

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

MARCELO TEIDER LOPES  
STEFAN CAMPANA FUCHS

**DOCUMENTO DE ESPECIFICAÇÃO E ESTUDO DA  
PLATAFORMA**

APS

**CURITIBA**

**2014**

MARCELO TEIDER LOPES  
STEFAN CAMPANA FUCHS

## **DOCUMENTO DE ESPECIFICAÇÃO E ESTUDO DA PLATAFORMA**

APS de Sistemas Embarcados apresentada ao Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção dos títulos de “Bacharel em Sistemas de Informação” e “Engenheiro em Computação”.

Orientador: Prof. Douglas Paulo Bertrand Renaux

**CURITIBA**

**2014**

## LISTA DE FIGURAS

FIGURA 1 – Diagrama de blocos do sistema. ....	6
--	---

## SUMÁRIO

<b>1</b>	<b>ESPECIFICAÇÃO DO PRODUTO</b>	<b>5</b>
1.1	INTRODUÇÃO	5
1.2	OBJETO	5
1.3	DOMÍNIO DO PROBLEMA	5
1.4	CONTEXTO	6
1.5	INTERFACES	6
1.5.1	Comunicação serial	6
1.6	ESPECIFICAÇÃO FUNCIONAL	6
1.7	ESPECIFICAÇÃO NÃO-FUNCIONAL	8
<b>2</b>	<b>ESTUDO DA PLATAFORMA</b>	<b>9</b>
2.1	UART	9

# 1 ESPECIFICAÇÃO DO PRODUTO

## 1.1 INTRODUÇÃO

Elevadores são dispositivos que estão presentes na maioria dos edifícios, sendo em muitos casos essenciais para o acesso das pessoas aos andares mais elevados. Um elevador, de forma genérica, possui sensores (botões de cada andar, sensores de posição) e atuadores (motor de subida/descida, motor das portas e luzes dos botões de andar).

Este projeto tem o objetivo de desenvolver um controlador de elevador com o Kit LPC1768, usando o sistema operacional CMSIS-RTOS. O controlador será responsável por gerenciar todos os aspectos da lógica de funcionamento do elevador.

## 1.2 OBJETO

O produto tem o propósito de ser um controlador de elevador, recebendo requisições de usuários nos botões de cada andar e nos botões internos, e efetuando o controle dos atuadores (luzes de botões, motor de subida/descida e portas) conforme essas requisições. De forma eficiente, o elevador deve atender às demandas dos usuários.

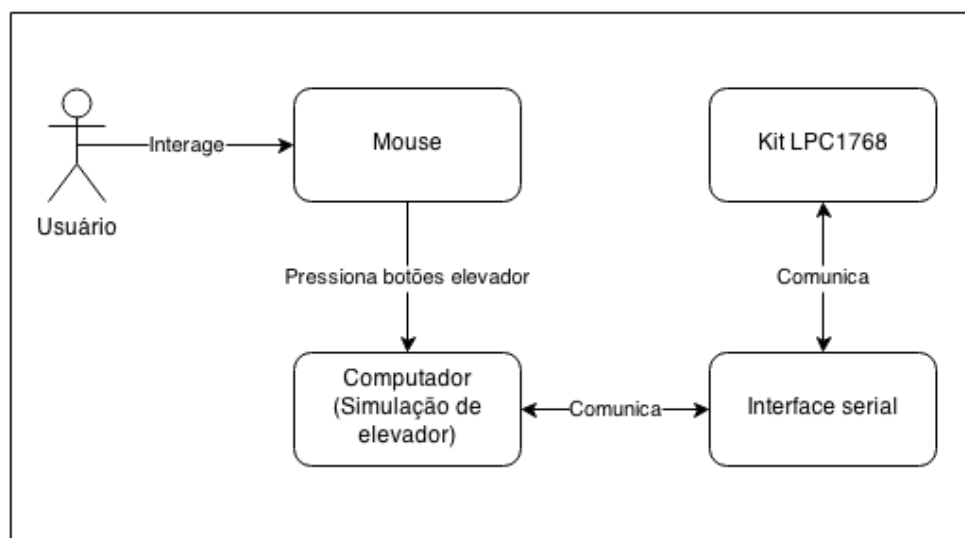
## 1.3 DOMÍNIO DO PROBLEMA

Para o gerenciamento de um elevador, há aspectos importantes que devem ser observados com relação ao seu funcionamento. O elevador possui limitações quando à sua aceleração máxima. Portanto, há um intervalo mínimo de tempo para ele conseguir parar quando recebe uma requisição de um usuário. De forma mais detalhada, estas limitações temporais estão expostas na seção 1.7.

Com relação à segurança, as portas somente devem ser abertas quando o elevador está parado em algum andar (posicionado dentro de uma faixa de segurança). Também, o elevador só deve mover-se com as portas fechadas.

## 1.4 CONTEXTO

A Figura 1 apresenta o diagrama de blocos com uma visão geral do sistema. Um computador executa continuamente uma simulação de elevador. O usuário interage diretamente com esta simulação através do mouse, pressionando botões (internos e externos) do elevador. O Kit LPC1768 faz o papel de controlador do elevador, gerenciando a lógica de movimentação e de abertura/fechamento de portas. O Kit e o computador comunicam-se via interface serial, sendo que o Kit controla o comportamento do elevador na simulação, e o computador envia ao Kit comandos de botões do usuário.



**Figura 1:** Diagrama de blocos do sistema.

## 1.5 INTERFACES

[TODO: INSERIR FIGURA DO SIMULADOR]

### 1.5.1 COMUNICAÇÃO SERIAL

O computador (executando a simulação de elevador) e o Kit LPC1768 se comunicarão através de uma interface serial. Os dados serão transmitidos a uma taxa de 115200 bits por segundo. A transmissão, recepção e interpretação dos dados (em ambos os lados) será feita com caracteres ASCII.

## 1.6 ESPECIFICAÇÃO FUNCIONAL

Os requisitos funcionais levantados para o *software* são:

- RF 1. O sistema deverá ligar a luz de um botão quando este for pressionado.
- RF 2. O sistema deverá desligar a luz de um botão quando o elevador parar no andar correspondente ao botão.
- RF 3. O sistema deverá abrir as portas quando o elevador parar em um andar.
- RF 4. O sistema deverá impedir que as portas sejam abertas quando o elevador não estiver posicionado em um andar.
- RF 5. O sistema deverá impedir que as portas sejam abertas quando o elevador estiver em movimento.
- RF 6. O sistema deverá impedir que o elevador se movimente quando as portas estiverem abertas.
- RF 7. O sistema deverá atender a requisições de andar, feitas através dos botões internos e externos.
  - RF 7.1. O sistema deverá manter a direção de seu movimento (subida/descida) atual, atendendo todas as requisições que estejam no trajeto.
    - RF 7.1.1. O sistema deverá parar o elevador quando ele estiver passando por um andar que o botão interno correspondente tenha sido pressionado.
    - RF 7.1.2. O sistema deverá parar o elevador quando ele estiver passando por um andar que o botão externo tenha sido pressionado, se estiver indo na mesma direção que a requisição foi feita.
  - RF 7.2. O sistema deverá inverter a direção de movimento (subida/descida) se houverem requisições na direção contrária, porém somente quando tiver atendido a todas as requisições do trajeto atual.
  - RF 7.3. O sistema deverá parar o elevador quando não houverem requisições pendentes.
  - RF 7.4. O sistema deverá fechar as portas do elevador quando ele estiver ocioso.
- RF 8. O sistema deverá esperar no mínimo 5 segundos após as portas terem sido abertas antes de fechá-las novamente.
- RF 9. O sistema deverá esperar meio segundo após fechar as portas antes de deslocar o elevador.
- RF 10. O sistema deverá esperar meio segundo após o elevador parar antes de abrir as portas.

## 1.7 ESPECIFICAÇÃO NÃO-FUNCIONAL

Os requisitos não-funcionais levantados para o *software* são:

- RNF 1. O sistema deverá parar o elevador após chegar no andar em no máximo 40 milissegundos.
- RNF 2. O sistema deverá responder a uma requisição em no máximo 40 milissegundos.
- RNF 3. O sistema deverá acender a luz correspondente em no máximo 10 milissegundos após um botão ter sido pressionado.



## 2 ESTUDO DA PLATAFORMA

### 2.1 UART

A UART é o periférico presente no Kit LPC1768 responsável por efetuar a comunicação serial. Ela efetua a conversão de dados de I/O entre a interface serial e a CPU, gerenciando o processo de transmissão e recepção de dados. A UART possui filas FIFO de recepção e transmissão, configuráveis, permitindo a geração de interrupções por número de caracteres na fila ou por *timeout*.

Os registradores utilizados para a inicialização da UART são utilizados como se segue (considerando-se que a UART0 da placa foi utilizada no projeto): O registrador PINSEL é utilizado para selecionar os pinos da UART, e o PINMODE é utilizado para configurar os modos dos pinos. O registrador PCONP controla a alimentação da UART, sendo que o bit 3 deve ser ativado para ligá-la. O registrador PCLKSEL0 controla a saída do clock para a UART, sendo que os bits 7:6 devem ser habilitados para que o clock seja ativado. O baudrate é ativado pelo registrador U0LCR, sendo que o bit 7 deve ser setado. Os registradores DLL (Divisor Latch LSB register) e DLM (Divisor Latch MSB register) podem então ser utilizados para configurar o valor de baudrate, de acordo com a fórmula definida na documentação do Kit LPC1768.

O registrador U0LCR é então utilizado para configurar o tamanho da word (bits 1:0), stop bits (bit 2) e paridade (bit 3 para habilitar ou não, bits 5:4 para selecionar o tipo). O registrador U0FCR é utilizado para habilitar e limpar as FIFOs de RX e TX (setando os bits 2:0) e para configurar o número de caracteres para trigger de interrupção (bit 7). As interrupções da UART são então ativadas setando-se os bits 1:0 de U0LCR.