Smart City: Networked traffic controller for autonomous vehicle

1st Sheikh Muhammad Adib Bin Sh Abu Bakar
Fachhochschule Dortmund
ME. Embedded Systems Engineering
sheikh.binshabubakar001@stud.fh-dortmund.de

2nd Mohammed Rizwan
Fachhochschule Dortmund
ME. Embedded Systems Engineering
mohammed.mohammedrizwan009@stud.fh-dortmund.de

3rd Aditya Kumar
Fachhochschule Dortmund
ME. Embedded Systems Engineering
aditya.kumar001@stud.fh.dortmund.de

4th Farhaad Sheikh Mohammed
Fachhochschule Dortmund
ME. Embedded Systems Engineering
farhaad.sheikhmohammed002
@stud.fh-dortmund.de

5th Mykyta Konakh
Fachhochschule Dortmund
ME. Embedded Systems Engineering
mykyta.konakh001@stud.fh-dortmund.de

***Abstract*—This paper presents the design, implementation, and validation of a networked traffic control system for autonomous vehicles for traffic management in smart cities, outlining requirements, architecture, processes, hazards, and solutions.**

***Index Terms*—Smart city, Traffic controller, Autonomous vehicle**

I. Introduction

A smart city is an urban area where technology and data collection improve the quality of life, sustainability, and efficiency of city operations [1]. Smart city technologies include information and communication technologies (ICT) and the Internet of Things (IoT). These technologies increasingly play a crucial role in areas such as transportation, energy, and infrastructure [2]. Smart city technologies, such as smart transportation systems, can optimize traffic flow, reduce congestion, and improve the quality of life for city residents and commuters using real-time data. Although implementing such a system, especially with the rise of autonomous vehicles, can be challenging, A networked traffic control system is needed to integrate these vehicles into urban infrastructure. This paper explores the design, implementation, and validation of a networked traffic control system for autonomous vehicles in smart cities. It identifies key requirements, use cases, system contexts, and constraints and outlines the architecture, roles, and processes of TCUs (Traffic Controller Units) and RSUs (Road Side Units). The system's performance is evaluated through test cases, and potential risks are identified, including communication failures, software and hardware malfunctions, and power fluctuations. Solutions are proposed to enhance system resilience and safety.

II. Analysis

This section will cover problem domain analysis, use case identification for the system to be developed, mapping of various requirements, and identifying potential constraints on the system.

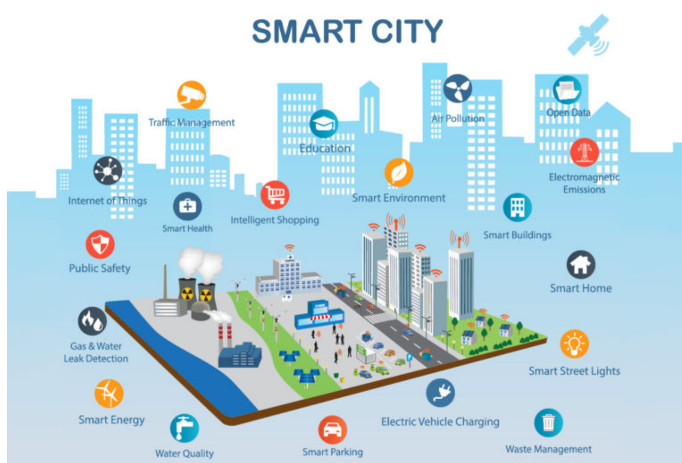

Fig. 1. Smart City Technologies [3]

We start by identifying the basic requirements needed to develop networked traffic control for autonomous vehicles. The following 10 requirements will be considered in the analysis

1) All of the controllers in the system must be able to communicate with one another and work together as a single, decentralized system.
2) The controllers must be able to communicate with each other as well as other devices in the network.
3) The system must be able to allow traffic authorities to monitor and adjust traffic signals.
4) The system must follow standards that other users can use.
5) The system must also be able to handle real-time data, make quick decisions, and manage traffic flow in real-

time.

6) The system must be able to schedule traffic at the intersection.
7) The system should establish public transportation schedules and prioritize public transportation when necessary.
8) The system should also have the capability to communicate with emergency vehicles.
9) The system should provide emergency vehicles with priority access to the intersection.
10) The system controller should be able to schedule pedestrian crossings.

Based on the requirements above we can define the use case for the system.

A. System Use Case

Based on the requirements above, we can define the use case for the networked traffic controller in Figure 2. We consider three main actors that will interact with the system: the administrator, the pedestrian, and the autonomous vehicle. The administrator can monitor the traffic. The pedestrian can use the system to request a crossing while the system ensures pedestrian safety. The system should optimize traffic flow and be able to dynamically adjust traffic signals. It should coordinate traffic at the intersection in real-time and avoid collisions between vehicles. It should encourage public transport and prioritize emergency vehicles.

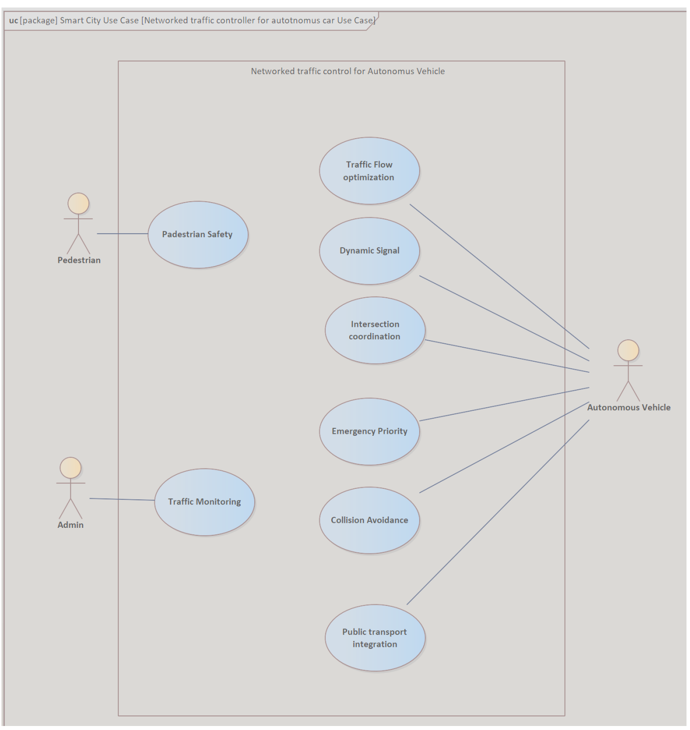

Fig. 2. Use Case Diagram

B. System Context

The system requirements diagram in Figure 5 describes the relationship between different requirements blocks and

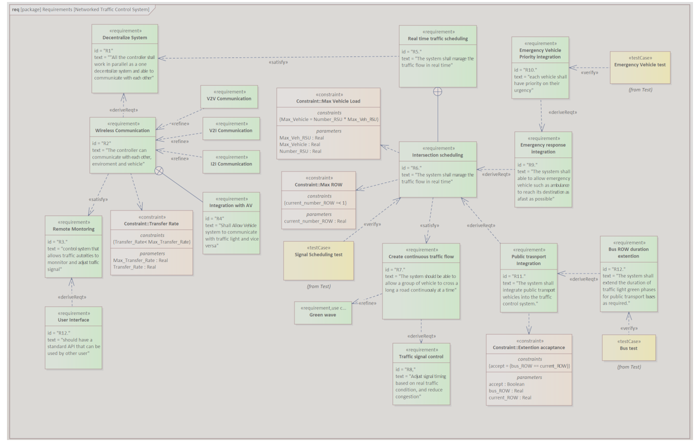

Fig. 3. Requirements Diagram

system constraints. The system requires all controllers to communicate and work together as a decentralized system; therefore, the system should have wireless communication capabilities and should support vehicle-to-vehicle, vehicle-to-infrastructure, and infrastructure-to-infrastructure communication. It should satisfy remote traffic monitoring. It must schedule traffic at intersections, prioritize public transportation, communicate with emergency vehicles, and schedule pedestrian crossings in real-time. It should also prioritize emergency vehicle access to the intersection.

C. Constraints

Some of the system limitations that we discovered during the process of developing the system context are shown in Fig. 4's constraints diagram. The system is required to schedule traffic for autonomous vehicles at the intersection, which it does by using the concept of ROW (right of way). The system determines the order in which vehicles can proceed through the intersection based on predefined rules and priorities. For our system, we consider four possible ROWs. The following are the five system constraints that we consider for our system:

1) The time extension for public transport is only possible when it has the same ROW as the one scheduled.
2) At a given time only one ROW is allowed to execute.
3) The vehicle can connect to the infrastructure when it is in a defined range.
4) The is a defined max load that the infrastructure can handle at a given time.
5) The transfer rate should be less than the defined limit.

III. Design and Implementation

In this section, the design and implementation stages were discussed. The fact that we combine the design and implementation stage is because we use agile method to develop the system prototype. At first, we refine our system architecture to have clear structure of the system, that will be explained in III-A, to make the development process much easier. After that, the structure and behaviour of each component were refined. This part are clarified in incoming subsection.

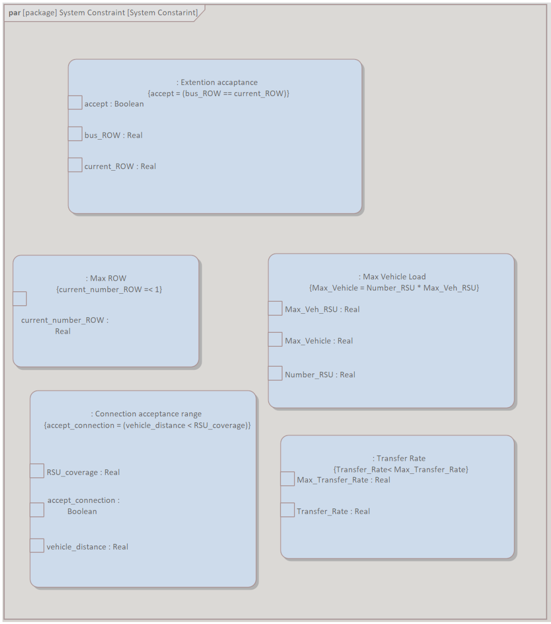

Fig. 4. System Constraint Diagram

A. System Architecture

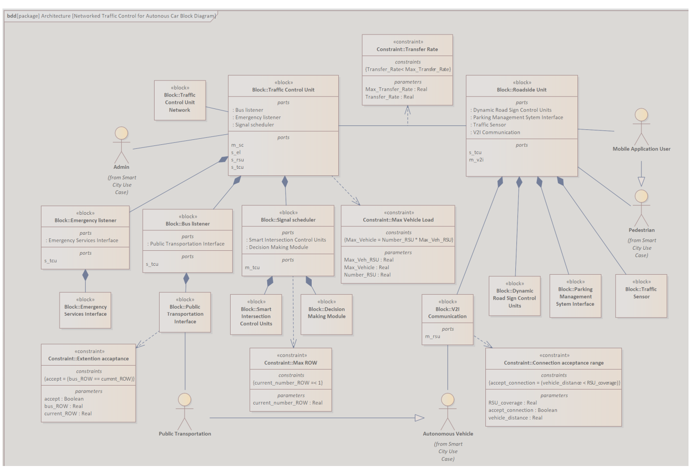

Fig. 5. System Block Diagram

The architecture in Figure 5 shows the networked traffic control system for autonomous vehicle. It describes how different components and sub-components work together to achieve specific functionalities. There are two main components in our system, TCU(Traffic Control Unit) and RSU(Roadside Unit). TCU control the traffic signals and communicate with traffic through RSU. Further working of both of these components are discussed in below sub-section in this paper. We are considering three main scenarios which are: Bus Listener, Emergency Listener and Signal Scheduler. Bus Listener receives and processes signals from the buses and the bus stop sign information sensor. It determines the arrival and departure times of the buses and adjusts the traffic signals accordingly. Emergency Listener handles emergency signals from the emergency vehicle information sensor. It gives priority to the emergency vehicles and alerts the RSU about the emergency situation. Signal Scheduler schedules the traffic signals based on certain criteria, such as traffic density, time of day, weather conditions, etc. It also communicates with the signal mixing module and the RSU to coordinate the traffic signals.

B. Allocation of TCU and RSU

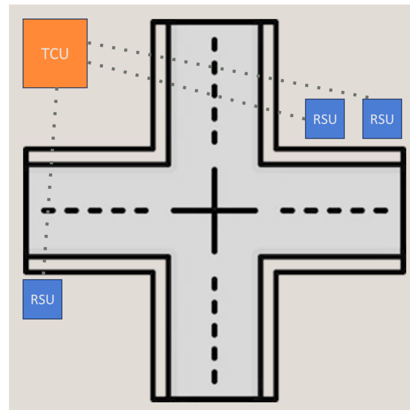

Fig. 6. Allocation of TCU & RSU

In Figure 6, we can see how Traffic Control Unit (TCU) and Roadside Units (RSUs) are positioned to monitor and manage traffic flow. The TCU is a central device that receives and processes data from multiple RSUs, which are devices that are installed at different locations on the roads. The RSUs communicate with vehicles using wireless signals, and collect real-time data such as location, direction, and requested direction etc. The TCU uses this data to analyze the traffic conditions and optimize the traffic signals, speed limits, lane changes, etc. The TCU does not communicate directly with the vehicles, but only with the RSUs. This system can improve traffic safety, efficiency, and environmental impact.

C. V2I Communication

The Figure 7 illustrates the process by which an autonomous vehicle requests permission to cross an intersection. The vehicle sends a request message to the nearest RSU, which acts as a relay between the vehicle and the TCU. The RSU forwards the request to the TCU, which is responsible for managing the traffic flow and signals at the intersection. The TCU consults with the Signal Scheduler, which is a module that determines the right of way (ROW) for each vehicle based on various factors, such as traffic density, priority, and safety. The Signal Scheduler returns the ROW status to the TCU, which then sends an instruction message to the RSU. The RSU relays the instruction to the vehicle, which can either proceed or stop depending on the ROW status.

The architecture of the Intersection Management System consists of two main blocks: Traffic Control and Roadside Unit shown in Figure 8. The Traffic Control block encompasses the TCU and the Signal Scheduler, as well as other components, such as Emergency Listener, Signal scheduler and Smart

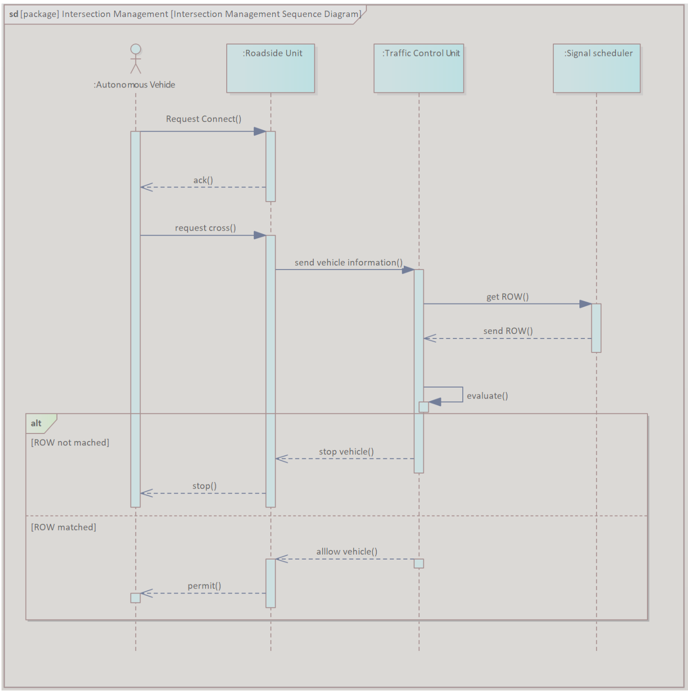

Fig. 7. Vehicle cross an intersection scenario

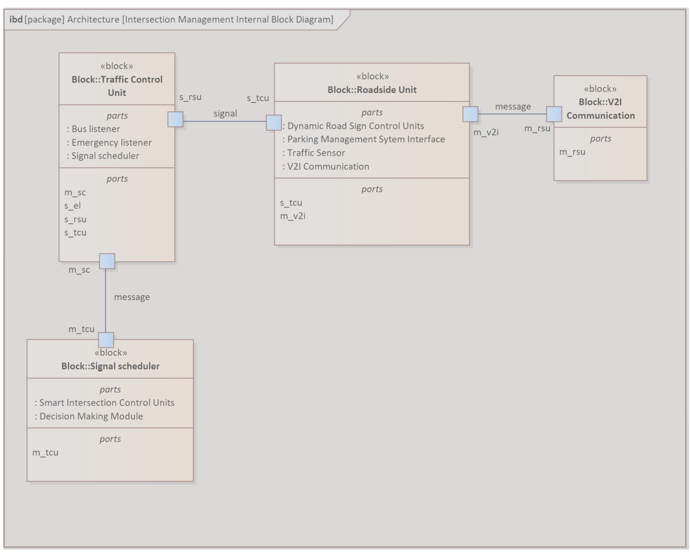

Fig. 8. System component interaction

Intersection Control Units, Decision Making Module. The Emergency Listener and Signal Scheduler is an interface that allows emergency vehicles, such as ambulances and fire trucks, to override the normal traffic signals and gain priority access to the intersection. The Smart Intersection Control and Units Decision Making Module are components that implement advanced algorithms and logic to optimize the traffic flow and signals at the intersection. The Traffic Sensor is a component that validates the traffic and the instruction received from the TCU and executes the appropriate action. The communication between these blocks and components is facilitated by messages exchanged via ports, as indicated by the arrows in the diagram. This diagram provides a detailed view of the Intersection Management System's architecture, and how it integrates different modules and interfaces to achieve safe and efficient navigation at the intersection.

D. TCU Processor

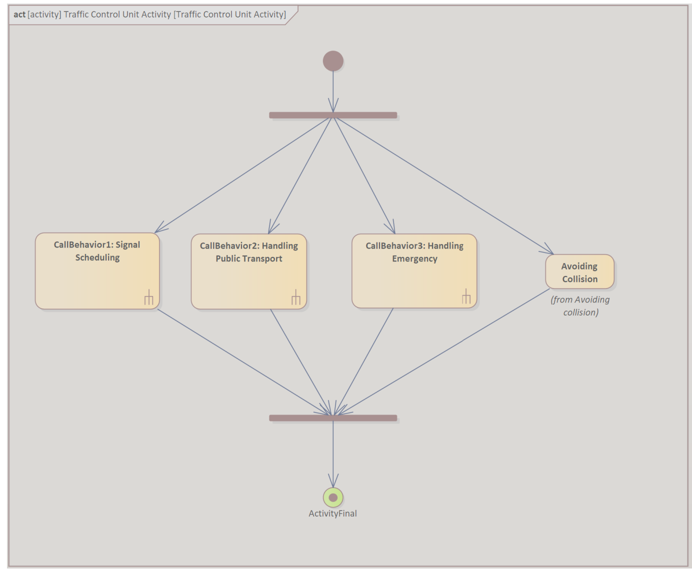

Fig. 9. Traffic Control Unit processes

The Traffic Control Unit (TCU) shown in Figure 9 is the key component for our networked traffic control system, and is responsible for optimizing traffic flow and improving road safety. It manages traffic lights to alleviate congestion, prioritizes public transportation to ensure schedule adherence, handles emergencies by redirecting traffic and assisting emergency vehicles, and uses sensor data to avoid collisions. The TCU receives data from the RSU first and then performs three major functions, which we are :

- Signal Scheduling ensures that vehicles operate smoothly and efficiently. Its goal is to reduce wait times and congestion through efficient traffic light coordination.
- Handling Public Transport: This process prioritizes buses and trams. The purpose is to ensure the timely flow and efficiency of public transportation.
- Handling Emergency: This process focuses on rerouting traffic and prioritizing emergency vehicles. The goal is to guarantee that emergency vehicles arrive at their destinations quickly.

E. RSU Behaviour

RSUs use three state machines, as shown in Figure 10, to manage traffic efficiently. The first state machine manages communication. In the 'Wait' mode, the RSU awaits incoming requests from cars or (TCUs). When it senses a request, it enters the 'Connect' state to establish communication with the requester. After successfully connecting, it enters the 'Receive'

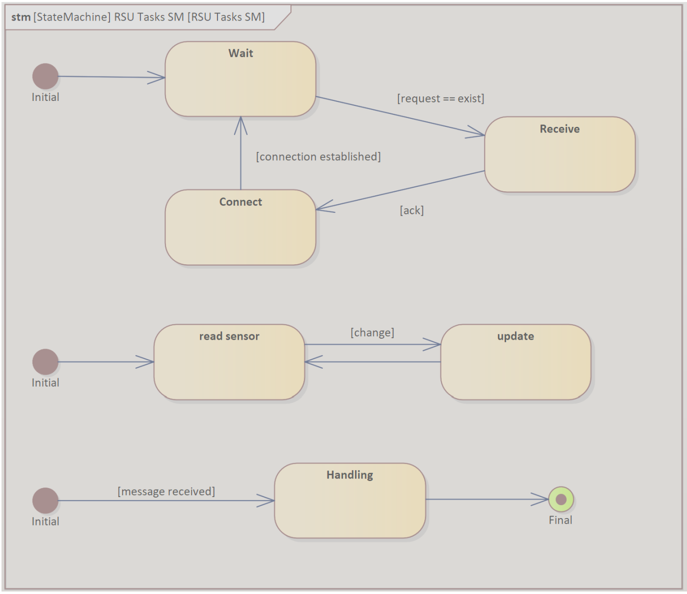

Fig. 10. RSU Processes

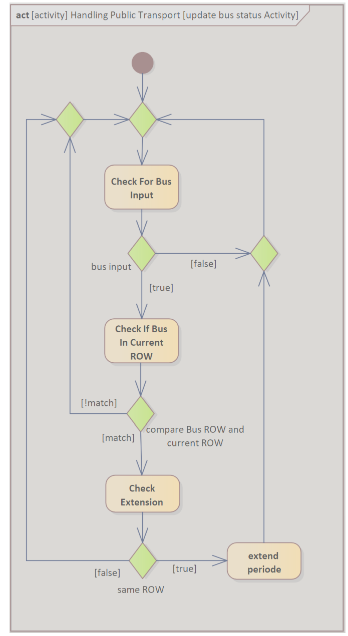

Fig. 11. Public Transport Integration (Bus) Activity

stage, where it is prepared to receive data or commands from the connected entity. The second state machine focuses on sensor inputs. The RSU continuously monitors sensor data, such as traffic conditions or environmental information. When it detects significant changes in this data, it enters the 'Update' state. The third handles message processing, transitioning from a passive to a 'Handling' state upon message receipt in order to process and act on the data. These techniques allow RSUs to effectively manage communication, sensor data, and messages, which is critical for reacting to and controlling real-time traffic circumstances.

F. Process 1: Public Transport Handler

One integral functionality within our Networked Traffic Control System involves the seamless integration of public transport vehicles, such as buses and trams, with the capability to communicate effectively with the TCU. Specifically, a noteworthy feature of our system is the ability of public transport vehicles to initiate communication with the TCU, thereby requesting an extension of the signal duration. This communication process is crucial for optimizing traffic flow. To elaborate on the mechanism, the public transport vehicle initiates the interaction by transmitting a signal to the TCU, requesting for an extended signal time.The process works as follows:

- In the initial phase, the TCU undergoes a verification process to determine if it has received any input from a public transport vehicle. In the absence of any input, the TCU diligently continues to monitor for incoming signals.
- Upon receiving an input, the TCU conducts a further assessment to verify if the vehicle input received from the ROW corresponds to the signal associated with the requested ROW extension. This stringent matching process ensures that the request aligns precisely with the anticipated response. In instances where a match is not successfully established, the current request is promptly discarded, and the TCU resumes its vigilance for new inputs.
- In the event of a successful match, the TCU proceeds to evaluate the possibility of extending the signal time. If the conditions allow for an extension, the TCU seamlessly executes the signal time extension. Conversely, if an extension is not permissible based on the existing criteria, the current request is promptly discarded, and the TCU resumes its continuous monitoring for any subsequent input.

The interaction between the public transport vehicle and TCU encompasses the dynamic adjustment of the idle scheduling within the traffic system as shown in Figure 8. Initiated by the Bus Listener, the process unfolds as follows: upon receiving input, the Bus Listener transmits a signal request to the TCU. Subsequently, the TCU engages in processing this signal request, evaluating the decision to be made.In instances where a signal extension is requested and deemed appropriate, the TCU issues a message instructing the adjustment of the signal schedulers behavior. This adjustment is based on the specified signal extension time.

G. Process 2: Emergency Handler

Emergency Handler is the second process which involves the detection and rapid response to emergency situations, typically triggered by the presence of emergency vehicles such as ambulances or fire trucks. In the event that the Traffic Control Unit (TCU) receives a signal from an emergency vehicle, it initiates a swift and high-priority response. Upon

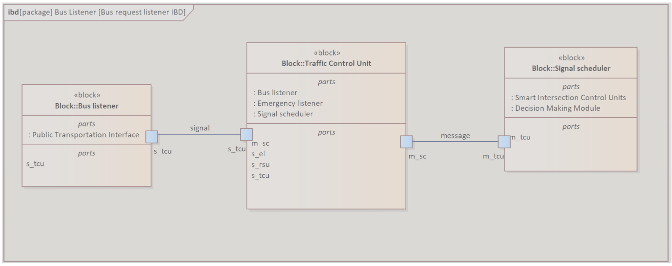

Fig. 12. Interaction between Bus Listener and Signal Scheduler through TCU

receiving the signal, the TCU processes the information and dynamically adjusts the idle signal scheduling state to transition into an emergency state. This modification ensures that the traffic control system promptly prioritizes and facilitates the smooth passage of the emergency vehicle through the affected area. This feature is pivotal in enhancing the system's responsiveness to critical situations, ultimately contributing to the efficiency of emergency services and the overall safety of the community. Illustrated in Figure 9:

- The Traffic Control Unit (TCU) consistently monitors for input pertaining to the presence of emergency vehicles.
- Upon receiving input indicating the presence of a high-priority vehicle, the TCU swiftly transitions to an emergency vehicle Right of Way (ROW) state. Simultaneously, the TCU modifies the standard signal scheduling to align with the exigencies of the emergency situation.

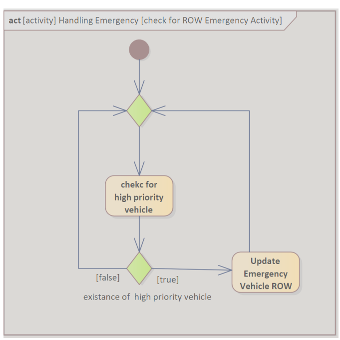

Fig. 13. Emergency Response Activity

The interaction between the emergency vehicle and the Traffic Control Unit (TCU) changes the the idle scheduling within the traffic system. Initiated by the Emergency Listener, the process unfolds as follows: upon receiving input, the Emergency Listener transmits a signal request to the TCU. Subsequently, the TCU engages in processing this signal request, evaluating the decision to be made. In instances where the existence of the emergency vehicle is determined, the TCU issues a message instructing the adjustment of the signal schedulers behavior. This adjustment is based on the Emergency Vehicle ROW. The interaction between these components is visually depicted in Figure 10, that contribute to the effective management and optimization of the traffic system.

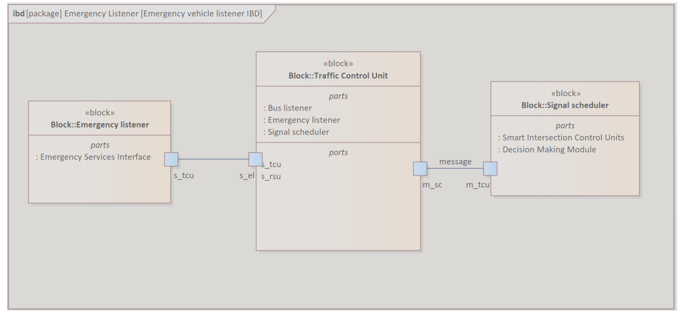

Fig. 14. Interaction between Emergency Listener and Signal Scheduler

H. Process 3: Signal Scheduler

Signal scheduling is the third process run by Traffic Control Unit (TCU). The behavior of this process can be grouped into two state, Idle Signal Scheduling state and ROW emergency state as visualized in Figure 15. In Idle Signal Scheduling state, the sequence of Right of Way (ROW) is fixed as depict in Figure 16.

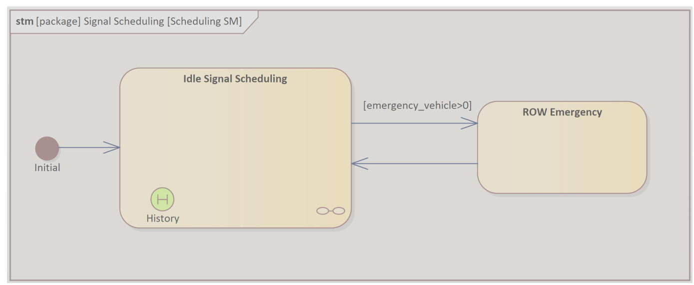

Fig. 15. Signal Scheduling Process

Whenever the value of emergency vehicle is greater the 1, the process will change its state to ROW Emergency state. The variable is a global variable that is updated by Emergency Response process as explained in section III-G. In ROW Emergency state, the process will change the ROW to the one that has emergency vehicle. For now, the duration for the process shall remain in the state within a fixed duration and shall turn back to Idle Signal Scheduling state.

Since Idle Signal Scheduling state is a history state, the process will back to previous ROW. This is to allow fairness between the ROWs. The process have the same procedure in every state even in ROW Emergency state because the state have fixed duration. The procedure is illustrated in Figure 17. It shows that the bus input is also a parameter used by the process that were explained in section III-F.

The TCU first updates its clock.Simultaneously, it actively monitors for the presence of an emergency state. If an emergency state is detected, the TCU proceeds to validate it. If the emergency state is valid, the TCU halts signal scheduling. In case the emergency state is invalid, the TCU concludes the emergency procedure and smoothly transitions back to the next appropriate state. When no emergency state is identified, the TCU proceeds to the next phase of validation.

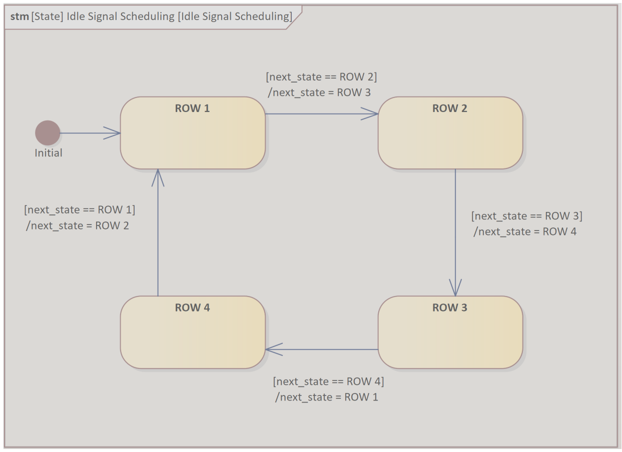

Fig. 16. Internal states of idle signal scheduling states

The TCU continues to validate its current state, and checking for duration extension or emergency input. If an emergency input is detected during this check, the TCU promptly shifts to the emergency state. In the absence of an emergency input, the TCU smoothly transitions to the next appropriate state in the sequence. Following state transition, the TCU re-evaluates the validity of its current state. If the current state is found to be valid, the TCU progresses to the subsequent state, updating the signal accordingly. In cases where the current state is invalidated, the TCU proceeds to wait.

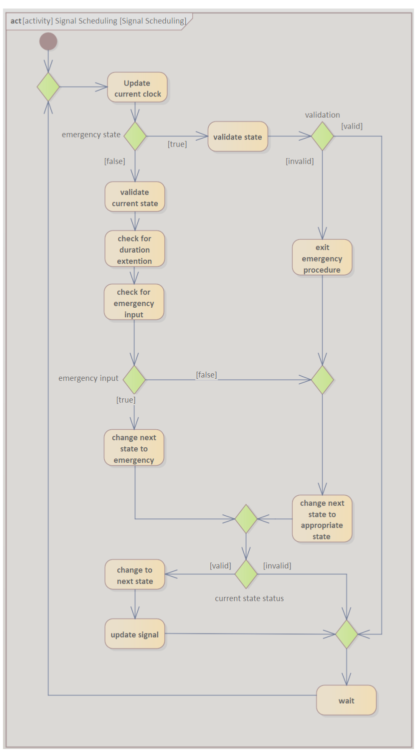

Fig. 17. Signal scheduling process' activity

In the next section, how these processes are executed in concurrent way on a chosen platform so that the system shall behave in predicted manner are explained.

TABLE I
TASK PARAMETER

Task	Period (µs)	Deadline (µs)	WCET (µs)	Capacity
Signal Scheduling	5000	5000	1692	0.34
Bus Listener	5000	5000	1248	0.25
Emergency Listener	5000	5000	1784	0.36

I. TCU Tasks scheduling

In the previous subsections, some of the processes run by the TCU were explained. In this paper, the word task and process are interchangeable. As our system must operate in real-time, ensuring the schedulability of the process is crucial to maintain predictability in the system's behavior. To realize this, we consider one processor scheduling problem that lead us to utilize Rate Monotonic Scheduling. The scheduler that we use in this project is FreeRTOS.

To schedule them we first set the parameter for each task that are needed by the scheduler as listed in Table I. The period and deadline of a task is the same since we use RMS. To gain the Worst Case Execution Time (WCET) for each task, we first find the longest possible path taken by a task and then try to run the task on the chosen platform, in our case is Arduino mega, and record the time it took to complete the task in micro seconds with a specific clock rate which is 16 MHz.

After that we set the period of the task in the way that they are schedulable. This schedulability is evaluated based on the capacity value using the utilization formula as in equation 1. The total capacity value is 0.94 which is less then 1. Meaning that they are scheduable with RMS. The result of the scheduler is visualized in Figure 18.

$$U = \sum C_i / D_i \tag{1}$$

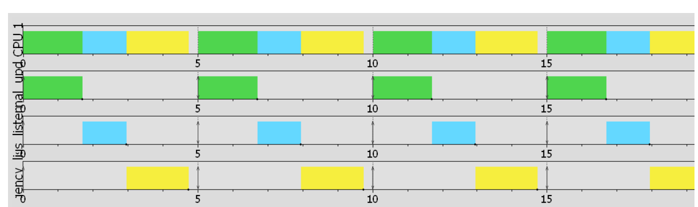

Fig. 18. Scheduler result.

In the next section we will discussed how we verify and validate our system.

IV. VERIFICATION AND VALIDATION

This chapter describes the procedures and methodologies used during the device and defect testing phase of the signal scheduling system's V&V process. The focus is on ensuring that the system meets design specifications and identifying any defects that may affect functionality, particularly in the areas

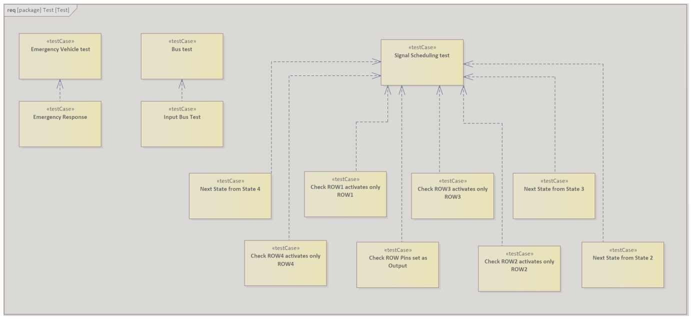

Fig. 19. Test case

of emergency response handling and signal scheduling.An overview of the test cases is illustrated in Figure 19

In order to make sure that our system will work properly in an emergency vehicle, we made three tests:

1) The first test verifies the ability of the system of processing into an emergency state or bus when a valid emergency signal is received. It ensures that the system will respond by entering the emergency state when an emergency signal or bus signal are received. The test confirms that the corresponding flag or state in the system is set to true, indicating that the system responds as expected.
2) The second test is designed to assess the system's tolerance to invalid alarms. The alarm trigger is set to an unexpected value outside the range of valid alarms. The purpose of this test is to ensure that the system does not erroneously transition to an emergency state when invalid data is received. The test verifies that the `emergency_state` remains false, demonstrating the system's capability to handle incorrect input signals without compromising its operational state.
3) The purpose of third test is to validate the logic of the system's transition to a normal state after an alarm occurs. It evaluates whether the system can correctly transition to the next state when an emergency occurs. This is very important for scenarios where the system has to cancel the normal sequence of actions due to an emergency. The test ensures that the system continues to transition from one state to the next as required after an emergency.

A. *Process 3: Signal Scheduler*

In order to test the signal scheduling system, we have written tests that check all the important functionality, detailed below:

1) Next State from State Tests: These tests are designed to verify the logic that governs state transitions within the signal scheduling system. They confirm that the system correctly transitions from one state to the next in a predetermined sequence:

- From state 1 to state 2,
- From state 2 to state 3,
- From state 3 to state 4, and
- From state 4 back to state 1.

Each test case focuses on a particular state transition, ensuring that the system's logic correctly handles the progression from one state to the next.

```
void test_ROW1ActivatesOnlyROW1Output(void) {
  signal_scheduling_task.SETUP();
    signal_scheduling_task.next_state = 1;

    signal_scheduling_task.job();

    TEST_ASSERT_EQUAL(HIGH, pin_value[ROW_1_OUTPUT]);
    TEST_ASSERT_EQUAL(LOW, pin_value[ROW_2_OUTPUT]);
    TEST_ASSERT_EQUAL(LOW, pin_value[ROW_3_OUTPUT]);
    TEST_ASSERT_EQUAL(LOW, pin_value[ROW_4_OUTPUT]);
}
```

Fig. 20. Testing ROW1 individually (sample code).

2) ROW Activation Tests: Each of these test cases checks the individual activation of the ROW (Right of Way) outputs. They are designed to confirm that when a particular ROW, such as a traffic lane or signal light, is activated, only that specific ROW is active while all others remain inactive. This selective activation is crucial to prevent conflicting signals that could potentially lead to traffic problems. The procedure for these tests is exemplified in Figure 20

Similar tests are conducted for ROW2, ROW3, and ROW4, ensuring that each ROW's activation is handled correctly by the system.

V. Hazard

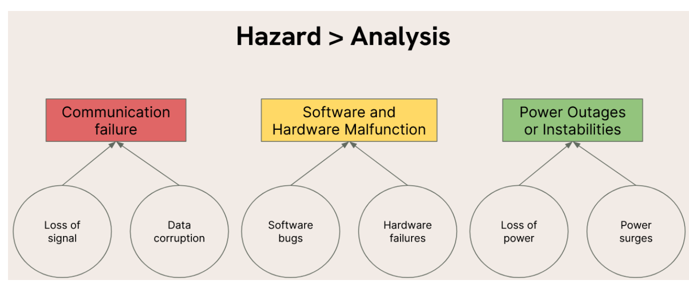

Fig. 21. Diagram outlining the main categories of hazards.

The diagram is organized into three main categories of hazards. An overview of the hazards illustrated in Figure 21

1) Communication Failure: This category identifies the risks associated with the failure of communication systems. Subcategories include:
   a) Loss of Signal: This can mean interruption of communication lines or signals, which can lead to a lack of coordination or response in a critical situation.
   b) Data Corruption: This hazard involves the alteration of data during transmission, which may result in incorrect information and action.

2) Software and Hardware Malfunction: This category refers to potential problems with software and hardware components of the system. Subcategories include:
   a) Software Bugs: Flaws or errors in software code that can cause unexpected behavior or system failures.
   b) Hardware Failures: Physical malfunctions or breakdowns of hardware components that can cause system malfunctions or improper operations.
3) Power Failures or Fluctuations: This category focuses on the power supply and the risks associated with its failures or fluctuations. Subcategories include:
   a) Power Loss: Complete power failures that can shut down an entire system or individual critical components.
   b) Power Surges: Power surges that can damage equipment, corrupt data, or cause erratic system behavior.

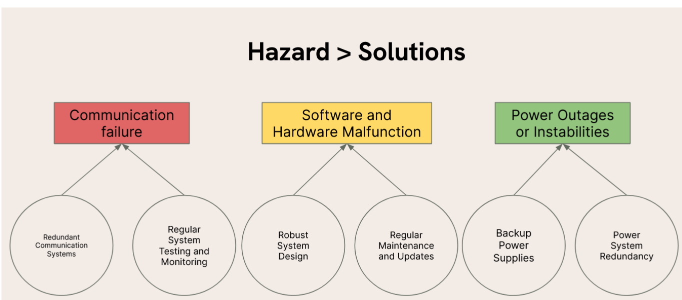

Fig. 22. Solutions to address the identified hazards.

Solutions illustrated in Figure 22.Solutions to address these hazards include:

1) Solutions to Address Communication Failures:
   a) Redundant Communication Systems: Implementing redundant systems ensures continuous and reliable communication by having backup channels or methods that can take over in case the primary communication path fails.
   b) Regular System Testing and Monitoring: Regular testing and monitoring of communication systems help detect and correct faults before they lead to failure, thus preventing signal loss or data corruption.
2) Solutions for Software and Hardware Faults:
   a) Robust System Design: A robust system design includes using high-quality components, developing fault tolerance, and implementing error-handling procedures to prevent complete system failure if a component fails.
   b) Regular Maintenance and Updates: Regular maintenance and software updates can address hardware problems and software bugs, preventing them from leading to failures.
3) Solutions for Power Outages or Instabilities:
   a) Backup Power Supplies: Backup power solutions, like Uninterruptible Power Supplies or generators, provide immediate power to ensure that critical components remain operational during outages.
   b) Power System Redundancy: Power system redundancy, involving alternative power sources or redundant power delivery paths, ensures the availability of power to critical system components at all times.

VI. Summary and Outlook

In this paper we have discussed how we develop out system prototype from refining our system use case and designing the system to the simulation of the system behaviour partially for verification and validation. We also consider the Hazards to improve system safety. Since there are many domains that need to be explored where it is beyond our ability, the system we designed has many flaws such as not considering communication failure. However, we focus on our approach to building a system. In the future, many stake holder shall involve in this project to improve the system requirement,thus improve the system design to make it more dependable.

References

[1] Team, L. (2023, March 10). How IoT is enhancing the development of smart cities. Lvivity. https://lvivity.com/iot-and-smart-cities
[2] What is a smart city? — IBM. (n.d.). https://www.ibm.com/topics/smart-city
[3] Van Nisselrooij, E. (2023, July 10). Was ist eine Smart City? Und welche Ziele hat sie? - Secure Insights. Secure Insights. https://www.axis.com/blog/secure-insights-de/was-ist-eine-smart-city/

Contributors

link to GitHub repository: https://github.com/ESE-Smart-City/Main/tree/main

Member and responsibility:

1) Sheikh Muhammad Adib
   - Signal Scheduling
   - Task Scheduling
   - Code and System Architecture
2) Mohammad Rizwan
   - System Architecture
   - Allocation of TCU and RSU
   - V2I Communication
   - TCU Processor
   - RSU Behaviour
3) Aditya Kumar
   - System Analysis: Use Case, Requirements, Constraints
   - Emergency Handling
4) Farhaad Sheikh Mohammed
   - Process 1: Public Transport Handler
   - Process 2: Emergency Handler
   - Process 3: Signal Scheduler
5) Mykyta Konakh
   - Verification and Validation
   - Hazards

AFFIDAVIT

We, Sheikh Muhammad Adib, Aditya Kumar, Mohammed Rizwan, Farhaad Sheikh Mohammed, Mykyta Konakh herewith declare that we have composed the present paper and work ourselves and without the use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form has not been submitted to any examination body and has not been published. This paper was not yet, even in part, used in another examination or as a course performance.