EMITOR: Jurnal Teknik Elektro

Development of a User Interface for Traffic Light Control Using PLC at a Four-Way Intersection

Linda Sartika[1], Abdul Muis Prasetia[*,1], Siti Nurholifah[1], Rahma Nur Amalia[2]
Department of Electrical Engineering: – University of Borneo Tarakan[1] – Politeknik Negeri Malang[2]
Tarakan & Malang, Indonesia
[*]prasetia.electric@gmail.com

Abstract – *In general, traffic light regulations currently use fixed times, where the lights are set to work alternately at predetermined intervals without considering real-time traffic flow conditions. For example, at four-way intersections, the green, yellow, and red light durations are uniform in each direction. This can lead to inefficient traffic management, such as long queues on one side and minimal queues on another, resulting in unnecessary delays. This research adopts an experimental method consisting of procurement, assembly, and trial phases. A Programmable Logic Controller (PLC) and User Interface (UI) are used to control and monitor the traffic light system at intersections. The system is designed and simulated using GX-Developer and GT-Designer3 software. The PLC automatically controls the operation of the lights based on commands from the UI. Light timing settings can be directly configured via the UI system. Comparative testing showed relative differences between the programmed timer, calculated timer, and stopwatch timer. The actual stopwatch readings for the green light duration were 41.79 seconds for intersections 1 and 2, 25.77 seconds for intersection 3, and 21.89 seconds for intersection 4. Yellow lights remained on for 3 seconds, and red lights for 1 second as a transition delay.*

Keywords – *Traffic Lights; PLC; UI; GX-Developer; GT-Designer3.*

I. Introduction

TRAFFIC lights are signals employed to regulate and direct the efficient flow of vehicular movement on roadways. According to Law Number 22 of 2009 regarding Road Traffic and Transportation, traffic signals are classified as Traffic Signaling Devices (APILL). Traffic lights are typically installed at crossroads, crossings, and other areas to regulate traffic flow and reduce congestion and accidents [1].

The traffic growth rate in Indonesian cities continues to rise annually, but this increase is not matched by road capacity expansion [2]. The growing number of road users, including motorcycles and cars, exacerbates this issue, especially at intersections where vehicle volumes exceed capacity [3]. This leads to traffic jams, particularly during morning and evening rush hours [4].

Current traffic light systems generally use a fixed-time configuration, where signal intervals are pre-set regardless of actual traffic conditions. This causes unnecessary delays, indicating a need for a smarter control system [5]. A static timing approach fails to adapt to dynamic traffic flow patterns, often leading to inefficiencies and driver frustration, especially when certain lanes remain empty while others experience long queues.

With rapid advances in technology, conventional control methods such as relay-based systems are being replaced by digital control systems like the Programmable Logic Controller (PLC) [6]. Technological innovations have produced tools that improve workflow across sectors, including industry [7]. Today, computers serve not only for documentation but also for processing data and interacting with machines [8]. According to NEMA, a PLC is a digital electronic device with memory for storing programs that perform control functions, including logic, sequencing, timing, counting, and arithmetic operations [9].

A traffic light generally uses three indicators: red for stop, green for go, and yellow as a warning [10]. For optimal function, traffic lights should respond to traffic density using a system known as the Automatic Traffic-Light Control System (ATCS) [11]. Some systems apply fuzzy logic to determine signal timing based on vehicle volume [12]. However, such adaptive systems

The manuscript was received on November 9, 2024, revised on March 17, 2025, and published online on March 28, 2025. Emitor is a Journal of Electrical Engineering at Universitas Muhammadiyah Surakarta with ISSN (Print) 1411 – 8890 and ISSN (Online) 2541 – 4518, holding Sinta 3 accreditation. It is accessible at https://journals2.ums.ac.id/index.php/emitor/index.

may be too complex or costly to implement in mid-scale cities or for training operators. As an alternative, semi-automatic or manually adjustable systems using user interfaces (UI) can offer sufficient flexibility and practicality for local traffic management [13].

The UI improves interaction between users and systems via graphic elements such as text, color, lines, buttons, and images [14]. Mobile applications are good examples of how UI can provide fast and efficient services [15]. By incorporating Human-Machine Interface (HMI) features, UI systems allow operators to visually monitor status and adjust operational parameters in real-time, making them ideal for traffic light management.

This study proposes designing and simulating a PLC-based traffic light system at a crossroad. The ladder diagram will be programmed using GX-Developer [16], and the UI will be created using GT-Designer3 [17]. This design integrates both hardware and software components to produce a functional model capable of controlling traffic lights based on adjustable timers, ensuring flexibility, ease of monitoring, and improved response to traffic conditions. The UI enables operators to manually monitor and control traffic conditions, reduce accidents, and address congestion during peak hours.

II. Research Methods

This section delineates the design, observation, and data gathering processes employed to evaluate the performance of the system developed and implemented utilizing programmable logic controllers (PLC) and user interfaces (UI). Additionally, the workflow of this research is succinctly delineated.

i. Programmable Logic Controller

A PLC is essentially a computer specifically engineered to regulate a process or machinery. The regulation of this regulated process involves the recurrent selection of components, akin to a servo system, or encompasses binary control (On/Off) executed frequently [18]. A PLC is a device that substitutes the functions of a sequential relay circuit within a panel control system [19]. A PLC is a programmable control device that utilizes a computer or laptop to manage the logic within the CPU (Central Processing Unit). The program format generated on the PLC is referred to as a ladder diagram, and instructions within the ladder diagram can be interpreted using Mnemonic Programming. Programmable Logic Controllers (PLCs) are widely used in industrial automation and control systems, offering flexibility, reliability, and robustness [20, 21]. PLCs utilize ladder diagram programming, which is easily understood by control professionals [22]. Recent advancements include energy management systems [23], smart home applications [20], and direct FPGA synthesis for faster response times [24]. To improve interoperability and software reuse, semantic models like OntoPLC have been developed [25]. User-friendly verification approaches, such as specification mining, enhance safety verification efficiency [26]. As the Internet of Things (IoT) era demands more complex control and networking capabilities, soft-PLCs with enhanced features like multi-threading and improved sensor interfacing are being designed to meet these challenges [27]. These developments aim to increase productivity, safety, and adaptability in industrial automation systems. The ladder diagram has input and output elements beside a voltage source, organized in a ladder-like configuration, and operates consecutively [28].

Ladder diagram is a programming language that employs logical symbols to illustrate the flow of logic in a PLC program, represented in the format of a ladder. The logic symbols include contact NO (Normally Open), NC (Normally Closed), and logical operators such as AND, OR, and NOT, along with specialized functions like Timer, Cost, and MOV. Mnemonic Code is a textual programming language utilized as a shorthand to delineate instructions and functionalities in PLC programs [29].

ii. User Interface

The user interface (UI) is an integral component of the application, facilitating seamless interaction between the application and the user. To obtain the outcomes of a design interface, it is necessary to utilize software such as GX-Developer for creating ladder diagrams intended for the PLC, as well as GT-Designer3 for designing a user interface system that will subsequently operate the traffic light program.

iii. Research Phases

This stage delineates the requisite processes in research that align with the study objectives. Literature review to analyze diverse sources or theories pertinent to the topic. Design a control system for traffic lights that can be operated manually to address research challenges. Upon completing the system design, proceed to test the traffic light control system; the initial test involves evaluating the ladder diagram, followed by the second test, which focuses on the user interface system. Following successful system testing, calculation analysis and comprehensive system analysis are conducted to ensure that the findings align with the specified objectives. Upon analysis, conclusions can be derived from the results, indicating that the research objectives align with those established.

iv. System Planning Traffic Lights

The system built in this research is a traffic light control system, through a laptop a system design will be created that can be controlled and monitored via the UI. This system uses a Mitsubishi PLC Type FX3U, on ladder PLC has a programming language that is used to create programs as a PLC system controller to produce a output that the user wants to produce as a control system traffic lights.

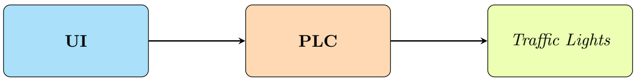

Figure 1: System Block Diagram

Figure 1 illustrates that the user interface (UI) functions as a visual system capable of presenting a designed display that is immediately perceivable by the user. In this user interface, the PLC is crucial for regulating the operating system by evaluating inputs and configuring output conditions as per the designer's specifications. The PLC section contains a CPU that organizes and processes commands, enabling the memory to store data with programmed instructions. These instructions can execute tasks such as timer functions or power-on configurations for traffic lights, generating signals that enter the PLC to produce output in the form of flashing traffic lights.

III. Results and Discussion

This research will elucidate the methodologies employed for design, experimentation, and data gathering. This is conducted to evaluate system testing in managing the duration. Traffic lights utilizing a PLC and a UI system designed as a monitoring tool for the human management of traffic lights in accordance with predetermined objectives.

i. PLC Configuration for Traffic Lights

This configuration is necessary for the planning of wiring traffic lights to streamline the device networking process. This concept utilizes the Mitsubishi brand PLC Type FX3U-64M, which features 32 inputs and 32 outputs, for traffic light systems. It employs a user interface as input and utilizes 12 output connections to connect the PLC with the traffic light module.

Figure 2 illustrates a PLC equipped with a voltage source, with an indicator light that conveys the status of the PLC as standby, off, or operational. This PLC

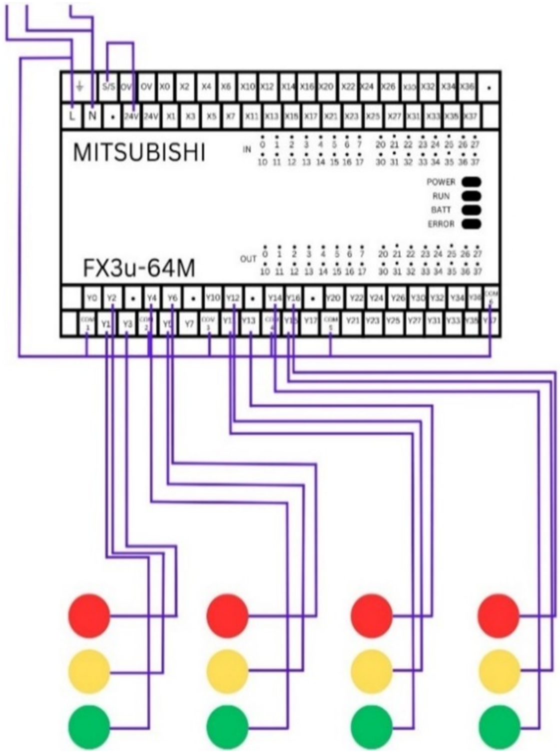

Figure 2: Wiring Traffic Lights

regulates the traffic light system prototype, with inputs and outputs managed by an uploaded program. The input utilized is a user interface system linked to the PLC with ON and OFF parameters. When the PLC is in the ON state, the program will access the PLC and retrieve program data, which will activate and supply output to the traffic light system prototype. The indicator lights will illuminate according to each address: Y1 for green, Y2 for yellow, and Y3 for red, corresponding to the connectors for each address in the traffic light wiring.

ii. PLC Address Identification

The traffic lights system was developed utilizing GX-Developer and GT-Designer3 software in the format of a ladder diagram. Several functions utilized in traffic light programs with ladder diagrams for identifying PLC addresses.

1. `X0`: Push button (OFF)
2. `X1`: Push button (ON)
3. `K`: Timer value
4. `M`: Internal memory
5. `Y1`: Green light at intersection 1
6. `Y2`: Yellow light at intersection 1
7. `Y3`: Red light at intersection 1
8. `Y4`: Green light at intersection 2
9. `Y5`: Yellow light at intersection 2
10. `Y6`: Red light at intersection 2
11. `Y11`: Green light at intersection 3
12. `Y12`: Yellow light at intersection 3

13. Y13: Red light at intersection 3
14. Y14: Green light at intersection 4
15. Y15: Yellow light at intersection 4
16. Y16: Red light at intersection 4
17. Timer 0: Green time 1
18. Timer 1: Yellow time 1
19. Timer 2: Red time 1
20. Timer 3: Yellow/red time 2
21. Timer 4: Green time 2
22. Timer 5: Yellow time 2
23. Timer 6: Red time 2
24. Timer 7: Yellow/red time 3
25. Timer 8: Green time 3
26. Timer 9: Yellow time 3
27. Timer 10: Red time 3
28. Timer 11: Yellow/red time 4
29. Timer 12: Green time 4
30. Timer 13: Yellow time 4
31. Timer 14: Red time 4
32. Timer 15: Yellow/red time 1
33. Timer 16: Reset program
34. Timer 17: Timer ON flip flop
35. Timer 18: Reset flip flop
36. Timer 24: Timer adds running time
37. Timer 25: Waiting time for green 1 before red 2
38. Timer 26: Green 2 waiting time before red 3
39. Timer 27: Waiting time for green 3 before red 4
40. Timer 28: Waiting time for green 4 before red 1

iii. I/O Module Traffic Lights Test Results

From Table 1, the following I/O traffic lights module test results show a ladder diagram, as well as an explanation of the I/O test results.

Table 1: Traffic lights I/O module test results

No	Component	Address	Result
1	Green indicator light at intersection 1	X000	Active
2	Yellow indicator light at intersection 1	X001	Active
3	Red indicator light at intersection 1	Y001	Active
4	Green indicator light at intersection 2	Y002	Active
5	Yellow indicator light at intersection 2	Y003	Active
6	Red indicator light at intersection 2	Y004	Active
7	Green indicator light at intersection 3	Y005	Active
8	Yellow indicator light at intersection 3	Y006	Active
9	Red indicator light at intersection 3	Y011	Active
10	Green indicator light at intersection 4	Y012	Active
11	Yellow indicator light at intersection 4	Y013	Active
12	Red indicator light at intersection 4	Y014	Active
13	Green indicator light at intersection 1	Y015	Active
14	Yellow indicator light at intersection 1	Y016	Active

Figure 3 illustrates that when the push button is activated, the indicator light signifies that input X001 is active until it generates output Y000, which can execute programs. Upon the initiation of the program, the

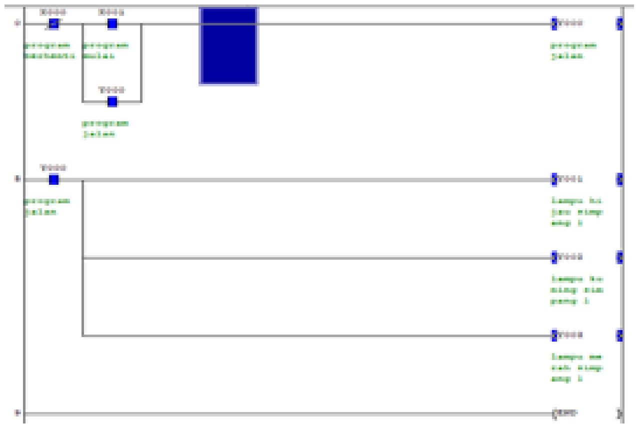

Figure 3: Ladder diagram for I/O module traffic lights testing

green, yellow, and red lights can be activated, namely Y001 for the green light at intersection 1, Y002 for the yellow light at intersection 1, and Y003 for the red light at intersection 1. The program may cease operation when the push button (off) with input X000 is activated, remaining inactive until the program halts or is not operational, which prevents output Y000 from executing. The I/O test generates 12 output lights that may illuminate from intersection 1 to intersection 4.

Table 2: Traffic light signal times

Approach	Red (s)	Yellow (s)	Green (s)	All Red (s)	Cycle Time (s)
North	102	3	42	1	148
East	102	3	42	1	148
West	118	3	26	1	148
South	122	3	22	1	148

The results of the timing study for the traffic lights above can be utilised for signal timing in the testing ladder diagram and user interface, with a timer configured via the UI system, allowing the green, yellow, and red lights at each intersection to be allocated time as specified in Table 2.

iv. User Interface Testing at Simpang Empat

This is a representation of the UI program for traffic lights and the corresponding ladder diagram, created with GX-Developer and GT-Designer3, utilising the signal timing data from Table 2, which has been preconfigured.

Figure 4 illustrates the procedure for inputting a timer value. In the user interface system, predefined relative values are utilised, namely values of timer multiplied by ten. For instance, if the desired timer value is 42 seconds, then 42 multiplied by 10 equals 420. Consequently, the configured timer value will be utilised in the timer user interface system, and to obtain the actual timer value on the stopwatch, the setting value must be

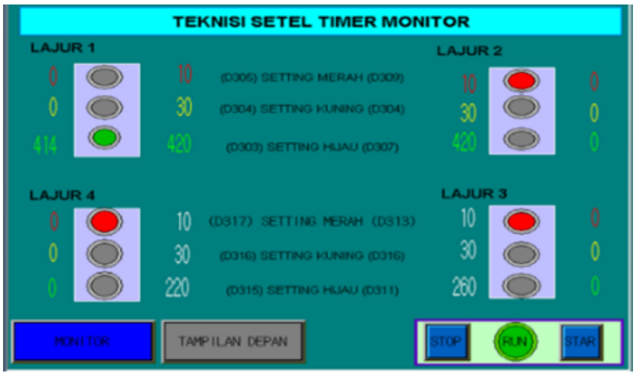

Figure 4: Display of determining UI timer settings at each intersection

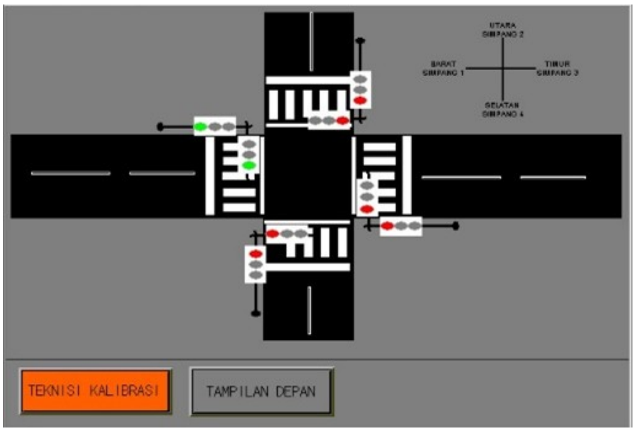

Figure 5: Traffic lights UI display at intersection 1 with green light on

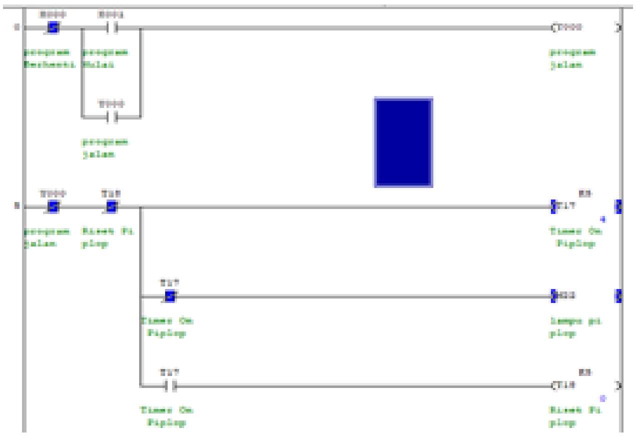

Figure 6: Ladder diagram display with flashing yellow light

Figure 7: Ladder diagram test of flashing yellow light on the traffic lights module

multiplied by 0.1 seconds.

The recorded durations at the green light for intersection 1 is 41.79 seconds, for intersection 2 is 41.79 seconds, for intersection 3 is 25.77 seconds, and for intersection 4 is 21.89 seconds. The yellow light duration at all intersections is 3 seconds, while the red light duration at all intersections is merely 1 second, representing the delay between the red and green lights at each intersection. By inputting a timer value for the green, yellow, and red lights at each intersection, the program can be promptly triggered and will operate according to the traffic light design depicted in Figure 5, with the green light illuminated at intersection 1.

Figure 6 illustrates that the Push Button functions as a Normally Closed (NC) contact and a Normally Open (NO) box. When the NC Y000 contact is engaged, the NC T17 contact activates, triggering the output light flip flop, which produces a flashing yellow indicator light at each intersection.

When T17 is a NO contact, T18 will activate to reset the yellow indicator light at each intersection, contingent upon the completion of a count timer. To ensure that the yellow light on each road remains illuminated when the yellow indicator light is blinking, contact M22 serves as an auxiliary contact to activate the yellow indicator light at each intersection during flashing conditions. Figure 7 illustrates the test documentation ladder diagram for the flashing yellow light on the traffic light module.

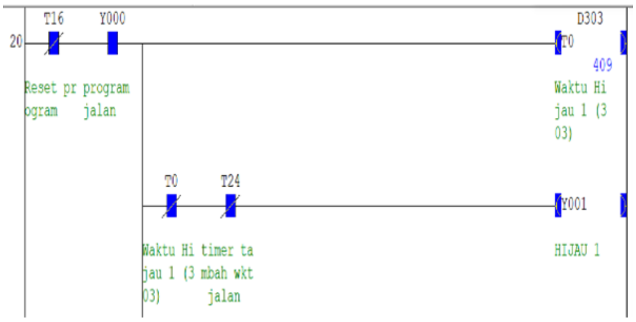

Figure 8: Ladder diagram view of green light at intersection 1

Figure 8 illustrates that at program initiation, T0 will be activated, subsequently activating Y001, the green light at intersection 1, allowing vehicles in lane 1 to proceed while halting vehicles in lanes 2, 3, and 4. Figure 9 is a test documentation ladder diagram

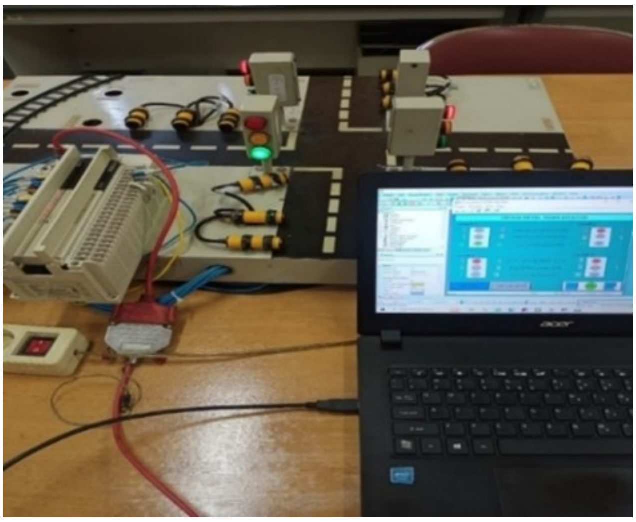

Figure 9: Testing the green light ladder diagram for intersection 1 on the traffic lights module

depicting a green light at intersection 1 on the module for traffic lights.

Figure 10: Ladder diagram view of yellow light at intersection 1

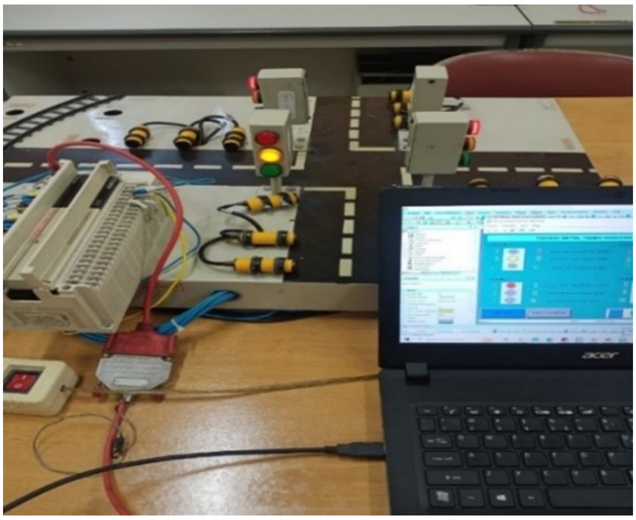

Figure 11: Testing the yellow light ladder diagram for intersection 1 on the traffic lights module

Figure 10 illustrates that when T0 deactivates, Y001, the green light at intersection 1, will similarly extinguish, and T0 will thereafter activate T1, which will in turn illuminate Y002, the yellow light at intersection 1. Figure 11 is a test documentation ladder diagram for the yellow light at intersection 1 on the traffic light module.

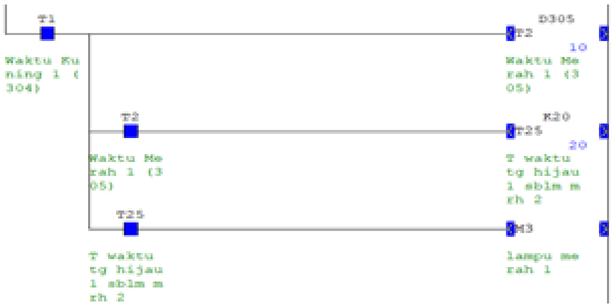

Figure 12: Ladder diagram view of red light at intersection 1

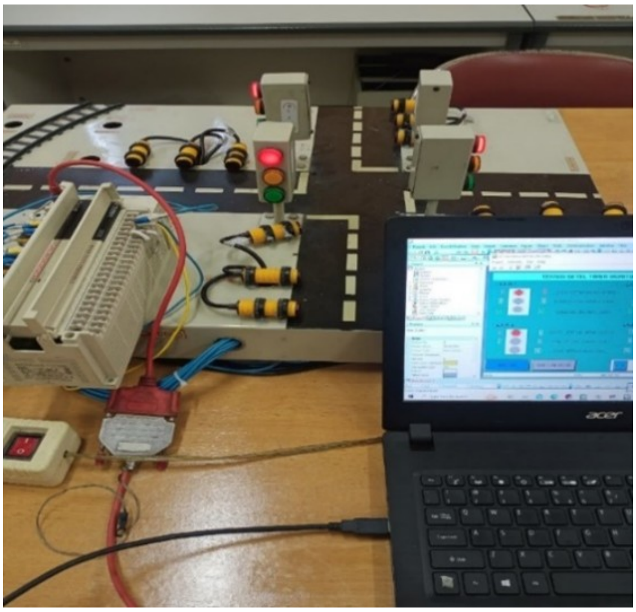

Figure 13: Testing the red light ladder diagram for intersection 1 on the traffic lights module

Figure 12 illustrates that when T1 deactivates, Y002, the yellow light at intersection 1, will similarly extinguish, and T1 will thereafter activate T2, which will in turn illuminate the red light at intersection 1. Figure 13 illustrates the test documentation ladder diagram for the red light at intersection 1 within the module for traffic lights.

v. UI System Timer Testing and Final Results

Table 3 elucidates the procedure for inputting a value timer. In the user interface system, predefined relative values are employed, namely values of timer multiplied by ten.

From the UI implementation depicted in Figure 14, the timing system was successfully configured using

Table 3: Testing the timer on the traffic lights module

Information	Timer	Result (s)	Stopwatch (s)
Green light at intersection 1	420	42	41.79
Yellow light at intersection 1	30	3	3
Red delay light at intersection 1	10	1	1
Green light at intersection 2	420	42	41.79
Yellow light at intersection 2	30	3	3
Red delay light at intersection 2	10	1	1
Green light at intersection 3	260	26	25.77
Yellow light at intersection 3	30	3	3
Red delay light at intersection 3	10	1	1
Green light at intersection 4	220	22	21.89
Yellow light at intersection 4	30	3	3
Red delay light at intersection 4	10	1	1

Figure 14: Traffic light UI System in Traffic light Module

values determined through analysis as outlined in Table 3. The results of UI testing with timer settings at each intersection are summarized in Table 4.

IV. Conclusion

The design employs a Mitsubishi PLC Type FX3U-64M, featuring 32 inputs and 32 outputs, for a traffic light system. It utilizes a user interface as input and connects to a traffic light module via a 12-output PLC connection. The achieved cycle time is 148 seconds, including green time for the north and east directions at 42 seconds, south at 26 seconds, and west at 22 seconds, with a yellow time of 3 seconds and a red delay of 1 second. The timing differences between the PLC program, the calculation results, and the stopwatch values indicate that timer input must be multiplied by 10 in the UI system, and then multiplied by 0.1 to reflect actual seconds. The system functioned as intended. The traffic light control and UI system performed efficiently, offering a monitoring interface that helped to ensure traffic flow based on pre-set timings.

Table 4: UI test results by determining the timing of traffic lights

No	Component	Information
1	Green light at intersection 1 on for 41.79s	Vehicles on lane 1 may go, lanes 2-4 stop
2	Yellow light at intersection 1 on for 3s	Turns on yellow light
3	Red delay at intersection 1 on for 1s	Red light delay between cycles
4	Green light at intersection 2 on for 41.79s	Vehicles on lane 2 may go, others stop
5	Yellow light at intersection 2 on for 3s	Turns on yellow light
6	Red delay at intersection 2 on for 1s	Red light delay between cycles
7	Green light at intersection 3 on for 25.77s	Vehicles on lane 3 may go, others stop
8	Yellow light at intersection 3 on for 3s	Turns on yellow light
9	Red delay at intersection 3 on for 1s	Red light delay between cycles
10	Green light at intersection 4 on for 21.89s	Vehicles on lane 4 may go, others stop
11	Yellow light at intersection 4 on for 3s	Turns on yellow light
12	Red delay at intersection 4 on for 1s	Red light delay between cycles

Acknowledgment

Gratitude is extended to the supervisor and faculty members of the Department of Electrical Engineering at the University of Borneo Tarakan for their guidance and support throughout this research project.

References

[1] F. Haradongan, “Kajian manajemen rekayasa lalu lintas di simpang perawang-minas kabupaten siak,” *Jurnal Penelitian Transportasi Darat*, vol. 21, no. 2, pp. 191–198, Dec. 2019.

[2] Y. P. Probo, P. Sigit, and M. Imam, “Pengaturan arus lalu lintas di kawasan pltu karangkandri cilacap (studi kasus: Ruas jalan lingkar timur cilacap),” *Jurnal Penelitian Transportasi Darat*, vol. 23, no. 1, pp. 1–17, Jun. 2021.

[3] S. Haryati and Najid, “Analisis kapastitas dan kinerja lalu lintas pada ruas jalan jenderal sudirman jakarta,” *JMTS: Jurnal Mitra Teknik Sipil*, vol. 4, no. 1, pp. 95–108, Feb. 2021.

[4] U. Zikra, “Perancangan prototype traffic light menggunakan arduino mikrokontroller berbasis antrian pada sebuah route persimpangan,” Ph.D. dissertation, Universitas Islam Negeri (UIN), 2022.

[5] Nursalim, "Desain sistem kontrol penanggulangan kemacetan lalu lintas adaptif berbasis plc," *Jurnal Media Elektro*, vol. 11, no. 1, Apr. 2022.

[6] W. M. Silaen, "Perancangan trainer plc omron cp1e sebagai sarana pembelajaran di laboratorium plc," Ph.D. dissertation, Universitas Medan Area, 2021.

[7] A. M. Prasetia, T. Hariyanto, A. Huda, L. Sartika, and Fitriani, "Monitoring dan kendali kecepatan motor universal menggunakan human machine interface (hmi)," *Elektrika Borneo (JEB)*, vol. 9, no. 1, pp. 28–35, Apr. 2023.

[8] C. E. W. Utomo and M. Hariadi, "Strategi pembangunan smart city dan tantangannya bagi masyarakat kota," *Jurnal Strategi dan Bisnis*, vol. 4, no. 2, Oct. 2016.

[9] D. Supriadi, "Rancang bangun sistem pembersihan dan pembilasan penampung air (toren) otomatis berbasis plc," *TEDC*, vol. 11, no. 3, Sep. 2017.

[10] Zulfikar, Tarmizi, and A. Adria, "Perancangan pengontrolan traffic light otomatis," *Jurnal Rekayasa Elektrika*, vol. 9, no. 3, Apr. 2011.

[11] I. Maulidya, "Optimisasi kinerja automatic traffic control system (atcs) di simpang dome kota balikpapan," *Warta Penelitian Perhubungan*, vol. 34, no. 2, pp. 119–128, 2022.

[12] M. R. Fauzi, "Implementasi fix timer berbasis kelompok waktu pada kasus persimpangan dalam kota (mengimplementasikan fungsi comparison)," Ph.D. dissertation, Universitas Komputer Indonesia, 2019.

[13] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, "Control of robot manipulators," *IEEE Control Systems Magazine*, vol. 27, no. 5, pp. 80–86, 2007. [Online]. Available: https://doi.org/10.1109/MCS.2007.904707

[14] M. A. Muhyidin, M. A. Sulhan, and A. Sevtiana, "Perancangan ui/ux aplikasi my cic layanan iinformasi akademik mahasiswa menggunakan aplikasi figma," *JURNAL DIGIT*, vol. 10, no. 2, pp. 208–219, Nov. 2020.

[15] D. A. Anggara, W. Harianto, and A. Aziz, "Prototipe desain user interface aplikasi ibu siaga menggunakan lean ux," *Jurnal Teknologi, Informasi dan Industri*, vol. 4, no. 1, Mar. 2021.

[16] A. Aziz, "Rancang bangun simulator traffic light berbasis plc mitsubishi type fx3u dan fx3u-enet-l," Ph.D. dissertation, Universitas Borneo Tarakan, 2017.

[17] A. D. Prabowo, "Pengaplikasian plc (programmable logic controller) untuk monitoring cara kerja pada modul pneumatik double acting cylinder," Ph.D. dissertation, Institute of Technology Surabaya, 2018.

[18] I. Setiawan, *Programmable Logic Controller dan Teknik Perancangan Sistem Kontrol*. Yogyakarta: Andi, 2006.

[19] F. E. Ahmad and E. Fitriani, "Penggunaan sistem outseal plc pada pemilih otomatis dan penghitung otomatis," Ph.D. dissertation, Universitas Palembang Indonesia, 2020.

[20] S. K. S. and S. Abdel-Nasser, "Design and Implementation of a Reliable and Secure Controller for Smart Home Applications Based on PLC," *Journal of Robotics and Control (JRC)*, 2022. [Online]. Available: https://journal.umy.ac.id/index.php/jrc/article/download/15972/7762

[21] A. S. Martin, L. Marten, W. M., U. I., W. M., N. Joerg, H. Stephan, N. M., and A. L. Edward, "Programmable Logic Controllers in the Context of Industry 4.0," *IEEE Transactions on Industrial Informatics*, 2021.

[22] S. Surawan and A. B. F. Edo, "Mengganti Mikrokontroller Mesin Dust Collector pada PT. xxx dengan PLC OMRON CP1E-NA20CD-R," *Journal of Electrical and Electronics Engineering*, 2021.

[23] A. K. A., "Management OF ENERGY CONSUMPTION USING PROGRAMMABLE LOGIC CONTROLLERS (PLC'S)," *Proceedings on Engineering Sciences*, 2021.

[24] A. Milik, M. Kubica, and D. Kania, "Reconfigurable Logic Controller—Direct FPGA Synthesis Approach," *Applied Sciences*, vol. 11, no. 18, p. 8515, sep 14 2021. [Online]. Available: http://dx.doi.org/10.3390/app11188515

[25] A. Yameng, Q. Fei-wei, C. Baiping, S. R., and W. Huifeng, "Ontoplc: Semantic Model of PLC Programs for Code Exchange and Software Reuse," *IEEE Transactions on Industrial Informatics*, 2021.

[26] X. Jiawen, Z. Gang, H. Yanhong, and S. Jianqi, "A User-Friendly Verification Approach for IEC 61131-3 PLC Programs," *Electronics*, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/4/572/pdf?version=1586328528

[27] M. Masoud, Y. Jaradat, A. Manasrah, and B. Taleb, "Designing of a General Purpose Soft Programmable Logic Controller (PLC) for the Internet of Things (IoT) Era," *International Review of Automatic Control (IREACO)*, vol. 13, no. 4, p. 153, jul 31 2020. [Online]. Available: http://dx.doi.org/10.15866/ireaco.v13i4.19328

[28] S. Sadi, "Implementasi human machine interface pada mesin heel lasting chin ei berbasis programmable logic controller," *Jurnal Teknik: Universitas Muhammadiyah Tangerang*, vol. 9, no. 1, pp. 18–24, Jan. 2020.

[29] I. G. S. Widharma, "Kajian analisis sistem kendali berbasis plc dalam dunia industri," Ph.D. dissertation, Politeknik Negeri Bali, 2021.