**VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY**

**UNIVERSITY OF TECHNOLOGY FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**DEPARTMENT OF ELECTRONICS**

🙞∙∙∙☼∙∙∙🙜



**Embeded System Design**

**HK242**

**Topic:** Wireless Communicate

Traffic Light

**Hồ Chí Minh City – 2025**

|  |  |
| --- | --- |
| **Full Name** | **Student’s ID** |
| Đào Nguyên Khôi | 2151215 |
| Nguyễn Hà Phong Thịnh | 2151143 |
| Nguyễn Đình Huy | 2211208 |

**Lecturer’s Sign:**

**Table of Contents**

[**I)** **Introduction:** 3](#_Toc197506712)

[**II)** **System requirements:** 4](#_Toc197506713)

[**III)** **Bill of Materials (BOM):** 6](#_Toc197506714)

[**IV)** **Diagram and Shematic:** 9](#_Toc197506715)

[**1)** **System Diagrams:** 10](#_Toc197506716)

[**2)** **System Schematics:** 11](#_Toc197506717)

[**3)** **System PCB:** 13](#_Toc197506718)

[**V)** **Design Architecture:** 17](#_Toc197506719)

[**1)** **Overview of the System Architecture:** 17](#_Toc197506720)

[**2)** **Master Unit Setup and Functionality:** 17](#_Toc197506721)

[**3)** **Slave Unit Setup and Functionality:** 22](#_Toc197506722)

[**VI)** **Result:** 25](#_Toc197506723)

[**1)** **Result and Conclusion:** 25](#_Toc197506724)

[**2)** **Future Improvements** 25](#_Toc197506725)

1. **Introduction:**

The wireless traffic light system project is designed to create an efficient and reliable embedded system for managing traffic flow in a modern, scalable manner. By utilizing a Master-Slave architecture, the system ensures synchronized control of traffic lights across two directions, addressing the need for effective traffic management in both urban and rural environments. The primary objective is to integrate cost-effective hardware and software components to deliver a robust solution that enhances road safety and driver convenience.

At the heart of the system, STM32 microcontrollers act as the central control units for both the Master and Slave, coordinating all system operations. To provide clear visual feedback, double 7-segment LEDs are employed to display the remaining time for each traffic light phase. These LEDs are interfaced with 74HC595 shift registers using the SPI protocol, enabling precise and efficient communication to ensure drivers are informed of timing changes.

Real-time synchronization is achieved through the DS3231 real-time clock (RTC) module, which plays a critical role in maintaining accurate timing. The DS3231 sends a pulse every second via its SQW pin to the STM32, triggering updates to the system’s timekeeping. This time data is displayed on a 16x2 LCD, which is integrated with a PCF8574T module for I2C communication. The LCD not only shows the current time but also indicates the active traffic light mode, providing clear and immediate status updates.

Wireless communication between the Master and Slave units is facilitated by HC-12 modules, which utilize UART to exchange data reliably. The Master transmits a string specifying the selected operating mode—such as Normal, Heavy, Night, or Control—and the Slave interprets this string to execute the corresponding traffic light sequence. This project showcases the seamless integration of embedded systems, demonstrating a practical and innovative approach to traffic management with potential for future enhancements.

1. **System requirements:**

The wireless traffic light system is designed to deliver a reliable, efficient, and user-friendly solution for managing traffic flow in various scenarios. To achieve this, the system must meet a comprehensive set of requirements that ensure safety, robust communication, accurate timekeeping, clear information display, and flexible operation. These requirements were carefully defined to balance performance, cost, and practicality, guiding the selection of components and the overall system design. By addressing critical aspects such as power stability, wireless data transmission, real-time synchronization, and mode flexibility, the system aims to provide a seamless experience for both operators and drivers.

The following sections outline the specific requirements, organized into six key categories: System Safety and Reliability, Communication and Data Transmission, Information Display, Traffic Light Control, Real-Time Synchronization, and Operation Mode Switching. Each category includes detailed specifications to ensure the system operates effectively under diverse conditions, from noisy environments to varying traffic demands. These requirements serve as the foundation for the system’s development, ensuring it meets the project’s goals of reliability, functionality, and ease of use.

|  |  |
| --- | --- |
| **1. System Safety and Reliability** | |
| ID | **Requirement** |
| 1.1 | The system must be powered stably using a 9V adapter regulated to 5VDC via ASM1117 |
| 1.2 | All connections must be properly placed and secured on the breadboard to ensure stable operation |
|  |  |
| **2. Communication and Data Transmission** | |
| ID | Requirement |
| 2.1 | 1 Master must use an HC-12 module to wirelessly transmit data to 1 Slave |
| 2.2 | HC-12 module communicate with STM32 via UART interface |
| 2.3 | Master and Slave are assigned at channel 88 |
| 2.4 | The communication speed must be at 9600 bps |
| 2.5 | The system must maintain stable communication at distances under 50 meters in noisy environments |
| 2.6 | The HC-12 module must operate at a transmit power of 20 dB (100 mW) to ensure sufficient communication range |
|  |  |
| **3. Information Display** | |
| ID | Requirement |
| 3.1 | The system must include a 16x2 I2C LCD to display real time |
| 3.2 | The LCD must display: real time and mode status |
| 3.3 | The display must be updated at least every 1 second |
|  |  |
| **4. Traffic Light Control** | |
| ID | Requirement |
| 4.1 | The Master must control Red, Yellow, and Green LEDs using GPIO as well as Slave |
| 4.2 | In Normal Mode, the Green light must stay on for 18 seconds, Yellow for 3 seconds, and Red for 20 seconds |
| 4.3 | In Heavy Mode, the Green light must stay on for 45 seconds, Yellow for 5 seconds, and Red for 50 seconds |
| 4.4 | In Night Mode, the Yellow light blinks every 0,5 second |
| 4.5 | Control Mode allows traffic light direction change |
|  |  |
| **5. Real-Time Synchronization** | |
| ID | Requirement |
| 5.1 | The system must use a DS3231 module for real-time clock (RTC) functionality |
| 5.2 | The DS3231 must communicate via I2C, sharing the bus with the LCD |
| 5.3 | The DS3231 update time every 1 second using interrupt |
| 5.4 | The DS3231 support automatic mode switching |
|  |  |
| **6. Operation Mode Switching** | |
| ID | Requirement |
| 6.1 | The system must support 5 operation modes controlled via 5 push buttons |
| 6.2 | Button 1 switches to **Normal Mode** |
| 6.3 | Button 2 switches to **Heavy Mode** |
| 6.4 | Button 3 switches to **Light Mode** |
| 6.5 | Button 4 switches to **Control Mode** (manual operation) |
| 6.6 | Button 5 is used to **manually toggle traffic light states** in Control Mode |
|  |  |

1. **Bill of Materials (BOM):**

The wireless traffic light system relies on a carefully selected set of components to ensure its functionality and performance. This section lists the Bill of Materials (BOM), detailing each component, its part number or type, quantity, and purpose within the system. These items were chosen to meet the project’s requirements for reliability, cost-effectiveness, and ease of prototyping on a breadboard. The selection process prioritized components that align with the system’s need for wireless communication, real-time display, and precise control, forming the foundation for the system’s design and implementation.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Component Name** | **Part Number / Type** | **Quantity** | **Notes** | **Link** |
| 1 | Microcontroller | STM32F103C8T6 | 2 | Main control unit | <https://www.thegioiic.com/mach-stm32f103c8t6-micro-usb> |
| 2 | Wireless communication module | HC-12 | 2 | For wireless data transmission | <https://hshop.vn/mach-thu-phat-rf-uart-si4463-433mhzkhoang-coch-1km> |
| 3 | Display module | LCD 16x2 I2C | 1 | Displays time and traffic light status | [https://hshop.vn/lcd-text-lcd1602-xanh-lo https://hshop.vn/mach-chuyen-giao-tiep-lcd1602-lcd1604-lcd2004-sang-i2c](https://hshop.vn/lcd-text-lcd1602-xanh-lo) |
| 4 | Real-time clock module | DS3231 | 1 | Maintains accurate real time | <https://hshop.vn/mach-thoi-gian-thuc-rtc-ds3231> |
| 5 | 5mm LEDs | 2 Red / Yellow / Green (Traffic Light Module) | 2 | 3 colors per direction | <https://hshop.vn/mach-hien-thi-led-den-giao-thong> |
| 7 | Push buttons | 5 Tactile Buttons | 5 | For selecting operation modes | <https://www.thegioiic.com/nut-nhan-6x6mm-cao-4-3mm-4-chan-xuyen-lo> |
| 8 | Voltage regulator | 2 AMS1117-5.0 / 3.3 module | 2 | Steps down from 9V to 5V / 3.3V | <https://hshop.vn/mach-cap-nguon-3-3vdc-5vdc-asm1117> |
| 10 | Power adapter | 2 9V 1A DC | 2 | Supplies power to the system | <https://caka.vn/nguon-adapter-9v-1a> |
| 12 | Breadboard / PCB | — | 2 | For prototyping | <https://hshop.vn/test-board-cammb-102> |
| 13 | Jumper wires | — | Many | For breadboard connections | <https://caka.vn/hop-day-cam-board-test-140-soi-nhieu-kich-thuoc> |

The wireless traffic light system’s components were selected to meet the project’s technical requirements while balancing cost, reliability, and ease of prototyping. Each component was chosen for its specific capabilities, ensuring the system could achieve efficient traffic management, reliable communication, accurate timekeeping, and clear visual feedback. The selection process also prioritized widely available parts compatible with breadboard prototyping, supporting rapid development and testing within the project’s constraints.

The STM32F103C8T6 microcontroller was chosen as the core of both Master and Slave units due to its robust ARM Cortex-M3 processor, which provides sufficient computational power for real-time tasks like GPIO control, UART, I2C, and SPI communication. Its support for multiple interfaces made it ideal for managing the HC-12 module, DS3231 RTC, and 7-segment displays, while its affordability and popularity in the maker community ensured accessibility and ample documentation for development. The HC-12 wireless module was selected for its reliable long-range communication capabilities at a low cost, operating effectively at 9600 bps and 20 dB power, which supports stable data transfer up to 50 meters even in noisy environments, aligning with the system’s synchronization needs. The DS3231 real-time clock module was picked for its high accuracy and low power consumption, with features like the SQW pin interrupt enabling energy-efficient time updates, crucial for maintaining consistent traffic light sequences across modes.

For display purposes, the 16x2 LCD with a PCF8574T I2C expander was chosen to provide a compact and user-friendly interface for showing real-time and mode status. The PCF8574T reduces the pin count needed for the STM32, simplifying the wiring on a breadboard, while the LCD’s clarity ensures operators can easily monitor the system. The 74HC595 shift register was selected to drive the double 7-segment LEDs, offering an efficient way to control multiple display segments using the SPI protocol, which minimizes the STM32’s pin usage and supports rapid updates for countdown timers. Red, Yellow, and Green LEDs were chosen for their simplicity and visibility, providing clear traffic signals, while five push buttons were included for their straightforward GPIO integration, enabling intuitive mode selection and manual control at a low cost. Finally, the AMS1117 voltage regulator was selected to ensure a stable 5V/3.3V supply from a 9V adapter, protecting the system from power fluctuations and supporting consistent operation of all components during prototyping and testing.

1. **Diagram and Shematic:**

To understand the design and implementation of the wireless traffic light system, this section provides four images: two system diagrams and two schematics. These visuals the integration of the STM32 microcontrollers, HC-12 wireless modules, DS3231 real-time clock (RTC), double 7-segment LEDs with 74HC595 shift registers, and the 16x2 LCD with PCF8574T I2C interface. The system diagrams offer a high-level view of the Master-Slave architecture, while the schematics focus on the precise hardware connections. Each illustration includes annotations to explain component interactions.

1. **System Diagrams:**

The system diagrams give a clear overview of the Master-Slave setup, showing how components connect and data moves between them. They focus on the wireless communication via HC-12 modules, timekeeping with the DS3231, and displays on the 7-segment LEDs and LCD.

The first diagram covers the Master unit. It shows the STM32 microcontroller linked to the HC-12 module for wireless communication (UART), the DS3231 RTC for time updates, the PCF8574T-driven LCD for status display, the 74HC595-driven 7-segment LEDs for countdowns, and push buttons for mode selection. This diagram explains the Master’s role in controlling the system.

A diagram of a computer

AI-generated content may be incorrect.

**Figure 1: Master Diagram**

The second diagram focuses on the Slave unit. It illustrates the Slave STM32 receiving instructions through its HC-12 module and controlling the 7-segment LEDs via the 74HC595 shift register. This shows how the Slave follows the Master’s commands to keep the traffic lights in sync.

A diagram of a computer

AI-generated content may be incorrect.

**Figure 2: Slave Diagram**

These diagrams together clarify how the Master and Slave units coordinate to manage the traffic light system.

1. **System Schematics:**

A diagram of a circuit board

AI-generated content may be incorrect.

**Figure 3: STM32F103C8T6 Interal Schematic**

The schematics provide detailed wiring instructions for building the system on a breadboard. They show exact pin connections, helping with assembly and troubleshooting.

The Master schematic details the wiring for the Master unit. It includes the STM32 connected to the HC-12 module (UART), DS3231 RTC and PCF8574T-driven LCD (I2C), 74HC595-driven 7-segment LEDs (SPI), push buttons (GPIO), and AMS1117 voltage regulator for 5V/3.3V power. This schematic guides the setup of the Master’s hardware.

**A diagram of a computer circuit

AI-generated content may be incorrect.**

**Figure 4: Master Schematic**

The Slave schematic shows the wiring for the Slave unit. It covers the Slave STM32 linked to the HC-12 module (UART), 74HC595-driven 7-segment LEDs (SPI), and AMS1117 voltage regulator. This diagram highlights the Slave’s simpler setup for responding to the Master.

**A diagram of a computer

AI-generated content may be incorrect.**

**Figure 5: Slave Schematic**

These schematics are practical tools for constructing the system, ensuring all connections are correct and functional.

1. **System PCB:**

The PCB design for the Master unit of the wireless traffic light system integrates an STM32F103C8T6 microcontroller with peripherals, focusing on reliability and signal integrity. This section details the two-layer PCB design, including the top layer for routing, vias for connectivity, and the bottom layer for ground plane.

The two-layer PCB supports the Master unit’s functions: wireless communication via HC-12, real-time synchronization with DS3231, mode selection through push buttons, and display on a 16x2 LCD with PCF8574T. Power is regulated from 9V to 5V/3.3V using an AMS1117 regulator.

The component positions are placed and group also set some Design Rules so that the wiring can be properly routed and are shown as shown in the figure below:

A blue circuit board with many wires

AI-generated content may be incorrect.

**Figure 6: Master PCB**

A green circuit board with a green screen

AI-generated content may be incorrect.

**Figure 7: Master PCB 3D**

The PCB uses a two-layer design:

* Top Layer: Contains all signal and power traces (yellow in the schematic). I2C lines to DS3231 and LCD, GPIO lines to push buttons, and power lines (5V/3.3V) are routed here. Vias are strategically placed to connect traces to the bottom layer for ground.
* Bottom Layer: Fully dedicated to a ground plane to reduce noise and improve signal integrity. Vias ensure all ground pins of components (e.g., STM32, DS3231) are connected to this plane. The layout minimizes trace lengths for I2C and GPIO signals, avoiding overlap to reduce interference.

For the Master, the group drew the internal schematic of the STM32 in detail. However, for the Slave, group used a module to make wiring easier and to be closer to the reality that group used a module to complete the project.

A blue screen with many wires

AI-generated content may be incorrect.

**Figure 8: SLAVE PCB**

A green circuit board with white and blue lines

AI-generated content may be incorrect.

**Figure 9: SLAVE PCB 3D**

1. **Design Architecture:**
2. **Overview of the System Architecture:**

The wireless traffic light system is engineered with a Master-Slave architecture to provide efficient and synchronized traffic management across two directions, addressing a variety of traffic scenarios from busy intersections to quieter roads. The STM32F103C8T6 microcontrollers serve as the central processing units for both the Master and Slave units, delivering the computational power required to handle communication, timing, and control tasks with precision. Wireless communication is a key feature, facilitated by HC-12 modules that enable the Master to transmit mode instructions to the Slave over a reliable UART interface, eliminating the need for physical wiring and enhancing system flexibility. This setup ensures that both units can operate in unison, adapting to the demands of real-world traffic conditions.

The system’s reliability is further supported by the DS3231 real-time clock (RTC) module, which provides accurate timekeeping by delivering precise updates to keep both units synchronized. This is essential for maintaining consistent traffic light sequences across different modes. Display outputs enhance usability, with a 16x2 LCD displaying real-time information and mode status, and double 7-segment LEDs showing countdown timers for each traffic light phase. Together, these components enable the system to support a range of operating modes like Normal, Heavy, Night, and Control, allowing it to adjust effectively to varying traffic demands while ensuring safe and reliable traffic management.

1. **Master Unit Setup and Functionality:**

To illustrate the operational flow of the Master unit, a flowchart is provided below. It outlines the sequence of tasks, including checking each push button (PB5 to PB9) for mode selection using decision points, synchronizing time with the DS3231, transmitting mode data to the Slave, controlling traffic lights, and updating displays, offering a clear visual guide to the Master’s role.

A diagram of a program

AI-generated content may be incorrect.

**Figure 10: Master Operation Flow Chart**

The Master unit is built around the STM32 microcontroller, which acts as the central hub for coordinating all system functions. This powerful microcontroller handles the processing of inputs, manages communication protocols, and controls the output devices, ensuring the system operates smoothly under various traffic conditions. Its setup includes connections to multiple peripherals, each playing a critical role in the unit’s operation, with power stability maintained by an AMS1117 voltage regulator that steps down the 9V power adapter input to a reliable 5V/3.3V supply. This foundational configuration enables the Master to serve as the decision-making core, sending instructions to the Slave and managing local traffic light operations effectively.

The DS3231 real-time clock module plays a vital role in ensuring accurate timing for the Master unit. It is set to operate in "alarm per one second" mode, producing a signal from its SQW pin that connects to the STM32 via an external interrupt (EXTI). This configuration greatly lowers power usage by removing the need for the microcontroller to constantly query the DS3231 for time, which could otherwise lead to energy drain or operational delays. Instead, the interrupt-based method allows the STM32 to act only when a new second is signaled, promoting efficient performance. The time data from the DS3231 is transferred to the STM32 through the I2C interface, where it is processed and then displayed on the 16x2 LCD. This arrangement guarantees that the Master unit maintains precise timekeeping, crucial for coordinating traffic light sequences and updating system status.

The DS3231 is configured to enable the one-second alarm, a choice made to save energy and optimize the system’s code efficiency. By generating an interrupt every second, the STM32 avoids continuous polling of the DS3231, reducing unnecessary processing overhead and power consumption. The module’s I2C address is set to 0x68, which is left-shifted by 1 bit (resulting in 0xD0 for write operations) to align with the I2C protocol’s 7-bit addressing format. To enable the alarm, specific values are written to the DS3231’s registers: the control register at address 0x0E is set to activate the alarm interrupt and configure the SQW pin for output, while the status register at 0x0F is cleared to reset any existing alarm flags, ensuring a clean operation. The alarm registers are also adjusted to define the one-second interval. Similarly, to set or read the time—including seconds, minutes, hours, day of week, day of month, month, and year—the same approach is used, writing or reading values to the timekeeping registers starting at address 0x00, with data converted to binary-coded decimal format for compatibility. This setup ensures the DS3231 provides a reliable and efficient timekeeping mechanism for the Master unit, supporting the overall functionality of the traffic light system.

The 16x2 LCD serves as the primary display for the Master unit, delivering real-time information to users about the current time and the active mode. It is interfaced with the STM32 through a PCF8574T I2C expander, which reduces the number of pins needed for connection and facilitates efficient data transfer. The STM32 updates the LCD at least every second, utilizing time data received from the DS3231 via the I2C bus. This consistent refresh allows operators and observers to accurately monitor the system’s status, whether it shows the current time or indicates the active mode—such as Normal, Heavy, Night, or Control. The clear and reliable display plays a crucial role in enhancing the system’s usability, serving as an essential tool for effective traffic flow management.

The setup of the LCD involves configuring the PCF8574T I2C expander with the address 0x4E, which is used by the STM32 to communicate with the display. After retrieving the real-time data from the DS3231, the STM32 processes this information and sends it to the LCD at the specified address to update the display with the current time. This address ensures precise targeting of the PCF8574T, enabling seamless data transfer and accurate rendering of time and mode information on the 16x2 LCD. The integration of this address into the system’s I2C communication framework supports the regular updates required for maintaining an effective and user-friendly interface.

The Master unit incorporates five push buttons (PB5 to PB9), connected to the STM32’s GPIO pins with pull-down resistors, to handle mode selection and manual control. The STM32 continuously monitors these inputs to detect when a button is pressed, triggering a change in the operating mode or a specific action. For instance, pressing PB5 selects Normal Mode, PB6 selects Heavy Mode, PB7 selects Night Mode, PB8 activates Control Mode, and PB9 allows manual toggling of traffic light states when in Control Mode. This input mechanism allows the system to adapt to different traffic conditions, with the STM32 processing each button press to determine the appropriate response. The use of pull-down resistors ensures stable input detection, preventing false triggers and enhancing the reliability of mode transitions.

The double 7-segment LEDs provide a visual countdown of the remaining time for each traffic light phase, offering drivers clear feedback on when the lights will change. These LEDs are driven by a 74HC595 shift register, which interfaces with the STM32 over the SPI protocol. When a mode is selected via the push buttons, the STM32 calculates the timing for the Red, Yellow, and Green lights—such as 30 seconds for Green, 5 seconds for Yellow, and 25 seconds for Red in Normal Mode—and sends this data to the shift register. The 74HC595 then updates the 7-segment displays accordingly, ensuring accurate and dynamic timing information is presented. This setup leverages the SPI interface’s speed and efficiency, making it ideal for real-time updates and enhancing the system’s visibility to users.

To display numbers effectively on the 7-segment LEDs, a table of values is constructed to map each digit to its corresponding segment pattern. This table allows the STM32 to send the appropriate data via the SDIO (Serial Data Input/Output) line of the SPI interface to the 74HC595 shift register. The SPI clock (SCLK) is used to synchronize the data transfer, while the NSS (Not Slave Select) pin, configured as the LOAD signal, triggers the update of the display with the new number once the data is fully shifted in. This method ensures precise control over the 7-segment LEDs, enabling the system to reflect the calculated timing accurately and maintain a responsive visual interface for drivers.

The HC-12 wireless module enables communication between the Master and Slave units, interfaced with the STM32 through the UART protocol. After the STM32 identifies the selected mode from the push button inputs, it sends this information to the Slave unit via the HC-12 module. This wireless connection ensures reliable operation, allowing the Slave to replicate the Master’s traffic light sequences without the need for wired connections. The UART protocol supports rapid and efficient data transfer, enabling the Master to communicate mode details in real time. This functionality is essential for maintaining synchronization throughout the system, ensuring the Master can control the operational state and achieve cohesive traffic management across both units.

The HC-12 module requires specific configuration to ensure optimal performance in the system. To set up the HC-12, the SET pin is driven low to enter configuration mode, granting access to send AT commands (ACK) for adjusting its settings. In this mode, the module is configured to operate at a baud rate of 9600 bps to match the system’s communication speed, and the channel is set to 0x88 to minimize interference and reduce noise in the wireless environment. Additionally, the transmit power is set to 20 dB, the maximum power level, to ensure a strong signal for reliable communication over distances up to 50 meters, even in noisy conditions. Once the configuration is complete, the SET pin is left floating during normal operation, allowing the HC-12 to function seamlessly in the circuit and maintain consistent wireless communication between the Master and Slave units.

1. **Slave Unit Setup and Functionality:**

The Slave unit’s operational flow is depicted in the flowchart below. It details the process of receiving mode instructions via UART interrupt from the Master through the HC-12 module, saving the mode, parsing the mode string, and controlling the traffic lights and display, providing a clear visual guide to the Slave’s responsive role in the system.

A diagram of a flowchart

AI-generated content may be incorrect.

**Figure 11: Slave Operation Flow Chart**

The HC-12 wireless module facilitates communication between the Master and Slave units, interfaced with the STM32 via the UART protocol. The STM32 uses a UART interrupt to read the incoming mode string, such as "normal" or "control", sent by the Master, immediately saving it to the system. This interrupt-driven approach prevents delays that could occur with continuous polling, ensuring timely and efficient mode updates. The STM32 then parses the saved string to identify the selected mode, enabling the Slave to adjust its traffic light sequence in real time. This wireless communication ensures synchronization without physical connections, maintaining system responsiveness.

The Slave unit controls its Red, Yellow, and Green LEDs via the STM32’s GPIO pins, operating in sync with the mode received from the Master. It works the same as the Master, executing sequences like 30 seconds Green, 5 seconds Yellow, and 25 seconds Red in Normal Mode, or blinking Yellow every 0.5 seconds in Night Mode, based on the parsed mode string.

The Slave unit includes a 7-segment LED display to show the remaining time for each traffic light phase, driven by a 74HC595 shift register interfaced with the STM32 over the SPI protocol. After parsing the mode string received via the HC-12, the STM32 calculates the timing for the Red, Yellow, and Green phases and sends this data to the 74HC595. The shift register then updates the 7-segment display to reflect the countdown, providing drivers with clear visual feedback. This SPI-based setup ensures efficient data transfer, keeping the display accurate and responsive to the selected mode’s timing requirements.

1. **Result:**
2. **Result and Conclusion:**

The wireless traffic light system effectively managed traffic flow using a Master-Slave wireless module with STM32F103C8T6 microcontrollers. It achieved stable wireless communication via HC-12 modules at 9600 bps over a 50-meter range in noisy environments. The DS3231 RTC ensured precise synchronization, updating every second via I2C, with real-time and mode status displayed on a 16x2 LCD. Double 7-segment LEDs, driven by 74HC595 shift registers via SPI, provided clear countdowns for traffic phases. The system supported multiple mode including, Heavy, Night and Control adapting to diverse traffic scenarios. The two-layer PCB design, with a top layer for signal routing and a bottom ground plane, minimized noise and ensured reliable operation.

This project successfully delivered a stable and scalable wireless traffic light system, enhancing road safety and driver convenience in both urban and rural residents. The integration of embedded systems, wireless communication, and clear displays proved robust and reliable. However, limitations such as communication range, lack of traffic monitoring, and absence of emergency vehicle prioritization were identified.

1. **Future Improvements**

To address these limitations, the following enhancements are proposed:

* Upgrade Communication Range: Replace the HC-12 module with a LoRa module to extend the communication range beyond 50 meters, enabling broader coverage for larger intersections.
* Add Traffic Monitoring: Integrate a surveillance camera with image processing to detect license plates and red-light violations, enhancing security and providing evidence for traffic enforcement.
* Prioritize Emergency Vehicles: Incorporate a priority signal detection module (e.g., RF, LoRa, or sound-based) to automatically adjust traffic lights for emergency vehicles like ambulances or police cars, reducing response times.

These improvements will enhance the system’s functionality, making it more adaptive and efficient for modern traffic management needs.