



PROJETO PRÁTICO DE ESTRUTURAS DE DADOS

Sistema de Análise de Distância do Rádio LoRa (Armazenamento em ABB)
Boanerges Junior, Eduardo Cezar, Leonardo Junio

LAVRAS – MG
2019

SUMÁRIO

- Introdução**
- Componentes**
- Descrição das Estruturas**
- Código**
- Conclusões e retratações**

INTRODUÇÃO

O projeto desenvolvido é um protótipo de geolocalização junto com a verificação da qualidade da rede LoRa em decorrência de envios e chegadas de pacotes de dados.

Para isso, estamos utilizando o dispositivo Heltec ESP32 LoRa como meio de comunicação, e a estrutura de dados Árvore binária de busca (ABB) para armazenar e ordenar os dados coletados.

A inserção é feita toda chegada de pacote, já a ordenação é feita pelo método “Ordem” da classe ABB, que imprime nossos dados já ordenados.

Todo o projeto foi executado seguindo às necessidades e aos requisitos propostos pelos professores.

COMPONENTES

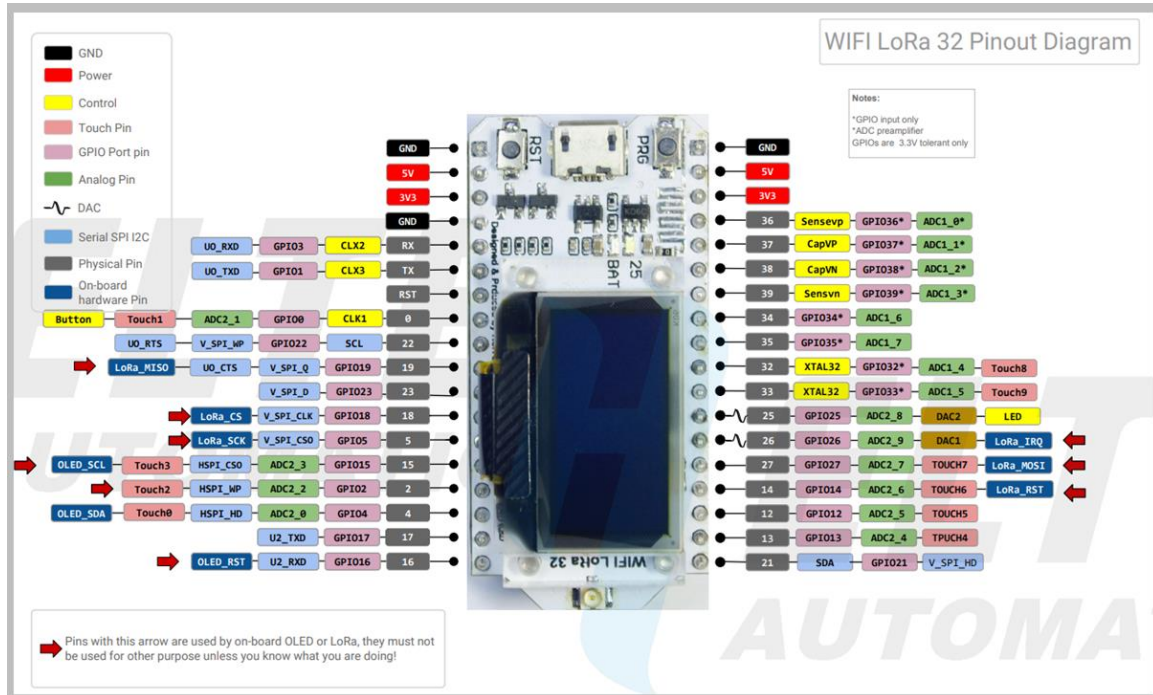
Seguem abaixo os componentes que fizeram parte do processo de execução e implementação do projeto:

Heltec ESP32 LoRa



Dispositivo utilizado para fazer a comunicação em rede para envio e recebimento de dados, feitos através da LoRa WAN, também é responsável pelo tratamento dos dados.

Pinagem e esquemático do ESP32 LoRa



DESCRIÇÃO DAS ESTRUTURAS

Main_reciever: o programa é utilizado em um dos LoRas configurado como 'recedor' dos dados.

Iniciamos com os include e bibliotecas necessárias para o funcionamento do código e do dispositivo, em seguida há a definição dos pinos padrões.

Em SETUP, nós configuramos o display OLED do dispositivo para resetar, escrever, informar se o dispositivo falhou ou se encontrou.

Em LOOP é onde acontece o recebimento dos pacotes, a variável packetSize recebe o tamanho do pacote e, caso tenha a confirmação que recebeu um pacote, chamamos o método cbk, onde é feito a decodificação e separação do pacote.

O método cbk é responsável por recuperar o conteúdo do pacote, porém tudo que está no pacote são strings, por isso criamos o método convert, que recebe a string que indica a qualidade do pacote e converte para int utilizando a tabela ASCII como referencia. Logo em seguida é chamada através de um objeto do tipo ABB o insere, passando por referencia o valor dB (qualidade do pacote).

Para finalizar o método loraData apenas imprime nossas informações recebidas no display OLED do dispositivo.

Main_sender:

Tem o papel de concatenar informações em um pacote e o transmitir via rede LoRa.

ABB: A estrutura utilizada foi uma árvore binária de busca, temos as classes padrões 'noh' e 'ABB' e os respectivos construtores, onde o construtor do noh recebe o dB provindo do dispositivo, há a inserção no método 'insere()' e para debug e ordenação dos dados utilizado o método 'Ordem()' e seus auxiliar 'OrdemAux()'

O código da estrutura foi feito de maneira a suportar remoção de dados, tal como seu rearranjo após remover algo (código comentado, pois não há necessidade de remoção até o momento).

Códigos

Os códigos desenvolvidos estão em anexo no documento com extensões .ino ; .h ; .cpp e PDF.

Conclusões e retratações

Conforme desenvolvíamos o projeto encontramos dificuldades em utilizar a estrutura de dados HEAP proposta inicialmente, em arquivo .cpp não conseguimos garantir a remoção de maneira adequada, portanto decidimos migrar para ABB, em seguida aprovado pelos professores.

Em relação ao projeto, até o momento temos a capacidade de identificar a qualidade de envio e recebimento de um pacote através da rede LoRa WAN, possível identificar quais os melhores locais para instalar o dispositivo para uma possível transferência de dados já que o dispositivo é amplamente utilizado para serviços IOT porque possui rede própria.