

# Eletrônica Embarcada

## Projeto Final - Controle para Elevador via Bluetooth

Erik Lopes Targino  
190086823  
Engenharia Eletrônica - UnB  
190086823@aluno.unb.br

Julia Beatriz Santos Pereira  
190090081  
Engenharia Eletrônica - UnB  
190090081@aluno.unb.br

João Pedro Bandeira Raposo  
190089709  
Engenharia Eletrônica - UnB  
190089709@aluno.unb.br

**Resumo - Este relatório descreve o desenvolvimento de um software para gerenciar um elevador controlado pelo microcontrolador PIC16F1827. O software permite aos usuários escolher andares de destino via Bluetooth e exibe informações úteis em um painel de LED.**

### I. INTRODUÇÃO

Este relatório detalha o desenvolvimento de um software específico para gerenciar um elevador. Este elevador é comandado pelo microcontrolador PIC16F1827 e opera com uma tensão de 5V. O software foi projetado para que os usuários possam selecionar facilmente o andar que desejam ir. Após a escolha, o elevador se move para o andar desejado, monitorando e informando o andar atual durante a viagem. Ao chegar no andar escolhido, o elevador pausa brevemente por 2 segundos para aguardar possíveis novas chamadas. Se alguém chamar o elevador nesse período, ele se movimenta para o novo destino. Caso contrário, ele retorna ao térreo.

O painel de controle do elevador mostra informações úteis como o andar atual, a direção em que está se movendo e os andares de destino em uma matriz de LEDs. O movimento do elevador é controlado pelo microcontrolador, que recebe comandos através de uma interface de comunicação serial. Dependendo do comando recebido, o sinal "DIR" é alterado, fazendo com que o elevador suba ou desça.

Além disso, o sistema é equipado para coletar informações importantes como a posição atual e a velocidade do elevador, utilizando para isso um *Encoder* e um sensor óptico. A temperatura do ambiente é monitorada por um sensor LM35, e a posição exata do elevador é determinada com a ajuda de sensores de efeito Hall A3144.

### II. PROJETO

#### A. Princípios de Operação

Desenvolvemos um sistema de controle para o elevador que é capaz de responder automaticamente a pedidos feitos de qualquer andar.

Quando o elevador está no térreo e não recebe nenhum pedido de deslocamento, ele permanece estacionado. Porém, se alguém solicitar o elevador do primeiro andar, ele se movimenta para cima até alcançar esse andar. Ao chegar, as portas se abrem e o elevador espera por dois segundos antes de decidir seu próximo movimento. Se não houver mais pedidos,

ele retorna ao térreo. Caso contrário, ele responde à nova solicitação e se desloca para o andar requerido.

Em uma situação onde há pedidos para subir e descer simultaneamente, o elevador primeiro sobe, faz uma breve parada de dois segundos no andar solicitado e, em seguida, desce. Se o elevador está no térreo e recebe pedidos para múltiplos andares, ele segue para o andar mais alto e depois desce, parando em cada andar que foi solicitado.

No que diz respeito à comunicação Bluetooth, utilizamos o software HTerm em computadores ou o aplicativo Serial Bluetooth Terminal em smartphones. Esses dispositivos se conectam com o módulo Bluetooth HC-06, permitindo a transmissão eficiente de dados para o controle do elevador.

#### B. Dinâmica de Operação do Elevador

O método utilizado para atender aos pedidos de movimentação do elevador é guiado por uma série de critérios, como o estado de movimento do elevador, o andar em que ele está e as solicitações recebidas. Esses fatores são analisados para decidir se o elevador deve subir ou descer. Assim, o elevador não segue para um destino pré-definido; em vez disso, a cada interrupção causada pelos sensores, o elevador reavalia a situação para tomar a próxima decisão. Isso é feito seguindo um fluxograma de estados específico.

Em cada parada, o elevador verifica se existem pedidos de subida ou descida. Se houver, ele atende esses pedidos antes de entrar no ciclo de decisão. Quando no térreo, o elevador verifica continuamente por novas solicitações e, se encontradas, ele inicia a subida. Nos primeiros e segundos andares, o processo de decisão é influenciado por uma hierarquia de prioridades: Estado de movimento, solicitação ativa e destino solicitado.

Essa estratégia permite que o percurso do elevador seja otimizado, considerando todas as solicitações feitas, independentemente de sua ordem ou do tempo em que foram feitas. No terceiro e último andar, após atender às solicitações, a única opção restante para o elevador é descer.

#### C. Medição de Temperatura e Velocidade

Neste projeto, estamos coletando dois tipos principais de dados: a temperatura e a velocidade. Para medir a temperatura, usamos um sensor LM35, que está conectado ao pino RA2 do nosso microcontrolador. Esse pino é responsável por

receber sinais analógicos. Para processar corretamente esses sinais, configuramos o conversor analógico-digital (ADC) do microcontrolador para usar uma fonte de relógio de FOSC/16, garantindo um Tempo de Aquisição Digital (TAD) de aproximadamente 1 microssegundo. Além disso, ajustamos a referência de tensão positiva no regulador de tensão fixo (FVR) em 2.048 volts e a negativa em VSS.

O LM35 é capaz de medir temperaturas que variam de 0°C a 150°C, correspondendo a uma faixa de tensão de 0 a 1,5 volts. Entretanto, no esquemático a faixa de temperatura exposta é de 0°C a 100°C, e a tensão de 0 a 889 milivolts. Com um ADC de 10 bits, que tem 1024 valores possíveis, calculamos a resolução do ADC como sendo 2 milivolts. Dessa forma, a resolução da nossa medição de temperatura é de aproximadamente 0,225°C para cada incremento no ADC.

Para medir a velocidade, usamos um Encoder acoplado a um motor e um sensor óptico. O Encoder está conectado ao pino RA4 do microcontrolador. Para capturar os dados de velocidade, configuramos o periférico CCP4 em modo de captura e o Timer1 com uma fonte de relógio FOSC/4 e um prescaler de 1:4, estabelecendo um período mínimo de 1  $\mu$ segundos. Isso nos permite medir o intervalo de tempo entre cada pulso emitido pelo Encoder. Com base nesse intervalo de tempo e na distância percorrida por cada pulso, calculamos a velocidade como 0.837 milímetros dividido pelo tempo capturado em segundos.

#### D. Display da Matriz de LEDs

Para controlar a matriz de LEDs do elevador, criamos variáveis específicas que representam o andar atual e o estado do elevador (se está subindo, descendo ou parado). Essas informações são exibidas na matriz para informar os passageiros.

À medida que o elevador passa por cada andar, ele compara o andar atual com o andar solicitado. Dependendo dessa comparação, as primeiras quatro colunas da matriz mostram o número do andar, e as três últimas colunas indicam a direção do movimento do elevador e se há alguma espera.

Em cada parada do elevador, uma interrupção é acionada para atualizar a matriz de LEDs. Para isso, escrevemos um número na memória flash do microcontrolador PIC. Utilizamos um vetor de 8 palavras de 8 bits para controlar a matriz, onde um bit alto acende o LED e um bit baixo apaga. Cada palavra de 8 bits controla uma coluna inteira da matriz.

Configuramos o microcontrolador para enviar os dados para a matriz de LEDs. Habilitamos os periféricos MSSP1 e MEMORY no Microchip Code Configurator (MCC) do projeto.

Os dados são enviados para a matriz usando a interface SPI (Serial Peripheral Interface). Criamos uma função no arquivo `main.c`, chamada `matrixUpdate()`, que por sua vez chama a função `txSpi(uint8_t *data, size_t dataSize)`. Essa função transmite os dados para a matriz via SPI, utilizando a função `SPI1_ExchangeBlock(data, dataSize)` para armazenar os endereços na memória flash

e enviar para a matriz, onde serão exibidos conforme programado.

#### E. Comunicação

Nosso sistema de comunicação do elevador opera por meio de dois métodos principais: comunicação serial e Bluetooth.

Para a comunicação serial, que transmite informações como a localização atual, temperatura, velocidade e o andar em que o elevador se encontra, usamos o periférico EUSART configurado para comunicação assíncrona serial. Definimos a velocidade de transmissão de dados (Baud Rate) em 19.600 bps. O envio dos dados é feito por meio de interrupções geradas pelo Timer0, que estão programadas para ocorrer a cada 300 milissegundos.

#### F. Fluxogramas

O diagrama de fluxo 1 ilustra graficamente o conceito que elaboramos para o sistema de estados do projeto, o qual abrange os movimentos do elevador.

### III. SIMULAÇÕES

As simulações evidenciaram resultados funcionais no processamento das solicitações do elevador. Utilizando o analisador lógico, observamos tanto a transmissão de dados quanto a ativação do pino 'CS'. As interrupções foram gerenciadas de forma eficiente em cada andar. No entanto, algumas questões surgiram relacionadas ao funcionamento do SDO, com a exibição de mensagens de erro ao longo da simulação, causadas pela definição do pino como output. Todavia, o elevador se manteve eficiente no processo de antedimento de solicitações.

### IV. CONCLUSÃO

A partir do desenvolvimento do código e das simulações realizadas para os testes de funcionamento do elevador, foi possível desenvolver um código funcional e eficiente para o equipamento, de modo que todas as solicitações sejam atendidas da forma mais eficiente possível.

Em relação às problemáticas do projeto, houve um problema referente às medições de altura, temperatura e velocidade do elevador, no qual o programa acusava uma dificuldade de alocação de memória dos comandos de envio destes dados. Não foi possível desenvolver também a parte do código em que são mostrados os próximos destinos do elevador devido ao tempo exigido para testes de eficiência do movimento.

Sendo assim, pelo fato de as partes faltantes do código não comprometerem o funcionamento do ascensor, é possível concluir que este opera em condições extremamente satisfatórias e seguras, obtendo êxito naquilo que se propõe.

## V. ANEXO I

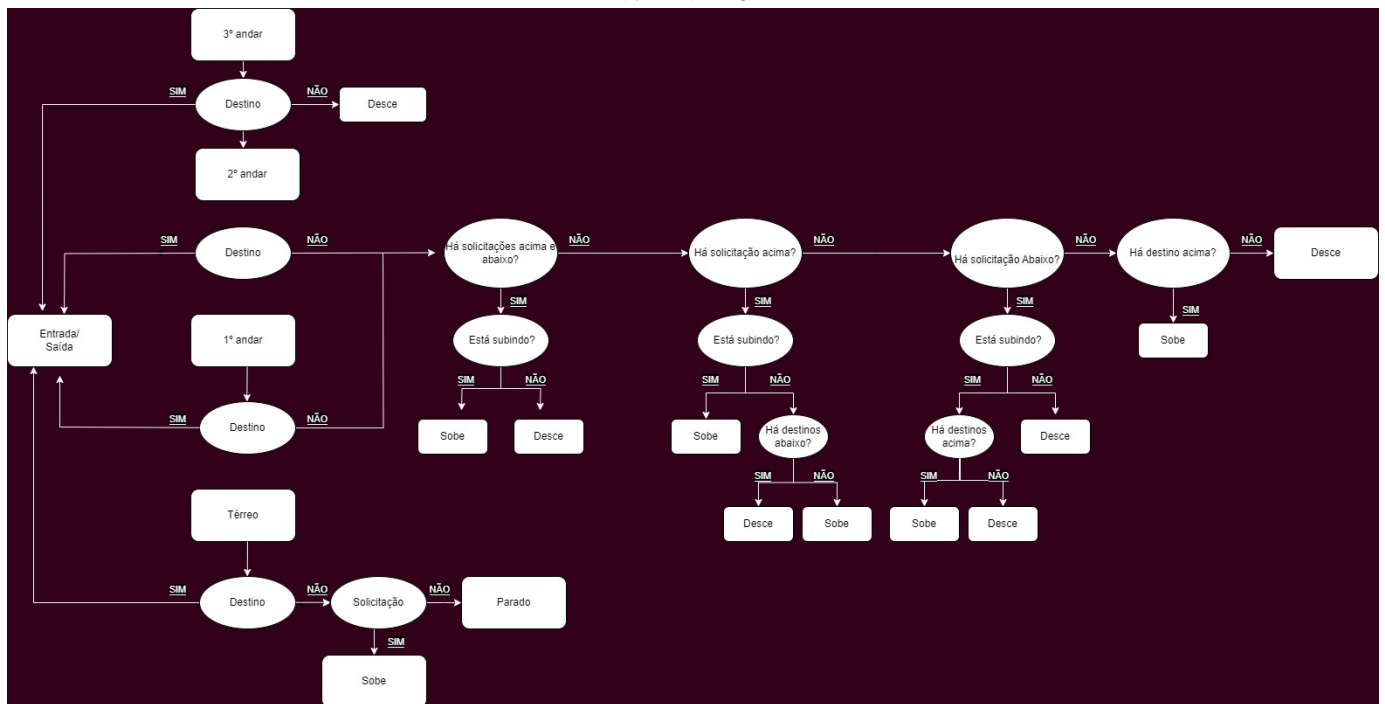


Figura 1: Fluxograma do Elevador