

Documentação de Projeto – Parte 2

Design, Estudo da Plataforma

Projeto: Nome do Projeto

Autores: André Luiz Poloni Rizzato, Isabela Amorim Siqueira
e Victor Hugo Polli Neves

Versão: 07-Dez-2023

Parte 2a – Design

1 Introdução

Esse documento se trata de uma continuação do documento 1 apresentado anteriormente, que trouxe diversos aspectos do sistema escolhido para ser projetado: Controlador de elevadores. Cabe ressaltar os tópicos abordados em cada seção do documento passado, auxiliando na interpretação deste documento a ser trabalhado.

Na parte A foram desenvolvidos os conceitos de operação (CONOPS) através dos seguintes tópicos: breve contextualização e introdução ao tema, descrição do sistema, interação com o usuário, identificação dos stakeholders, exposição dos requisitos de Stakeholders e apresentação dos cenários de operação.

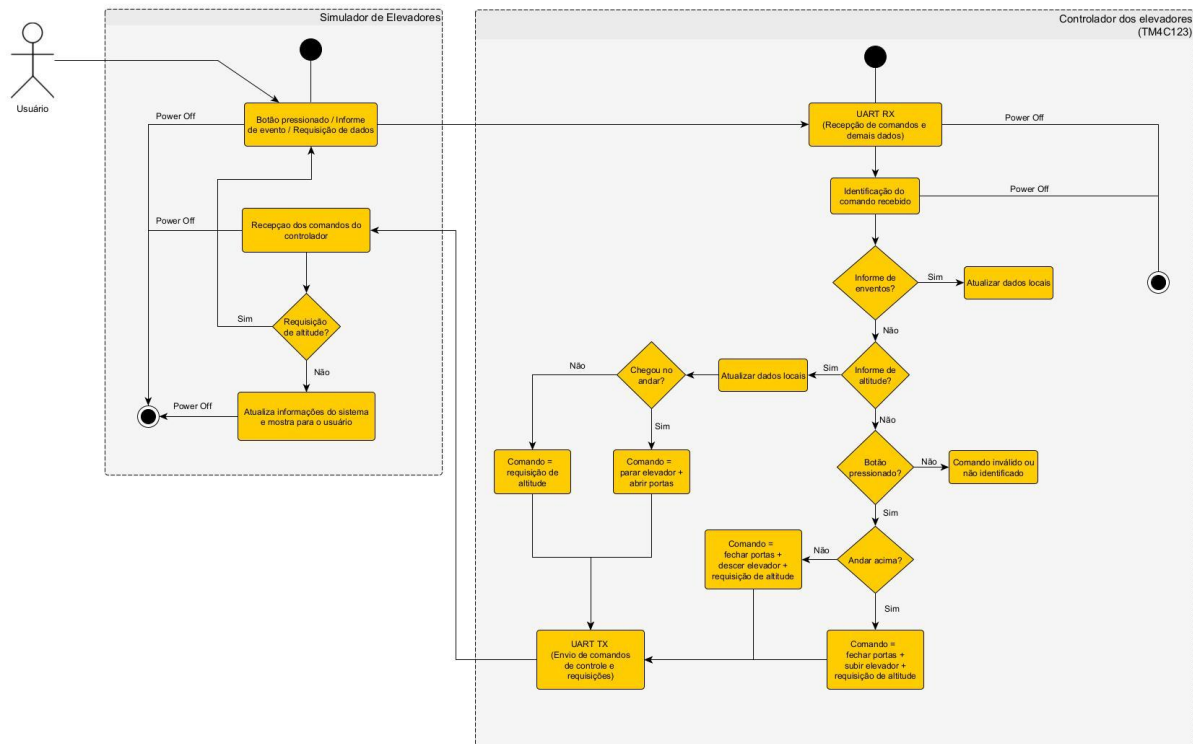
Já na seção B foi realizado o reconhecimento do domínio do problema, mostrando quais áreas de conhecimento são necessárias e onde o sistema a ser desenvolvido se insere.

Por fim, na parte C foi elaborada a especificação do sistema, a explicação da arquitetura funcional, e as especificações funcionais e não funcionais de fato além das restrições.

Diferentemente do documento 1, neste o objetivo será de colocar informações mais detalhadas e técnicas, além de falar sobre o estudo da plataforma.

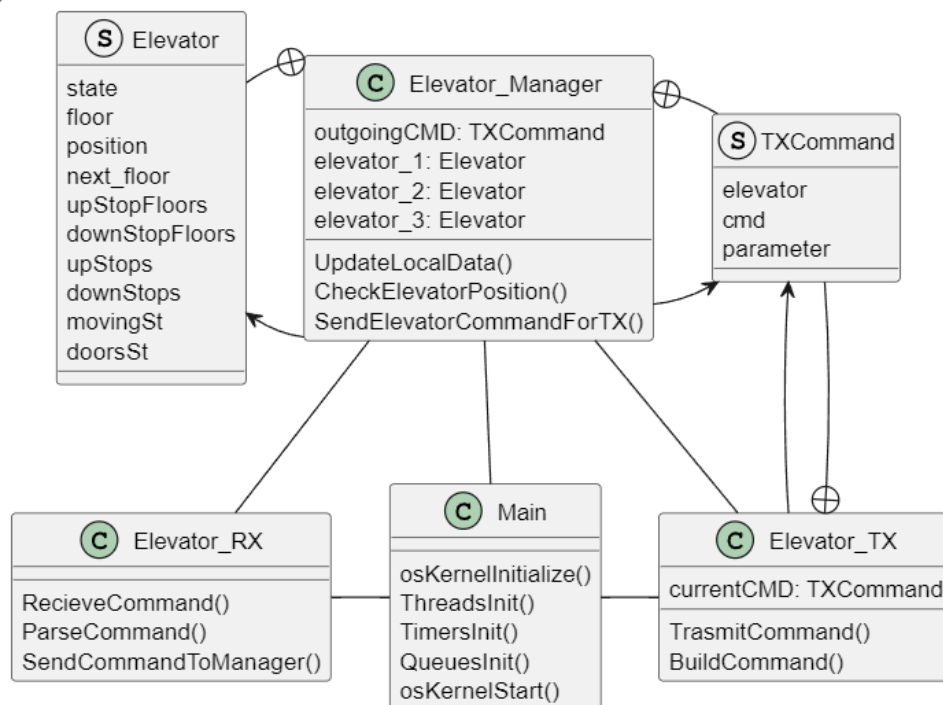
2 Arquitetura Funcional

O diagrama da arquitetura funcional foi atualizado para detalhar melhor o processo de tratamento de comandos por parte do controlador de elevadores:



3 Arquitetura Física

O diagrama estrutural abaixo mostra a estrutura planejada do programa do controlador de elevadores, destacando como as 3 threads (tarefas) são criadas pela Main (código principal - inicialização). A thread **Elevator_RX** é responsável por receber os dados da serial, enviados pelo simulador, e repassá-los à thread **Elevator_Manager**, que por sua vez é responsável por identificar e validar os comandos/dados recebidos, atualizar dados locais do controlador e enviar comandos de controle/requisições para a thread **Elevator_TX**. Esta última é responsável por transmitir os pacotes recebidos via serial.



4 Interface com o Usuário

A interface do usuário é o simulador de elevadores fornecido. Nele o usuário é capaz de monitorar e interagir 3 elevadores, além de visualizar o prédio de 15 andares onde estão os elevadores. A interface do simulador disponibiliza informações como andar, posição, estado das portas e da cabine para cada um dos três elevadores. O usuário controla os elevadores clicando nos botões dos mesmos, sendo 16 botões que representam cada andar.

Ao iniciar o simulador, o usuário deve especificar a porta serial COM em que a placa TM4C (controlador) está conectada, assim como a velocidade da porta. O usuário é então apresentado a interface de controle do simulador, como mostra a imagem abaixo:



5 Mapeamento da Arquitetura Funcional à Arquitetura Física

Abaixo, o mapeamento da arquitetura funcional (colunas) à arquitetura física (linhas):

	Recepção de comandos e demais dados do simulador	Identificação do comando recebido	Atualizar dados locais e monitorar estados dos elevadores	Envio de comandos de controle e requisições	Recepção dos comandos do controlador	Atualizar Interface visual para o usuário	Envio de comandos e dados para o controlador
Elevator_RX (MCU UART)	X	X					
Elevator_Manager (MCU)			X				
Elevator_TX (MCU UART)				X			
Simulador					X	X	X

6 Arquitetura do Hardware

O hardware do controlador é composto apenas pela placa de desenvolvimento TM4C123XGL, que conta com a MCU TM4C123GH6PM, baseada na arquitetura Arm Cortex-M4F. Este microcontrolador oferece uma ampla variedade de periféricos, porém o projeto requer apenas o uso da interface de comunicação serial (UART), utilizada para estabelecer a interação (troca de dados e comandos) entre o controlador e o simulador de elevadores.

A programação do microcontrolador foi facilitada com o uso do SDK TivaWare, desenvolvido e disponibilizado pela Texas Instruments, que fornece drivers para todos os periféricos e códigos de exemplo para dezenas de aplicações para o TM4C123G.

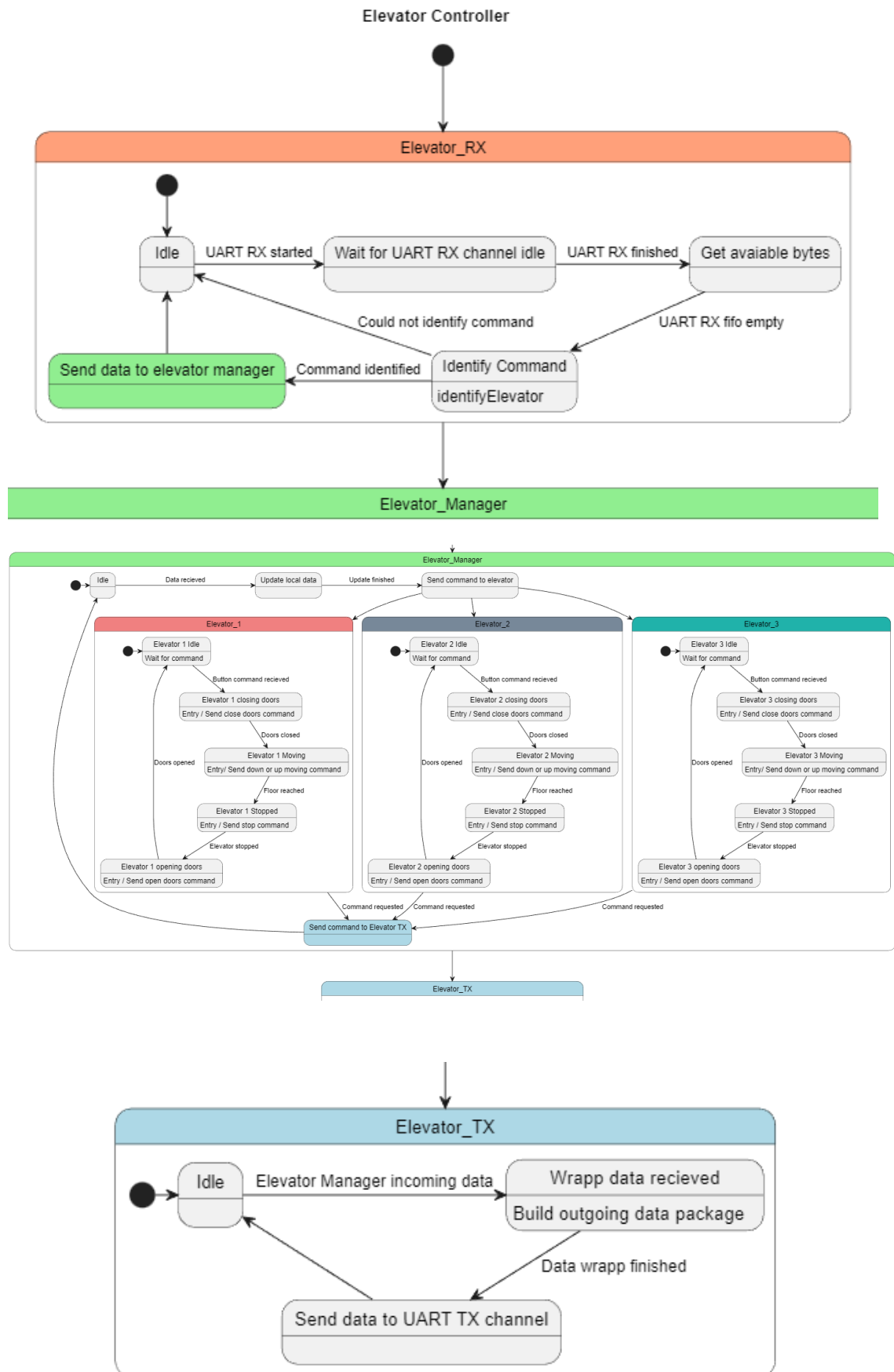
7 Design Detalhado

A tarefa Elevator_RX é responsável por receber os comando da UART, identificá-los e validá-los e então enviá-los à tarefa Elevator_Manager.

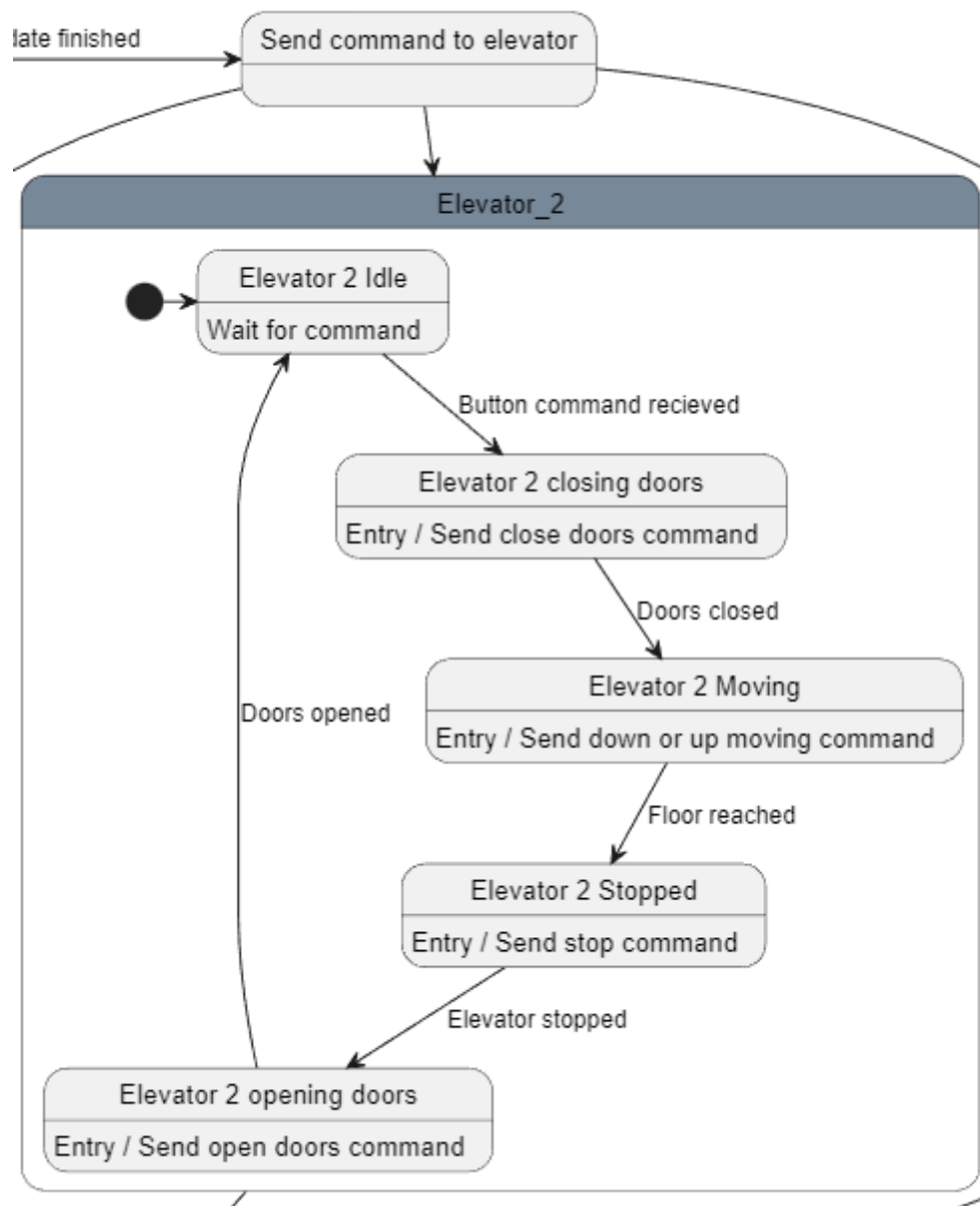
A tarefa principal será o Elevator_Manager, que será responsável por armazenar e atualizar os dados e estados dos três elevadores, monitorando suas altitudes enquanto se movem, estado das portas, botões, etc. Além disso, esta tarefa é responsável por requisitar o envio de mensagens, se comunicando com a terceira tarefa, Elevator_TX.

A tarefa Elevator_TX é responsável por construir as mensagens requisitadas pela tarefa Elevator_Manager, e então enviá-las pela porta serial UART.

Abaixo o diagrama de estados das três tarefas que compõem o programa do controlador:



Como pode-se observar, a tarefa Elevator_Manager é composta pelas máquinas de estado dos elevadores, que podem ser melhor visualizadas abaixo:



Parte 2b – Estudo da Plataforma

Esta parte do documento representa o estudo da placa de prototipação, IDE, bibliotecas, periféricos e protocolos necessários para o desenvolvimento do projeto.

1 TIVA C TM4C123GXL

A placa de desenvolvimento TIVA C TM4C123GXL é a base da solução, oferecendo um microcontrolador Arm Cortex-M4F, 256 kB de memória flash, 32 kB de RAM e diversos periféricos integrados. Estes incluem módulos USB, ADC, UART, SPI, I2C dentre outros.

2 FreeRTOS

O FreeRTOS é um sistema operacional de tempo real (RTOS) que fornece um ambiente de execução robusto para o desenvolvimento de sistemas embarcados. Ele oferece escalonamento de tarefas, gerenciamento de memória e outros recursos essenciais para operações em tempo real.

3 Simulador de Elevadores

O simulador de elevadores é uma parte crítica da plataforma, proporcionando um ambiente virtual para testar e validar o controlador. Ele permite simular o comportamento do sistema de elevadores conforme as especificações fornecidas.

4 IDE Keil uVision

O Keil uVision é uma IDE (Integrated Development Environment) amplamente utilizada para o desenvolvimento de sistemas embarcados. Ele oferece recursos como edição de código, depuração, compilação e simulação, facilitando o desenvolvimento e teste de aplicações embarcadas.

5 UART

A comunicação serial UART é uma parte fundamental do sistema, permitindo a troca de dados entre o controlador e o simulador de elevadores. O uso eficiente da UART é essencial para garantir a comunicação adequada entre os componentes do sistema.

6 plantUML

A ferramenta plantUML é empregada para criar diagramas UML que auxiliam na visualização e comunicação da arquitetura do sistema. Esses diagramas são valiosos para entender a estrutura do software e a interação entre os diferentes módulos.