

**COUNTDOWN TIMER**

PROJECT REPORT

# Submitted by

**Abhiram.D [192210154]**

**Y.Krishna karthikeya [192210017]**

*Under the guidance of*

**Dr. Ganesh ramachandran**

(Professor, block chaining and cyber security)

***In partial fulfilment for the completion of course***

**CSA0814 - PYTHON PROGRAMMING FOR SOLUTION DEVELOPMENT**



**SIMATS ENGINEERING**

**THANDALAM**

**FEB 2024**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**COUNTDOWN TIMER**” is the bonafide work of "**Abhiram.D [192210154], Y.Krishna karthikeya [192210017]**"

who carried out the project work under my supervision as a batch. Certified further, that to the best of my knowledge the work reported herein does not form any other project report.

Date: Project Supervisor Head of the Department

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CONTENT** | **PAGE NO.** |
| 1 | ABSTRACT | 4 |
| 2 | INTRODUCTION | 5 |
| 3 | METHODOLOGY | 5-6 |
| 4 | IMPLEMENTATION | 6 |
| 5 | ALGORITHM | 7-8 |
| 6 | CODE EXPLANATION | 9-12 |
| 7 | OUTPUT | 12 |
| 8 | RESULTS AND DISCUSSIONS | 12-13 |
| 9 | CONCLUSION | 14 |
| 10 | FUTURE  ENHANCEMENT | 14 |
| 11 | REFERENCE | 15-16 |

# ABSTRACT

This project presents a Python-based countdown timer application with a graphical user interface (GUI) representation. The application allows users to set a specific duration for countdown and provides a visual representation of the time remaining. Developed using the Tkinter library for GUI, the program offers an intuitive interface for setting countdown parameters and initiating the timer. The GUI displays the remaining time dynamically, updating in real-time as the countdown progresses. Additionally, users can customize the appearance and functionality of the timer to suit their preferences. This project serves as a practical implementation of GUI programming in Python, offering users a simple yet effective tool for managing timesensitive tasks with visual feedback.

**Keywords :-** Python, Countdown timer, GUI, Tkinter, Graphical user interface, Time management, Real-time updating, User customization, Programming.

# Introduction

In the real time management and productivity enhancement, countdown timers serve as invaluable tools for users across various domains. Whether it's for boosting productivity during work sessions, managing time-sensitive tasks, or simply staying organized, countdown timers offer a tangible representation of time that aids users in staying focused and efficient. Python, a versatile and widely-used programming language, provides developers with the flexibility to create such countdown timer applications, often complemented by graphical user interfaces (GUI) for enhanced user interaction. In this project, we explore the development of a Python-based countdown timer with a GUI representation, leveraging the Tkinter library, a popular choice for creating GUI applications in Python.

The countdown timer application developed in this project offers users a seamless experience for setting and tracking time durations. Through the intuitive interface provided by Tkinter, users can input their desired countdown duration in hours, minutes, and seconds, enabling precise customization to suit their specific needs. Upon submission of the countdown parameters, the application dynamically updates the GUI to display the remaining time, allowing users to visually track the countdown progress. This real-time updating feature ensures that users are constantly informed about the time remaining, facilitating efficient time management and task prioritization.

Furthermore, the countdown timer application incorporates functionalities to enhance user experience and usability. Error handling mechanisms are implemented to ensure that users provide valid input, preventing runtime errors and enhancing the robustness of the application. Additionally, upon completion of the countdown, the application displays a message box alerting users that the time is up, providing a clear indication of task completion. Through this project, users gain insight into the development of GUI-based countdown timer applications in Python, empowering them with a practical tool for improving time management and productivity in various contexts.

**2. Methodology:**

The development of the countdown timer application involved both hardware and software design aspects (please refer Fig. 1). The hardware components utilized for this project included a standard computing device capable of running Python scripts, such as a desktop computer. In the software design phase, Python 3 served as the primary programming language for implementing the countdown timer functionality.

PySimpleGUI and tkinter libraries were employed to create the graphical user interface (GUI) for user interaction.

Parameters considered during the design and implementation of the countdown timer application included:

**Timer Setting:** Users can input the desired countdown duration in hours, minutes, and seconds through the GUI.

**Countdown Progress:** The application dynamically updates the GUI to visually represent the remaining time as the countdown progresses, providing real-time feedback to the user.

**Error Handling:** Mechanisms were implemented to handle invalid user inputs, ensuring the robustness of the application.

**Completion Alert:** Upon reaching zero, the application triggers a message box alert to notify the user that the countdown has concluded.

The software design encompassed the creation of user-friendly interfaces using PySimpleGUI and Tkinter, allowing users to easily set and monitor the countdown timer. The Python scripting facilitated the logic behind the countdown functionality, including the calculation of time remaining, updating the GUI interface, and triggering alerts upon completion. Throughout the development process, emphasis was placed on ensuring the reliability, accuracy, and user-friendliness of the countdown timer application.

1. **Implementation:**

The implementation of the countdown timer application involved setting up the hardware components, designing the software logic, and integrating both aspects to create a functional system. This section outlines the experimental setup, including the hardware and software configurations, as well as the program logic for managing parameters and controlling the countdown timer functionality.

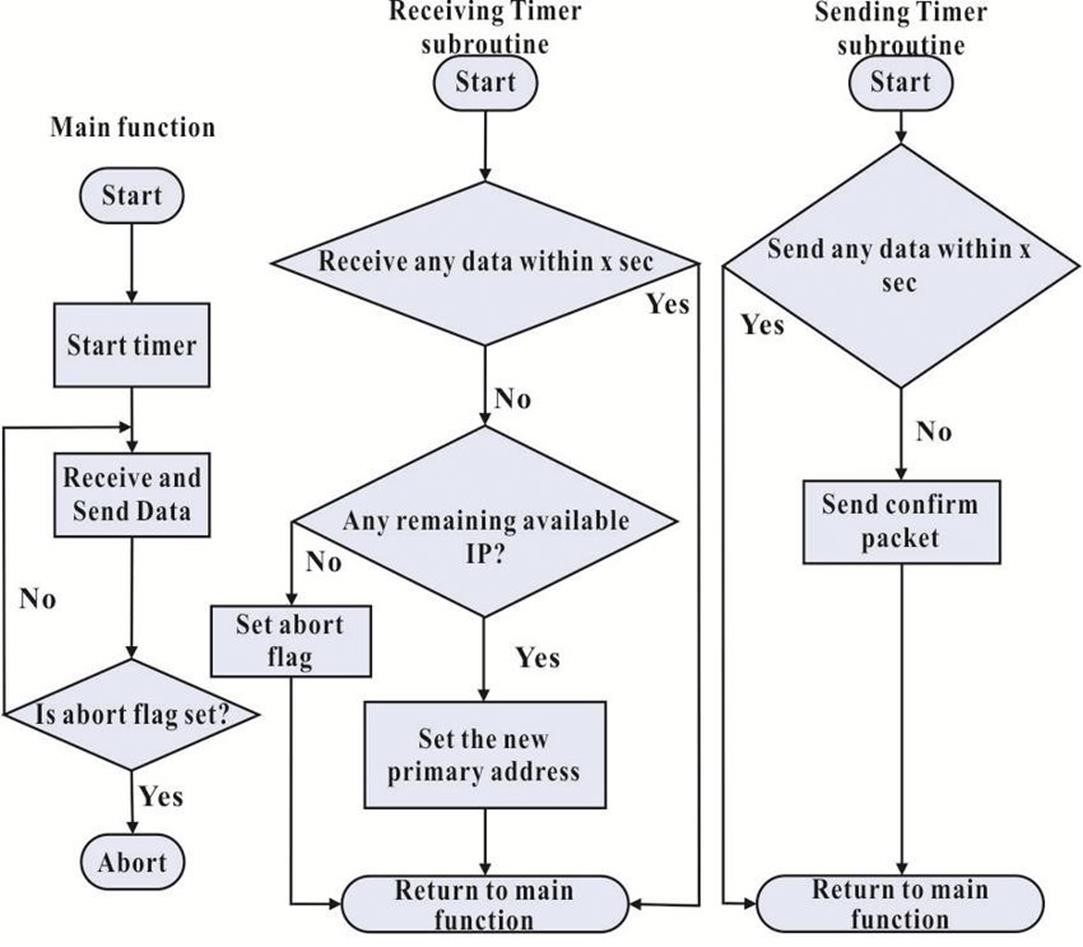
* 1. **Experimental Setup:**

The experimental setup for the countdown timer application comprised configuring the hardware components, such as the computing device (e.g., Raspberry Pi), I/O modules, and relays, as well as installing the necessary software dependencies, including Python 3, PySimpleGUI, and Tkinter libraries. The hardware components were connected according to the specified wiring diagrams, ensuring proper communication and functionality.

* 1. **Program Logic for Parameters:**

The program logic for managing parameters in the countdown timer application involved capturing user inputs, validating the input values, and updating the countdown timer interface accordingly. Upon receiving input from the user through the GUI interface, the application parsed the input values, converted them to the appropriate data types, and performed error checking to ensure validity. Subsequently, the countdown timer logic utilized the validated parameters to initiate and manage the countdown process, dynamically updating the GUI interface to reflect the countdown progress in real-time.

1. **Algorithm:**



**Figure 1.** Flowchart of countdown timer

**gui.py Algorithm:**

Create GUI:

Create Tkinter root window.

Set geometry, title, and background color. Return the root window. **main.py Algorithm:**

**Main:**

Create Tkinter root window using create\_gui().

Initialize StringVar variables (hour, minute, second).

Create Tkinter Label and Entry widgets.

Create Tkinter Button triggering countdown\_timer. Enter the Tkinter main loop.

**Timer\_logic.py Algorithm:**

Countdown Timer:

Try to parse user input to seconds.

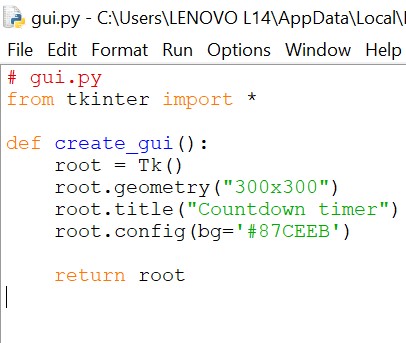
● Show warning if unsuccessful.

Enter loop while user\_input >= 0:

1. Convert user\_input to hours, minutes, seconds.
2. Update hour, minute, second variables.
3. Update current\_time\_label with current time.
4. Update Tkinter window.
5. Sleep for 1 second.
6. If user\_input is 0, show a time-over message.
7. Decrement user\_input by 1.

**5. Code:**

**Create GUI:**



**Figure 2.** GUI of countdown timer

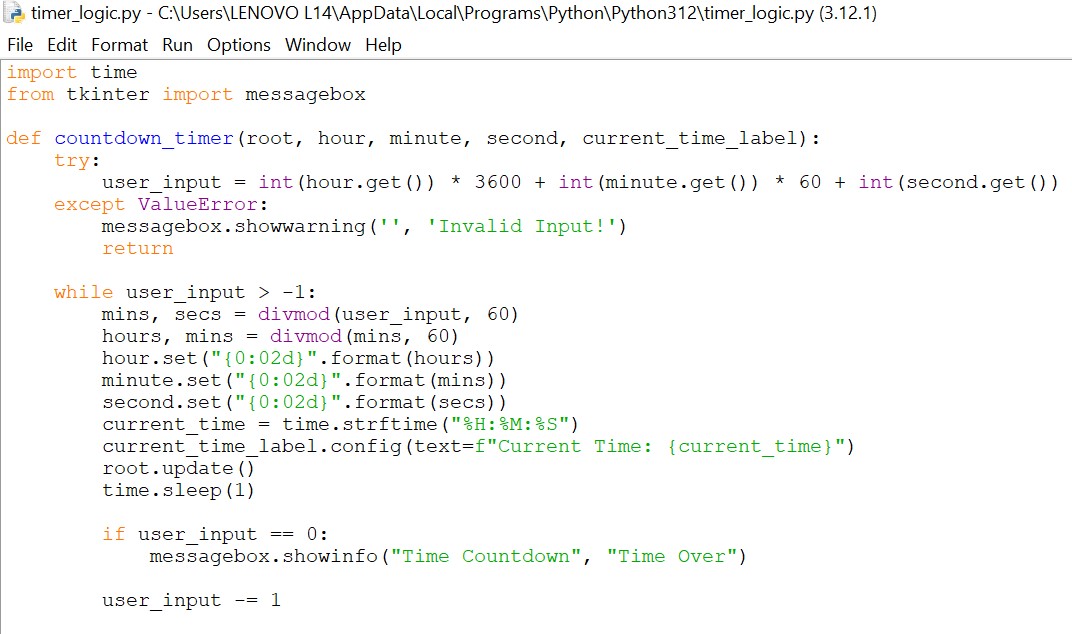
This algorithm starts by creating a graphical user interface (GUI) using Tkinter, a Python library for creating GUI applications.

It initializes a Tkinter root window.

The geometry, title, and background color of the root window are set.

Finally, the root window is returned to be used in the main.py script.

**Timer\_logic.py Algorithm:**



**Figure 3.** Timer Logic in COUNTDOWN TIMER.

**Countdown Timer:**

This algorithm implements the logic for the countdown timer functionality.

It attempts to parse user input into seconds to determine the countdown duration.

If parsing is unsuccessful, a warning message is displayed to the user.

The algorithm enters a loop while the countdown duration (user\_input) is greater than or equal to 0. Within the loop:

1. The user\_input is converted into hours, minutes, and seconds.
2. The hour, minute, and second variables are updated accordingly.
3. The current time is displayed in the GUI using a label widget.
4. The Tkinter window is updated to reflect the changes.
5. The algorithm pauses execution for 1 second using the sleep function to simulate realtime countdown.
6. If the user\_input reaches 0, a time-over message is displayed to notify the user.
7. The user\_input is decremented by 1 in each iteration of the loop until the countdown completes.

**Main**:

A. import time

def countdown\_timer(seconds):

while seconds:

mins, secs = divmod(seconds, 60)

hours, mins = divmod(mins, 60)

timer = f'{hours:02d}:{mins:02d}:{secs:02d}'

print(timer, end="\r")

time.sleep(1)

seconds -= 1

print("Time's up!")

if \_\_name\_\_ == "\_\_main\_\_":

try:

total\_seconds = int(input("Enter the time in seconds: "))

countdown\_timer(total\_seconds)

except ValueError:

print("Please enter a valid number.")

**Figure 4.** Main program for countdown timer.

* 1. This algorithm serves as the main entry point for the countdown timer application.
  2. It starts by creating a Tkinter root window using the create\_gui() function from the gui.py module.
  3. String Var variables (hour, minute, second) are initialized to store user input.
  4. Tkinter Label and Entry widgets are created to display and input the countdown duration.
  5. A Tkinter Button is created to trigger the countdown timer function.
  6. The script enters the Tkinter main loop, where it waits for user interaction and updates the GUI accordingly.

1. **Output:**

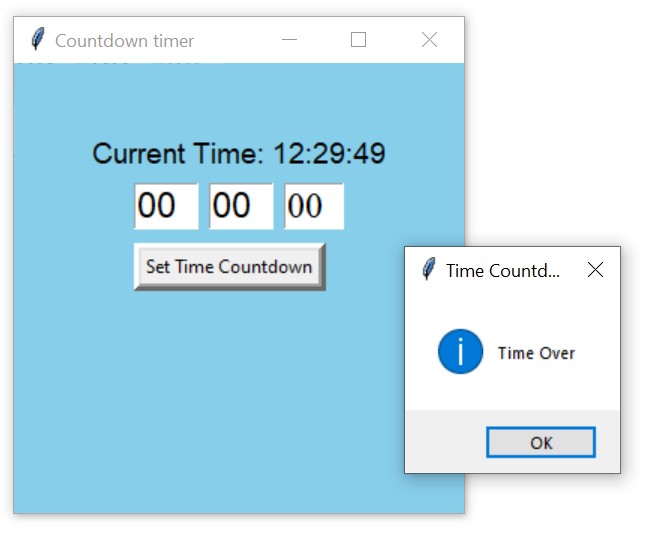


Figure 5. Displaying COUNTDOWN TIMER.

1. **Results and Discussion:**

In the development of the countdown timer application, key results include the creation of a user-friendly machine utilization dashboard and intuitive GUI display screens. This section presents an overview of these outcomes and discusses their implications for enhancing user experience and productivity.

* 1. **Machine Utilization Dashboard:**

The machine utilization dashboard provides users with comprehensive insights into the operational status and efficiency of the system. Through graphical representations and statistical data, users can monitor parameters such as machine running hours, breakdown hours, part count, and overall machine utilization. This real-time visualization of machine utilization metrics enables users to identify patterns, optimize operational workflows, and make informed decisions to improve productivity and efficiency.

By incorporating the countdown timer functionality into the machine utilization dashboard, users gain additional flexibility and control over timesensitive tasks and production schedules. The countdown timer feature allows users to set specific time durations for tasks, track countdown progress, and receive timely alerts upon completion. This integration enhances the functionality of the dashboard, transforming it into a multifunctional tool for time management and productivity enhancement.

* 1. **GUI Display Screens:**

The GUI display screens of the countdown timer application provide users with an intuitive and interactive interface for configuring countdown parameters and monitoring countdown progress. The GUI layout is designed to be user-friendly and visually appealing, with clear labels, input fields, and progress indicators for easy navigation and comprehension.

The countdown timer GUI displays screens dynamically updated in realtime to reflect the remaining time, allowing users to visually track countdown progress and adjust settings as needed. Error handling mechanisms are implemented to provide feedback and guidance to users in case of invalid inputs or runtime errors, ensuring smooth operation and user satisfaction.

Overall, the GUI display screens of the countdown timer application enhance user experience by providing a streamlined interface for setting countdown parameters, monitoring countdown progress, and receiving timely alerts. The intuitive design and functionality of the GUI screens contribute to the usability and effectiveness of the countdown timer application, enabling users to manage time-sensitive tasks with ease and efficiency.

1. **Conclusion and Future Enhancement:**

In conclusion, the development of the countdown timer application has resulted in a valuable tool for time management and productivity enhancement. Through the integration of user-friendly graphical interfaces and robust countdown timer functionality, the application offers users a convenient and efficient solution for managing time-sensitive tasks and improving workflow efficiency.

The countdown timer application provides users with the ability to set specific time durations, track countdown progress, and receive timely alerts upon completion. The machine utilization dashboard and GUI display screens offer comprehensive insights into operational metrics and facilitate informed decisionmaking to optimize productivity. Moving forward, several avenues for future enhancement and expansion of the countdown timer application can be explored. This includes:

**Integration with additional features:** Enhancing the functionality of the countdown timer application by integrating features such as task scheduling, calendar integration, and notification reminders.

**Customization options:** Providing users with the ability to customize the appearance and behavior of the countdown timer application, including themes, notification sounds, and countdown styles.

**Multi-platform support:** Extending the compatibility of the countdown timer application to support multiple platforms, including mobile devices, tablets, and web browsers, to enhance accessibility and usability.

**Advanced analytics:** Implementing advanced analytics and reporting capabilities within the machine utilization dashboard to provide users with deeper insights into productivity trends, performance metrics, and optimization opportunities.

Overall, the countdown timer application serves as a versatile tool for enhancing time management and productivity in various contexts. By continually refining and expanding its features, the application can further empower users to streamline workflows, meet deadlines, and achieve their goals with greater efficiency and effectiveness.

## REFERENCES

1. Chen, 1. C., Chang, K. K., Chang, C. C., and Lai, C. H. The impact evaluation of vehicular signal countdown displays. Research Report Institute of Transportation 2007, Ministry of Transportation and Communications, Taiwan.

1. Chiou, Y. C., and Chang, C. H. Driver responses to green and red vehicular signal countdown displays: Safety and efficiency aspects. Accident Analysis & Prevention 2010: 42-4, p. 1057-1065.

1. Fujita, M., Suzuki, K., and Yilmaz, C. Behavior and consciousness analyses on effect of traffic signals including countdown device for vehicles. Journal of the Eastern Asia Society for Transportation Studies 2007:7, pp. 2289-2304. 4. Ibrahim, M. R., Karim, M. B., and Kidwai, F. A. The effect of digital count-down display on signalized junction performance.

1. Krishna, K., C. V. Effect of countdown display on queue discharge characteristics at signal controlled intersections. M.Tech. Dissertation 2014, Indian Institute of Technology (IIT) Roorkee, Roorkee 247667, Uttarakhand, India.

1. Li, K., Sun, J., Dong, S., and Yu, X. Study on the influence of signal countdown device on traffic safety of intersections. International Conference on Measuring Technology and Mechatronics Automation 2009, Zhangjiajie, Hunan Province, China.

IEEE.

1. Limanond. T., Prabjabok, P., and Tippayawong, K. Exploring impacts of countdown timers on traffic operations and driver behavior at a signalized intersection in Bangkok.

Transport Policy 2010:17-6, p. 420-427.

1. Long, K., Han, L. D., and Yang, Q. Effects of countdown timers on driver behavior after the yellow onset at Chinese intersections. Traffic Injury Prevention 2011: 12-5, p. 538-544. 9. Lum, K. M., and Halim, H. A before-and-after study on green signal countdown device installation. Transportation Research Part F: Traffic

Psychology and Behaviour 2006:9-1, p. 29-41. 10. Ma, W., Liu, Y., and Yang, X. Investigating the impacts of green signal countdown devices: empirical approach and case study in China.

Journal of Transportation Engineering, ASCE 2010:136-11, p. 1049-1055. 11.

Raksuntom, W. The Effects of Countdown Signals on Intersection Capacity. International Transaction Journal of Engineering,

Management, & Applied Sciences & Technologies 2012-3-2, p. 159-165. 12. Rijavec, R., Zakovšek, J., and Maher, T. Acceptability of countdown signals at an urban signalized intersection and their influence on drivers behaviour. PROMET-Traffic & Transportation, 2013:25-1, p. 63-71.

1. Sharma, A., Vanajakshi, L.., and Rao, N. Effect of phase countdown timers on queue discharge characteristics under heterogeneous traffic conditions. Transportation Research Record: Journal of the Transportation Research Board, 2013: No. 2130, p. 93100.

1. Sharma, A., Vanajakshi, L., intersections. Journal of Transportation Engineering, ASCE 2012:138-4, p. 467-478

Girish, V., and Harshitha, M. S. Impact of signal timing information on safety and efficiency of signalized

1. Wu, W. J., Juan, Z. C., and Jia, H. F. Drivers' behavioral decision-making at signalized intersection with countdown display unit. Systems Engineering - Theory & Practice, 2009:29-7, p. 160-165.