



# International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

*(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)*



**Impact Factor: 8.524**

**Volume 14, Issue 1, January 2025**

# Indigenously Designed Programmable Timer Circuit for High-Temperature Furnace Control

Sharad Dange<sup>1</sup>, Savita Dange<sup>2</sup>

Associate Professor, Jai Hind College, Churchgate, Mumbai, India<sup>1</sup>

Associate Professor, The Institute of Science, Dr Homi Bhabha State University, Mumbai, India<sup>2</sup>

**ABSTRACT:** Timer circuits are widely available in the market; however, designing tailored and cost-effective solutions for specific industrial applications often necessitates an innovative approach. In this work, we present an indigenously designed and programmed timer circuit, specifically developed to control a high-temperature furnace, without employing a Real-Time Clock. The circuit is built using ATmega328 microcontroller, utilizing its readily available resources to maintain simplicity and affordability. The system features a user-friendly interface comprising a 4x3 matrix keyboard and a 16x2 liquid crystal display, designed to accommodate the operational needs of the furnace. The timer's resolution is achieved to the millisecond level, with its display represented intuitively in hours, minutes and seconds. A solid-state relay, used as per the technical specifications of the furnace, ensures reliable operation under heavy electrical loads. The modular design of this circuit allows for adaptation to other applications, offering flexibility for customization based on environmental and functional requirements.

**KEYWORDS:** ATmega328, Programmable Timer, Timer calibration, system clock, millis(), real-time clock, Crystal Oscillator.

## I. INTRODUCTION

Timer circuits play a vital role in various industrial and consumer applications, ranging from household appliances to advanced process control systems. These circuits provide precision timing and automation capabilities, often relying on Real-Time Clocks (RTCs) or other standardized components. Modern timer circuits typically feature user-friendly interfaces and are designed for generalized use, which limits their ability to be customized for specific applications. Moreover, the cost and complexity associated with integrating specialized timing modules can make these solutions less desirable for bespoke systems, particularly in scenarios that demand affordability and adaptability.

In this context, our work introduces an indigenously designed timer circuit tailored to meet the requirements of controlling a high-temperature furnace. Unlike conventional solutions, our design eliminates the need for an RTC by employing the Arduino ATmega328 microcontroller, leveraging its internal resources and an external crystal oscillator for efficient and precise timing. The user interface has been specifically crafted for ease of operation, using a 4x3 matrix keyboard and a 16x2 liquid crystal display (LCD) to provide a seamless and intuitive experience. The system supports millisecond resolution with time display in hours, minutes and seconds, ensuring accuracy while maintaining simplicity. A solid-state relay (SSR) is incorporated to handle heavy electrical loads, enhancing the circuit's reliability and robustness.

The presented timer circuit exemplifies a low-cost, customizable, and adaptable solution, capable of being modified for diverse applications beyond furnace control. Its modular architecture makes it particularly well-suited for environments that demand a balance between precision, reliability, and application-specific configurations. This development showcases the potential of microcontroller-based systems to offer innovative, simple and affordable alternatives to traditional timer circuits.

## II. SYSTEM DESIGN

### A. HARDWARE COMPONENTS

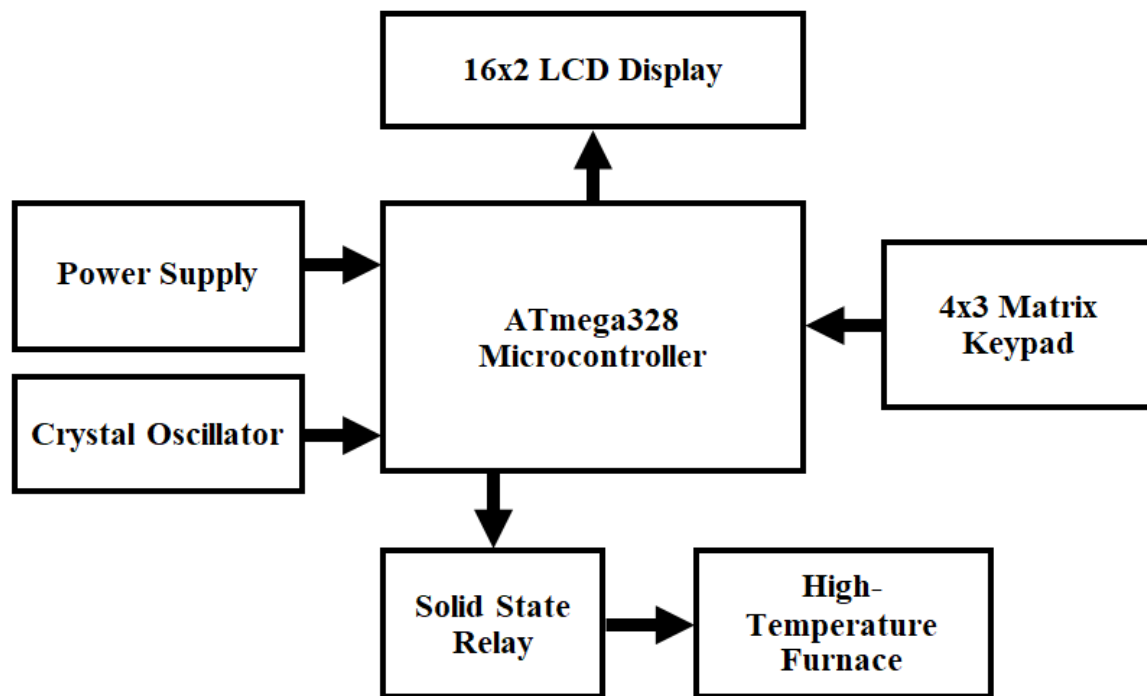
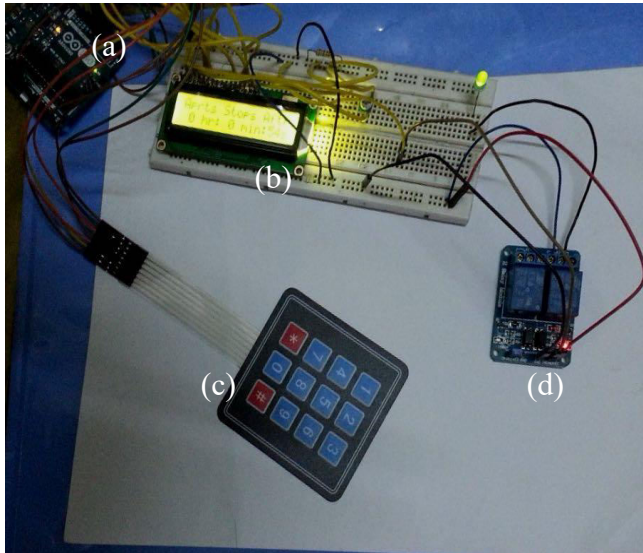


Figure 1: Schematic Diagram of the Timer Circuit

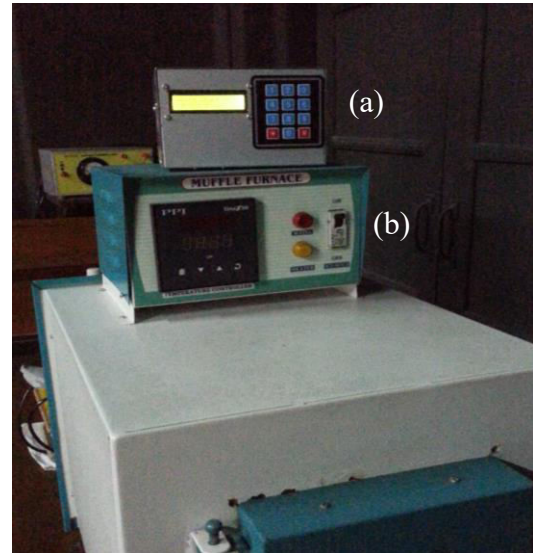
The figure 1 shows the schematic diagram of the timer circuit. It consists of the following major components:

- **ATmega328 Microcontroller:** It serves as the core processing unit, managing inputs, processing timing functions, and controlling outputs. The ATmega328 is an 8-bit microcontroller with 32KB of Flash memory, 2KB of SRAM, and 1KB of EEPROM. A 16MHz external piezoelectric oscillator is employed to clock the CPU.
- **Power Supply:** The Timer Circuit is powered by a 5 Volt DC power supply separately provided in the timer module.
- **4x3 Matrix Keypad:** It facilitates user input for setting start and stop times. The keypad allows users to input numerical values and commands, which are interpreted by the microcontroller to set the desired timing parameters.
- **16x2 LCD Display:** It provides real-time feedback to the user, displaying prompts, countdowns, and status messages. The display is interfaced with the microcontroller to show relevant information, enhancing user interaction and system transparency.
- **Solid-State Relay:** It controls the power supply to the furnace, enabling safe and efficient switching of high-power loads. The SSR was selected for its ability to handle the high electrical load of a furnace, ensuring both safety and longevity.
- **Crystal Oscillator:** The timer circuit in this work employs a 16 MHz external crystal oscillator as the primary clock source for the ATmega328 microcontroller, ensuring improved timing accuracy compared to internal oscillators.

## B. OPERATION PROCEDURE



**Figure 2:** Operation of Timer Circuit using a Bread Board (a) Arduino UNO board (b) 16x2 LCD display (c) 4x3 Matrix Keyboard (d) Relay Module



**Figure 3:** (a) Timer module (b) Temperature controller of Furnace

Figure 2 shows operation and testing of the timer circuit using a bread board and a simple relay module. The figure 3(a) shows the actual Timer Module developed from the timer circuit and connected to the temperature controller of high temperature furnace shown in figure 3(b).

The system operation is as follows:

1. **System Initialization:** Upon powering the system, the LCD displays a welcome message for 2 seconds, indicating that the system is ready for operation.
2. **User Prompt:** The user is prompted to press '\*' to skip the timer or '#' to set the timer. This allows flexibility in operation, enabling immediate start or customized timing as per user requirements.
3. **Setting Start Time:** If '#' is pressed, the user is prompted to enter the start delay time in hours and minutes, indicating when the apparatus should start. The input is validated to ensure correctness, and the entered time is displayed for confirmation.
4. **Setting Stop Time:** The user is then prompted to enter the operation duration in hours and minutes, indicating how long the apparatus should operate. This ensures precise control over the operation period of the furnace.
5. **Confirmation :** The user can press '\*' to reset the input times or '#' to confirm and proceed. This step allows correction of any input errors before the timer sequence begins.
6. **Countdown to Start:** The system displays the remaining time before activation, with the seconds portion flashing every second to indicate progress. This visual feedback keeps the user informed about the time remaining until the furnace starts.
7. **Activation:** Once the start delay elapses, the SSR is activated, powering the furnace. The system ensures that the SSR is switched on precisely at the end of the countdown, initiating the furnace operation.
8. **Operation Countdown:** The system displays the remaining operation time, with the seconds portion flashing every second. This provides real-time monitoring of the operation duration, ensuring that the process adheres to the set parameters.
9. **Completion:** After the operation duration elapses, the microcontroller turns off the SSR, and the LCD displays a completion message indicating that the apparatus has stopped. This ensures accurate operation, as the furnace is turned off precisely as scheduled, enhancing safety and efficiency.

### C. SOFTWARE IMPLEMENTATION

The system's software was developed in C++ using the Arduino Integrated Development Environment (IDE). The program initializes hardware components, manages user inputs, and controls timing sequences. For timekeeping, the Arduino millis() function is employed, providing millisecond resolution which operates based on the system clock. The accuracy of this function is influenced by the precision of the clock source. In this implementation, the program was uploaded to a standalone ATmega328 microcontroller using an Arduino UNO board. The microcontroller was then integrated into the timer circuit module, utilizing an external crystal oscillator to enhance timing accuracy.

#### Algorithm:

##### 1. Initialization:

Initialize the LCD display, keypad, and SSR.  
Set SSR to the 'off' state.

##### 2. Display Welcome Message:

Show a predefined message on the LCD for 2 seconds.

##### 3. User Prompt:

Prompt the user: "Press '\*' to Skip Timer, '#' to Set Timer."

- Wait for user input.

##### 4. Input Handling:

If '\*' is pressed:

- Bypass the timer.
- Activate the apparatus immediately by setting SSR to 'on' state.
- Proceed to the operation completion sequence.

If '#' is pressed:

- Prompt the user to set the start time (delay before activation) in hours and minutes.
- Display the entered time as "XXhr:XXmin".
- Prompt the user to set the stop time (duration of operation) in hours and minutes.
- Display the entered time as "XXhr:XXmin".
- Confirm the settings by prompting: "Press '\*' to Reset, '#' to Continue."

##### 5. Confirmation Handling:

If '\*' is pressed:

- Reset the timer settings and return to the start time input step.

If '#' is pressed:

- Proceed to the countdown sequence.

##### 6. Countdown Sequence:

Display the message: "Apparatus Starts After XXhr:XXmin:XXsec".

Flash the seconds portion every second to indicate progress.

Once the start time countdown reaches zero:

- Activate the apparatus by setting SSR to 'on' state.
- Display the message: "Apparatus Stops After XXhr:XXmin:XXsec".
- Flash the seconds portion every second.
- Begin the stop time countdown.

##### 7. Operation Sequence:

During the stop time countdown, the apparatus remains active.

The timer continues to count down with millisecond resolution.

The LCD updates the remaining time in real-time.

##### 8. Completion Sequence:

Once the stop time countdown reaches zero:

- Deactivate the apparatus by setting SSR to 'off' state.
- Display the message: "Timer Complete, Apparatus Stopped".

##### 9. End:

The system halts and can be powered off.

### III. CALIBRATION AND ERROR

The accuracy of the millis() function is directly influenced by the system clock's precision. Arduino boards equipped with ceramic resonators typically exhibit a  $\pm 0.5\%$  error, leading to noticeable time discrepancies over extended periods. By contrast, crystal oscillators provide significantly better accuracy, with errors typically ranging from 20 to 50 parts per million (ppm). These deviations can result from factors such as initial tolerance, temperature variations, or aging.

To verify timing accuracy, a known external time reference clock was utilized. The system clock (F\_CPU) was synchronized with the 16 MHz crystal oscillator's frequency, and software corrections were applied to address deviations in the timer's output.

Testing over extended durations, with comparisons against an external reference clock, confirmed reliable timing performance. After calibration and software corrections, the observed time drift was approximately 0.005%.

### IV. RESULTS AND DISCUSSION

The programmable timer circuit was successfully tested and implemented with a high-temperature furnace. The system demonstrated reliable timing control, with the user interface providing clear and intuitive interaction. The elimination of an external RTC module reduced the overall cost and complexity of the circuit. This simplification is particularly advantageous for applications requiring precise but cost-effective timekeeping, where the accuracy of millis() provides an optimal balance between performance and affordability. In future work, we aim to integrate temperature compensation techniques by exploring the implementation of temperature-compensated crystal oscillators (TCXOs) to improve timing precision in highly sensitive applications.

### V. CONCLUSION

This work demonstrates an indigenously developed low-cost, programmable timer circuit that combines precision, user convenience, and application adaptability, establishing its potential for broader industrial applications.

### REFERENCES

- [1] K. S. C. Kotcherlakota, V.B.R.Chinta "Smart Home Automation Based On 555 Timer" Int. Journal of Engineering Research and Application Vol. 6, Issue 5, pp.77-80, 2016.
- [2] A.M. Imam, "Design and Implementation of A 24-Hour, Minutes and Seconds Display Digital Clock" International Journal of Scientific and Research Publications, Volume 5, Issue 12, pp. 565-570, 2015.
- [3] A. L. Durga, B. Prasanthi, K. Maheshwari, and B. Greeshma, "Time Operated Electrical Appliances Control System Using Arduino," International Research Journal of Engineering and Technology (IRJET), vol. 5, no. 2, pp. 556-559, 2018.
- [4] L. A. Ajao, M. A. Adegboye, E. M. Dogo, S. O. Aliyu, and D. Maliki, "Development and Implementation of Microcontroller-based Improved Digital Timer and Alarm System," in Proceedings of the International Conference on Information and Communication Technology and Its Applications (ICTA), 2016.
- [5] T. Pan, "Development and Implementation of Microcontroller-based Digital Clock," Journal of Academy of Science and Technology, vol. III, Myanmar, 2008.
- [6] Arduino, "Arduino Documentation," [Online]. Available: <https://docs.arduino.cc/>.



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  [ijirset@gmail.com](mailto:ijirset@gmail.com)



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details