UNIVERSITY
OF TWENTE.

Soft RTOS Design for Simulated Sensor Network Coordination

Muhammad Arief Kurniawan

April 2025

Contents

Chapter 1

Introduction

This document presents the project objectives and roadmap. This project combines several theoretical fundamentals that revolve around real-time operating systems. It consists of the application of scheduling theory, real-time system design, and distributed system simulation and implementation.

1.1 Objectives

The table below showcases the high-level objectives of the project.

No.	Objective	Description
1	Design and implement soft RTOS	Designing an RTOS in user space
2	Simulate a dynamic sensor network	Create "virtual" sensors that replicate a network of sensors with diverse task priorities and deadlines
3	System analysis	Perform analysis of system performance, timing constraints, and message latency
4	Load handling	Demonstrate how system performs under load

Table 1.1: Table of Project objectives

1.2 Problem Statement

Design and simulate a soft real-time operating system that manages a dynamic set of sensor nodes. Each sensor periodically generates and sends data to the central node. A **Real-Time Operating System (RTOS)** ensures tasks are scheduled and completed as close to deadlines.

1.3 Purpose

The purpose of this project is to design and simulate a soft real-time operating system tailored for managing distributed sensor networks under timing constraints. Such systems are deployed in applications like environmental monitoring and industrial automation where tasks must be completed within a certain time bounds. With simulating sensor nodes, real-time task scheduling, and inter-task communication, this project enables exploration of RTOS design principles and the trade-off between system responsiveness, resource allocation, and deadline adherence.

Chapter 2

Requirement Analysis

This section presents the analysis of required system components and tasks. Additionally, it presents the analysis of functional and non-functional requirements.

2.1 System Components

No.	Component	Description
1	Sensor Nodes (simulated)	Virtual processes generating data peridocially
2	RTOS Kernel Simulator	Core layer managing task scheduling, queues, clocks, and logging
3	Scheduler	Handles task prioritisation and execution
4	Inter-Process Communication (IPC)	Message queues for parsing sensor data
5	Logger	Tracks execution statistics, deadlines, and timing behaviour
6	Visualisation Module	Graphically show Gantt charts or task timelines

Table 2.1: Table of System Components

2.2 Task Classification

No.	Type	Characteristic	Example
1	Periodic	Fixed intervals	Sensor data read
2	Sporadic	Irregular, but with known minimum interval	Motion detection alert
3	Aperiodic	Unpredictable, triggered externally	Network diagnostic task

Table 2.2: Table of Task Classification

2.3 Functional Requirements

No.	Requirement
1	Create and manage simulated sensor nodes with deadlines
2	Implement soft real-time scheduler supporting: - Round-robin (baseline) - Optional: EDF or Rate Monotonic
3	Handle task creation, execution, and inter-task communication
4	Log metrics
5	Support dynamic addition/removal of sensor nodes

Table 2.3: Table of Functional Requirements

2.4 Non-Functional Requirements

No	Requirement
1	Modular and extensible code structure using C++
2	Runtime configurability of task frequencies and deadlines
3	Deterministic task handling for reproducibility in experiments
4	Lightweight simulation suitable for academic environments

Table 2.4: Table of Non-Functional Requirements

2.5 Initial Simulation Parameters

No.	Parameter	Value
1	Number of sensors	3 (scalable)
2	Sensor period range	50ms to 100ms
3	Simulation duration	60 seconds (dynamic)
4	Soft deadline tolerance	10% margin

Table 2.5: Table of Simulation Parameters

Chapter 3

System Architecture

This section showcases an overview of the distributed sensor network and the proposed system architecture.

3.1 Distributed Sensor Network

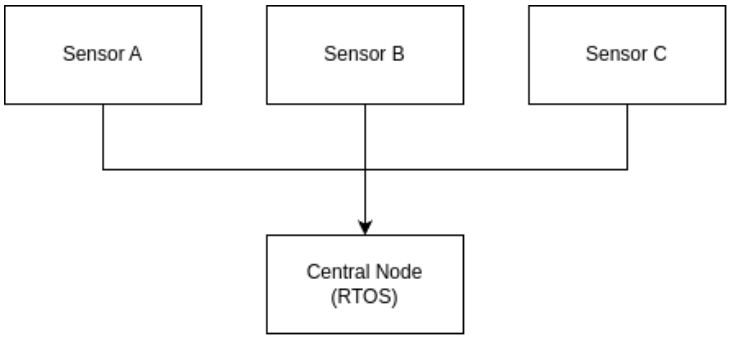

Figure 3.1: Distributed Sensor Network

3.2 RTOS Architecture

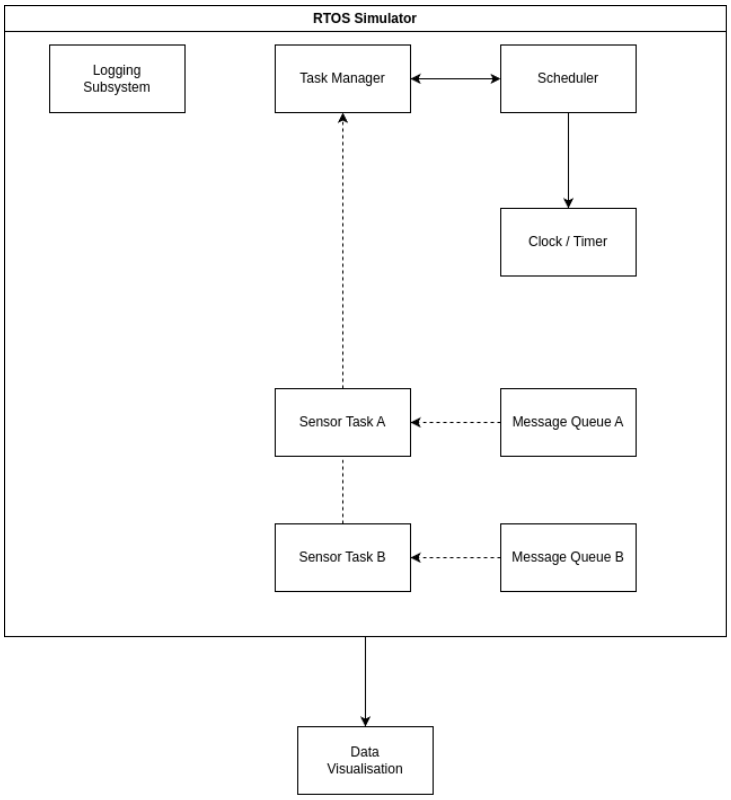

Figure 3.2: System Architecture Diagram

Chapter 4

Timeline

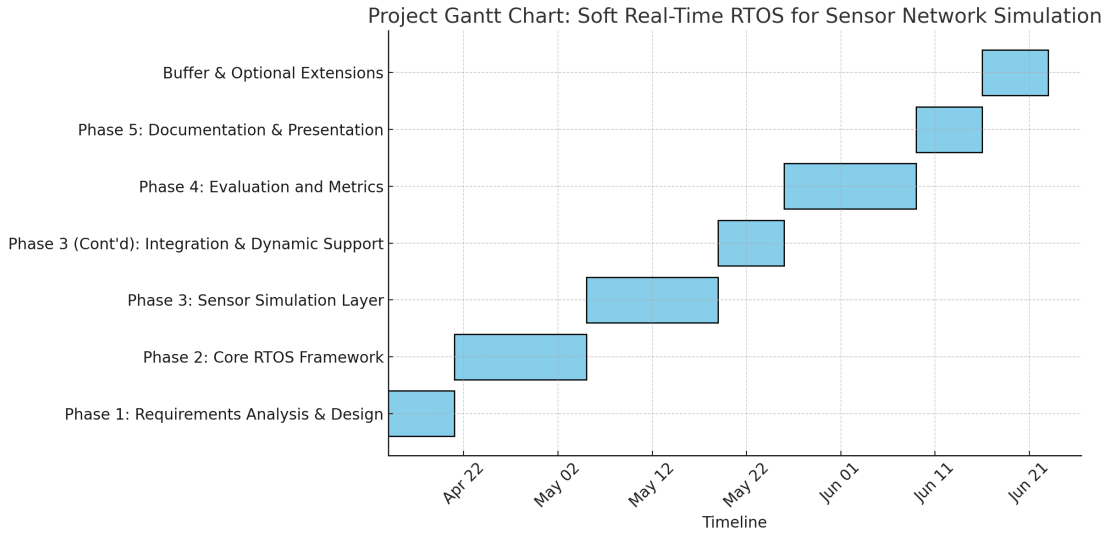

Figure 4.1: Gantt Chart for Project Timeline