

CONTROLE DE UM ELEVADOR DE 4 ANDARES POR BLUETOOTH

1. OBJETIVOS

Permitir ao aluno integrar os tópicos estudados durante o curso num modelo que emula uma aplicação real. Durante o projeto, o aluno trabalhará em equipe para desenvolver o firmware do microcontrolador e atender os requisitos definidos no roteiro.

2. FUNDAMENTOS TEÓRICOS

Para uma melhor compreensão do projeto, vocês precisam pesquisar sobre os temas a seguir:

- Funcionamento dos elevadores. Optimização do percurso.
- Geração dos sinais de modulação por largura de pulso (PWM) para o controle da velocidade e sentido de rotação de motores de corrente contínua (CC) com ponte H.
- Funcionamento da ponte H TB6612FNB.
- Método óptico para medir a posição e a velocidade de rotação de um motor.
- Representação gráfica com matriz de LEDs com controlador MAX7219.
- Comunicação UART/Bluetooth pelo módulo HC06.

A bibliografia básica sugerida para estas temáticas está listada no final do texto.

3. DESCRIÇÃO DO PROCESSO E REQUISITOS DO SISTEMA DE CONTROLE

No projeto você desenvolverá um sistema de aquisição de dados e controle para uma maqueta (modelo) de um elevador que atende 4 andares. O movimento do elevador é controlado por um microcontrolador PIC16F1827 alimentado com 5V, que atua sobre um motor de corrente contínua. Na Figura 1 se exibe um digrama em blocos do sistema e na Figura 2 as imagens do modelo e detalhes das partes.

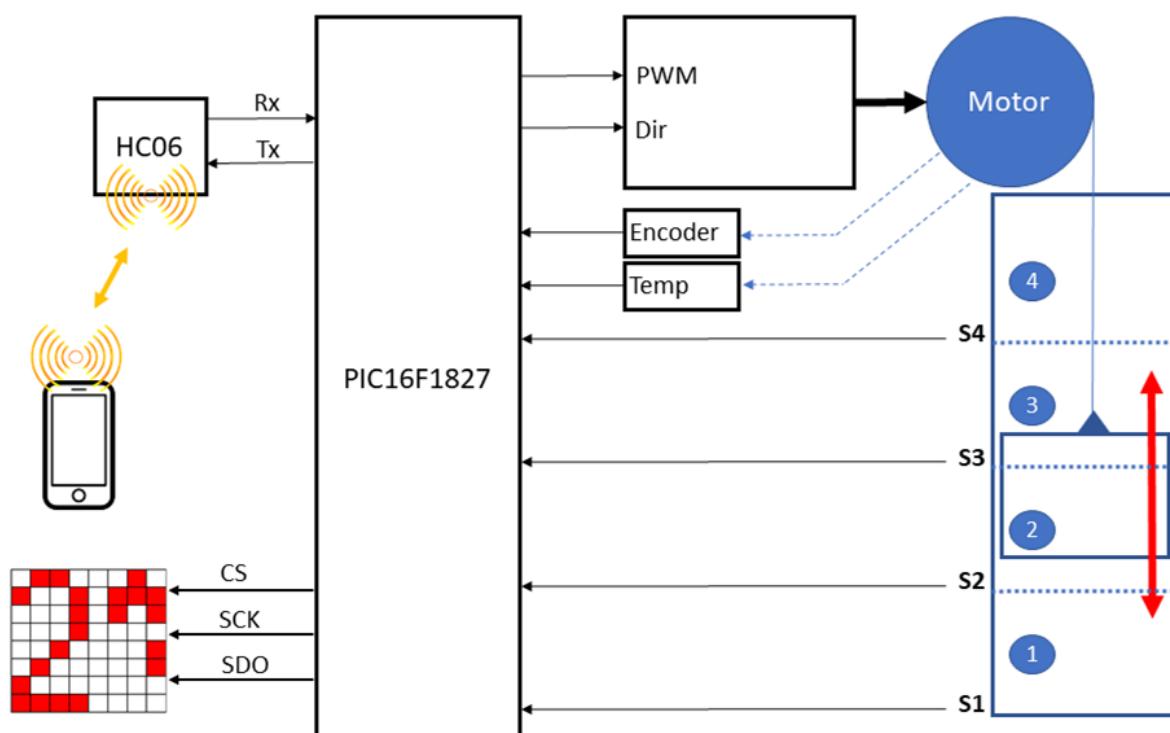


Figura 1 Diagrama em blocos.

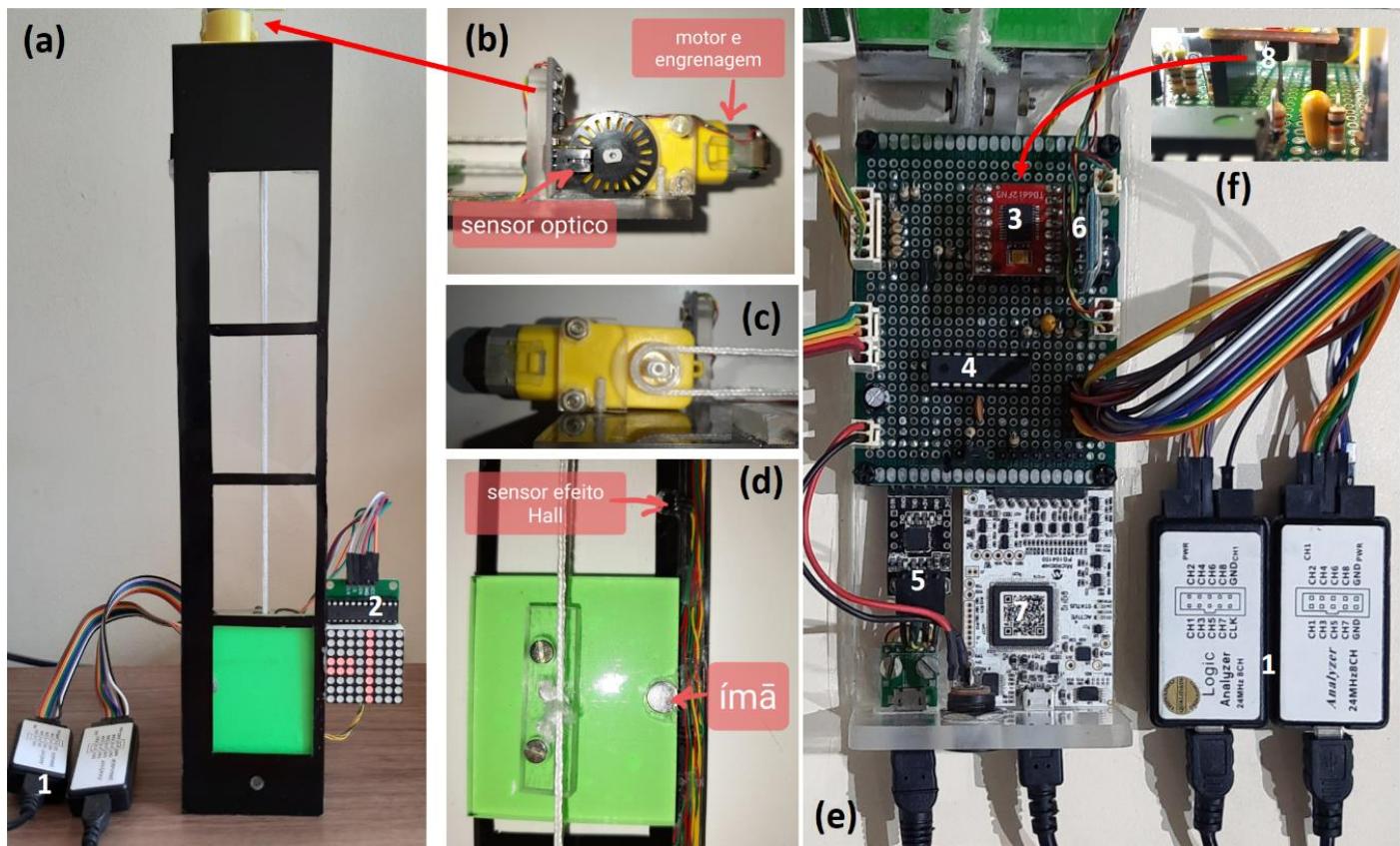


Figura 2 Imagens do modelo e circuitos de controle: (a) Estrutura em acrílico do modelo; (b) Motor e encoder óptico; (c) Motor e polia; (d) Caixa do elevador; (e) Placas eletrônicas; (f) Sensor de temperatura localizado embaixo da ponde H; (1) Analisadores lógicos; (2) Matriz de LEDs; (3) Ponte H; (4) PIC16F1827; (5) Conversor Serial/USB; (6) Conversor Serial/Bluetooth; (7) Programador (SNAP) e (8) Sensor de temperatura.

No esquema da Figura 3 se exibe o esquemático da placa de controle e na Figura 4 os componentes externos a ela.

O circuito que comanda o motor (driver de potência) utiliza uma ponte H (TB6612FNG) que muda seu ciclo útil e a polaridade da alimentação do motor. O circuito tem como entrada dois sinais. Um deles é a modulação por largura de pulso (**PWM**), que define o ciclo útil da tensão de alimentação do motor, atingindo 10 bits de resolução. O outro é um sinal digital, que define o sentido de movimento do motor (**Dir**). Quando **Dir** está em “1” o movimento do elevador é ascendente e quando está em “0” é descendente. Os dois trabalham na faixa 0-5V.

O motor movimenta a caixa do elevador entre os 4 andares e em cada andar existe um sensor magnético de efeito Hall A3144 (**S1...S4**). Um ímã fixo na caixa do elevador vai ativar o sensor de cada andar quando estiver próximo de ele. A saída de cada sensor é do tipo coletor aberto e fica em condução (saturação) quando a caixa atingir esse andar, e em corte quando se afasta de essa posição. Para garantir a parada do motor no momento certo, os sensores devem ser atendidos por interrupção. No caso dos sensores S1 e S2 podem ser atendidos por IOC, e S3 e S4 pela interrupção dos comparadores analógicos.

Para conseguir diferenciar facilmente entre os sensores S1 e S2 na interrupção, deve ser desmarcada a opção *Single ISR per interrupt* na janela Interrupt Module. Com isso o MCC gerará uma função de atendimento à IOC para cada pino.

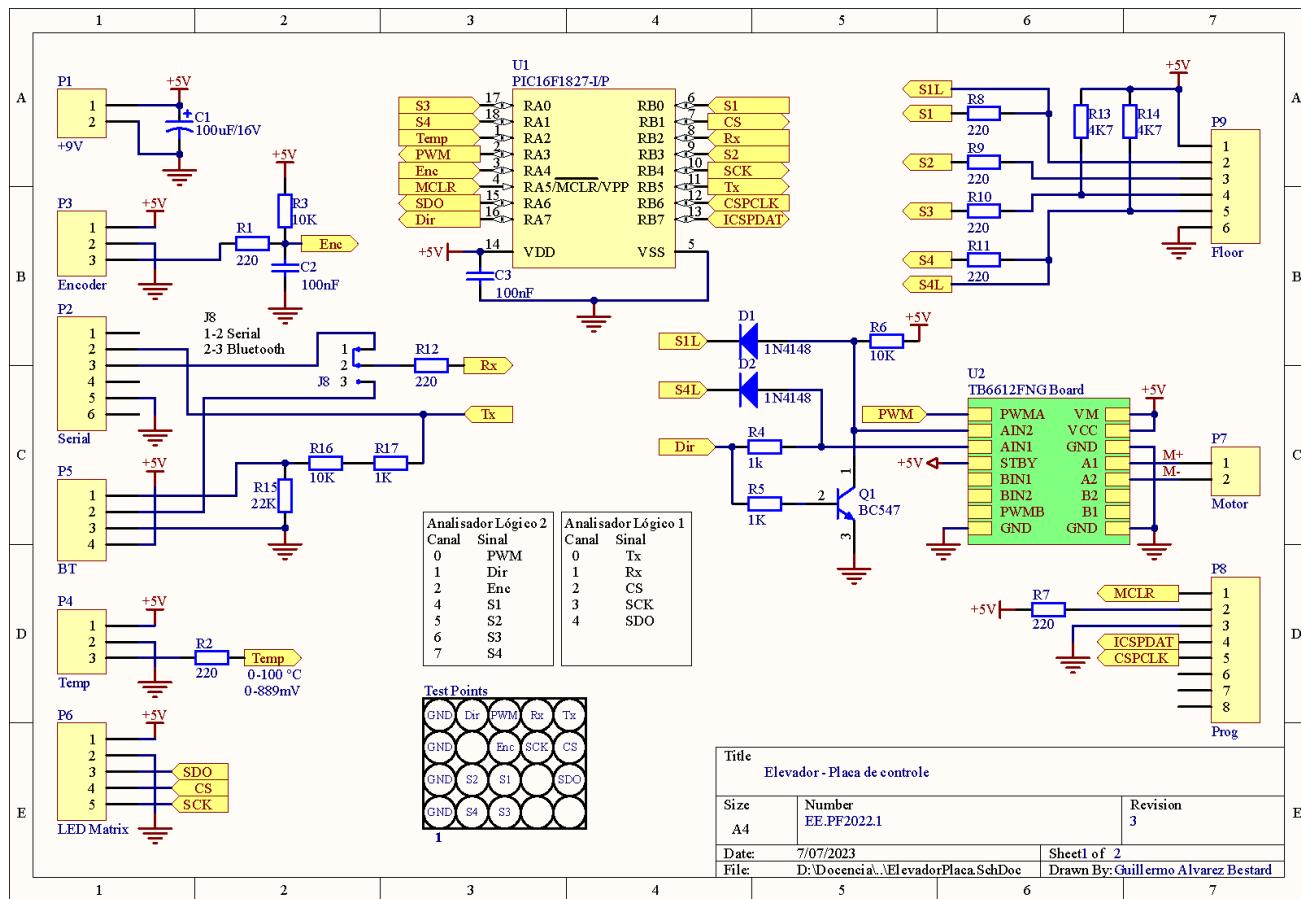


Figura 3 Esquemático da placa de controle.

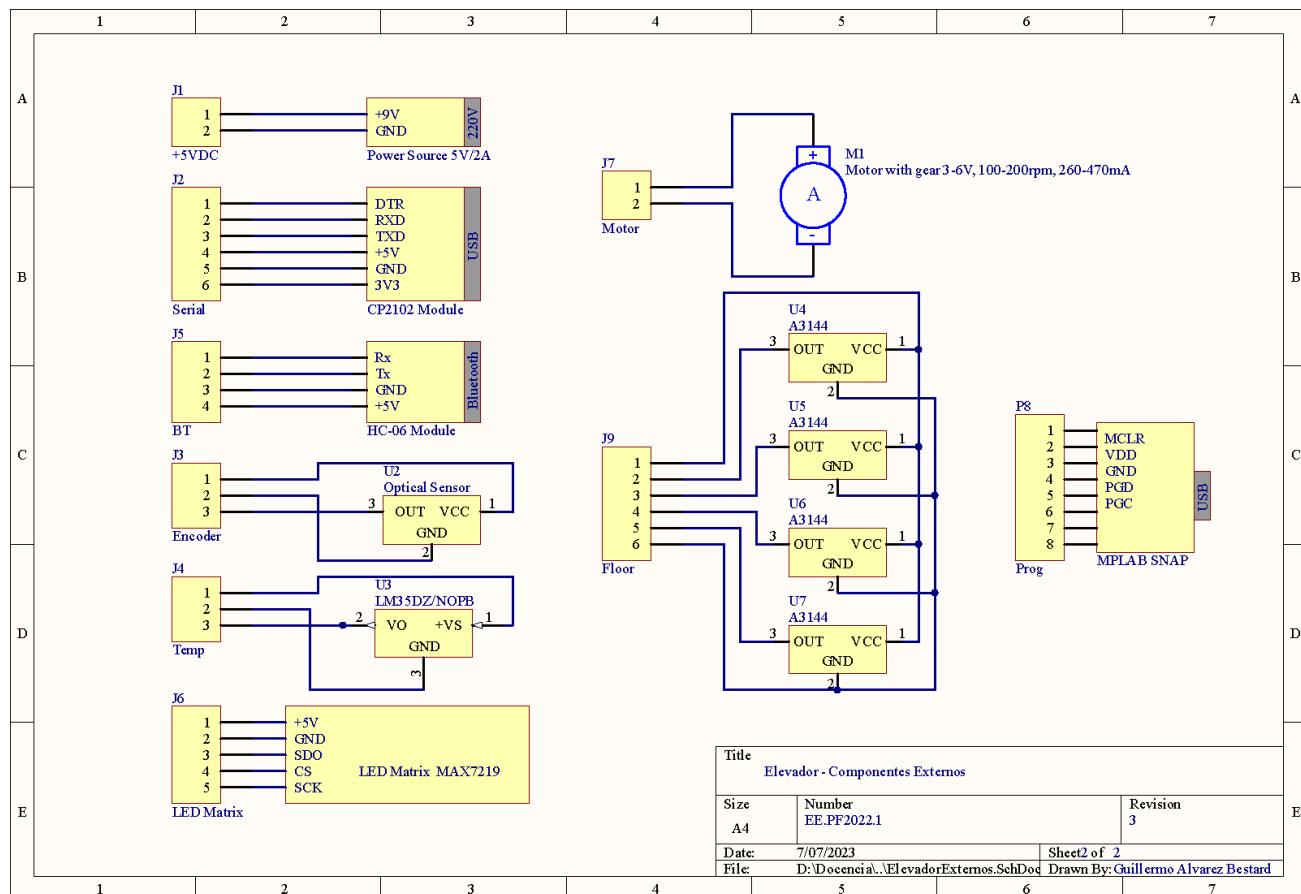


Figura 4 Esquemático dos componentes eletrônicos externos.

Independentemente do sinal **PWM**, os sensores do andar inferior (S1) e superior (S4) são utilizados num sistema de proteção que para o motor quando o elevador atinge um desses andares. Dependendo do estado do sinal **Dir**, eles colocam em condição de parada a ponte H.

A posição e a velocidade do elevador podem ser obtidas de um sensor óptico de fim de curso (usado como encoder) e um disco encoder acoplado no eixo da engrenagem do motor. Serão usadas as unidades milímetros (mm) e milímetros por segundo respectivamente (mm/s). A saída **Enc** fornece 215 pulsos no percurso total do elevador.

O percurso total do elevador é de 180 mm (240 mm dos 4 andares menos 60 mm da altura do elevador). A posição inicial pode ser inicializada usando o sensor de efeito Hall do andar inferior (S1).

A medição da velocidade vai ser usada apenas para indicação. Pode ser usado um ciclo útil fixo entre 40% e 50% na alimentação do motor. O deslocamento entre o andar 0 e 3, com 40% de ciclo útil, demora aproximadamente 9 segundos.

Uma espera de 2 segundos deve ser feita quando chegar no destino, para permitir a entrada ou saída do usuário. Depois de isso, o elevador deve voltar ao andar inferior (posição de repouso) se não tiver outras solicitações.

Para não danificar o motor e as engrenagens, uma espera de 500 ms deve ser feita antes de mudar a direção de movimento (se for uma parada para isso apenas), como pode acontecer se uma nova solicitação chegar durante o retorno para a posição de repouso.

Um sensor de temperatura LM35 será usado para obter a temperatura (**Temp**) da ponte H através do conversor ADC de 10 bits.

Numa matriz de LEDs, controlada por um CI MAX7219 através do protocolo SPI no modo zero a uma frequência de 1 MHz (para conseguir acompanhar no analisador lógico), deve ser exibido o último andar que foi atingido pelo elevador, o sentido de movimento e os andares solicitados. Nas primeiras 4 colunas será exibido o número do último andar atingido. Nas últimas 3 colunas, na parte superior, deve ser desenhada uma seta indicando o sentido de movimento e uma linha ou um rombo quando estiver parado. Na parte inferior serão ligados 4 pontos indicando os andares que estão no percurso e que ainda não foram atendidos. Um exemplo se exibe na Figura 5.

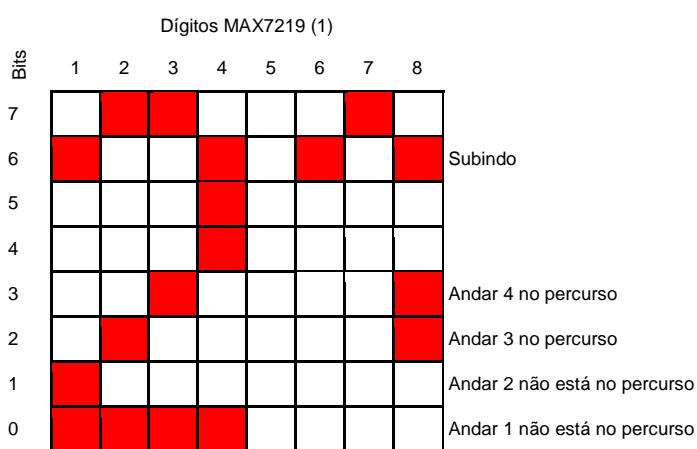


Figura 5 Sugestão de representação dos dados na matriz de LEDs. Exemplo para o elevador subindo e passando pelo andar 2 e com destino nos andares 3 e 4.

Um aplicativo móvel (Serial Bluetooth Terminal) no telefone celular ou o software HTerm no computador, podem se comunicar com o microcontrolador através de uma conexão Bluetooth, usando um conversor Bluetooth/Serial Assíncrono ligado ao microcontrolador no qual deve ser usado o UART.



O microcontrolador do elevador deve enviar, a cada 300 ms, as informações listadas na Tabela 1. Os dados devem ser enviados e recebidos no microcontrolador em formato ASCII, por comunicação serial assíncrona com um Baud Rate de 19200 BPS, 1 bit de início, 8 bits de dados, 1 bit de parada e sem paridade.

O aplicativo vai exibir o quadro de comunicação conforme enviado, sem nenhum tipo de processamento. Na Tabela 1 também se exibe o quadro de comunicação na transmissão e na recepção.

Tabela 1. Quadro de comunicação entre o microcontrolador e o aplicativo no telefone.

Variável	Símbolo	Faixa	Formato	Exemplo	Comentário
Origem	O	0...3	Inteiro 4 bits	1	Andar origem
Destino	D	0...3	Inteiro 4 bits	3	Andar destino
Andar	A	0...3	Inteiro 4 bits	2	Último andar atingido
Motor	M	0...2	Inteiro 2 bits	1	Estado do motor (0-parado, 3-movimiento ascendente e 2-movimiento descendente)
Posição	HHH	0...180	Inteiro 8 bits, decimal	035	Altura em mm do elevador respeito ao solo. No exemplo seria 35 mm
Velocidade	VV.V	0...51,1	ponto fixo, uma casa decimal (9 bits), decimal	20.3	Velocidade em mm/s do elevador. No exemplo seria 20,3 mm/s
Temperatura	TTT.T	0...99,9	ponto fixo, uma casa decimal	035.1	Temperatura em °C da ponte H. No exemplo seria 35,1 °C

Sentido da transmissão

Aplicativo -> Elevador

Elevador -> Aplicativo

Quadro de comunicação

\$OD<cr>

\$A,D,M,HHH,VV.V,TT.T<cr>

Exemplo

\$13<cr>

\$2,3,5,1,035,20.3,035.1<cr>

3.1. ALGORITMO DE ATENDIMENTO ÀS SOLICITAÇÕES

Deve ser implementado um algoritmo que atenda até 5 solicitações dos usuários. O sistema deve recolher ao usuário no andar de origem e deixar ele no andar de destino, mas para isso deve otimizar o percurso no máximo possível. Num mesmo sentido de movimento devem ser realizadas todas as paradas necessárias para atender as solicitações selecionadas, sempre respeitando o origem e destino de cada solicitação, sem inverter ele. O elevador deve permanecer no andar 0 (posição de repouso) até chegar uma solicitação e voltar para ele quando atender todas.

3.2. COMUNICAÇÃO BLUETOOTH

O computador ou o telefone celular devem ser conectados via Bluetooth à interface HC-06, mas pode ser usado apenas um deles por vez, após desconectado o outro. No computador pode ser usado o HTerm e no celular deve ser instalado o aplicativo Serial Bluetooth Terminal. Tanto o telefone quanto o computador, devem ser pareados com a placa HC-06 antes de estabelecer a conexão Serial. O pareamento precisa ser feito apenas uma vez em cada novo dispositivo.

Para o pareamento no celular e no computador, deve ser procurado o menu Bluetooth e, dentro da lista dos dispositivos disponíveis (sem parear), deve ser selecionado o dispositivo Elevador. No computador deve ser selecionada a opção de adicionar um dispositivo Bluetooth. Caso não encontrar o dispositivo, procure um dispositivo com endereço 0022:09:01A63D ou um dispositivo desconhecido e teste parear ele. Uma senha será solicitada e deve ser colocado 1234.



Uma vez concluído o pareamento, no Serial Bluetooth Terminal procuramos na opção *Devices* o dispositivo Elevador novamente. Em seguida, na tela do aplicativo, podem ser enviadas e recebidas as mensagens seriais desde e para o microcontrolador. No computador abrimos o HTerm e procuramos uma nova porta COM criada no pareamento do dispositivo (duas portas COM serão adicionadas, mas apenas uma pode ser usada para a comunicação). O HTerm pode ser usado como habitualmente fazemos.

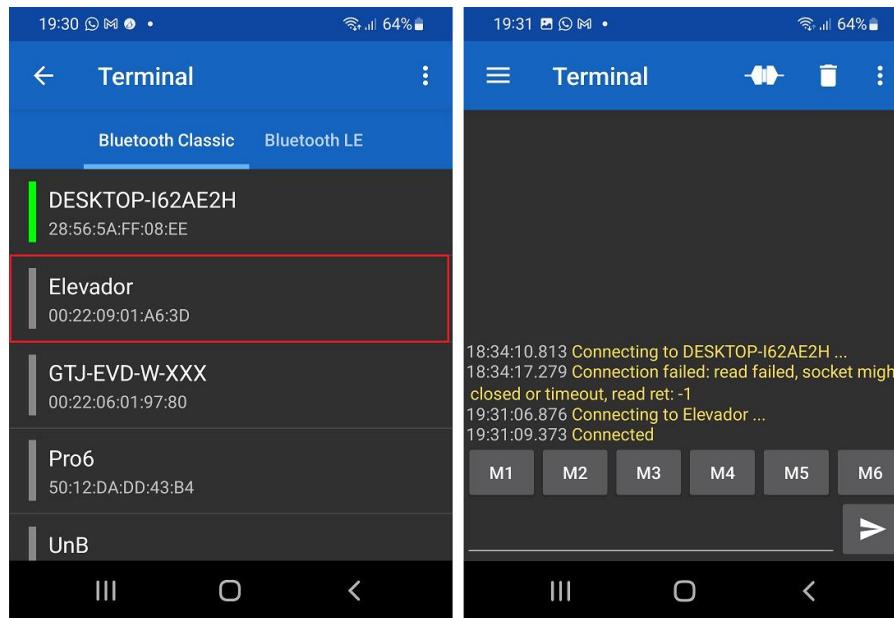


Figura 6 Telas do aplicativo no pareamento e conexão.

Antes do pareamento, os LEDs azul e vermelho do HC-06 permanecem piscando. Após do pareamento param de piscar. Para conectar outro telefone ou o computador, o anterior deve ser desconectado no aplicativo.

O aplicativo deve ser configurado para separar as linhas quando chegar um CR, conforme a imagem a seguir. Essa configuração também permitirá que se inclua CR no final.

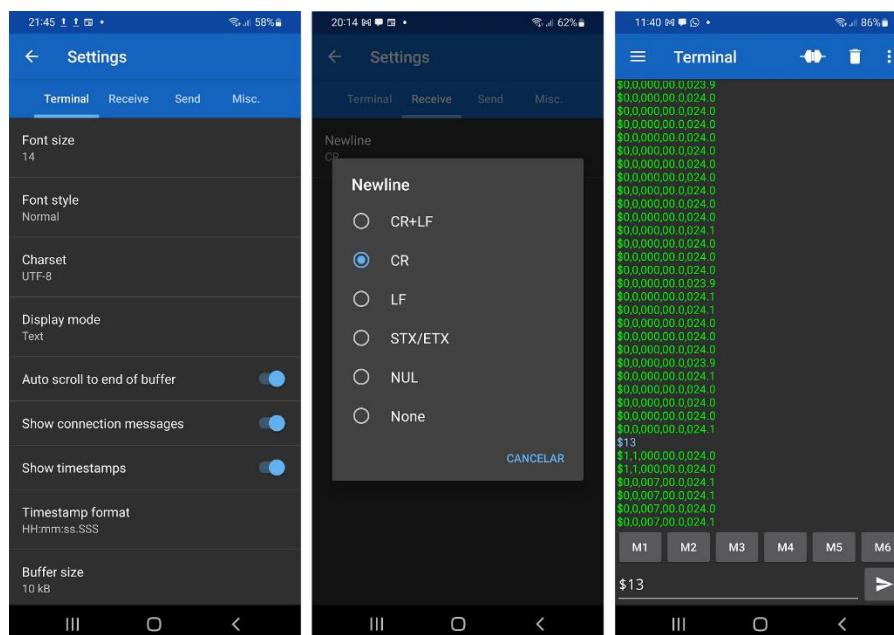


Figura 7 Telas do aplicativo na configuração e transmissão de dados.

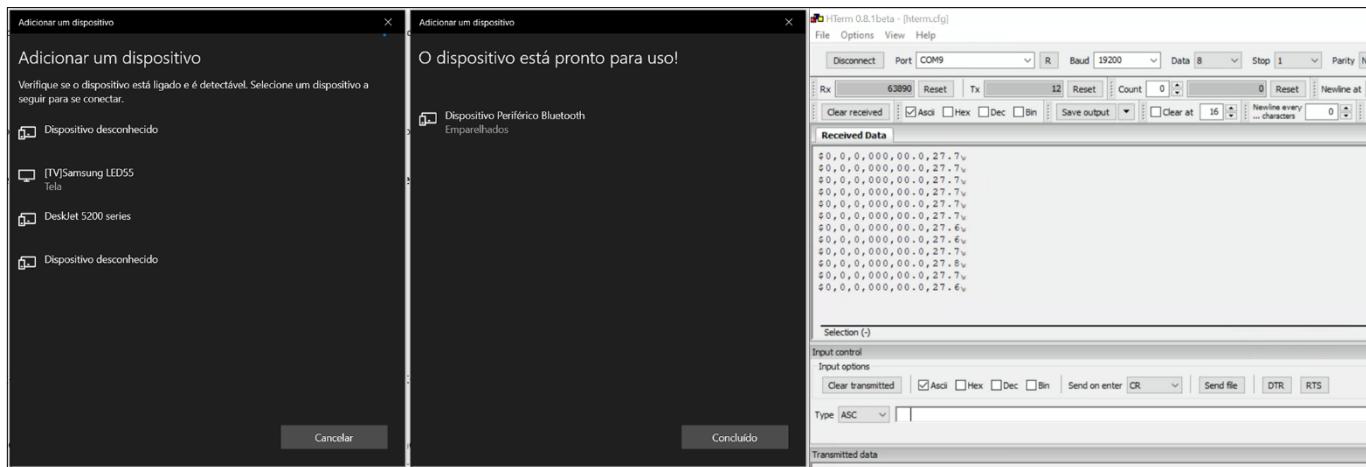


Figura 8 Telas do computador no pareamento e transmissão de dados.

4. PROCEDIMENTO DE TRABALHO

O projeto final pode ser desenvolvido em grupos de até 4 alunos (não serão aceitos grupos de 5). Cada estudante deverá desenvolver e testar uma parte do código. É importante fazer testes rigorosos no simulador para conseguir eliminar a maior quantidade dos erros antes da integração das partes.

Desenhe o fluxograma (ou os fluxogramas) necessários para atender as especificações descritas anteriormente (podem usar <https://www.diagrams.net/>). Com base nisso, desenvolva o firmware necessário no microcontrolador. Coloque comentários no código descrevendo o funcionamento de forma clara.

As variáveis globais e funções que são usadas por várias partes, devem ser definidas entre todos os integrantes da equipe no início do projeto com o intuito de garantir a integração final.

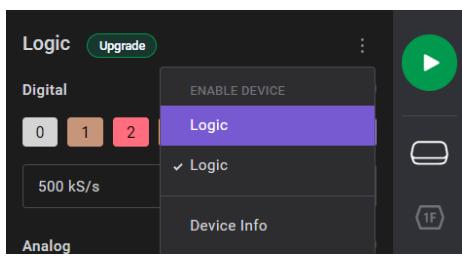
O projeto todo será testado na simulação observando cada detalhe. Apenas depois de verificar o correto funcionamento em simulação, o teste poderá ser realizado no Lab Remoto (a primeira semana) ou no Lab NEI2 (a partir da segunda semana), onde ficará disponível o modelo do elevador.

Elabore um relatório de até 5 páginas com as atividades desenvolvidas pela equipe e por cada participante. Especifique os estímulos utilizados e os resultados obtidos na simulação.

Programe o microcontrolador com o código simulado usando a interface MPLAB Snap In-circuit Debugger. Teste o funcionamento do sistema com o Snap no modo debugger e com os dois Analisadores Lógicos conectados ao sistema.

No software para configurar e obter os sinais digitais (Logic), que mede cada o Analisador Lógico, precisa selecionar um deles e depois carregar o profile correspondente. Para o analisador com número de série **A448875BC25A7C58**, conectado aos sinais de comunicação (Tx, Rx, CS, SCK e SDO) carregar o preset **EE_PF_Elevador_Com**. Para o analisador com número de série **A12C9BA34F81F038**, conectado aos sinais de movimento (PWM, Dir, Enc, S1, S2, S3 e S4)), carregar o preset **EE_PF_Elevador_Mov**. Na Figura 8 se exibe onde selecionar o analisador e mudar o preset.

Seleção do Analisador



Troca do preset

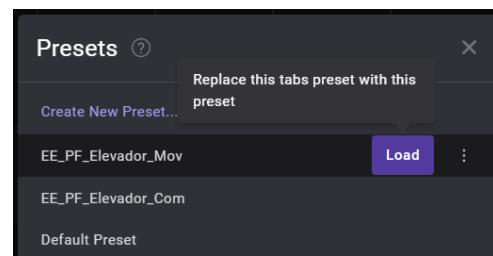


Figura 8 Seleção do Analisador Lógico e troca do Preset.

Um exemplo do funcionamento se exibe na Figura 9, onde se observa o deslocamento do elevador no analisador lógico.

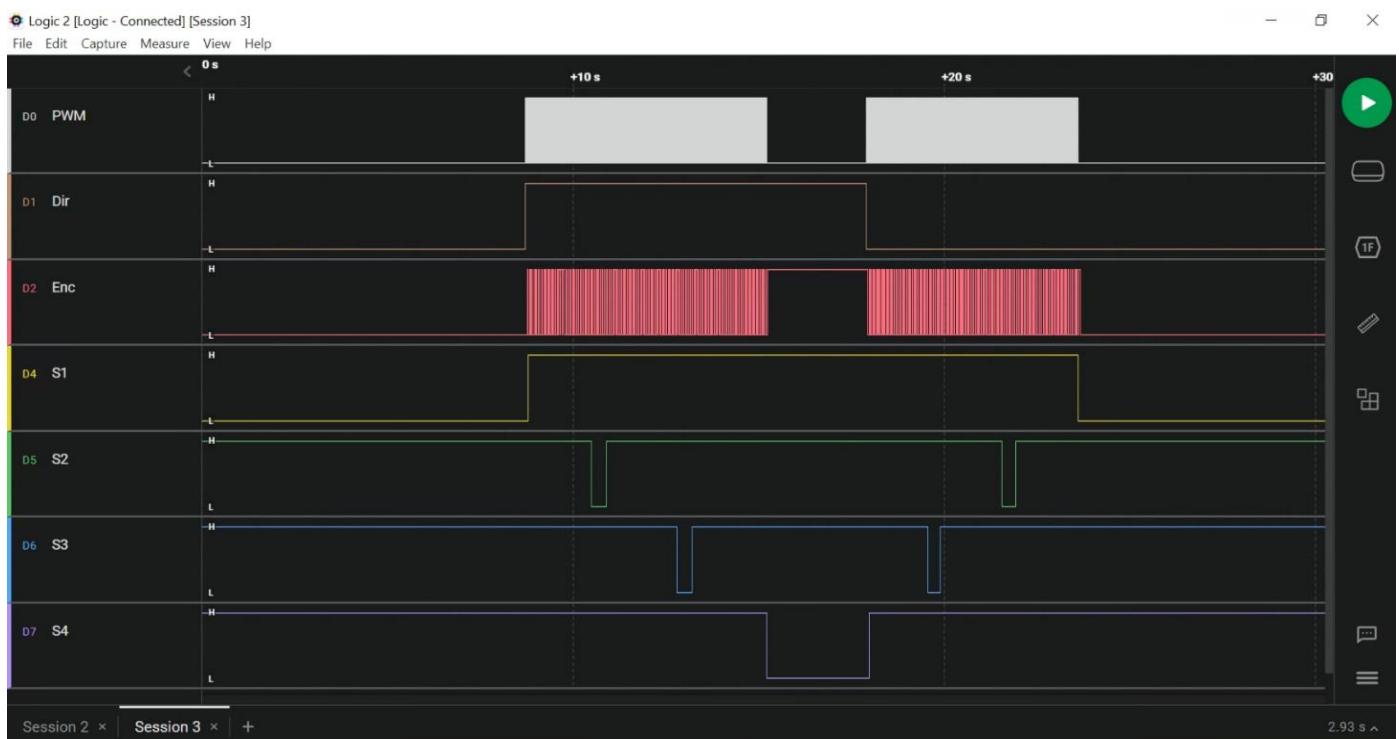


Figura 9 Sinais que exibem o deslocamento do elevador do andar 0 até o 3 e posteriormente retornando para o andar 0, com a ativação de cada sensor.

Quando estiver pronto faça um vídeo do funcionamento exibindo de forma clara. Explique o funcionamento de cada parte do código, a utilização do debugger e do Analisador Lógico nos testes, assim como o funcionamento com o software. Todos os integrantes devem participar na elaboração do vídeo. A duração não deve superar os 20 minutos.

Mantenham atualizado o repositório GitHub.

5. QUE DEVE SER ENTREGUE

A equipe deve criar um repositório privado no GitHub contendo:

- Readme com uma descrição básica do projeto e de como foi implementado.
- Um fluxograma que descreva o funcionamento do sistema. Pode ser desenvolvido no [Mermaid](#) e integrado no Readme como imagem ou com a diretiva ````mermaid` no início do diagrama.
- O firmware do projeto.
- Documentação gerada no Doxygen no formato HTML.



O acesso do professor deve ser garantido pelo e-mail guillermo@unb.br

Será avaliado apenas o que esteja no último commit do branch principal ou main até a data de entrega. Serão consideradas na avaliação, as contribuições de cada integrante no histórico de cada branch.

No Aprender deve ser entregue:

- O arquivo PDF com o relatório da simulação e os resultados dos testes.
- O link do vídeo postado (deve ficar disponível até 1 ano).
- O link do repositório GitHub deve ser entregue no Aprender.

6. AVALIAÇÃO.

A avaliação do projeto será através dos arquivos enviados, conteúdo do repositório no GitHub até a data de entrega e pela apresentação em sala de aulas (esse dia o modelo estará disponível). Será conferida a simulação, o funcionamento, e que este coincida com o código enviado. **Podem ser feitas perguntas a cada membro da equipe sobre as técnicas utilizadas no desenvolvimento, independentemente da parte que ele desenvolveu.**

As dúvidas serão esclarecidas em sala de aulas, pelo e-mail, WhatsApp, com os monitores ou com o professor.

7. DATAS IMPORTANTES

Data	Atividade
08/12/2023 24:59h	Data limite para o envio das equipes
19/12/2023 08:00h	Entregue da documentação do projeto final
20/12/2023 16:00	Avaliação do projeto (perguntas do professor e monitores)

8. LINKS PARA ESTUDAR.

- Como o elevador funciona? Saiba absolutamente tudo! <https://meuelevador.com/como-o-elevador-funciona/>
- Guia Definitivo sobre Elevadores, seus componentes e mecanismos. <https://www.atualizeelevadores.com.br/guia-definitivo-sobre-elevadores-seus-componentes-e-mecanismos/>
- O que é PWM <http://www.newtoncbraga.com.br/index.php/robotica/5169-mec071a>
- Ponte H TB6612FNG <https://toshiba.semicon-storage.com/eu/semiconductor/product/motor-driver-ics/brushed-dc-motor-driver-ics/detail.TB6612FNG.html>
- Motor DC <https://www.huinfinito.com.br/motores/1110-motor-com-reducao-3-6v-180-graus.html>
- Sensor óptico de fim de curso <https://www.huinfinito.com.br/modulos/1173-modulo-chave-optica-para-encoder.html>
- Disco do sensor óptico (<https://www.huinfinito.com.br/acessorios/1569-disco-encoder-20-furos.html>)
- Método óptico para medir a posição e velocidade de rotação de um motor. <http://www.newtoncbraga.com.br/index.php/artigos/54-dicas/6083-art762.html> <https://www.youtube.com/watch?v=T9JjSYzbPH4>
- MAX7219-MAX7221.pdf <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>



- Modulo-matriz-led-8x8
 - <https://www.huinfinito.com.br/home/1117-modulo-matriz-led-8x8-com-max7219.html>
 - <https://www.embarcados.com.br/modulo-matriz-de-leds-com-max7219/>
- ASCII
 - <https://pt.wikipedia.org/wiki/ASCII>
 - <https://www.embarcados.com.br/tabela-ascii/>
- Conversor Serial/Bluetooth
 - <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>
- Serial Bluetooth Terminal
 - https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal&hl=pt_BR&gl=US
- SNAP
 - <https://www.microchip.com/en-us/development-tool/PG164100>)
- Analisador lógico
 - <https://support.saleae.com/getting-started>)
- Doxygen
 - <https://www.doxygen.nl/files/doxygen-1.9.7-setup.exe>
 - <https://www.doxygen.nl/download.html>
 - <https://www.doxygen.nl/manual/index.html>
- Markdown Tutorial
 - https://github.com/luong-komorebi/Markdown-Tutorial/blob/master/README_pt-BR.md
- Mermaid
 - <https://www.mermaidchart.com/>
 - <http://mermaid.js.org/ecosystem/integrations-community.html>
 - <https://stackedit.io/app#>