Task Scheduler using TIVA Microcontroller

Aishwarya Pitchumani, Meenakshi, Nila Sangamitra Arul, Panchumarthi Mohithi

3rd year, Computer Science and Engineering

Amrita School of Computing, Amrita Vishwa Vidyapeetham

Ettimadai, Coimbatore- 641112

***Abstract* — This project presents a task scheduler designed for the Tiva TM4C123GH6PM microcontroller, enabling multitasking capabilities through a round-robin scheduling method. Leveraging the SysTick timer and GPIOs, the scheduler efficiently manages concurrent tasks, exemplified by controlling LEDs on GPIO Port F with distinct toggling intervals. The implementation demonstrates effective task management, showcasing the scheduler's potential for multitasking in embedded systems.**

I. Introduction

In the realm of embedded systems, the need for effective multitasking mechanisms is paramount to handle diverse and concurrent operations. This project delves into the development and implementation of a task scheduler specifically tailored for the Tiva TM4C123GH6PM microcontroller, aiming to facilitate seamless multitasking capabilities within embedded applications.

At its core, the task scheduler orchestrates the execution of multiple tasks, leveraging the TM4C123GH6PM's innate features such as the SysTick timer and GPIO functionalities. The primary objective is to manage these tasks concurrently, ensuring equitable execution through the utilization of a round-robin scheduling algorithm.

The significance of multitasking lies in its ability to allow the microcontroller to efficiently allocate processing time among various tasks, thereby optimizing resource utilization and enhancing overall system responsiveness. To showcase the scheduler's functionality, individual tasks have been designed to control specific LEDs connected to GPIO Port F. Each task exhibits distinctive blinking patterns with varying time intervals, providing a tangible demonstration of the scheduler's ability to switch between tasks effectively.

By emphasizing the utilization of the Tiva microcontroller's hardware features and its capacity for multitasking, this project endeavors to establish a robust framework for managing parallel operations within embedded systems. The developed task scheduler serves as a foundational cornerstone, paving the way for further advancements in complex embedded applications that necessitate synchronized task management, improved efficiency, and streamlined resource allocation.

Through this endeavor, the project aims to not only showcase the capabilities of the TM4C123GH6PM microcontroller in multitasking scenarios but also to provide a scalable and adaptable solution that can be utilized and expanded upon in a myriad of embedded applications.

II. Primary Objectives

.

- Task Scheduler Development
- Multitasking Implementation
- Hardware Utilization
- LED Control Demonstration
- Framework Establishment

III. Hardware Requirements

- Tiva TM4C123GH6PM Microcontroller

IV. Software Requirements

- Keil µVision Software

V. Methodology For the Project

1. System Understanding and Requirements Analysis:

   To gain a comprehensive understanding of the Tiva TM4C123GH6PM microcontroller's architecture, including its peripherals like GPIOs and SysTick timer.
   To define the requirements for the task scheduler, including multitasking capabilities, LED control, and task switching mechanisms.

2. Environment Setup and Configuration:

   The Keil µVision Integrated Development Environment (IDE) was utilized for developing the task scheduler targeting the Tiva TM4C123GH6PM microcontroller. Keil µVision was chosen for its user-friendly interface and comprehensive support for ARM-based microcontrollers.

3. Task Definition and Design:

   Define individual tasks, specifying their functionalities, execution patterns, and interaction with GPIOs for LED control.
   Allocate stack memory for each task to maintain their contexts during execution.

4. Scheduler Implementation:

   Implemented the task scheduler logic utilizing the SysTick timer for timekeeping and interrupt-based task switching.
   Developed a round-robin scheduling algorithm to manage task execution, ensuring fairness in processing time allocation among tasks.

5. GPIO Configuration and LED Control:

   Configured GPIO Port F pins for controlling the LEDs, setting directions for output and enabling digital functionality.
   Implemented task functions to control specific LEDs, demonstrating distinct blinking patterns with varying time intervals.

6. Interrupt Handling and Context Switching:

   Set up interrupt handlers, specifically SysTick_Handler, to manage the SysTick timer interrupts for task scheduling.Developed context switching routines, saving and restoring task contexts when switching between tasks.

7. Code Compilation and Debugging:

   Compile the code, ensuring it is free from syntax errors and conforms to the Tiva microcontroller's architecture.
   Utilize simulation or debugging tools for testing code functionality, identifying and resolving any runtime issues or logical errors.

8. Deployment and Demonstration:
   Flashed the compiled code onto the Tiva TM4C123GH6PM microcontroller's flash memory using a suitable flashing tool.
   Demonstrated the functionality of the task scheduler by observing the LEDs controlled by individual tasks exhibiting varied blinking patterns.

9. Testing and Optimization:
   Performed comprehensive testing to validate the scheduler's multitasking capabilities, LED control, and task switching functionalities.
   Optimization of the code for efficiency, ensuring minimal resource consumption and effective task management.

VI. Pseudocode

Define Constants:
```
   MAX_TASKS = 4
   TICK_HZ = 1000  // Tick frequency in Hz
   LED1_TOGGLE_DELAY = 1000  // LED1 toggle delay in milliseconds
   LED2_TOGGLE_DELAY = 500   // LED2 toggle delay in milliseconds
   LED3_TOGGLE_DELAY = 250   // LED3 toggle delay in milliseconds
   LED4_TOGGLE_DELAY = 125   // LED4 toggle delay in milliseconds
```

Define Tasks:
```
   Task1:
      Loop:
         Toggle LED1
         Delay(LED1_TOGGLE_DELAY)

   Task2:
      Loop:
         Toggle LED2
         Delay(LED2_TOGGLE_DELAY)

   Task3:
      Loop:
         Toggle LED3
         Delay(LED3_TOGGLE_DELAY)

   Task4:
      Loop:
         Toggle LED4
```

```
    Delay(LED4_TOGGLE_DELAY)
```

Initialize Scheduler:

```
InitializeStacksForTasks()    // Allocate stack memory for tasks
SetSysTickTimer(TICK_HZ)   // Initialize SysTick timer for task scheduling
SetTaskPriorities()        // Set priorities for tasks (if required)
```

SchedulerLoop:

```
Loop:
  if SysTick Interrupt:
    UpdateTickCounter()
    UnblockTasks()  // Unblock tasks waiting for specific ticks
    ScheduleNextTask()
```

TaskDelay(TickCount):

```
currentTask.blockCount = currentTick + TickCount
currentTask.state = BLOCKED
```

SysTickInterruptHandler:

```
IncrementTickCounter()
UnblockTasks()    // Check and unblock tasks waiting for current tick
ScheduleNextTask()
```

ScheduleNextTask():

```
SelectNextReadyTask()
SwitchContextToSelectedTask()
```

UnblockTasks():

```
Loop through tasks:
  if task.state == BLOCKED and task.blockCount == currentTick:
    task.state = READY
```

SwitchContextToSelectedTask():

```
SaveCurrentTaskContext()
LoadSelectedTaskContext()
```

SaveCurrentTaskContext():

Save CPU registers and context to the task's stack

LoadSelectedTaskContext():

Load CPU registers and context from the selected task's stack

InitializeStacksForTasks():

Allocate stack memory for each task and initialize task contexts

SetSysTickTimer(Frequency):

Configure SysTick timer with the specified frequency for task scheduling

VII. Conclusion

The successful implementation of the task scheduler on the Tiva TM4C123GH6PM microcontroller marks a significant milestone in enabling multitasking capabilities within embedded systems. The scheduler, built upon the SysTick timer and GPIO functionalities, demonstrates efficient task management and context switching, allowing concurrent execution of multiple tasks.

Key Achievements:

1. **Multitasking Efficacy**: The scheduler adeptly manages multiple tasks, showcasing fair task allocation and synchronized execution through a round-robin scheduling approach.

2. **LED Control Demonstration**: Tasks controlling distinct LEDs on GPIO Port F exhibit varied blinking patterns, highlighting the scheduler's ability to orchestrate diverse operations concurrently.

3. **Resource Utilization**: Efficient use of stack memory and interrupt-driven task switching optimizes resource utilization, ensuring minimal overhead and effective multitasking.

4. **Reliability and Stability**: Rigorous testing and debugging ensure the scheduler's stability, mitigating runtime issues and guaranteeing reliable task execution.

Acknowledgment

We express our sincere gratitude to all those who contributed to the successful development of the task scheduler for the Tiva TM4C123GH6PM microcontroller. Special thanks to our faculty for their invaluable guidance and expertise and our Institution for providing essential resources Lastly, heartfelt thanks to friends, family, and peers for their unwavering encouragement and understanding throughout this endeavor.

REFERENCES

[1] https://www.udemy.com/course/embedded-system-programming-on-arm-cortex-m3m4/learn/lecture/19392444#overview

[2] Texas Instruments Tiva TM4C123GH6PM Microcontroller Data Sheet

[3] Furber SB. ARM system-on-chip architecture. pearson Education; 2000.

[4] Martin T. The Insider's guide to the Philips ARM7-based microcontrollers. Coventry, Hitex, UK, Ltd. 2005.