SPRINTS

Report On Designing A Real-Time System

July 2023

Presented by
Abdelrhman Walaa

Presented to
Sprints

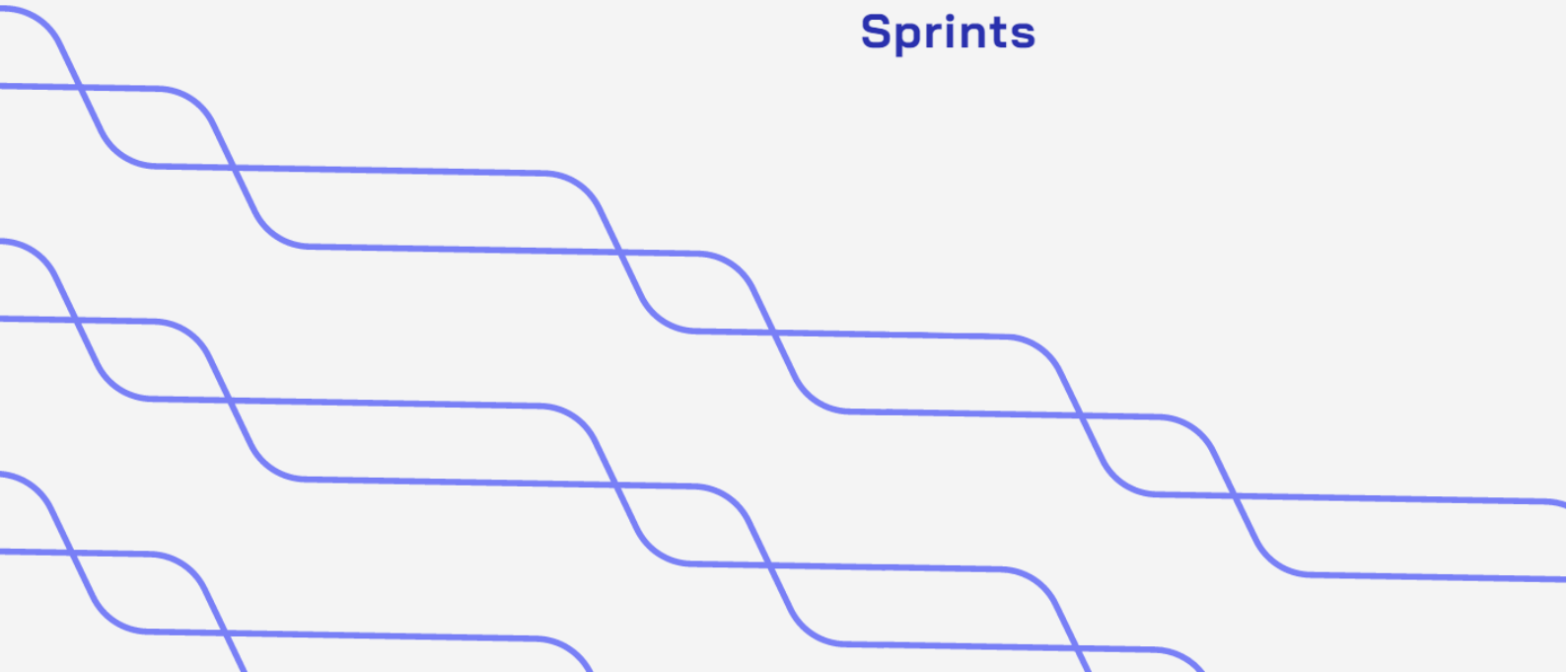

Table Of Content

Report on
Designing a Real-Time System

1. Introduction

Designing a healthcare system using an RTOS (Real-Time Operating System) involves integrating multiple components and ensuring real-time data processing.

1.1. RTOS Task Design

To meet the real-time requirements, organize the system's functionalities into separate tasks, each with a specific priority and execution time:

1.1.1. LCD Task

- Periodically read 4-byte commands from the touch LCD via UART with a speed of **9600** bps.
- Process the commands within **2** ms.

1.1.2. Blood Pressure Task

- Continuously read blood pressure data from the sensor every 25ms.
- Process the data within **3** ms to calculate blood pressure values.

1.1.3. HeartBeat Task

- Continuously read heart beat data from the detector every 100ms.
- Process the data within **1.5** ms to determine heart rate and detect abnormalities.

1.1.4. Temperature Task

- Continuously read temperature data from the sensor every 10ms.
- Process the data within **2.5** ms to monitor body temperature.

1.1.5. Siren Control Task

- Receive commands from the other tasks to activate or deactivate the alert siren.
- Complete siren control operations within **1** ms.

2. System Design

2.1. System Flowchart

2.1.1. Definition

A *Flowchart* (*Figure 1*) is a visual representation or diagram that depicts the sequence of steps or actions in a process or system. It is a graphical tool used to illustrate the flow of control, data, or information through a series of interconnected steps, decision points, and processes.

2.1.2. Flowchart Design

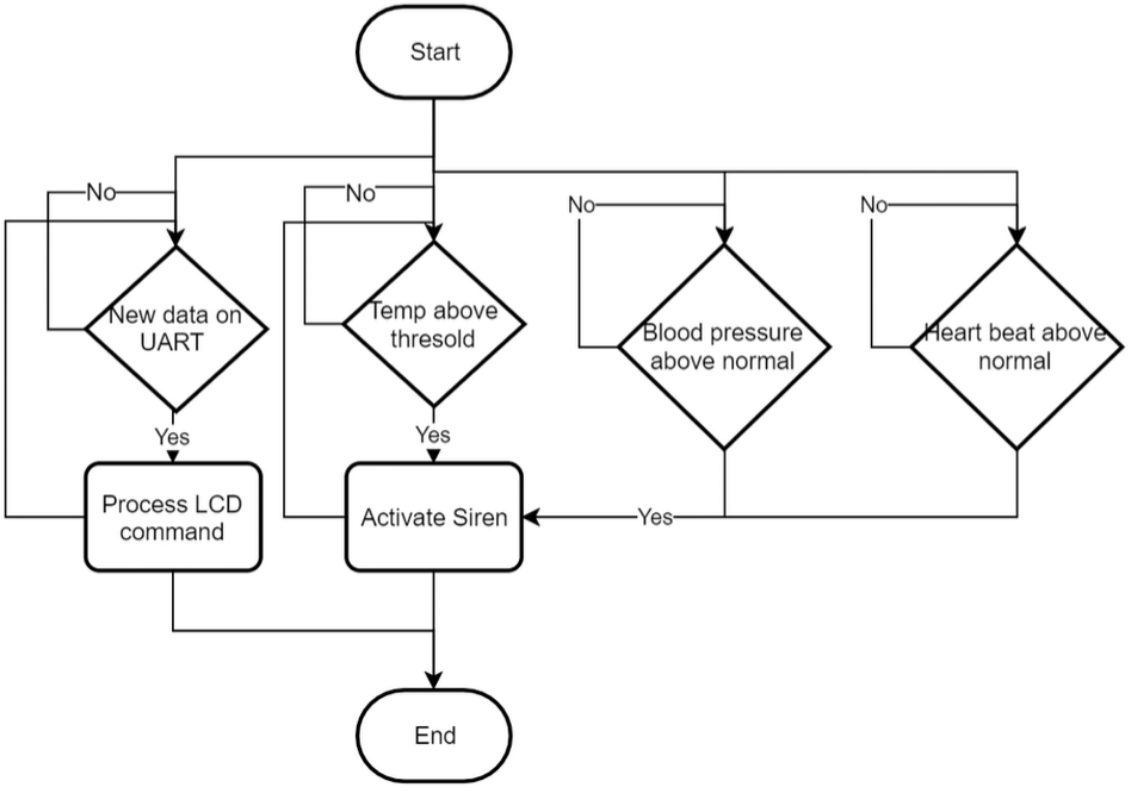

Figure 1. Flowchart Design

2.2. System Analysis

2.2.1. System Data

Task	Task Type	Periodicity (ms)	Deadline (ms)	Execution Time (ms)	Priority
LCD Task	*Periodic*	*100*	*100*	*2*	*1*
Blood Pressure Task	*Periodic*	*25*	*25*	*3*	*3*
HeartBeat Task	*Periodic*	*100*	*100*	*1.5*	*2*
Temperature Task	*Periodic*	*10*	*10*	*2.5*	*4*
Siren Control Task	*Aperiodic*	*-*	*10*	*1*	*5*

Calculations,

- **Total Execution Time** (ms) = 2 + 3 + 1.5 + 2.5 + 1 = ***10*** ms.
- *Since*: **SysTick Value** > **Total Execution Time of All Tasks**, and *since* “Siren Control Task” is an aperiodic task with execution time of ***1*** ms.
- *Therefore*: **SysTick Value** = ***10*** ms.
- *Since*: **Hyperperiod Value** = **LCM(pi)**, where (pi) is all tasks periodicities.
- *Therefore*: **Hyperperiod Value** = ***100*** ms.
- *Since*: **CPU Load** = **Execution Time of Tasks** / **Hyperperiod value**.
- *Therefore*: **CPU Load** = (2 + (3 x 4) + 1.5 + (2.5 x 10) + 1) / 100 = ***0.415***

2.2.2. System Timeline

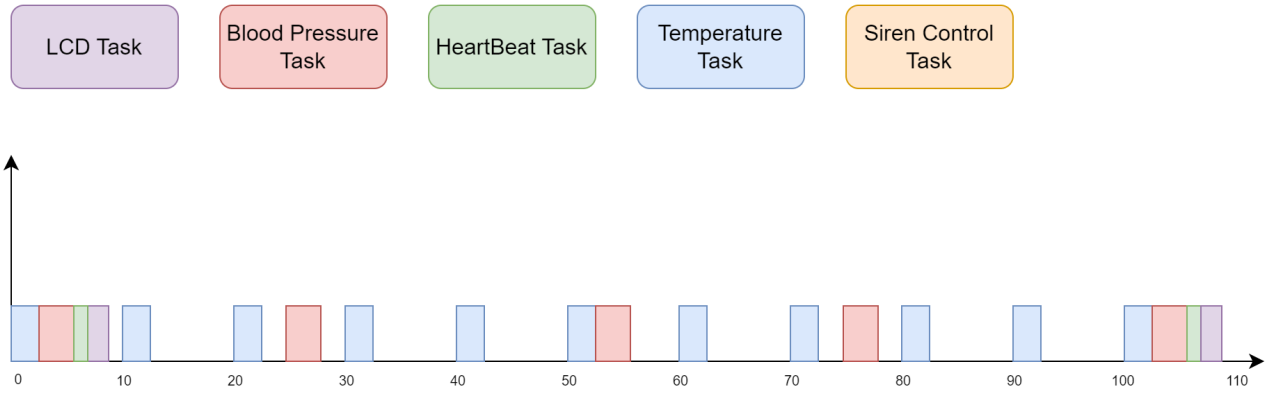

Figure 2. System Timeline

2.2.3. System Simulation

The *SimSo Real-Time Scheduling Simulator* demonstrates that the system is healthy and schedulable. The calculated **CPU Load** of ***0.4167*** aligns closely with the manual calculations. Additionally, the system exhibits schedulability as each task successfully meets its **deadline**.

	Total load	Payload	System load
CPU 1	0.4167	0.4167	0.0000
Average	0.4167	0.4167	0.0000

Figure 3. CPU Load Using SimSo Simulator

All tasks are able to meet their respective deadlines. This means that each task completes its execution within its allocated time frame, ensuring that all deadlines are satisfied.

Based on this observation, it can be concluded that the system is schedulable. Schedulability refers to the property of a system where all tasks can be scheduled and completed within their specified deadlines.

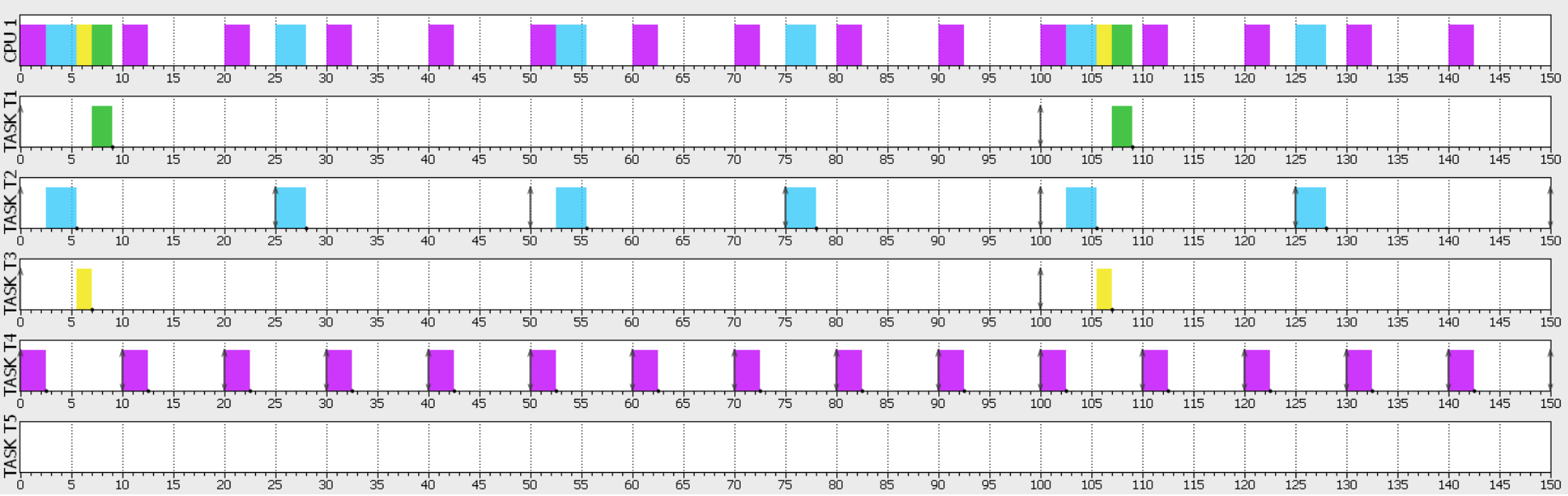

Figure 4. System Timeline Using SimSo Simulator