System Technology Report

Toby Aldred

Contents

1.0 Introduction 2

2.0 Relevant Background Information 3

3.0 Methodology 5

4.0 Final design solution 8

5.0 Conclusion 11

6.0 References 12

7.0 Appendix 12

1.0 Introduction

1.1 The problem: Current urban traffic infrastructure heavily relies on traditional traffic management systems. These systems are typically constructed using simple timers that operate independently of real-world variables. This creates systemic inefficiencies, leading to unnecessary congestion during lowtraffic periods. Furthermore, static systems lack responsiveness to environmental hazards, which increases fuel consumption, elevates carbon emissions, and reduces overall road safety.

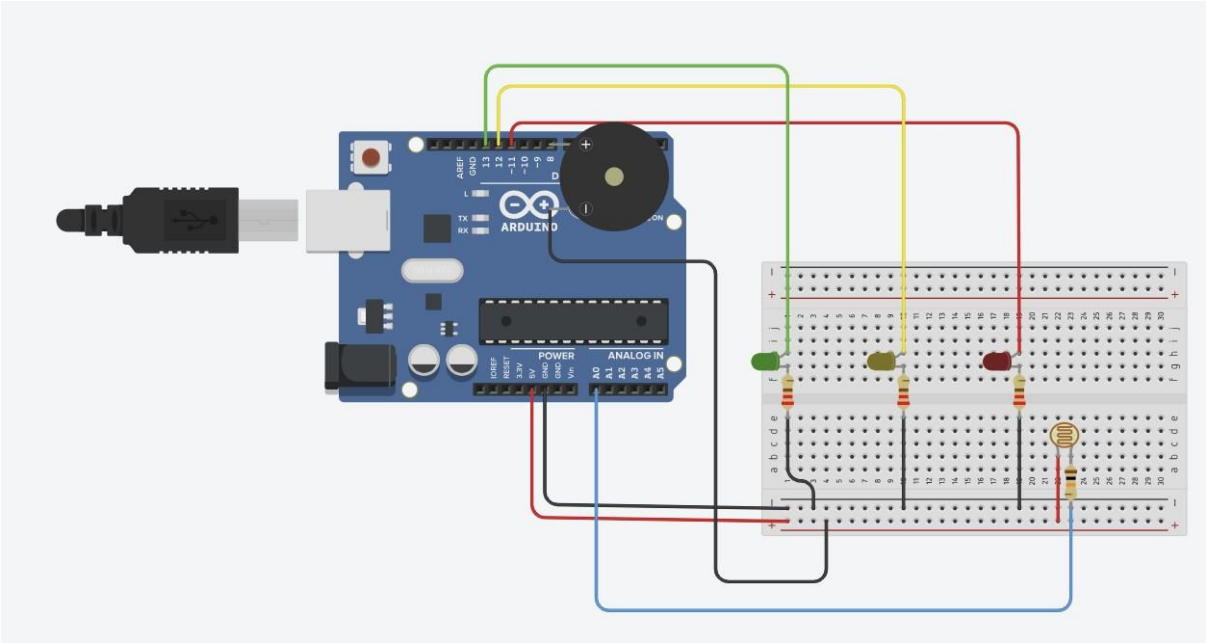


Figure 1 Description: *Full hardware architecture of the single Intelligent Traffic Light Node, featuring the Arduino Uno, integrated sensor array, and dual-mode actuators.*

1.2 The objective: The primary aim of this project is to design and implement an Intelligent Traffic Light Controller (ITLC) system. To achieve this, an Arduino Uno Microprocessor is utilised as the central control unit (simulated via Tinkercad). The system integrates light dependent resistors to sense ambient conditions and adjust the traffic light system signals in a logical way. This design prioritises efficiency through data driven timing and safety through buzzers (Piezo) for pedestrians to know when to safely cross.

1.3 The scale: The project covers a single embedded node (The built singular example) and a network of 10 nodes (Theoretical city network of multiple systems).

1.4 Project Scale: Node and Network

The report encapsulates the details and development of the system across two scales. While traditional prototypes focus on a single 4-way junction using multiple LED arrays, this project prioritises the development of a Singular Intelligent Node. This approach allows for deeper exploration of sensor integrations (LDR) and multi-modal actuators (Buzzer/LEDs). The single node

serves as the functional blueprint for the theoretical 10-node Small Area Network (SAN), demonstrating how localised intelligence can be scaled across mesh topology

- **Embedded implementation**: The embedded system represents the design of a singular working node in the designed system, to simulate this I have used Tinkercad to demonstrate the hardware interface and the logical operations of the system.

- **Network Architecture**: The network architecture demonstrates the theoretical application of the system being implemented across a small 10 node Small Area Network (SAN). Which explores how multiple intelligent nodes communicate across a city-wide topology to optimise traffic flow efficiency through communication and shared data.

2.0 Relevant Background Information

2.1 The operating environment: The proposed system is designed for implementation in high-density urban junctions. These environments are characterised by fluctuating traffic volumes and varying environmental conditions, including fog, heavy rain, and low visibility during nighttime operations. Additionally, the system must also operate 24/7 and provide clear systems to motorists and pedestrians, particularly in busy areas with high foot traffic and school zone proximity.

2.2 Design Constraints: Developing an intelligent system requires working with technical and safety limitations, which can real world effects.

- **Power Consumption:** In a large-scale city network, maximising power usage and efficiency is crucial, using low power microprocessors and LEDs. The technology makes sure that the system is restricted to low power consumption.
- **Response Time:** The system must process data gathered from sensors (LDR) and update the traffic light states in real time to prevent accidents.
- **Fail Safe Technology:** In case of a network failure each embedded node will have a default mode that consists of a standard timing system to ensure that in result of network failure it does not lead to an uncontrolled traffic system rather resorts to the default timed technology acting as a failsafe.
- **Accessibility:** Legal standards require that traffic systems accommodate users with various disabilities, to ensure that we follow these standards all the systems have mandatory auditory feedback.
- **Safety Overrides:** The system must prioritise emergency signals over the standard timing loops to reduce response times for emergency services.
- **Network synchronisation:** In a 10-node small area network (SAN), if one node detects an ambulance, it must broadcast this to the second node on the network, so by the next junction the light is already green by the time the emergency service vehicle arrives.

- **Accessibility for Auditory impairment:** While the piezo buzzer provides auditory feedback, the system is design with a multi sensor redundancy. Allowing for pedestrians with hearing loss to visually see that the red light shown for motorists indicates a safe stop. which acts as the primary cross signal. For future improvements of the single embedded nodes, they may incorporate a tactile actuator (a spinning physical object) for physical sensory – ensuring that the SAN remains inclusive for all users including those with different forms of sensory loss.
- **Automated pedestrian detection:** current traffic management systems contain a physical button as a manual form of input to trigger pedestrian crossing phase. This embedded system design utilises the LDR sensor to prioritises automated environmental response. Which ensures the system reacts to real time variables without requiring manual intervention. **2.3 Comparison of Systems**

System Type	Characteristics of the system	Difference compared to my system
Fixed Timers	Reliable, simple, Boolean values (on/off)	They are static and cannot see the environment, my design uses LDRs to adjust based on light levels
Inductive Loops	Detect cars using metal sensors in the road	They are expensive and require digging up roads – additionally make road maintenance more difficult. My design is a modular embedded solution that is easy to install and maintain
Smart Centralised Lights	Large scale Artificial intelligence control systems	Dependent on constant high speed internet link – affecting latency. My design uses small area networks allowing local nodes to make decisions independently rather than on a joint server

2.4 Emergency vehicle pre-emption (EVP): Enhances public safety, the design includes a priority logic sub system in the embedded nodes for emergency services (ambulances, fire services, and police)

2.4.1 How it works.

The Sensor: The system makes use of the LDR (Light dependent resistor) not only for the night, but additionally to detect the specific high intensity strobe patterns of emergency service vehicles.

The Logic: When the LDR detects a rapid light level fluctuation (emergency vehicle strobe lights) the Arduino triggers an interrupt in the code.

The action once detected:

1. Immediate transition: The current green light, immediately moves to amber then red.
2. The lane where the emergency vehicle is detected turns to green.
3. Audio warning: The buzzer emits a unique high frequency tone to warn pedestrians that there is an emergency vehicle approaching at speed.

3.0 Methodology

3.1 Hardware Interface and Component Selection

- **The Microprocessor:** an Arduino Uno is used as the centralised control unit due to its robust input output capabilities(I/O) and suitability for prototyping embedded system logic due to its "plug and go" USB interface. The microprocessor acts like the brain of the system.
- **Visual Signals (Output):** The system is made up of Three LEDS (Red, Amber, Green) which are interfaced via digital pins 13,12,11 on the Arduino Microprocessor board.

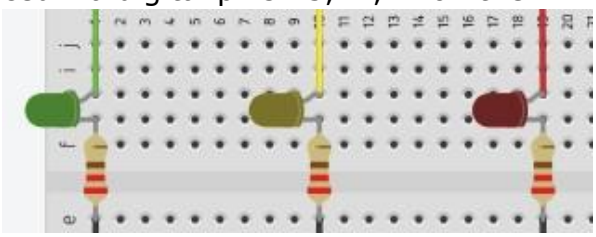




Figure 2 Description: *Digital Output Sub-system: Wiring configuration of the Red, Amber, and Green LEDs with 220Ω current-limiting resistors to protect the MCU I/O pins.*

- **Current Limiting Resistors:** Each LED is paired with a 220Ω resistor. The LEDs were paired with this chosen value to ensure that the current stays within the safe operating range (20ma) which prevents damage to the LED bulbs and the Arduino pins.
- **Auditory Alert (Output):** A piezo Buzzer (Anode [+]) is connected to digital pin 8 on the Arduino board. Providing audio feedback for pedestrian safety and emergency service vehicle alerts

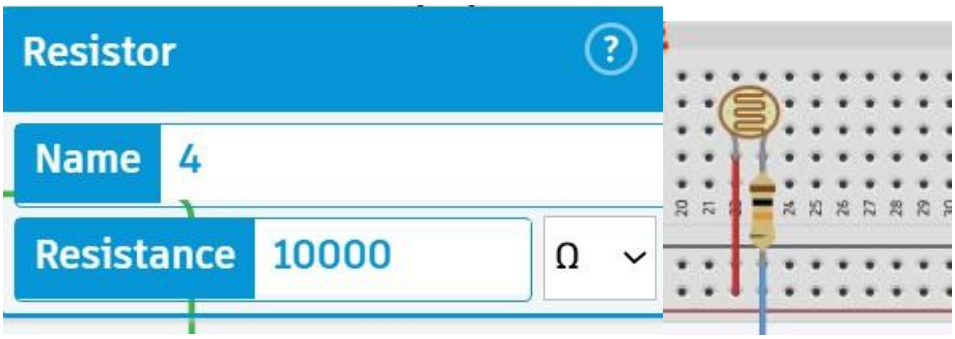


Figure 3 Description: *Analog Input Sub-system: LDR sensor configured as a voltage divider with a 10kΩ resistor to facilitate environment-aware logic processing.*

- **Intelligence Sensor (Input):** A light dependent resistor (LDR) is connected to Analog pin A0 Located on the Arduino Microprocessor. It uses a 10,000Ω pull down resistor to create a divide in the voltage circuit. This type of configuration is necessary to translate the changing resistance of the LDR into a readable voltage signal for the microprocessor itself.

3.1.2 Power distribution and Grounding

The design of the embedded system utilises a common grounding architecture consisting of all three 220Ω current limiting resistors and a 10k Ω pull down resistor, all of which are connected to the negative rail on the breadboard, which is linked by the Arduino GND pin. This architecture ensures that a complete return path is present for the current. The LDR is based by the 5V positive rail (breadboard) to maintain the integrity of the analog signal at pin A0.

3.2 System Operations and Logic

During the testing phase of the single node system, a signal issue was identified where the serial monitor reported a constant value of 1023 at pin A0 after debugging to find why the system was not working as planned. Through this process it was identified that the sensor was defaulting to a maximum brightness in the simulation. After this the system was successfully fixed by adjusting the strobe threshold logic to account for the environmental variables. This ensured that the EVP was only triggered under high intensity light fluctuation. The system operations and logic are as follows:

- **Ambient Light Processor:** This system samples the light level from Pin A0 (Located on Arduino). If the values indicate nighttime, the system enters an energy saving mode, for minimal energy consumption, leading to the LEDs to be dimmed and the cycle lengths to be adjusted to changes in the surroundings.
- **Emergency vehicle Pre-emption (EVP):** The light dependent resistor (LDR) is programmed to detect high intensity strobe light patterns (given off from emergency vehicles). Upon detection the microprocessors trigger the priority override feature. Immediately transitioning cross traffic to red and granting the emergency vehicle lane a green signal.

- **Pedestrian accessibility:** The buzzer is synchronised with the green signal phase. During the Emergency Vehicle Pre-emption (EVP), the buzzer frequency increases to a unique frequency to warn pedestrians of an approaching high-speed vehicle.

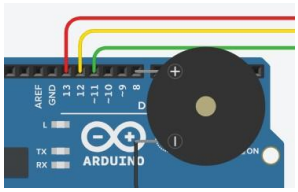


Figure 4 Description: *Piezo buzzer (Auditory actuator) highlights the connection to Digital Pin 8, enabling the system to deliver synchronized auditory signals for pedestrian safety and highfrequency alerts during Emergency Vehicle Pre-emption (EVP) phases.*

- **Integrated safety cycle:** The piezo buzzer is programmed to synchronise with the red LED phase. This ensures that the auditory walk signal only activates when the traffic flow is physically stopped. Providing users with a fail-safe environment.
- **Emergency Visual Warning:** During an EVP phase the transition between red and yellow LEDs provide a high visibility hazard warning for both motorists and pedestrians, who may not have heard the urgent alarm or have difficulty hearing.

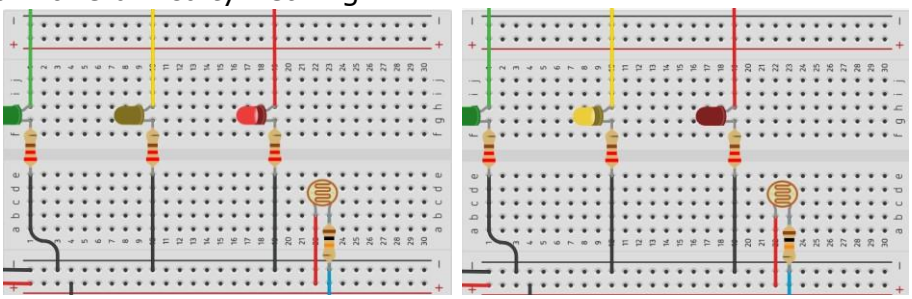


Figure 5 Description :
State A (Left): *The Red LED is active, signaling an immediate halt to standard traffic flow.*
State B (Right): *The system transitions to a Yellow "Caution" state. These two states alternate rapidly alongside the high-frequency buzzer to create a high-visibility warning. This ensures that pedestrians who are deaf or hard of hearing receive a clear visual signal that an emergency vehicle is present. Even if they cannot hear the auditory alarm.*

3.3 Theoretical Scalability

While the methodology focuses on a single node system, it is important to mention that the code and the systems are designed to be modular. This allows the single nodes data to be packaged and sent across a small area network (SAN) to the other 9 nodes in the system.

b

4.0 Final design solution

This section presents the final design of the intelligent traffic light node and provides the architectural framework that would be required to scale the system to a 10-node small area network (SAN).

The single embedded node which is simulated in Tinkercad forms the building blocks of the theoretical capabilities and application to a network of these systems (10 nodes). The configuration of the hardware ensures that all the systems are individually 'intelligent' and capable of local independent decision making, with system failure resorting to a preprogrammed timer to ensure traffic does not go unmanaged.

4.1.1 Single Node system Verification

The physical implementation of the single intelligent node system, evidenced in the Tinkercad simulation, successfully integrates all requirements and features into one unified control unit.

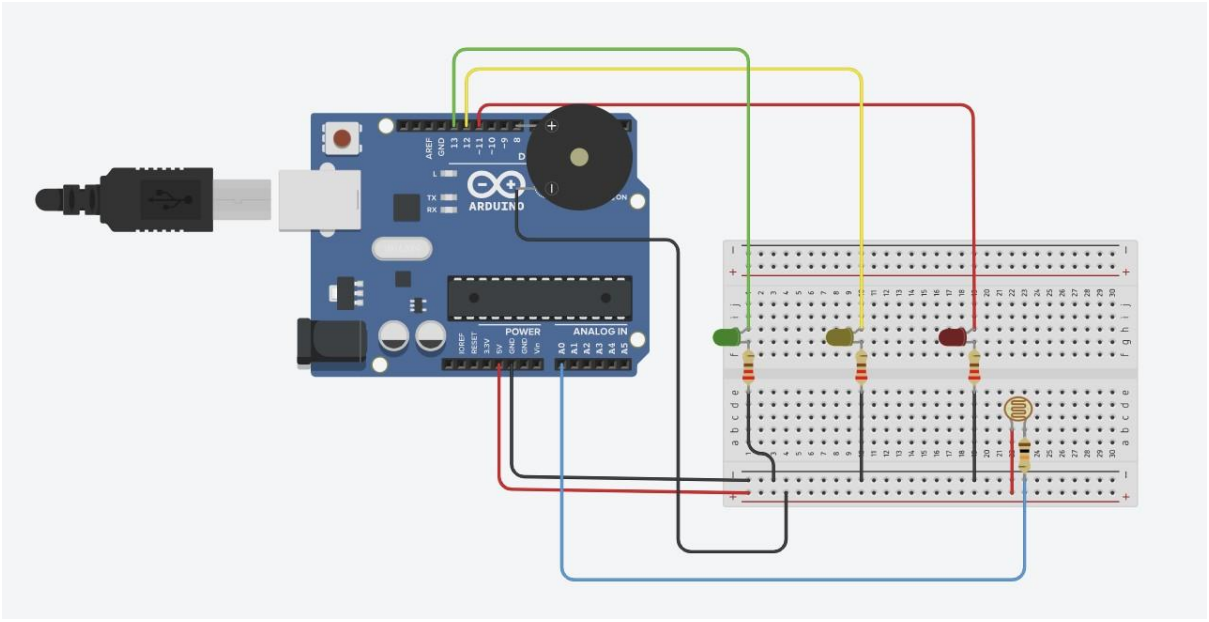


Figure 6 Description *Final Integrated Design Solution: Demonstrating the full integration of visual, auditory, and environmental sensing capabilities.*

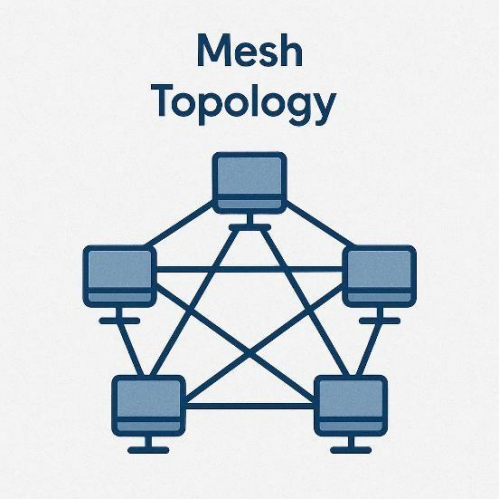
- **Component:** The system utilises the Arduino uno to process the analogy data from the LDR that triggers digital outputs via the array of LEDS and the piezo buzzer
- **Electrical integrity:** all components are protected by 220Ω resistors to avoid damaging components of the circuit. Meanwhile the analogy input utilises a 10,000Ω pull down resistor to maintain a stable voltage, ensuring that the system operates safely and to mitigate any potential risks or hazards that may come from not having a voltage divider.
- **Intelligence:** The hardware can perform multiple tasks, including environmental sensing for night mode, and a high frequency strobe detection aspect for the Emergency Vehicle pre-emption (EVP)

4.1.2 Hardware Signal Integrity

During the testing of the single node, a continuous issue was identified. That issue being where the amber and green LEDs failed to activate during the cycle. A test of the system confirmed that the red LED was correctly wired to the system, after reviewing the simulation of the node, it was identified that the remaining LEDs required realignment of the jumper leads to ensure that the circuit path for the LEDs and the digital pins were complete. Once this was corrected the full three phase traffic cycle was verified through the simulation of nodes visual feedback.

4.2 Small Area Network Design (SAN)

4.2.1 Proposed Topology: Mesh Topology



For the 10-node network, a Mesh topology is proposed. This is because in a traffic management environment, redundancy is crucial, unlike a star topology where a central hub failure would disable all 10 junctions, the Mesh topology allows all 10 nodes to communicate through multiple paths.

If one node fails then the EVP – Emergency vehicle pre-emption signal across the nodes can move to the other nodes on the network to reach its destination, which maintains the safe green corridor for the emergency vehicles.

4.2.2 Addressing scheme:

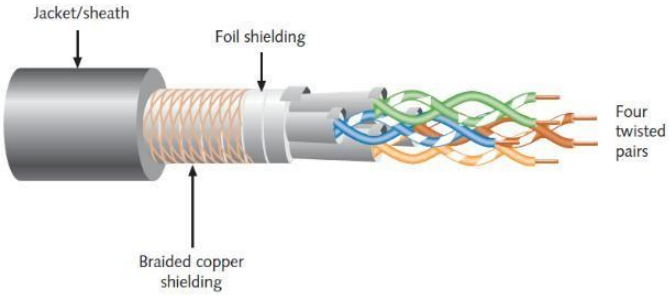
To accommodate the organised data exchange between nodes on the network and packet collision, the implementation of a static IPV4 addressing scheme will be implemented, allowing every node covering the junctions to be uniquely identified within the small area network (SAN). An example of this is shown below:

Node ID	**Physical Junction**	**Assigned IP Address**	**Subnet Mask**
Node 01	Train Station Junction	192.168.1.2	255.255.255.0
Node 02	Bus Station Junction	192.168.1.3	255.255.255.0
Node 03	Box Junction	192.168.1.4	255.255.255.0
Node 04	4-way Junction	192.168.1.5	255.255.255.0
…	…	…	…
Node 10	Roundabout Traffic Lights	192.168.1.254	255.255.255.0

(The assigned IPv4 address cannot feature 0 in the final octet, as this identifies the network itself. Additionally, the final octet cannot be 255, which represents the broadcast address. By convention, 1 is reserved for the network's default gateway)

4.2.3 Communication Media

Proposed media: Shield Twisted Pair (STP) – Industrial Grade



The Shield Twisted Pair (STP) would be the best ethernet medium for the core backbone of the system. This is since wired communication provides low latency which is required for real time safety overrides and emergency signals which would be used for the EVP feature. To guarantee low latency and minimal jitter during an Emergency Vehicle Preemption (EVP), the network utilizes strict QoS protocols. High-priority EVP packets are given maximum throughput over standard traffic status updates

Considerable Factors:

- **Electromagnetic Interference:** Urban environments often contain high voltage powerlines that could corrupt data signals between nodes. The use of the proposed media (STP) provides the network of nodes the essential shielding to maintain the signal integrity across the network.
- **Network Latency:** High volumes of data traffic could delay the crucial EVP Packets; to mitigate this risk, the system will include a priority-based protocol, this is used to ensure that the emergency interrupts from the EVP are processed before standard status updates are.

4.2.4 Design Justifications

- **Uniformed:** All 10 nodes utilise the same identical Arduino design shown previously in figure 1 to ensure seamless data transmission and compatibility across the network.
- **Fail Safe autonomous:** each node of the traffic light network possesses enough local processing power to revert to a default timer if network connection is lost to ensure that traffic never goes unmanaged if there is a link failure in the network.
- **Decentralised system logic:** the design of the network justifies the SAN and the mesh topology rather than a centralised server; this is because it eliminates the risk of single point failure affecting the functionality of the whole network. Furthermore, using the small area network approach reduces bandwidth costs associated with constant cloud service connectivity.

5.0 Conclusion

The aim of the project was to design and implement an intelligent traffic light controller (ITLC) system to address the inefficiency of traditional traffic light systems; set static timing. With Tinkercad and the successful simulation of a single embedded node the theoretical proposal of a 10-node small area network of these embedded systems has demonstrated a viable solution for a modern traffic light control system infrastructure.

The fully developed single node embedded system successfully integrates environmental sensing and auditory feedback to improve efficiency and safety with:

- **System intelligence logic**: The system utilises an Arduino uno and an LDR, meaning that the embedded system can dynamically adjust to light conditions and detect emergency vehicle strobe patterns.
- **Safety and Accessibility**: The implementation of a piezo buzzer ensures compliance with legal accessibility standards and provides critical crucial warnings during an EVP phase.
- **Precision**: The methodology verified that using specific components like the 220Ω and 10,000Ω resistors ensure that the hardware remains safe and within safe operation limits while maintaining the systems signal integrity throughout

5.1 Real world impact:

The transition from independently timed embedded systems to a decentralised mesh topology reduces the risk of total failure and eliminates the single point of failure risk, which is common in centralised system designs. The implementation of the Emergency vehicle pre-emption (EVP) logic addresses the objective of governments reducing response times for emergency services (UK Parliament Emissions – Reference) and potentially saving lives. Furthermore, this design of the system allows for a reduction in unnecessary congestion during low traffic volume periods, which in addition also contributes to lower fuel consumption and decrease in carbon emissions.

5.2 Future Improvements:

Tactile sensory feedback: future iterations of this system will include a tactile actuator for those who may be visually and audibly impaired for this a small spinning cone will be implemented. This provides a physical signal that can be felt by pedestrians of these impairments, ensuring that the 10 node SAN meets the maximum inclusivity and accessibility standards set by the government. (Gov UK Inclusivity – Reference)

Pedestrian demand functions: The current model is a sensor driven system: in future models adding a physical button where users can request – could provide users with a secondary sense of comfort for example in a high-density pedestrian area such as a school zone.

5.3 Implementation Risks

While the proposed design offers significant advantages, several implementation risks must be identified and mitigated prior to real-world deployment. A primary hardware risk is the physical vulnerability of the exposed LDR sensors to environmental damage, dirt accumulation, or vandalism, which could induce false ambient readings. Additionally, although the mesh topology provides high network redundancy, a localised power failure at a specific intersection could disable a node entirely before it can default to its failsafe timer. Lastly ensuring highly secure data transmission protocols will be vital to prevent the malicious interception or spoofing of the EVP emergency packets across the network

6.0 References

UK Parliament Emissions: https://commonslibrary.parliament.uk/research-briefings/cdp-2025-0042/#:~:text=A%20minimum%20of%2078%25%20of,conveyances%20and%20reduce%20handover%20delays.

Gov UK Inclusivity: https://assets.publishing.service.gov.uk/media/61d32bb7d3bf7f1f72b5ffd2/inclusive-mobility-a-guide-to-best-practice-on-access-to-pedestrian-and-transportinfrastructure.pdf

7.0 Appendix

7.1 Debugging and Error log

During the initial testing phase using the Tinkercad software, there were several initial critical errors that were identified and resolved to ensure that the system met the design requirements and works practically and as it should.

- **Syntax Error (Figure A):** The initial code execution failed, with a "[C] does not name a type" error. This was identified to be a header formatting issue, where the language identifier was included in the complier window, this error was resolved by reinitalising the script with clear variable declaring to avoid this error again.
- **Signal saturation (1023 Error – Figure B):** The serial monitor reported a constant value of 1023 at pin A0, which locked the embedded system into a permanent emergency system override phase. This was found to be a floating signal due to hardware misalignment ...hardware misalignment between the LDR sensor and its pull-down resistor, and the Arduino. Correcting this established the necessary voltage resistance, which allowed for an accurate environment sensing.
- **Multi Sync:** The buzzer had to be remapped to buzz on the red LED as it mistakenly was alerting pedestrians when the traffic light was green allowing vehicles to go. This error was remapped to priorities pedestrian safety, ensuring that the auditory walk signal is only active when motorists flow is physically stopped.

7.2 Screenshots of Errors:

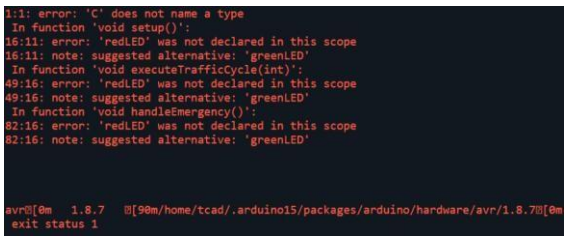


Figure A (Above): 'C' does not name a type error.

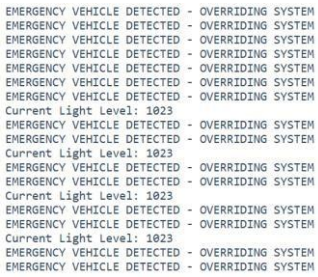


Figure B(Above): Signal Saturation (Floating Pin)

7.3 Source code:

```
// Pin config const int

redLED = 11;    const

int amberLED = 12; const

int greenLED = 13;

const int buzzer = 8;

const int ldrPin = A0;


// --- Light Level Thresholds ---

const int NIGHT_THRESHOLD = 300;     // Values below this trigger
Night Mode

const int STROBE_THRESHOLD = 900;    // High intensity strobe
detection int lightLevel = 0;

 void setup() {

pinMode(redLED, OUTPUT);

pinMode(amberLED, OUTPUT);

pinMode(greenLED, OUTPUT);

pinMode(buzzer, OUTPUT);


  // Initialize Serial Monitor for debugging

  Serial.begin(9600);
```

```
}  void loop() {   lightLevel =
analogRead(ldrPin);   // ADD THIS
LINE BELOW
  Serial.print("Current Light Level: ");
  Serial.println(lightLevel);
     if (lightLevel >
STROBE_THRESHOLD) {
handleEmergency();
  }

  // Constant monitoring of the environment
lightLevel = analogRead(ldrPin);

  // --- Action 1: EMERGENCY VEHICLE DETECTION ---
if (lightLevel > STROBE_THRESHOLD) {
handleEmergency();
  }
  // --- Action 2: NIGHT MODE ---   else if (lightLevel <
NIGHT_THRESHOLD) {     executeTrafficCycle(6000); // Extended
6-second red phase
  }
  // --- Action 3: STANDARD MODE ---   else {
executeTrafficCycle(3000); // Standard 3-second red phase
  }
}

// Function to manage the standard light sequence
// Function to manage the standard light sequence
void executeTrafficCycle(int redDuration) {   //
Green phase for motorists (Pedestrians wait)
digitalWrite(greenLED, HIGH);
```

```
digitalWrite(amberLED, LOW);
digitalWrite(redLED, LOW);   delay(3000);

  // Yellow phase
digitalWrite(greenLED, LOW);
digitalWrite(amberLED, HIGH);
delay(2000);

  // Red phase for motorists (Pedestrians safe to cross)
digitalWrite(amberLED, LOW);   digitalWrite(redLED,
HIGH);

  // Pedestrian audio walk signal synchronised with Red LED
for(int i = 0; i < 5; i++) {     tone(buzzer, 440);
delay(100);     noTone(buzzer);     delay(400);
  }

  // Wait out the rest of the red duration
delay(redDuration - 2500);
}

// Function for EVP void
handleEmergency() {
  Serial.println("EMERGENCY VEHICLE DETECTED - OVERRIDING SYSTEM");
digitalWrite(greenLED, LOW); // Ensure green is off

  // High-visibility hazard warning (Alternating Red and Yellow)
for(int i = 0; i < 20; i++) {     digitalWrite(redLED, HIGH);
    digitalWrite(amberLED, LOW);
tone(buzzer, 1000);
delay(50);
digitalWrite(redLED, LOW);
```

```
digitalWrite(amberLED, HIGH);
noTone(buzzer);     delay(50);
  }      digitalWrite(redLED,
LOW);   digitalWrite(amberLED,
LOW);
}
```