EMITOR: Jurnal Teknik Elektro

Design and Construction of Inverse Time & Definite Time Overcurrent Relay Based on Arduino Uno

Lucianus H. G., Ipniansyah, Abdurrahim, Toto N.*, Gusti F. F., Muhammad A. S. S., Rizky A. S.
Departement of Electrical Engineering – Samarinda State Polytechnic
Samarinda, Indonesia
*totonrh234@gmail.com

Abstract *– This system is designed to detect overcurrent faults, then disconnect them from the circuit. In inverse time overcurrent relay, the relay trip time will depend on the magnitude of the fault current. The greater the fault current, the shorter the relay trip time will be. Meanwhile, in definite time overcurrent relay, the relay trip time will work according to the specified time. This system consists of Arduino Uno, PZEM-004T, and relay module. PZEM-004T acts as a current sensor that can measure the current value flowing in the load. Arduino Uno is used as a microcontroller that can process data that has been measured through PZEM-004T. Then, if the current flowing in the load has exceeded the specified value, the Arduino Uno will activate the relay that can disconnect the current flowing in the load. The system was tested using an induction motor and variable resistor. The test results show that this system works effectively and efficiently, and can protect the load from damage due to overcurrent*

Keywords – *Arduino Uno; Overcurrent Relay (OCR); inverse time overcurrent relay; definite time overcurrent relay; PZEM-004T.*

I. Introduction

ELECTRICAL power systems are critical infrastructures that support industrial operations, commercial activities, and residential needs in modern society. The continuity, stability, and quality of electrical energy supply determine the operational reliability of many essential devices and services. Therefore, power system protection plays an important role in preventing equipment damage, ensuring safety, and minimizing outages caused by electrical disturbances.

One of the most common disturbances in distribution networks is overcurrent, which can occur due to short circuits, overloads, insulation failures, grounding faults, or network instability. If an overcurrent condition is not handled immediately, excessive thermal stress may occur, resulting in permanent damage to electrical components, reduced equipment lifespan, and even catastrophic hazards such as fires [1]. To overcome this issue, an appropriate protection system must be installed to detect and isolate the faulty section from the healthy system, preventing disturbance propagation.

Among the existing protection devices, Overcurrent Relay (OCR) is one of the most widely used technologies due to its simplicity, selectivity, and effectiveness as both primary and backup protection in distribution networks, transformers, generators, and transmission substations [2, 3]. OCR operation is carried out by comparing the measured current with a predefined threshold (pickup current). When the measured current exceeds this threshold, the relay will send a trip signal to disconnect the electrical load.

Two widely applied OCR operating characteristics include inverse time and definite time. Inverse Time Overcurrent Relay provides a shorter tripping duration as the fault current becomes higher, making it suitable for overload protection and protection coordination between multiple relay levels [4–9]. Conversely, Definite Time Overcurrent Relay operates with a constant tripping duration that is independent of the magnitude of the fault current, enabling precise and selective protection settings in multi-layered networks [10–13].

In line with the increasing need for practical learning media in protection technology, this study focuses on designing and implementing a prototype of an inverse time and definite time OCR using Arduino Uno,

The manuscript was received on November 5, 2025, revised on December 1, 2025, and published online on March 27, 2026. 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.

Copyright: © 2024 by the authors. This work is licensed under a Creative Commons Attribution–NonCommercial 4.0 International License (CC BY–SA 4.0)

integrated with PZEM-004T as current sensor, relay modules for switching, and an LCD display for monitoring. Apart from functioning as an overcurrent protection device, this prototype also serves as a training tool in educational laboratories to enhance understanding of microcontroller-based protective relay systems.

Experimental testing was carried out by applying various levels of fault currents to evaluate the accuracy of the current measurement and the consistency of the tripping characteristics based on theoretical curves. The performance evaluation aims to validate that the proposed system is able to operate quickly, selectively, and reliably according to inverse time and definite time protection principles.

II. Research Methods

The block diagram in Figure 1 represents the fundamental architecture of the microcontroller-based overcurrent protection system that has been designed and implemented in this research. The system is initiated by receiving electrical energy from a 220V AC mains source, which is the standard voltage level commonly used in residential and industrial power distribution networks. Since electronic control devices such as the Arduino Uno cannot directly operate at high AC voltage levels, an AC to DC conversion stage is placed at the front end of the system [14]. This converter plays a crucial role in transforming the high-voltage alternating current into a safe, stable low-voltage direct current supply. Failure to perform this conversion would not only render the system inoperable but could also cause severe damage to sensitive semiconductor components.

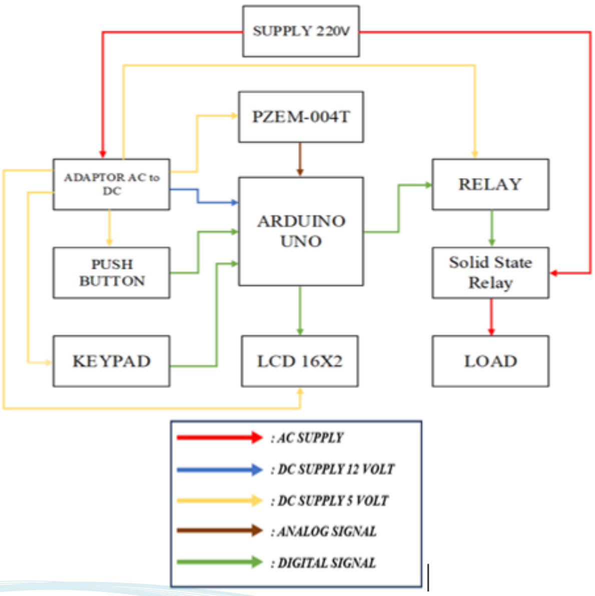

Figure 1: Block Diagram of the System

After the voltage is regulated, the DC power is distributed to various functional subsystems according to their operational requirements. As the central processing element, the Arduino Uno is responsible for acquiring real-time measurement data, executing computational algorithms, and commanding the actuation of the protection switching device [15]. Communication between the sensing subsystem and the controller is continuously established through digital serial communication, where the PZEM-004T sensor module transmits essential electrical characteristics such as load current, voltage, and power [16, 17]. These measurements serve as critical diagnostic inputs that form the basis of the system's decision-making logic in identifying abnormal and hazardous overload conditions.

To align the protection behavior with practical relay coordination strategies, the system provides human–machine interaction capabilities. A set of user-input components, including push buttons and a keypad, facilitate the configuration of key protection parameters. The operator may select between inverse time protection and definite time protection modes depending on the level of sophistication and coordination required in a given application. In inverse time mode, the system models traditional power system relays by adjusting the trip delay proportionately to the severity of the overcurrent. Conversely, in definite time mode, the relay executes a predetermined fixed delay once an overcurrent is detected. This duality enables the system to accommodate diverse fault scenarios and reinforces its applicability within both academic settings and practical field demonstrations.

Furthermore, the keypad allows users to set custom threshold current values and trip delays, offering flexibility in adapting the relay behavior to different types of electrical loads. To enhance operational transparency and monitoring accuracy, a 2 × 16 LCD with I2C communication is integrated to present real-time operating data, including actual load current levels, selected protection features, and the relay status. This ensures that the operator can monitor system performance continuously without the use of additional debugging equipment, allowing for efficient prototyping, testing, and field operation.

Once the Arduino detects that the measured current surpasses the configured protection threshold, it immediately activates the relay driver circuit. The relay then performs an electrical isolation process by disconnecting the load from the supply, preventing escalation of fault conditions such as excessive heating, component damage, and potential fire hazards. The disconnection event is also visually indicated through the LCD display, and the system enters a protective state while awaiting manual reset. This ensures that the system does not auto-reclose onto a still-existing fault, upholding safe restoration practices.

In summary, the block diagram in Figure 1 illustrates a compact yet highly functional protection scheme where sensing accuracy, intelligent programmable control, safety-oriented actuation, and intuitive user interaction are cohesively integrated. This emphasizes the potential of microcontroller-based relay systems to serve not only as effective learning tools for educational institutions but also as practical alternatives to conventional protection relays in small-scale installations requiring high flexibility, reliability, and cost efficiency.

i. Materials

This research utilizes a combination of essential hardware components that are carefully integrated to form an intelligent and autonomous overcurrent protection system. At the core of this system lies the Arduino Uno, a microcontroller development board powered by the ATmega328 architecture, equipped with 14 digital I/O pins and 6 analog input ports for sensor interfacing and control signal processing [18]. The Arduino Uno is responsible for performing computational tasks such as reading measurement data, executing relay protection algorithms, and determining trip decisions based on overcurrent conditions. Figure 2 shows the Arduino Uno used in this study, illustrating its structural design and pin configuration, which play a vital role in signal routing and correct wiring implementation.

For accurate monitoring of electrical parameters, the PZEM-004T module is employed as the main sensing component. This instrument is capable of measuring alternating current (AC), voltage, real-time power, and accumulated energy usage with a high degree of precision [19]. As shown in Figure 3(a), the measurement data generated from the PZEM-004T is transmitted digitally to the Arduino Uno, where the system further analyzes whether the detected current exceeds the predefined safety threshold. This continuous and dynamic data feedback ensures that the system can respond rapidly to abnormal load conditions.

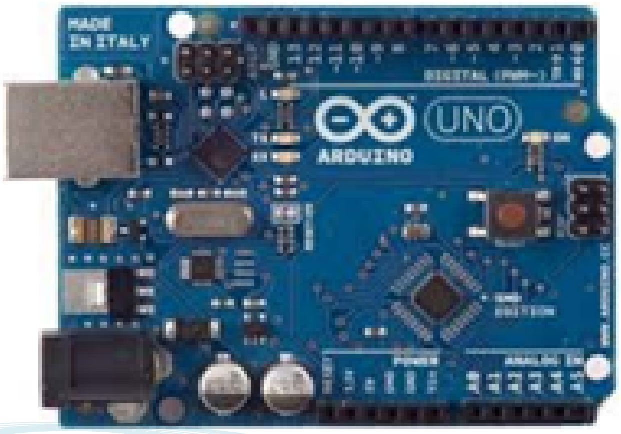

Figure 2: Arduino Uno as the main controller [18]

To support user interaction and enhance operational transparency, a 16x2 LCD module with I2C communication protocol is integrated, as displayed in Figure 3(b). The LCD not only serves as a visual display for current magnitude but also indicates system operation mode, trip status, and other critical messages [20]. This real-time display allows users or field operators to observe system behavior without needing external programming interfaces, making the prototype more suitable for practical field applications and academic demonstrations.

The switching mechanism of the protection function is controlled through a relay module illustrated in Figure 3(c). The relay operates as an electromechanical actuator that disconnects the power supply when an overcurrent fault occurs [21]. It is driven directly by the digital output signal from the Arduino, thus enabling a fast and automated response to potential electrical hazards.

As the supply input for this system originates from a standard 220V AC source, a reliable power conversion device is required to provide safe and stable DC voltage for all components. Therefore, a Switching Mode Power Supply (SMPS) is used to convert high-voltage AC into low-voltage DC, as shown in Figure 3(d). This module ensures that the microcontroller and sensing devices operate within their recommended voltage specifications, minimizing the risk of component failure due to voltage instability [22]. Furthermore, the XL6009 (CA-6009) DC-DC Step Up Module, shown in Figure 3(f), is employed to elevate the DC voltage from 5V to 12V whenever required, particularly for auxiliary components that demand higher voltage levels [23]. This flexibility in power adjustment improves modular compatibility and enhances system robustness.

To reinforce load isolation and ensure rapid switching for high AC current conditions, a Solid State Relay (SSR) is also included, illustrated in Figure 3(e). The SSR utilizes semiconductor-based switching, which reduces mechanical wear and provides faster operational response when compared to conventional relays [11]. Its integration ensures that the system can maintain stable performance even under frequent switching conditions, thereby increasing reliability, extending service life, and supporting safer protection functions.

In conclusion, all interconnected components presented in Figure 3 are assembled to form a unified protection system with strong measurement accuracy, efficient decision execution, and high operational reliability. This configuration supports the main objective of this research—developing an innovative, intelligent, and practical learning-based overcurrent relay prototype capable of enhancing safety in low-voltage electrical installations.

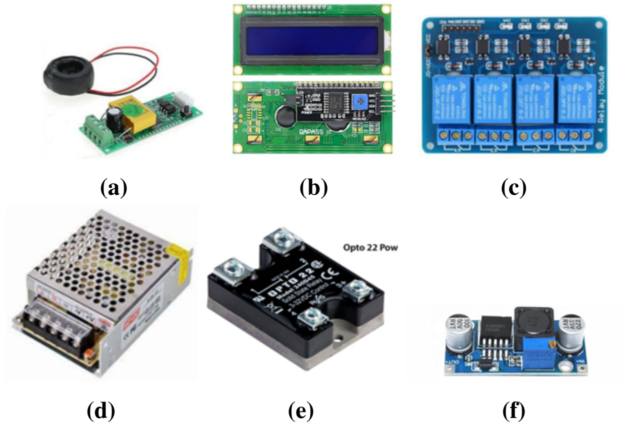

Figure 3: (a) PZEM-004T Module [19] (b) LCD I2C 16x2 [20] (c) Relay Module [21] (d) Switching Mode Power Supply (SMPS) (e) Solid State Relay (SSR) (f) XL6009 DC-DC Step Up Module

ii. System Workflow

Explanation of Figure 4: Figure 4 illustrates the operational workflow of the protection system, beginning with the activation of the device (Start), followed by continuous current monitoring through the input current sensor. When the sensed current exceeds the predetermined threshold limit, the system evaluates the selected operating mode to determine the appropriate protection response. If the inverse time mode is selected, the relay trip time becomes inversely proportional to the current magnitude, meaning that a larger fault current results in a faster relay trip as a measure of rapid protection against severe faults. Conversely, in the definite time mode, the relay operates based solely on a preset delay time regardless of the magnitude of the current, ensuring a consistent and predictable disconnection. Once the relay has operated and the fault has been cleared, the reset function restores the system to its normal monitoring status, ensuring continuous protection readiness. Thus, the workflow demonstrates a decision-based protection mechanism designed to maintain system safety and reliability under abnormal current conditions.

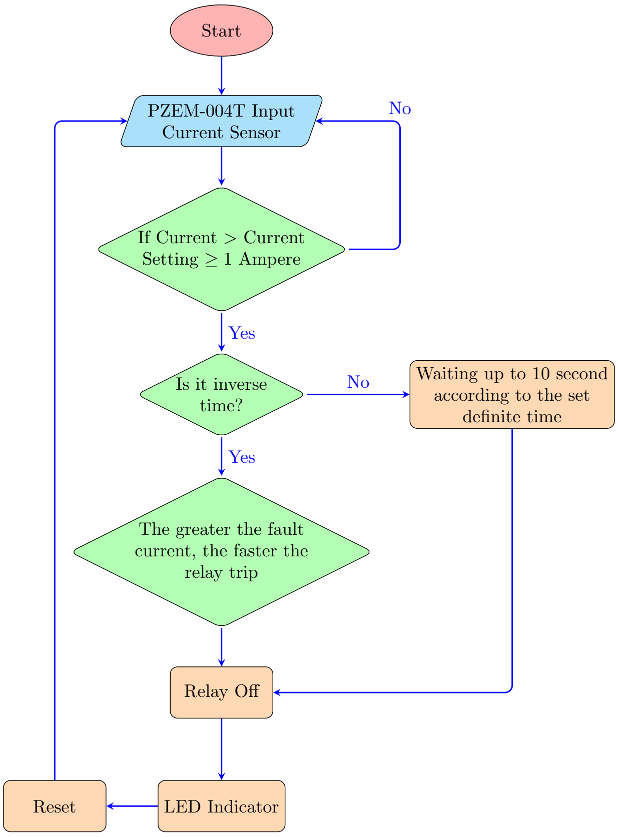
Start
PZEM-004T Input Current Sensor
No
If Current > Current Setting ≥ 1 Ampere
Yes
Is it inverse time?
No
Waiting up to 10 second according to the set definite time
Yes
The greater the fault current, the faster the relay trip
Relay Off
Reset
LED Indicator

Figure 4: System Workflow

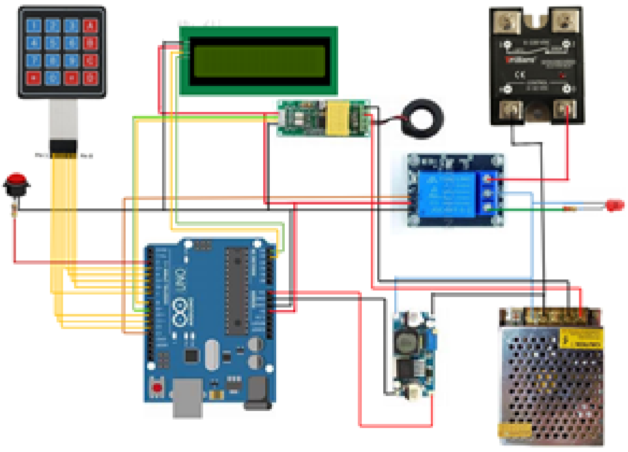

Figure 5: Wiring Diagram of the System

iii. Wiring Diagram

Figure 5 illustrates the overall wiring configuration that enables full integration between the Arduino Uno and all peripheral components, allowing the system to operate in accordance with the programmed relay protection logic. The Arduino Uno functions as the main microprocessor-based controller that continuously processes current data obtained from the PZEM-004T sensor. Once the real-time current value is read, the Arduino compares it with the user-defined setting to determine whether an overcurrent condition is present. If the current exceeds the set threshold, the Arduino immediately initiates the appropriate protection sequence by calculating the operating delay time based on the selected characteristic—either inverse time, where higher fault currents lead to faster tripping, or definite time, where the delay remains constant regardless of current magnitude. After the decision-making stage, the Arduino sends a control signal to the relay output to disconnect the electrical circuit and protect the load from damage. To support flexible system configuration, a keypad is interfaced with the Arduino, allowing users to input current limit values and choose between inverse or definite time modes. These parameters directly influence the behavior of the protection algorithm, ensuring the system can adapt to different load characteristics and safety requirements. The LCD 16x2 with I2C interface provides a real-time visual display of system conditions, including voltage values, current readings, selected mode, and relay status. This display improves operational awareness and simplifies user interaction since it helps users verify whether the input settings correspond to the expected behavior.

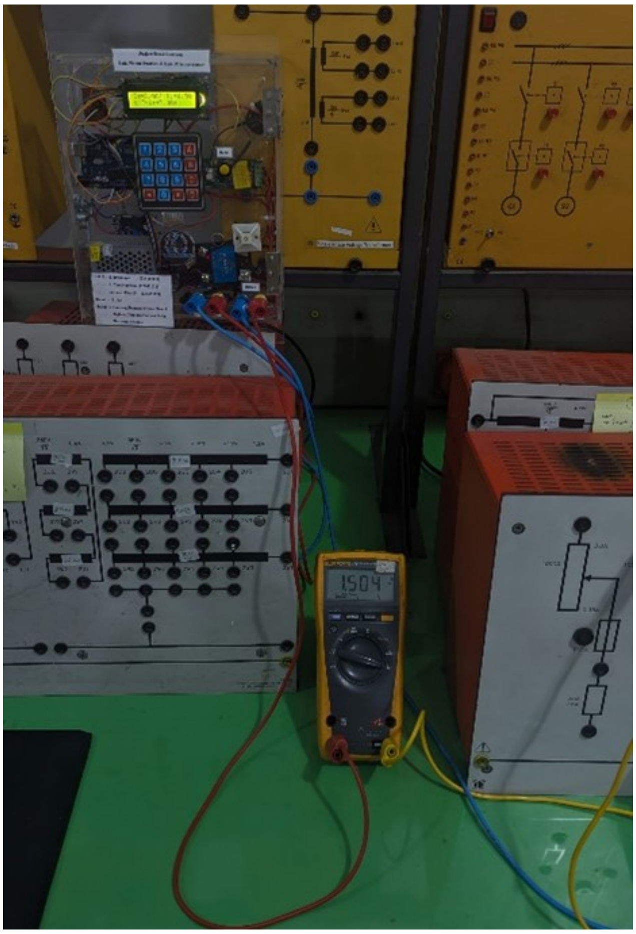

Figure 6: Inverse Time Testing with $I_s = 0.5$ A and $t_{I>} = 0.1$ s

The PZEM-004T module plays a crucial sensing role by delivering accurate current measurements directly to the Arduino. This information is essential because any error in detection may lead to delayed relay operation or unwanted disconnection. The relay module serves as the primary actuator in isolating the load whenever a fault occurs. Its COM, NO, and NC terminals direct the power flow, with the NO terminal connected to an LED indicator for visual confirmation of disconnection, and the NC terminal connected to the Solid-State Relay (SSR) for high-voltage load isolation. The SSR ensures safe switching of AC loads without mechanical wear, increasing overall system durability. Power distribution also plays a vital role in ensuring stable operation. A Switched Mode Power Supply (SMPS) converts high-voltage AC (220V) to low-voltage DC (5V), supplying the Arduino and sensors with stable power. However, some modules require a higher operating voltage; therefore, an XL6009 DC-DC Step-Up module is used to boost the voltage from 5V to 10V, fulfilling the needs of components requiring elevated supply levels. Additionally, a buzzer can be integrated to provide audible notification when the relay is triggered, confirming to the operator that the protection mechanism is actively responding to a fault.

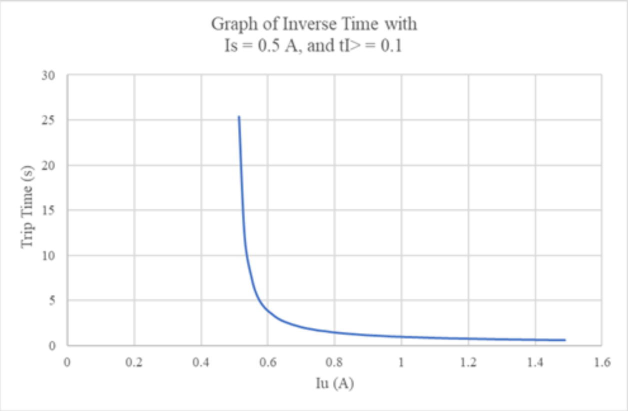

Figure 7: Graph of Inverse Time with $I_s = 0.5$ A and $t_{I>} = 0.1$ s

Through the combination of sensing, processing, configuration, indication, and controlled disconnection, the wiring system in Figure 13 demonstrates a reliable, automated overcurrent protection mechanism. Each component has a critical function that supports a logical chain of operation - from fault detection to safe load isolation - ensuring improved safety and system reliability in practical electrical applications.

III. Results and Discussion

In the inverse time test, a variable resistor is used as the load and a Fluke Digital Multimeter is used as the measuring instrument. The setting current value (I_s) is determined to be 0.5 A and $t_{I>} = 0.1$ s. The test was conducted by taking 50 measurement samples. The inverse time test results are shown in Table 1.

Table 1: Inverse time relay test results for current setting $I_s = 0.5$ A

No.	I_s (A)	I_u (A)	I_u on DMM (A)	$t_{\text{Trip Formula}}$ (s)	$t_{I>}$ (s)	t_{Trip} (s)	Error (%)
1	0.5	0.51	0.513	25.34	0.1	25.30	0.15
2	0.5	0.53	0.530	12.00	0.1	11.89	0.91
3	0.5	0.55	0.552	7.33	0.1	7.27	0.81
4	0.5	0.57	0.570	5.33	0.1	5.22	2.06
5	0.5	0.59	0.590	4.22	0.1	4.13	2.13
6	0.5	0.61	0.612	3.51	0.1	3.46	1.42
7	0.5	0.63	0.630	3.02	0.1	2.96	1.98
8	0.5	0.65	0.650	2.66	0.1	2.62	1.50
9	0.5	0.67	0.671	2.38	0.1	2.34	1.68
10	0.5	0.69	0.691	2.16	0.1	2.13	1.38
11	0.5	0.71	0.710	1.98	0.1	1.95	1.51
12	0.5	0.73	0.731	1.84	0.1	1.87	1.63
13	0.5	0.75	0.750	1.71	0.1	1.68	1.75
14	0.5	0.77	0.774	1.61	0.1	1.59	1.24
15	0.5	0.79	0.791	1.52	0.1	1.49	1.97
16	0.5	0.81	0.810	1.44	0.1	1.43	0.69
17	0.5	0.83	0.830	1.37	0.1	1.36	0.72
18	0.5	0.85	0.851	1.31	0.1	1.30	0.76
19	0.5	0.87	0.873	1.25	0.1	1.24	0.80
20	0.5	0.89	0.890	1.20	0.1	1.19	0.83
21	0.5	0.91	0.910	1.16	0.1	1.15	0.86
22	0.5	0.93	0.931	1.12	0.1	1.11	0.89
23	0.5	0.95	0.952	1.08	0.1	1.07	0.92
24	0.5	0.97	0.970	1.04	0.1	1.04	0.00
25	0.5	0.99	0.994	1.01	0.1	1.01	0.00
26	0.5	1.01	1.010	0.98	0.1	0.98	0.00
27	0.5	1.03	1.031	0.96	0.1	0.95	1.04
28	0.5	1.05	1.050	0.94	0.1	0.93	1.06
29	0.5	1.07	1.071	0.91	0.1	0.90	1.09
30	0.5	1.09	1.090	0.89	0.1	0.88	1.12
31	0.5	1.11	1.110	0.87	0.1	0.87	0.00
32	0.5	1.13	1.130	0.85	0.1	0.85	0.00
33	0.5	1.15	1.150	0.83	0.1	0.83	0.00
34	0.5	1.17	1.171	0.82	0.1	0.81	1.21
35	0.5	1.19	1.190	0.80	0.1	0.79	1.25
36	0.5	1.21	1.210	0.79	0.1	0.78	1.26
37	0.5	1.23	1.233	0.77	0.1	0.76	1.29
38	0.5	1.25	1.250	0.76	0.1	0.75	1.31
39	0.5	1.27	1.270	0.74	0.1	0.74	0.00
40	0.5	1.29	1.291	0.73	0.1	0.73	0.00
41	0.5	1.31	1.310	0.72	0.1	0.72	0.00
42	0.5	1.33	1.330	0.71	0.1	0.70	1.40
43	0.5	1.35	1.350	0.70	0.1	0.69	1.42
44	0.5	1.37	1.370	0.69	0.1	0.68	1.44
45	0.5	1.39	1.391	0.68	0.1	0.67	1.47
46	0.5	1.41	1.410	0.67	0.1	0.67	0.00
47	0.5	1.43	1.430	0.66	0.1	0.66	0.00
48	0.5	1.45	1.450	0.65	0.1	0.65	0.00
49	0.5	1.47	1.471	0.64	0.1	0.64	0.00
50	0.5	1.49	1.490	0.63	0.1	0.63	0.00

From Table 1, it can be seen that the larger the fault current, the faster the relay trips. The data relationship is shown in Figure 7. The results show that the system follows the inverse time characteristics, where the relay trip time decreases as the fault current increases. The average current reading error between the LCD and the Fluke Multimeter is 0.28%.

i. Definite Time Testing

In the definite time test, $I_s = 0.5$ A and the relay trip time is set to 5 s.

Measurement samples were taken 10 times. The results are presented in Table 2.

A graph of the resulting data is shown in Figure 9.

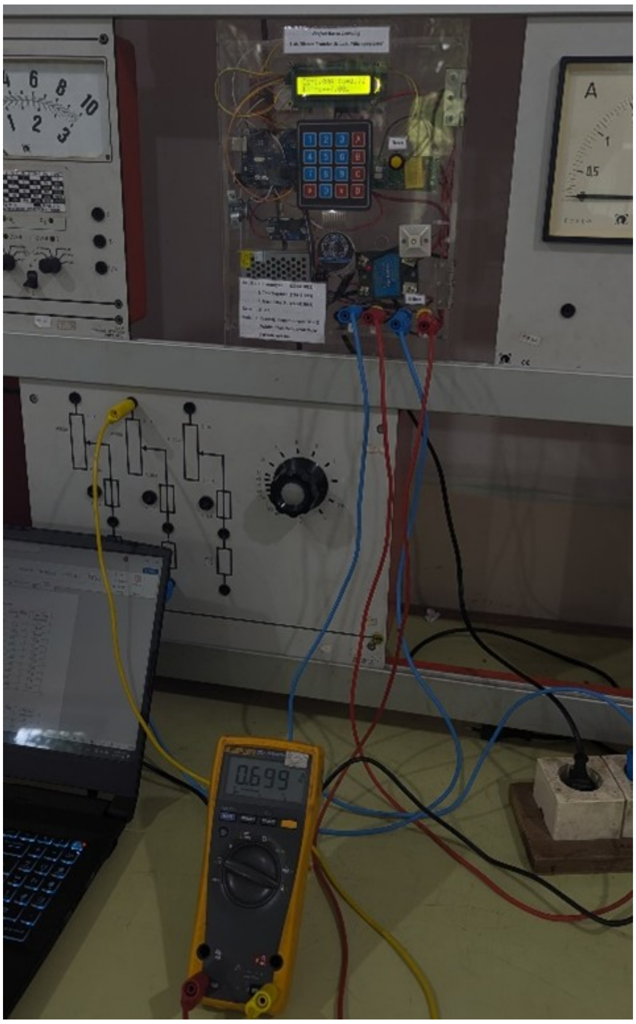

Figure 8: Definite Time Testing with $I_s = 0.5$ A and trip time = 5 s

Table 2: Definite Time Testing with trip time = 5 s

No.	I_s (A)	I_u (A)	I_u DMM (A)	t Trip (s)
1	0.5	0.61	0.635	5
2	0.5	0.72	0.713	5
3	0.5	0.82	0.814	5
4	0.5	0.91	0.902	5
5	0.5	1.05	1.027	5
6	0.5	1.12	1.12	5
7	0.5	1.23	1.124	5
8	0.5	1.31	1.30	5
9	0.5	1.42	1.43	5
10	0.5	1.52	1.52	5

Based on the results, the fault current magnitude does not affect the relay trip time, which always remains at 5 seconds. The difference between current measurements on the LCD and multimeter is also 0.28%, showing good system accuracy.

IV. Conclusion

Based on the results, the inverse time and definite time overcurrent relay based on Arduino Uno successfully

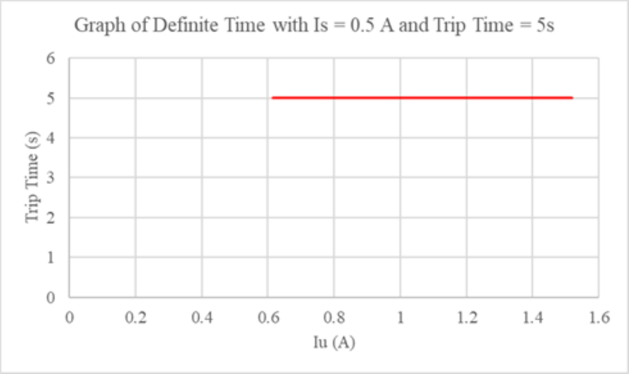

Figure 9: Graph of Definite Time with $I_s = 0.5$ A and trip time = 5 s

detects and disconnects overcurrent conditions.

For inverse time, trip duration is inversely proportional to the fault current. For definite time, trip duration remains constant regardless of current. The average current reading error is only 0.28%, indicating good measurement accuracy.

References

[1] M. Y. Suliman and M. Ghazal, "Design and implementation of overcurrent protection relay," *Journal of Electrical Engineering & Technology*, vol. 15, no. 4, pp. 1595–1605, 2020. [Online]. Available: https://doi.org/10.1007/s42835-020-00447-0

[2] M. Rojnić, R. Prenc, H. Bulat, and D. Franković, "A comprehensive assessment of fundamental overcurrent relay operation optimization function and its constraints," *Energies*, vol. 15, no. 4, p. 1271, 2022. [Online]. Available: https://doi.org/10.3390/en15041271

[3] M. Noman, I. Ullah, M. A. Khan, A. Qazi, W. Farooq, A. Saqr, and A. Elsheikh, "Analysis of overcurrent protective relaying as minimum adopted fault protection for small-scale hydropower plants," *International Journal of Environmental Science and Technology*, vol. 21, no. 4, pp. 4457–4470, 2024. [Online]. Available: https://doi.org/10.1007/s13762-023-05284-y

[4] N. El-Naily, S. M. Saad, and F. A. Mohamed, "Novel approach for optimum coordination of overcurrent relays to enhance microgrid earth fault protection scheme," *Sustainable Cities and Society*, vol. 54, p. 102006, 2020. [Online]. Available: https://doi.org/10.1016/j.scs.2019.102006

[5] D. Tarasov, R. Solopov, and I. Kindenkova, "Application of integral values in relay protection and emergency automation devices based on arduino," in *2025 International Russian Smart Industry Conference (SmartIndustryCon)*, 2025, pp. 561–565. [Online]. Available: https://doi.org/10.1109/SmartIndustryCon65166.2025.10986214

[6] A. Rauf, B. Y. Dewantara, and I. Winarno, "Design of protection relay prototype against arduino uno overload interference in electrical installations," *Fidelity: Jurnal Teknik Elektro*, vol. 5, no. 3, pp. 168–174, 2023. [Online]. Available: https://doi.org/10.52005/fidelity.v5i3.168

[7] M. A. Ibrahim, W. K. Ibrahim, and A. N. Hamoodi, "Design and implementation of overcurrent relay to protect the transmission line," *International Journal of Engineering Research and Technology*, vol. 13, no. 11, pp. 3783–3789, 2020.

[8] R. Rusdiansyah, R. A. Susilo, A. S. Kadang, S. M. Setiaji, and A. Ihsandi, "Design of constant time overcurrent relay single phase based on arduino uno," *International Journal of Current Science Research and Review*, vol. 7, no. 11, pp. 8559–8565, 2024. [Online]. Available: https://doi.org/10.47191/ijcsrr/V7-i11-42

[9] M. H. K. Khan, M. A. Al Rakib, and N. Sumaiya, "Design and performance evaluation of numerical relay for three-phase induction motor protection," *International Journal of Smart Grid*, vol. 7, no. 2, pp. 46–52, 2023.

[10] S.-T. Lim and S.-H. Lim, "Analysis on protective coordination between over-current relays with voltage component in a power distribution system with sfcl," *IEEE Transactions on Applied Superconductivity*, vol. 30, no. 4, pp. 1–6, 2020. [Online]. Available: https://doi.org/10.1109/TASC.2020.2968252

[11] R. A. Susilo, M. R. Sidiq, E. N. Hendra, A. H. Kurniawan, and I. A. Riyanto, "Design and development of overcurrent protection relay inverse definite minimum time type based on arduino uno," *International Journal of Current Science Research and Review*, vol. 7, no. 10, 2024. [Online]. Available: https://doi.org/10.47191/ijcsrr/V7-i10-01

[12] J. L. Santos, R. C. Miguel, A. I. Mendoza, and J. A. Garcia, "Coordinated overcurrent relay development with arduino and acs712," *Journal of Engineering, Technology, and Applied Science*, vol. 5, no. 3, pp. 113–120, 2023. [Online]. Available: https://doi.org/10.36079/lamintang.jetas-0503.387

[13] A. Hasibuan, M. Hafidzuddin, M. Jannah, D. R. Jintaka, and G. S. Kerimzade, "Development of 220v overcurrent relay protection system based on internet of things," *Andalas Journal of Electrical and Electronic Engineering Technology*, vol. 5, no. 1, pp. 17–22, 2025. [Online]. Available: https://doi.org/10.25077/ajeeet.v5i1.40

[14] C. S. Clemente, D. Davino, I. Iannone, and V. P. Loschiavo, "A real-time arduino based ac-dc boost converter for vibration energy harvesting devices," *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pp. 1–6, 2022. [Online]. Available: https://doi.org/10.1109/COINS54846.2022.9854992

[15] Anuj, Radhe Kishan, Shree Kishan, Shiwangi Singh, and Ashish Kumar Mishra, "Arduino-BASED INTELLIGENT INDUCTION MOTOR PROTECTION SYSTEM," *EPRA International Journal of Multidisciplinary Research (IJMR)*, pp. 717–720, apr 26 2025. [Online]. Available: http://dx.doi.org/10.36713/epra21243

[16] R. Silaban, M. E. Dalimunthe, and Z. , "Implementasi Sistem Proteksi dan Pemantauan Daya Rumah Berbasis IoT," *Power Elektronik : Jurnal Orang Elektro*, vol. 13, no. 3, pp. 321–325, oct 4 2024. [Online]. Available: http://dx.doi.org/10.30591/polektro.v13i3.7135

[17] Y. Prasetyo, S. Triwijaya, A. Khakim, D. N. Prakoso, and B. Winarno, "Integrated IoT System for Real-Time Electrical Load Monitoring," *Journal Geuthee of Engineering and Energy (JOGE)*, vol. 4, no. 1, pp. 17–23, may 26 2025. [Online]. Available: http://dx.doi.org/10.52626/joge.v4i1.57

[18] A. Darabi, M. Bagheri, and G. B. Gharehpetian, "Highly sensitive microgrid protection using overcurrent relays

with a novel relay characteristic," *IET Renewable Power Generation*, vol. 14, no. 7, pp. 1201–1209, 2020. [Online]. Available: https://doi.org/10.1049/iet-rpg.2019.0793

[19] M. N. Uddin, N. Rezaei, and O. E. Olufemi, "Adaptive and optimal overcurrent protection of wind farms with improved reliability," *IEEE Transactions on Industry Applications*, vol. 58, no. 3, pp. 3342–3352, 2022. [Online]. Available: https://doi.org/10.1109/TIA.2022.3147151

[20] J. P. Nascimento, N. S. D. Brito, and B. A. Souza, "An adaptive overcurrent protection system applied to distribution systems," *Computers & Electrical Engineering*, vol. 81, p. 106545, 2020. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2019.106545

[21] A. A. Hameed, A. J. Sultan, and M. F. Bonneya, "Design and implementation a new real time overcurrent relay based on arduino," in *IOP Conference Series: Materials Science and Engineering*, vol. 871, no. 1, 2020, p. 012005.

[22] A. A. Hameed, A. J. Sultan, and M. F. Booneya, "Design and implementation a new real time overcurrent relay based on arduino mega," in *IOP Conference Series: Materials Science and Engineering*, vol. 881, no. 1, 2020, p. 012142.

[23] A. Arbain, R. A. Susilo, A. S. Akbar, M. Wahyu, and M. Risky, "Arduino-based overcurrent relay design with very inverse type," *International Journal of Current Science Research and Review*, vol. 7, no. 11, pp. 8547–8558. [Online]. Available: https://doi.org/10.47191/ijcsrr/V7-i11-41