



# MEASUREMENT OF OPERATING TIME OF IDMT RELAY USING ARDUINO UNO

<sup>1</sup>Bharat Sorte, <sup>2</sup>Jayesh Chhabile, <sup>3</sup>Abhishek Deshmukh, <sup>4</sup>Mahima Jadhav, <sup>5</sup>Nitin Bardiya,

<sup>6</sup>Dr. Vyanktesh Dhotre

<sup>1</sup>Group Leader, <sup>2</sup>Group Member, <sup>3</sup>Group Member, <sup>4</sup>Group Member, <sup>5</sup>Group Member, <sup>6</sup>Guide,

<sup>1</sup>Electrical Engineering,

<sup>1</sup>Government College of Engineering Jalgaon, Jalgaon, India.

**Abstract:** This paper presents an in-depth exploration and implementation of an Arduino-based testing system to measure the operating time of an Inverse Definite Minimum Time (IDMT) relay. IDMT relays play a critical role in power system protection by offering time-delayed responses based on fault current magnitude. Traditional testing of these relays is often reliant on expensive and complex test equipment. The proposed system utilizes Arduino Uno and associated circuitry to provide a cost-effective, accessible solution for simulating faults and measuring relay operating time. The paper also discusses the full working principle, circuit design, Arduino programming, and expected outcomes. Applications in education, industry, and research are addressed, and the system's potential for future enhancements is highlighted. This approach supports a practical understanding of relay protection systems, enabling hands-on experimentation and learning.

**Keywords - IDMT Relay, Arduino Uno, Operating Time, Power System Protection, Overcurrent Relay, Fault Simulation, Electromechanical Protection.**

## I. INTRODUCTION

Power system protection is fundamental to the operation and safety of electrical grids. Inverse Definite Minimum Time (IDMT) relays are overcurrent protection devices whose operating time decreases as the magnitude of fault current increases [1]. These relays ensure timely fault clearance and proper coordination between protection devices. However, traditional methods for relay testing involve expensive kits and require expert knowledge, making them inaccessible in academic settings.

This paper proposes a low-cost, Arduino Uno-based platform to simulate fault conditions and accurately measure the relay's operating time. The system bridges the gap between theoretical studies and practical application, providing a hands-on tool for students and researchers alike.

## II. LITERATURE REVIEW

The measurement of relay operating time has long been a fundamental aspect of protection system engineering. Inverse Definite Minimum Time (IDMT) relays are widely used in medium-voltage systems for overcurrent protection [1] due to their inverse time-current characteristics. Traditionally, the operating time of these relays has been measured using analogue test kits, which rely on manual fault simulation, mechanical timers, and analogue circuitry. These analogue systems, while functional, present several limitations, especially in educational and evolving technological contexts.

### ➤ Limitations of Existing Analogue Kits

Analogue relay testing kits, though still used in many engineering institutions, are restricted in terms of precision, flexibility, and data handling. They typically involve:

- Manual activation of fault conditions using toggle switches or mechanical contacts.
- Timing mechanisms based on analogue stopwatches or mechanical counters are prone to human error.
- No data logging capability for trend analysis or long-term study.
- Lack of automation and difficulty in interfacing with modern digital systems or SCADA environments.

These factors limit the depth of experimentation and the accuracy of results, particularly in fast-switching scenarios where millisecond precision is crucial.

### ➤ Advancement through Microcontroller Technology

Recent technological advancements have led to a surge in the adoption of microcontroller platforms such as Arduino, Raspberry Pi, and ESP32 in protection system research and educational applications. These systems are programmable, low-cost, and highly flexible, offering precision time measurement using internal timers and interrupt-driven logic.

Studies such as Jadhav et al. (2018) and various projects across engineering institutions have demonstrated Arduino's capability in:

- Capturing precise time intervals (milliseconds or microseconds).
- Interfacing with LCDs and serial monitors for real-time output display.

- Logging relay timings and generating fault scenarios automatically.
  - Reducing cost drastically compared to commercial digital relay test kits.
- Arduino's `millis()` and `micros()` functions offer accurate time capture between two electrical events (such as fault initiation and relay tripping), which is essential for profiling the inverse time characteristics of IDMT relays.

### III. PROBLEM STATEMENT

Conventional testing of IDMT relays is hindered by:

- Low precision from manual methods.
- No real-time display or digital logging.
- High cost and complexity of professional test kits.
- Limited access for students and small industries.

➤ **Objective:** To build a system using Arduino that is low-cost, accurate, programmable, and capable of measuring relay operating time digitally in real-time.

### IV. METHODOLOGY

The project follows a structured approach:

1. **Design Planning:**  
Understand the IDMT relay characteristics and testing needs.  
Select suitable hardware and tools (Arduino Uno, LCD, current injector, etc.).
2. **Hardware Integration:**  
Develop a power supply circuit.  
Set up Arduino to measure time using the `millis()` function.
3. **Operation Flow:**  
Start the timer on fault initiation (via push button).  
Stop the timer on the relay trip signal.  
Display time on LCD.
4. **Test Cycles:**  
Vary the injected current using the injector kit.  
Measure and record trip time.  
Repeat tests to ensure consistency.

### V. COMPONENT USED

Table V(1): Component Used

Component	Specifications
Contactors	650 volt , 25 amp ,50 Hz, Single pole
SMPS	230 volt AC / 24 volt DC, 5 amp
Arduino UNO	Standard
LCD Display	16*2
Transformer	230/12 Volt
Relay Module	1 channel, 5 volt
Push Button	Standard

### VI. WORKING AND SYSTEM ANALYSIS

1. **Power Conversion:**  
230V AC is stepped down to 12V AC using a transformer.  
AC is rectified and regulated using a 7812 IC to provide DC for Arduino and peripherals.
2. **Fault Simulation:**  
A push button simulates fault conditions by starting the test.  
Arduino energises an electromagnetic relay, which in turn closes a contactor.  
A current injector pushes overcurrent into the circuit.
3. **Relay Operation:**  
The IDMT relay senses the fault current.  
Based on the fault level, it trips after a predefined inverse delay.

4. **Timing Logic:**  
Arduino starts timing when the fault begins.  
When the relay trips, it sends a feedback signal to the Arduino. The timer stops, and the result is shown on a 16x2 LCD.
5. **System Reset:**  
A reset button restarts the cycle for additional test rounds.

## VII. EXPERIMENTAL SETUP

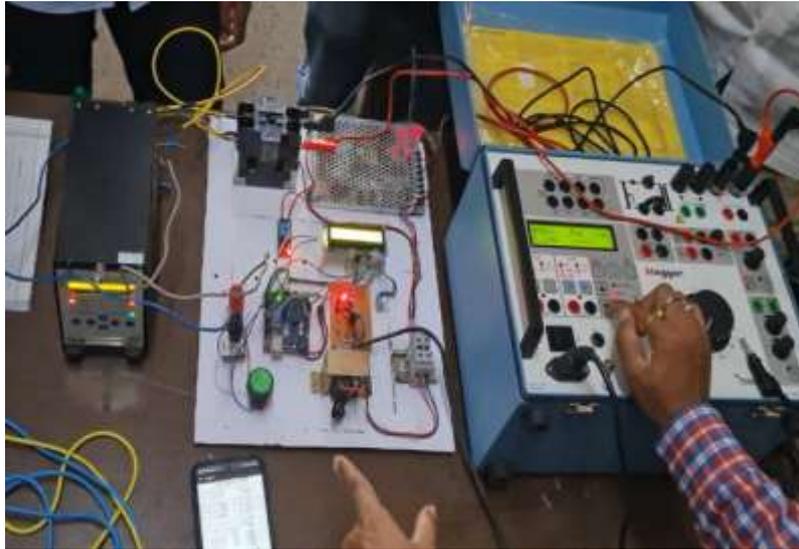


Fig VII(1): Experimental Setup

## VIII. DIGITAL CIRCUIT DIAGRAM AND BLOCK DIAGRAM

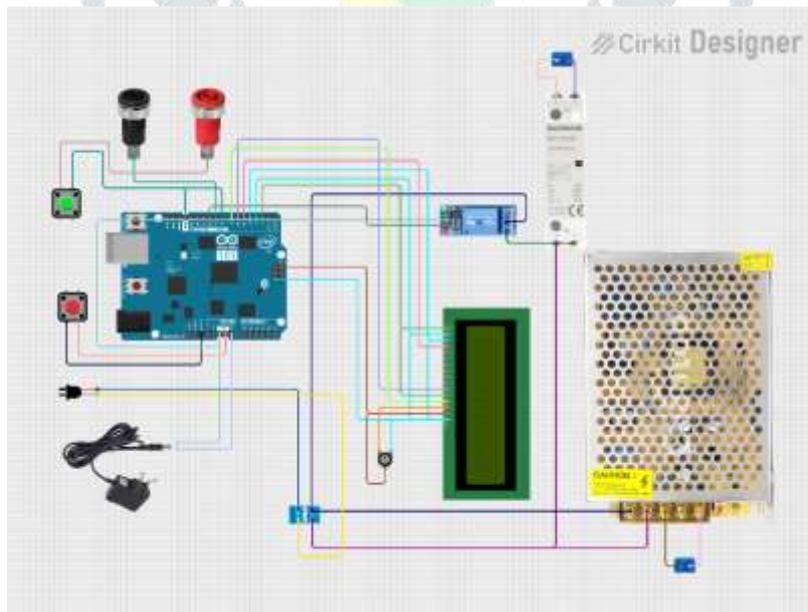


Fig VIII(1): Circuit Diagram

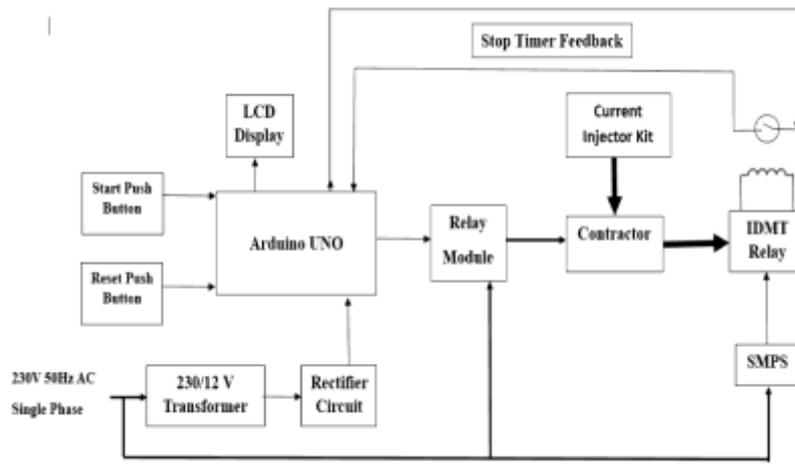


Fig VIII(1): Block Diagram

➤ **Signal Flow:**

1. Input → Transformer → Rectifier → Regulator → Arduino
2. Arduino I/O Pins ↔ Push Buttons + Relay Contacts
3. Relay Module → Contactor → Fault Injector → IDMT Relay

**IX. RESULTS AND DISCUSSION**

CT Ratio – 100:5  
 TMS - 1  
 Current Setting – 50%  
 Therefore N = 50% of 5 amp = 2.5 amp

Table IX(1): Curve 1

**Curve 'C1' - 3.0 sec**

Sr. no	Current	Time on Arduino Kit	Time on Analog Kit	Tolerance range
1.	2N = (2.5*2) = 5 A	11.03	11.03	8.77 – 11.28
2.	5N = (2.5*5) = 12.5 A	4.32	4.34	3.95 – 4.60
3.	10N = (2.5*10) = 25 A	2.96	2.96	2.82 – 3.11

Table IX(2): Curve 2

**Curve 'C2' – 1.3 sec**

Sr. no	Current	Time on Arduino Kit	Time on Analog Kit	Tolerance range
1.	2N = (2.5*2) = 5 A	4.36	4.35	3.76 – 4.83
2.	5N = (2.5*5) = 12.5 A	1.85	1.85	1.69 – 1.97
3.	10N = (2.5*10) = 25 A	1.29	1.29	1.20 -1.33

Table IX(3): Curve 3

**Curve C3 – 1.5 sec**

Sr. no	Current	Time on Arduino Kit	Time on Analog Kit	Tolerance range
1.	2N = (2.5*2) = 5 A	13.49	13.51	11.81 – 15.18
2.	5N = (2.5*5) = 12.5 A	3.40	3.39	3.12 – 3.62
3.	10N = (2.5*10) = 25 A	1.47	1.48	1.42 – 1.57

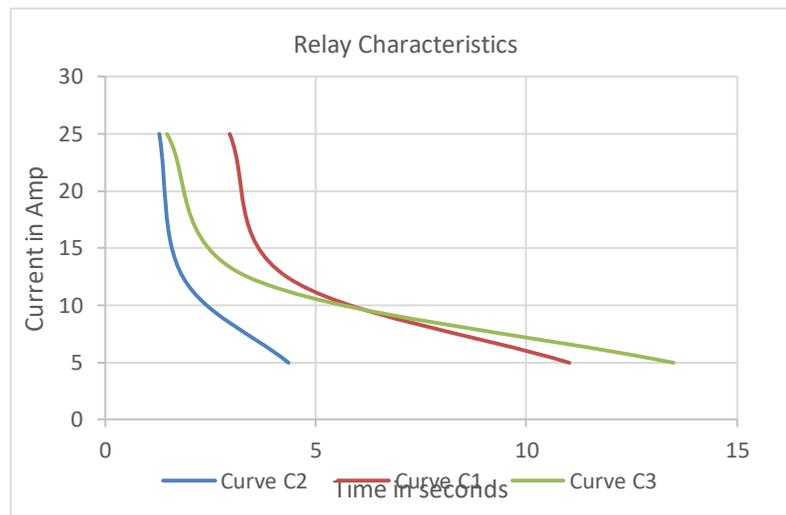


Fig:- XI(1)

Table IX(1): Discussion Points- Analogue kit vs Digital kit

Feature	Analog Kit	Arduino-Based Kit
Timing Accuracy	Low (manual)	High (digital, sub-millisecond)
Data Logging	Not supported	Easily supported (SD card, serial)
Real-Time Display	Absent or analogue meter	LCD / Serial / GUI
Automation	Not possible	Fully programmable
Future Integration (IoT/SCADA)	Not possible	Easily expandable
Educational Value	Limited interactivity	High engagement and versatility
Cost	Moderate	Low

## X. APPLICATION

- **Educational Institutions:** Training kits for engineering labs.
- **Industrial Maintenance:** On-site relay testing for small-scale facilities.
- **Relay Coordination Studies:** Real-time testing under different fault levels.
- **Automation Training:** SCADA/IOT integration simulations.
- **Smart Grid Testbeds:** Simulated fault response under load variation.

## XI. FUTURE SCOPE

- Integration with MATLAB/Python GUI for data logging.
- Auto fault injection via digitally controlled sources.
- IoT-enabled relay monitoring.
- SD card storage for test logs.
- SCADA-based fault visualization interface.
- Testing of multiple relays and relay coordination grading.

## XII. ACKNOWLEDGMENT

We would like to express our sincere gratitude to our esteemed guide and Head of Department, **Prof. V. Dhote**, for his continuous support, valuable suggestions, and expert guidance throughout the course of this project. His motivation and technical insights greatly contributed to the successful completion of this work.

We are also deeply thankful to **Mr. Rajesh B. Dhumal**, Assistant Executive Engineer, Testing Division, Jalgaon for not only providing us with the problem statement but also for his constant guidance and support during the development and testing phases. His practical insights and experience were instrumental in aligning our work with real-world applications.

We extend our appreciation to all faculty members, friends, and colleagues who directly or indirectly contributed to this research.

## REFERENCES

- [1] W. Elmore, *Protective Relaying Theory and Applications*, IEEE Press
- [2] B. Ravindranath and M. Chander, *Power System Protection and Switchgear*
- [3] Jadhav et al., "Arduino-Based Relay Testing System," *IJERT*, 2018
- [4] IEC 60255 – Measuring Relays and Protection Equipment
- [5] Arduino.cc – Arduino Uno Documentation
- [6] IDMT relay characteristics