

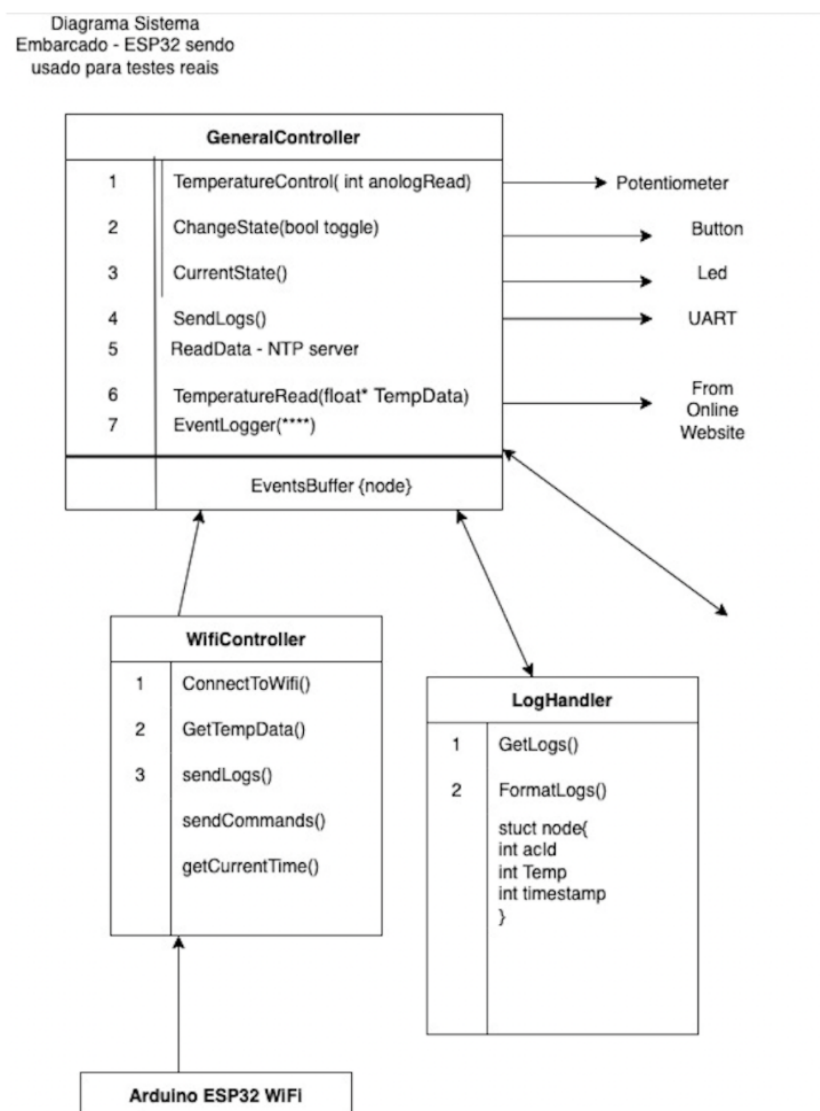
Controlador de ar-condicionado

Overview

Neste projeto foi utilizado o Espressif ESP32-DevKitC (como mostrado no site oficial da [tensorflowLite](https://www.tensorflow.org/lite)), um potenciômetro (o mesmo foi utilizado por apresentar mesmo função operacional que os dois botões, que o autor não possuía para implementar o projeto), um botão para ligar ou desligar o sistema do ar-condicionado , módulo WiFi embarcado na placa *ESP32* utilizada, a porta serial-UART para enviar os logs para o programa de *admin*, o código do software do pc foi desenvolvido no visualStudioCode em c++, o compilador escolhido foi o g++ e o padrão c++17 está sendo usado no projeto do computador, o sistema operacional é MacOS distribuição Monterey.

Sistemas Embarcados

Diagrama de classes



O trabalho avançou bem ao longo desta semana, para a entrega 3. No entanto, houveram algumas modificações efetuadas quanto ao trabalho por conta de problemas e bugs que demoram mais tempo que o julgado inicialmente para resolver.

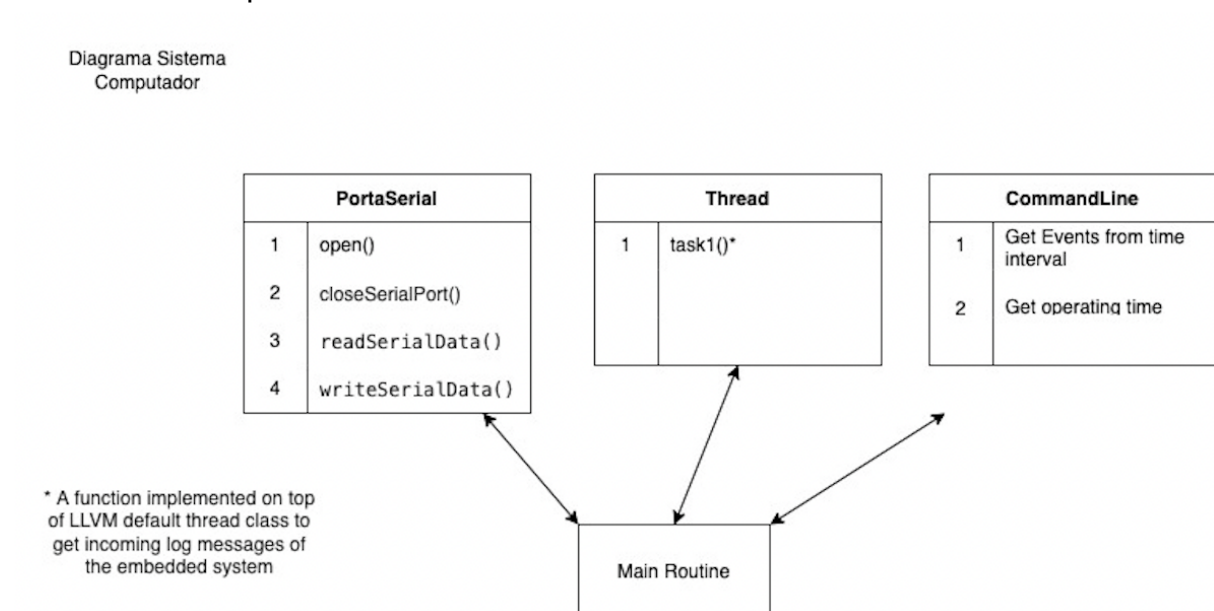
As sessões abaixo estão divididas nas classes principais

- Main Routine:
 - A rotina principal inicializa a GeneralController class
 - Também inicializa a classe de WiFi e módulo NTP pelo qual é possível pegar informações de horário em tempo real, como timestamp e o horário escolhido é GMT-3 (por conta de um servidor NTP apresentar dados de horário mais realistas e portanto fáceis de conferir no programa de Admin foi utilizado o servidor NTP ao invés da classe time/clock apresentada pelo professor)
- GeneralConttroller Class:
 - A classe de generalController está praticamente acabada.
 - A classe de LogHandler está implementando uma lista com um buffer circular que salva os dados de logs e envia os via serial quando o usuário desliga o ar condicionado ou quando a memória do aparelho está próxima de estar completa, os logs são enviados via serial(UART).
 - Não foi possível transformar o botão e nem potenciômetro em uma função de interrupção, devido a bugs e erros nas tentativas de implementação.
 - No entanto, tendo os dois no loop principal da rotina main, já é o suficiente para detectar qualquer variação de temperatura ou estado do ar condicionado inserida pelo usuário em questão de milissegundos.
 - A classe LogHandler será uma classe herdada pelo GeneralController, esse foi considerado o modelo mais eficiente de implementação dessa classe.
 - O protocolo de *handshake* implementado é bem simples e é explicado na parte do sistema do computador
- LearningPattern Class:
 - O código da rede neural utilizada como exemplo acabou gerando alguns erros no processo de conversão do tensorflow lite, os quais serão trabalhados ao longo desta última semana de desenvolvimento do trabalho

- O ML algo será adicionado através do tensorflow lite que é compatível com esta versão do esp32, o ML algo terá sua própria classe e será usado no loop principal.
- Pretende-se usar o exemplo deste link para executar a previsão do tempo da região do usuário e tomar decisões verificando a temperatura atual da CA. As decisões serão aumentar/diminuir a temperatura ou desligar o AC.
- ESP32 Arduino WiFi Class:
 - A classe WiFi está iniciando um cliente web para consultar dados de um servidor NTP adquirindo o horário atual em timestamp UNIX
 - O webApp para consultar dados de temperatura será:
<https://thecustomizewindows.com/2019/06/esp32-arduino-fetching-current-weather-data-no-json-parsing/>
 - A lógica para aquisição dos dados de temperatura ainda tem que ser implementada

Ainda faltam a lógica de aquisição dos dados de temperatura e a implementação da classe Learning Pattern a serem desenvolvidas no sistema embarcado.

Sistema do Computador



O sistema do computador utiliza basicamente uma interface UART com a parte serial do computador na qual o ESP32 esta conectado para conseguir escrever informações pedindo logs da aplicação e também receber estes logs após a solicitação ser efetuada, em uma *Thread* diferente para não ser perdido

informações, dado que a serial do *ESP32* é halfDuplex a thread contém toda a parte de escrita e leitura das informações enviadas ao sistema Embarcado.

Existe um protocolo simples para um *handShake* entre a aplicação administradora e o sistema embarcado, basicamente é enviado pelo sistema embarcado o tamanho de informação em bytes que será enviado via serial e o sistema administrador após receber esses dados envia uma resposta para o sistema embarcado. Após a resposta o sistema administrador espera 3 segundos para ver os dados no buffer serial do computador, mais que o suficiente para o envio de dados visto que a baud rate da porta serial é de 9600 bits/segundo

Após a implementação de *handshake* o sistema administrador conseguiu efetuar a leitura de mais de 90 eventos de mudança de temperatura e convertê-los de volta na struct utilizada no sistema embarcado sem problemas. Esse foi um dos maiores desafios na implementação do trabalho até agora.

Ao invés de uma interface lógica, o sistema de admin é todo desenvolvido por command line, sendo possível buscar os dados nos quais a temperatura do ar condicionado foi modificada, com um ID de identificação do aparelho e também com a timestamp do momento que o evento ocorreu.

Também é possível obter o intervalo de tempo no qual o sistema está ligado pelo painel de administrador.

A lógica central é aplicada na função main que inicializa o código.

Sistema Celular

Será implementado um cliente HTTP para executar queries para o servidor do sistemaEmbarcado, possibilitando desligar o mesmo e alterar a temperatura através de *POST* requests e *GET* requests para adquirir os dados de logs.

Como desenvolver para iOS com C++ no visualStudioCode está gerando diversos *bugs* e não foi possível nem executar um simples código de *HelloWorld*, foi decidido utilizar ReactNative para agilizar o desenvolvimento do projeto final da disciplina.

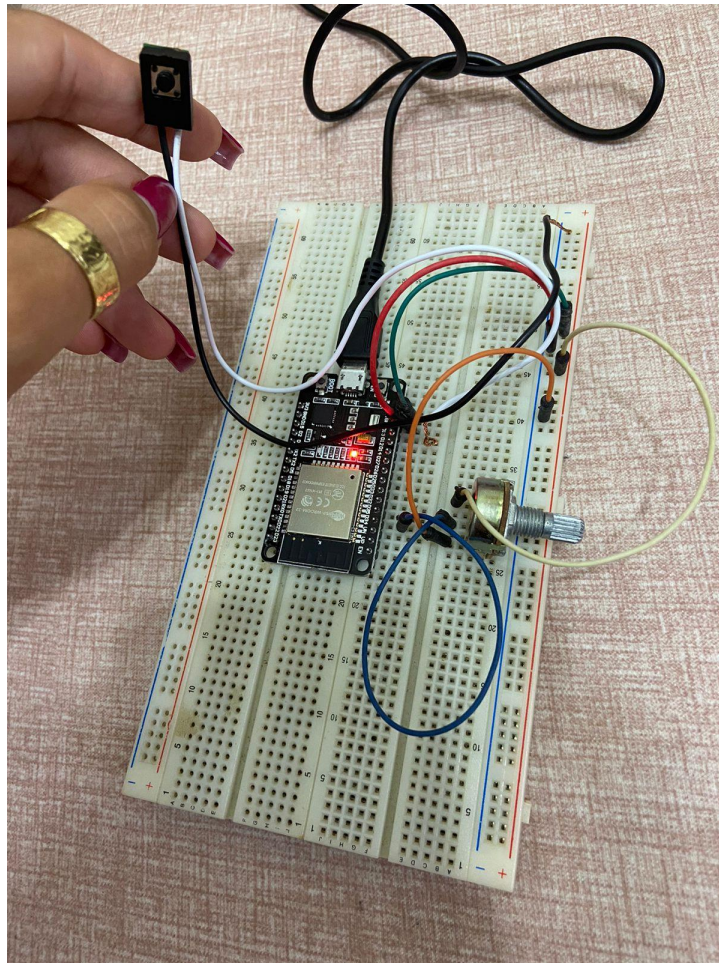


Foto do sistema de testes no estado atual deste relatório

NTP(Network Time Protocol)

NTP significa Network Time Protocol e é um protocolo de rede para sincronização de relógio entre sistemas de computador. Em outras palavras, ele é usado para sincronizar os horários do relógio do computador em uma rede.

