

Controlador de ar-condicionado

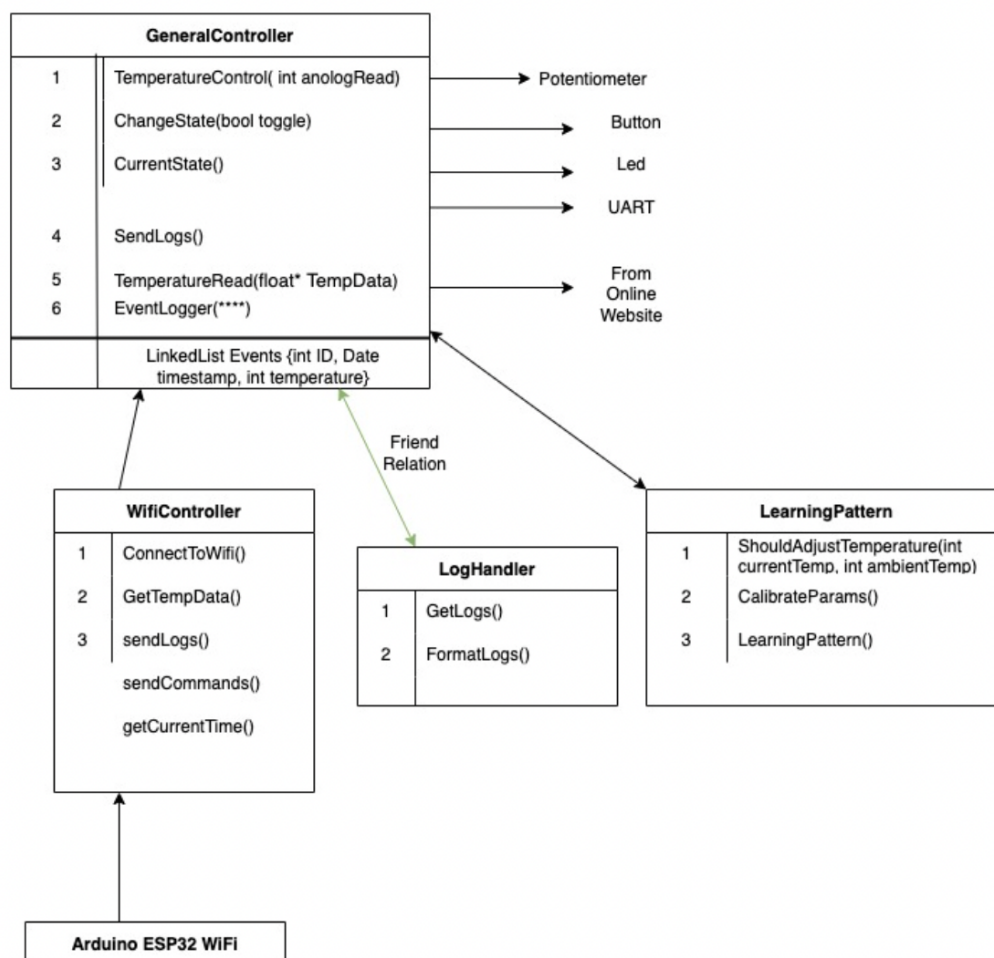
Overview

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

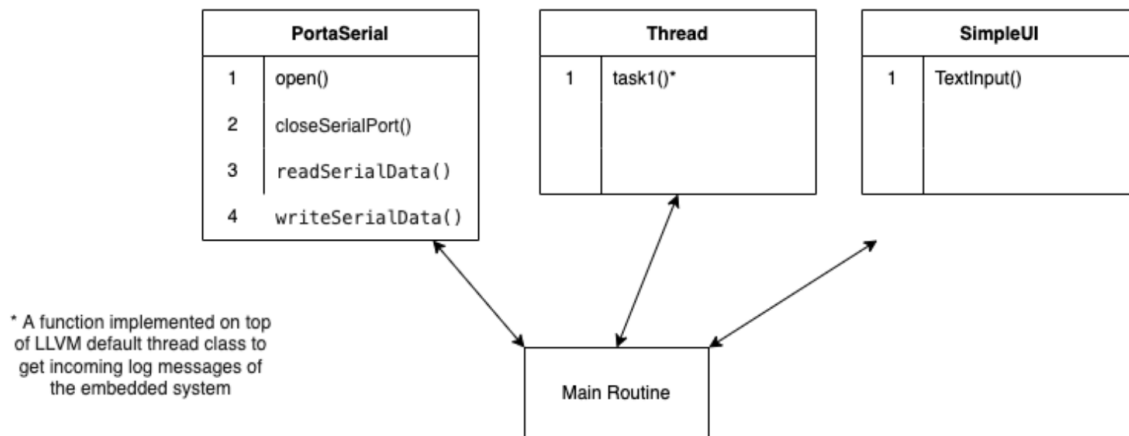
Diagrama Sistema
Embarcado - ESP32 sendo
usado para testes reais



Os estágios para implementar este trabalho completamente já foram avaliados separadamente e considerados viáveis (com alguns testes isolados), ao longo da próxima semana o autor estará implementando os itens remanescentes do planejamento do projeto que são descritos de forma sucinta na lista abaixo:

- GeneralConttroller Class:
 - A classe do generalConttroller está em progresso, haverá uma classe LogHandler que vai ser friend do generalController.
 - O LogHandlerClass vai implementar a linkedlist e estará salvando os logs que serão enviados via serial(UART) mediante solicitação do administrador no software do computador.
 - Adicionalmente estes logs estarão disponíveis para consulta através de um webServer inserido no módulo ESP32.
 - O botão será uma função de interrupção (portanto, mudará o estado quando o usuário solicitar) em uma interrupção.
 - O potenciômetro de temperatura também será inserido em uma interrupção.
- LearningPattern Class:
 - 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 será usada para inicializar um servidor web e um cliente para consultar dados da temperatura real e enviar informações para o aplicativo de celular.
 - O webApp para consultar dados temporários será:
<https://thecustomizewindows.com/2019/06/esp32-arduino-fetching-current-weather-data-no-json-parsing/>

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 sistemaEmbarcado.

A UI será simples e basicamente tem o propósito de facilitar o envio de comandos de intervalos de datas , convertidos em timestamp para serem enviados ao ESP32, para o sistema embarcado.

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 esta gerando diversos *bugs* e não foi possível nem executar um simples código de *HelloWorld*, o autor decidiu utilizar ReactNative para agilizar o desenvolvimento do projeto final da disciplina. Caso o professor esteja de acordo.

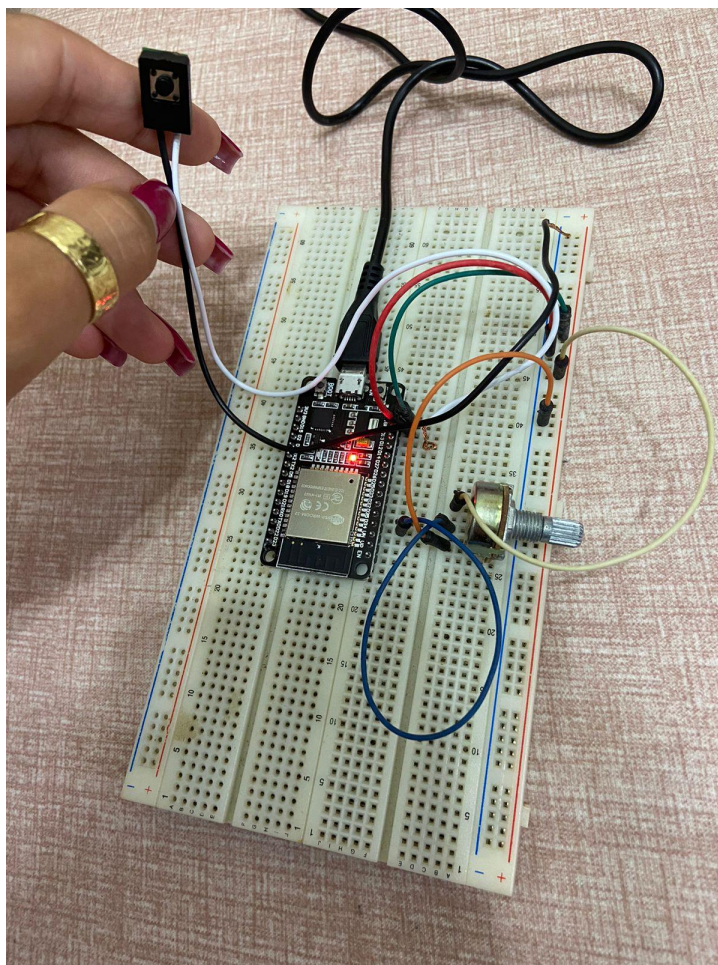


Foto do sistema de testes no estado atual deste relatório