# **TechUp Final Project Documentation**

**Project Title:** Traffic Light Control System with Pedestrian Crossing  
 **Author:** KWIHANGANA Enock (Group 5)  
 **Date:** September 2025

## **1. Objective**

The goal of this project is to simulate a **realistic traffic light sequence** with pedestrian crossing control using an **Arduino Uno**. The system cycles through standard traffic lights (Red → Yellow → Green) and allows pedestrians to safely cross when they press a push button. An LCD display provides visual feedback by showing **“WAIT”** or **“WALK”** messages.

## **2. Components Used**

* **Arduino Uno (1x)** – microcontroller to run the logic
* **LEDs (3x)** – Red, Yellow, Green for vehicle signals
* **Push Button (1x)** – pedestrian request button
* **16x2 I²C LCD (1x)** – display messages “WAIT” or “WALK”
* **Resistors (220Ω)** – current limiting for LEDs
* **Breadboard & Jumper Wires** – prototyping connections
* **Power Source** – Arduino USB or 9V adapter

## **3. System Workflow**

1. **Default cycle**:  
   * Green LED ON → Yellow LED ON → Red LED ON (repeat).
   * LCD shows **“WAIT”** for pedestrians.
2. **Pedestrian request**:  
   * When the button is pressed, the system detects the request.
   * Traffic completes its current cycle safely.
   * System switches to **Pedestrian Mode**.
3. **Pedestrian Mode**:  
   * Red LED ON (traffic stopped).
   * LCD shows **“WALK”**.
   * Pedestrians have ~5 seconds to cross.
4. **Resume cycle**:  
   * Pedestrian request is cleared.
   * System returns to the normal RED → GREEN → YELLOW cycle.

## **4. Block Diagram**

📌 See Diagrams/BlockDiagram.png

## **5. Circuit Diagram**

📌 See Simulation/Tinkercad\_Circuit.png

## **6. Arduino Code**

📌 Main source file: Code/TrafficLight\_Control.ino

## **7. Simulation Results**

* **Green phase:** Green LED ON, LCD = “WAIT”
* **Yellow phase:** Yellow LED ON, LCD = “WAIT”
* **Red phase:** Red LED ON, LCD = “WAIT”
* **Pedestrian phase:** Red LED ON, LCD = “WALK”  
   📌 See Simulation/SimulationResults.png

## **8. Reflection / Challenges Faced**

* **Tinkercad limitation:** Tinkercad does not support external I²C LCD libraries (LiquidCrystal\_I2C or Adafruit\_LiquidCrystal), making it impossible to simulate the LCD screen directly. We overcame this by testing the LCD code on **real hardware** and using placeholders in Tinkercad.
* **Non-blocking logic:** The initial implementation used delay(), which blocked the program and made the system unresponsive to button presses. This was solved by switching to a **state machine with millis()**, making the system responsive and professional.
* **Hardware timing:** Balancing traffic signal timing and pedestrian crossing duration required fine-tuning. Too short walk times were unsafe, too long blocked traffic unnecessarily.

## **9. Conclusion**

This project successfully demonstrates how **sequential control, timing, and human interaction** can be integrated into an Arduino IoT system. The use of **state machine logic, push button input, and LCD output** makes the system interactive and realistic.