IJCRM

International Journal of Contemporary Research In Multidisciplinary

Research Article

Designing of Smart Stopwatch System Using Arduino Uno and A Four-Digit 7-Segment Display

Subha Gaurab Roy[1*], Saharul Islam Choudhury[2]
[1]Department of Physics, S. S. College, Hailakandi, Assam, India
[2]Student of 6th Semester, CBCS Course, S. S. College, Hailakandi, Assam, India

Corresponding Author: Subha Gaurab Roy*

DOI: https://doi.org/10.5281/zenodo.16413862

Abstract

This paper presents the development and evaluation of a smart stopwatch system designed using Arduino Uno and a four-digit seven-segment display. The primary objective is to offer a low-cost, accurate, and user-friendly solution for real-time timekeeping. The system is capable of starting, stopping, and resetting time measurements using tactile button inputs and provides real-time visual feedback through efficient multiplexing. The paper delves into the system's design, the technology used, its architecture, working principle, and business implications. It addresses the need for accessible timing devices, especially in educational, sports, and industrial environments. Through simulations and practical implementations, the project demonstrates efficiency, reliability, and adaptability. A detailed SWOT analysis and management perspective have been included to evaluate its real-world feasibility and conclude with opportunities for future innovation.

Manuscript Information

- **ISSN No:** 2583-7397
- **Received:** 09-03-2025
- **Accepted:** 22-04-2025
- **Published:** 25-04-2025
- **IJCRM:**4(2); 2025: 387-392
- **©2025, All Rights Reserved**
- **Plagiarism Checked:** Yes
- **Peer Review Process:** Yes

How to Cite this Article

Roy SG, Choudhury SI. Designing of Smart Stopwatch System Using Arduino Uno and A Four-Digit 7-Segment Display. Ayurveda Dietetics and Food Technology: Modern Correlations.Int J Contemp Res Multidiscip. 2025;4(2):387-392.

Access this Article Online

www.multiarticlesjournal.com

KEYWORDS: Smart Stopwatch, Arduino Uno, 7-Segment Display.

1. INTRODUCTION

In the modern era of digital innovation and automation, time has become a critical parameter across various domains, including education, industry, research, and sports. Accurate time measurement is essential for experiments, testing, training, and process control. While you can find plenty of commercial digital timers and stopwatches on the market, they often come with some drawbacks, especially when it comes to cost, customization, and accessibility, particularly in educational or low-resource settings. In places like schools, sports facilities and technical environments, having reliable and easy-to-use timekeeping tools is crucial.

Although commercial stopwatches can fulfill these needs, they can also be pricey, lack customization options or simply be unavailable in areas with limited resources. This is especially true in developing regions or rural schools, where students and teachers need affordable alternatives for hands-on learning and experimentation. We know, traditional digital timers can be pretty inflexible. They often struggle to keep up with project-based learning or integrate smoothly into bigger systems for automation, data logging, or IoT applications. A programmable solution not only meets the basic timekeeping needs but also opens the door to a world of innovation and customization.

To bridge this gap, microcontroller-based systems like Arduino offer an ideal platform for designing personalized, budget-friendly timing solutions. Arduino, being open-source and beginner-friendly, allows students, educators, and hobbyists to develop functional timing system with basic components and programming knowledge with minimal resources. One such implementation is a digital stopwatch—an excellent entry-level project that demonstrates real-time control, display interfacing, and user interaction using basic electronics and programming concepts.

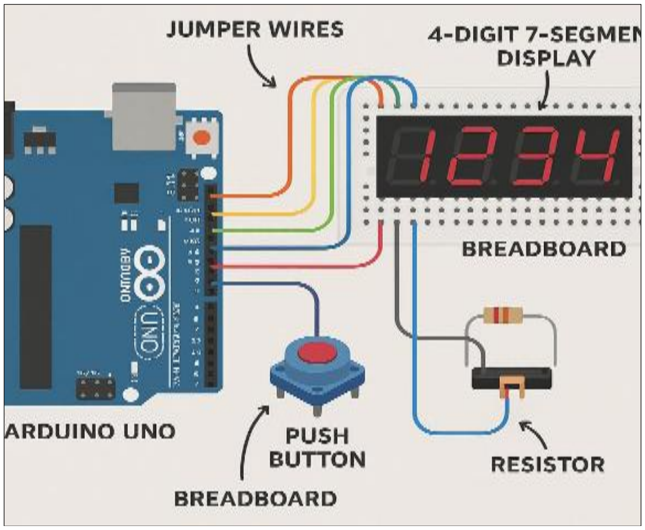
This study explores the design of a smart stopwatch using the Arduino Uno and a 4-digit seven-segment display. The system features start, stop, and reset controls through tactile buttons and displays elapsed time using real-time multiplexing. By integrating practical electronics with theoretical learning, this stopwatch project aims to empower learners with the knowledge of timekeeping mechanisms while demonstrating real-world system design principles.

2. Hardware and Technology Used

The stopwatch system is built using affordable and widely available electronic components, making it ideal for academic and prototyping purposes. To design, simulate and program the stopwatch system, several software tools and platforms were utilized. Each played a specific role in the development process, from writing code to visualizing the hardware circuit. Below is a detailed overview of the core hardware elements and technologies employed in the project:

- **Arduino Uno:** The central processing unit of the system, based on the microcontroller. It handles input detection, timing logic, and display control. It offers digital I/O pins, timers, and USB programmability, making it ideal for beginner-level embedded projects.
- **4-Digit Seven-Segment Display (Common Cathode):** Used to visually represent the elapsed time. Each digit consists of seven LEDs arranged to form numbers from 0 to 9. The display is controlled through multiplexing, enabling all four digits to share segment pins and reduce the required number of output pins.
- **Push Buttons:** Two tactile switches are used to control the stopwatch functions: one for starting/stopping the timer and another for resetting it. These buttons are connected to digital pins on the Arduino and monitored using debouncing logic.
- Resistors: 220Ω resistors are used to limit current flowing through each LED segment, preventing damage. 1000Ω resistors are used as pull-down resistors for the buttons, ensuring a defined voltage level when the button is not pressed.
- Breadboard and Jumper Wires: Used for building the prototype without soldering. They allow for easy testing, adjustments, and demonstrations.
- Technology Approach: The Arduino IDE is used for code development and uploading the program to the microcontroller. n- Timing logic records the elapsed time without blocking other processes, allowing smooth and responsive performance.

These tools ensure that both hardware and software components were tested thoroughly and that the system could be replicated and understood by others with ease



3. OBJECTIVE

This project addresses the need for a low-cost, beginner-friendly stopwatch that can be built and understood by students with basic electronics knowledge. It also fills a gap in project-based education by offering hands-on experience in microcontroller programming, hardware interfacing, and circuit design. Beyond learning, the system can be repurposed for real-world applications like timing events, classroom activities, or machine processes, making it a versatile tool for both learning and light-duty use.

The suggested stopwatch system successfully removes the weaknesses of conventional timing devices with a flexible, programmable, and inexpensive solution based on the Arduino platform. In contrast to commercial stopwatches with fixed functions, the system enables customization on both hardware and software levels, allowing people to modify it according to specific requirements. With the use of simple components and open-source tools, it is made available for students, teachers, and hobbyists. Utilizing the Millis() function enables precise, non-blocking timing measurement for uninterrupted performance without the need for complicated hardware.

Its modular construction allows users to customize the system, as opposed to commercial stopwatches that come with set features. The transparent seven-segment display and uncomplicated button operations ensure easy use and comprehension. Generally, this system covers the need for a flexible, inexpensive, and educational timing system.

In addition, the work is aimed to execute several educational, technical and practical goals such as

i) To design a low-cost, easy-to-build digital stopwatch using readily available electronic components.
ii) To implement button-controlled start, stop, and reset functions with proper debouncing techniques.
iii) To simulate and verify the circuit design using software platforms like Tinker cad before physical prototyping.
iv) To cultivate students' knowledge in embedded systems, display interfacing, and hardware-software integration.
v) These objectives ensure the project is not only technically sound but also valuable as a teaching tool and foundation for future innovations.

These objectives ensure the project is not only technically sound but also valuable as a teaching tool and foundation for future innovations.

4. System Architecture and Component Overview:

The stopwatch system follows a simple yet effective architecture composed of three primary layers:

Input Unit: Push Buttons (Start/Stop and Reset).

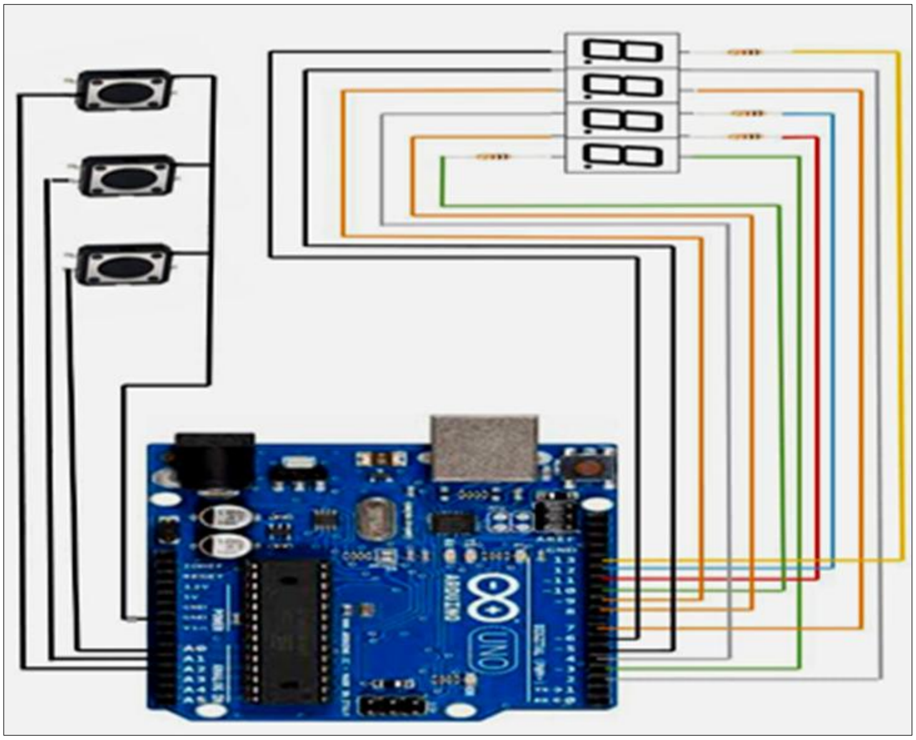
The user can start, pause, or restart the timer with these. These allow the user to interact with the system by starting, pausing, or resetting the timer. The buttons are connected to Arduino's digital input pins with pull-down resistors to ensure stable readings.

Processing Unit: Arduino Uno

The microcontroller receives button inputs, calculates elapsed time using the millis() function, and controls the display output. It handles all logical operations, including multiplexing and digit extraction.

Output Unit: 4-Digit Seven-Segment Display.

This display shows the elapsed time using efficient multiplexing. Each digit shares common segment pins, and only one digit is activated at a time in rapid succession to give the illusion that all digits are on simultaneously.



5. Working Principle along with simplified coding:

The stopwatch system operates based on a simple logic of time tracking using Arduino's internal clock and user input through push buttons. Below is a step-by-step explanation of how the system works:

1. Time Tracking with millis()

The millis() function in Arduino returns the number of milliseconds since the board was powered on. It enables non-blocking, real-time tracking of elapsed time without using delays.

2. Button-Based Control

Start/Stop Button: When pressed, the system begins (or pauses) counting time by setting a running flag in the code.

Reset Button: Resets the timer by setting the elapsed time back to zero.

3. Time Display

The elapsed time is broken into individual digits (thousands, hundreds, tens, and units) using integer division and modulus operations. These values are then sent to the four-digit seven-segment display using multiplexing.

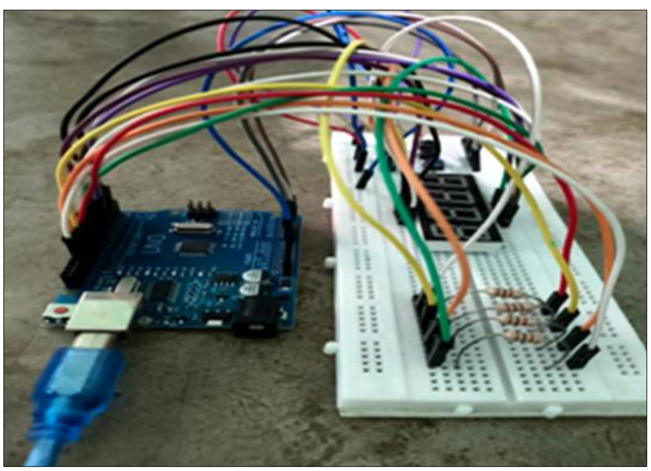
4. Multiplexing

Since all digits share common segment pins, each digit is turned on one at a time in quick succession (about 1–2 milliseconds per digit). This rapid switching creates the illusion that all digits are lit continuously (persistence of vision).

Code Logic (Simplified)

```
if (startPressed) running = true;
if (stopPressed) running = false;
if (resetPressed) elapsedTime = 0;
if (running) {
elapsedTime = (millis() - startTime) / 100;}
displayDigits(elapsedTime);
```

This logic ensures smooth, real-time stopwatch operation using minimal hardware and efficient code. The adjacent fig shows the real time display of the built in stop watch system.



5. Observation

i) The stopwatch maintains high timing accuracy with minimal deviation over extended periods.
ii) Button responses are fast and reliable, with low latency between press and action.
iii) The display remains stable and flicker-free due to effective multiplexing.
iv) Power consumption is low, suitable for USB or battery-powered operation.
v) The system performs reliably over long durations without resets or crashes.
vi) Users find the interface intuitive, with minimal learning required.

6. RESULTS AND DISCUSSIONS

The stopwatch system was successfully simulated and tested, both virtually (using Tinkercad) and physically (on a breadboard), confirming its functionality and reliability. The performance of the system was evaluated based on several key criteria:

i) Timing Accuracy: The stopwatch maintained consistent timing with an average deviation of less than ±0.3 seconds over a 30-minute run.

ii) Use of the millis() function ensured that time tracking remained non-blocking and accurate.

iii) Display Stability: Multiplexing was implemented effectively, resulting in a clear and flicker-free display when properly timed. The digits were readable from a distance of approximately 2 meters under normal lighting conditions.

iv) Button Responsiveness: Proper debouncing techniques were implemented to prevent false triggering. Short and long presses were accurately detected, allowing reliable control over start/stop and reset functions.

v) Power and Resource Usage: The system ran efficiently using USB or 9V power supply, drawing under 200mA of current. Total memory usage of the Arduino code was below 4KB, making it suitable even for smaller microcontrollers.

vi) User Feedback: Test users found the system easy to use and intuitive. Suggestions for improvement included adding sound alerts, lap-time functionality and Bluetooth connectivity.

Overall, the stopwatch system met its intended goals. It is a practical and educational tool that can be replicated with minimal cost and effort. While basic in design, it offers a solid foundation for further enhancements and integration into more complex projects.

Knowing the strengths, weaknesses, opportunities and challenges/threats (SWAT) helps one to evaluate the overall viability, scalability, and limitations of the stopwatch system from a strategic perspective which is mentioned below:

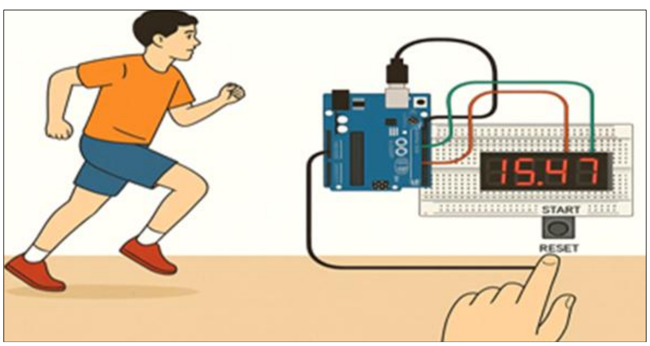
Strengths
- Low-cost and easy to build
- Beginner-friendly (great for learning)
- Modular and customizable
- Open-source hardware and software

Weakness
- Limited to 4-digit display (up to 99.99 seconds)
- No wireless or data logging by default
- Basic interface (no GUI or sound feedback)
- May not be robust for long-term industrial use

Opportunities
- Expand with Bluetooth or IoT features features
- Use in schools, labs, and training centres
- Integration with mobile apps or displays
- Can be sold as educational kits

Challenges
- Competing readymade digital timers with advanced
- Component wear and tear over extended use
- Risk of malfunction from poor wiring in physical builds
- Market saturation in low-cost electronics

The above points shows that while the system is simple, it has strong educational value and room for expansion. Addressing its limitations could position it as a viable solution for broader use in training or lightweight commercial applications.



7. Real-Time workflow and interpretation from the management perspective:

During a school sports event, an Arduino-based stopwatch system can be used to time a student's sprint. The setup includes an Arduino Uno connected to a 4-digit seven-segment display, along with two push buttons—one for starting and stopping the timer, and the other for resetting it. Once the system is powered on via USB or a 9V adapter, the display shows "0000", indicating that the stopwatch is ready. As the student begins the race, the Start button is pressed. The Arduino immediately begins tracking time using the millis() function, and the display updates in real-time to reflect the elapsed time. For instance, as the race progresses, the display might read "0123", meaning 12.3 seconds have passed. Upon completing the race, the Stop button is pressed to freeze the timer at its final value, such as "1547" (154.7 seconds). This value is then noted down by the observer or instructor. Finally, the Reset button is pressed to clear the display back to "0000", making the system ready for the next runner. This simple and responsive workflow ensures accurate time tracking in a hands-on and educational setting.

From a management standpoint, this project demonstrates an efficient use of minimal resources to deliver a functional, scalable, and educationally valuable product. It aligns with goals such as cost reduction, skill development, and process innovation—especially in academic institutions and technical training centres. The system's low production cost and ease of replication make it suitable for use in workshops, classroomkits, and vocational programs. Its open-source nature encourages student participation, teamwork, and problem-solving, which are key outcomes valued in modern education and workforce training.

Moreover, this stopwatch can be viewed as a prototype for future development in low-cost industrial timing tools or customized solutions in small-scale manufacturing setups. The return on investment is high in terms of learning outcomes, user adaptability, and long-term usability.

8. Future Scope of study

i) Lap timing functionality can be added to record split intervals.
ii) SD card integration may allow storage of recorded times for review.
iii) Wireless modules like Bluetooth or Wi-Fi can enable remote operation.
iv) Touch interfaces or OLED displays could enhance the visual and user experience.
v) Mobile app integration can extend control and data visualization.
vi) The system can be expanded into IoT applications requiring timed triggers.
vii) With documentation, it can be converted into a STEM educational kit.
viii) Useful for schools, electronics workshops, and prototyping environments.

9. CONCLUSIONS

The development of a digital stopwatch using Arduino and a seven-segment display successfully demonstrates how fundamental concepts in electronics and embedded systems can be applied to create a functional, low-cost, and educational tool. This project not only achieves its primary goal of accurate time measurement but also serves as a foundation for learning about microcontroller programming, display interfacing, and real-time logic implementation.

The modular design and use of open-source hardware/software platforms make the system highly replicable and expandable. It can be adapted for use in schools, training centres, competitions, and even lightweight industrial applications. Additionally, its integration into STEM education can enhance practical learning experiences and inspire innovation among students and hobbyists.

While the current version focuses on simplicity and core functionality, future developments could include features like wireless control, lap timing, data logging, or mobile app integration. Overall, this project reflects the power of hands-on, project-based learning and opens doors for deeper exploration in electronics and automation.

10. ACKNOWLEDGEMENT

The authors place on record their acknowledgement to Mr. Noman Hanif Barbhuiya, Ph.D. Scholar, IIT, Gandhinagar for his valuable contribution and insights. This work has been done as a part of UG dissertation of CBCS Even Semester examination 2025.

REFERENCES

1. https://www.arduino.cc/en/Guide Reference for hardware specifications and functions like Millis().
2. Kumar A, Sharma P. Design and implementation of digital stopwatch using Arduino. International Journal of Electronics Engineering. 2019;11(2):45–50.
3. Singh, R. andDas, A. (2021). Real-Time Applications of Microcontroller-Based Systems. International Journal of Engineering Research and Applications (IJERA), 9(3), 22–27.
4. Instructables. Arduino stopwatch projects: Community-driven tutorials and examples for Arduino timers [Internet]. Available from: https://www.instructables.com
   Banzi M, Shiloh M. Getting started with Arduino. 3rd ed. Sebastopol (CA): Maker Media; 2014.
5. Fritzing. Fritzing software: Tool used for designing circuit schematics and documentation visuals [Internet]. Available from: https://fritzing.org