1 second
6. Red light for 0.5 second
7. Give semaphore to traffic1\_2Task()

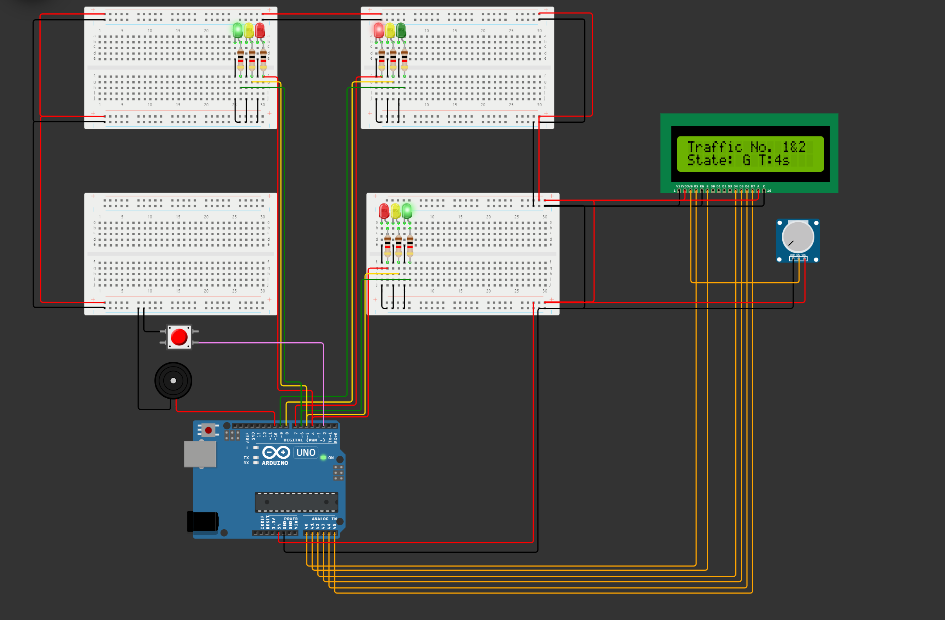
* The code also defines an interrupt function, button\_ISR(), which gives a semaphore to the InterruptTask that is triggered by the push button. When the button is pressed, the interrupt suspends the traffic1\_2Task() and traffic3Task() functions, and when the button is pressed again, the interrupt resumes the tasks.
* The LCD shows the current state of the traffic lights, as well as the status of the interrupt function. The display is updated in real-time using the LiquidCrystal library.
* Now here is the full code with comments for explanation:



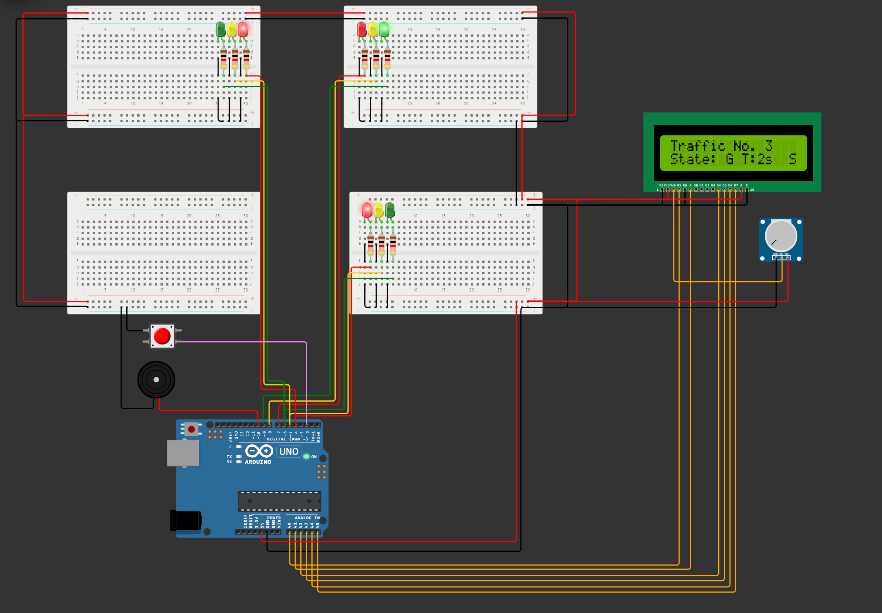
# Simulation and physical connection:

We use <https://wokwi.com> website for simulation, and here are some figures for the simulation:

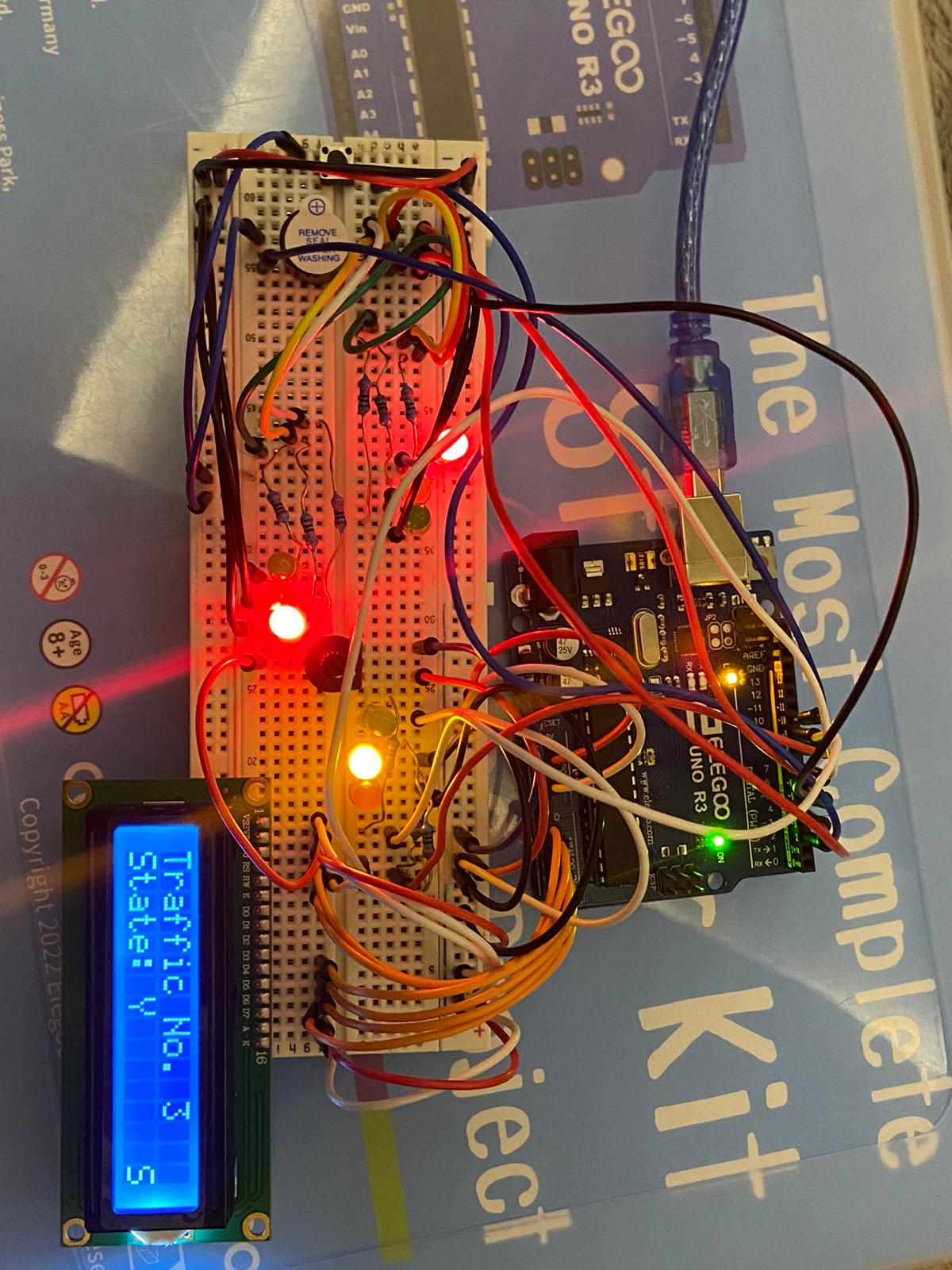
This is the like for our Project: [**https://wokwi.com/projects/364904810601550849**](https://wokwi.com/projects/364904810601550849)



* In this figure the LCD screen show to us that traffic 1&2 is executed and shows the state of the traffic (G: green, Y: yellow, R: reed) and the timer for the green state (T:4s)



* In this figure, the current traffic run is traffic 3 and the current state is G(Green), and S letters display on the LCD screen means the push button was pressed and the system stopped we must press the push button again to make the system continue operation.
* After we make sure that the system works well in the simulation, we will now use the physical component to implement the system:



# Answering The Question:

1. What are the advantages of using Arduino Uno and FreeRTOS in the design of a traffic light control system?

The advantages of using Arduino Uno are a powerful and flexible platform for developing embedded systems like traffic lights system. And FreeRTOS is suitable for synchronization between tasks and interrupts and resource handling.

1. What are the challenges and limitations of implementing a traffic light control system using Arduino Uno?

The limitations that we faced in this project were that if we want to add more traffic lights we need more pins that we have in the Arduino, and we cannot add more than two interrupts in Arduino Uno because there are only two pins for interrupt in Arduino Uno Pin2 and Pin3.

1. What are some potential future enhancements that could be made to the traffic light control system, and how could they be implemented?

We can use an 8-bit shift register (74HC595) or multiplexer for using more pins in Arduino Uno or we can use Arduino Mega to use more pins and interrupts.

# Conclusion:

In conclusion, the objectives of this project were to design and implement a functional traffic light system using Arduino Uno and FreeRTOS, and to simulate the system using Summation software. Firstly, we described the traffic light system and listed the components that will be used in the project, including the Arduino Uno board, LEDs, a buzzer, a push button, and an LCD display. Then, we designed and implemented the traffic light system using FreeRTOS and the Arduino Uno board. The system included two main tasks to control the traffic lights for three roads, as well as a push button interrupt and an LCD display for status updates.

Finally, we simulated the traffic light control system using the wokwi.com website. The simulation allowed us to test and refine the system's functionality before deploying it to the physical components. In summary, we achieve all the objectives and we designed the traffic lights control system successfully by using FreeRTOS and Arduino Uno which can be used in real-world scenarios. This demonstrates the potential for embedded systems in practical applications.