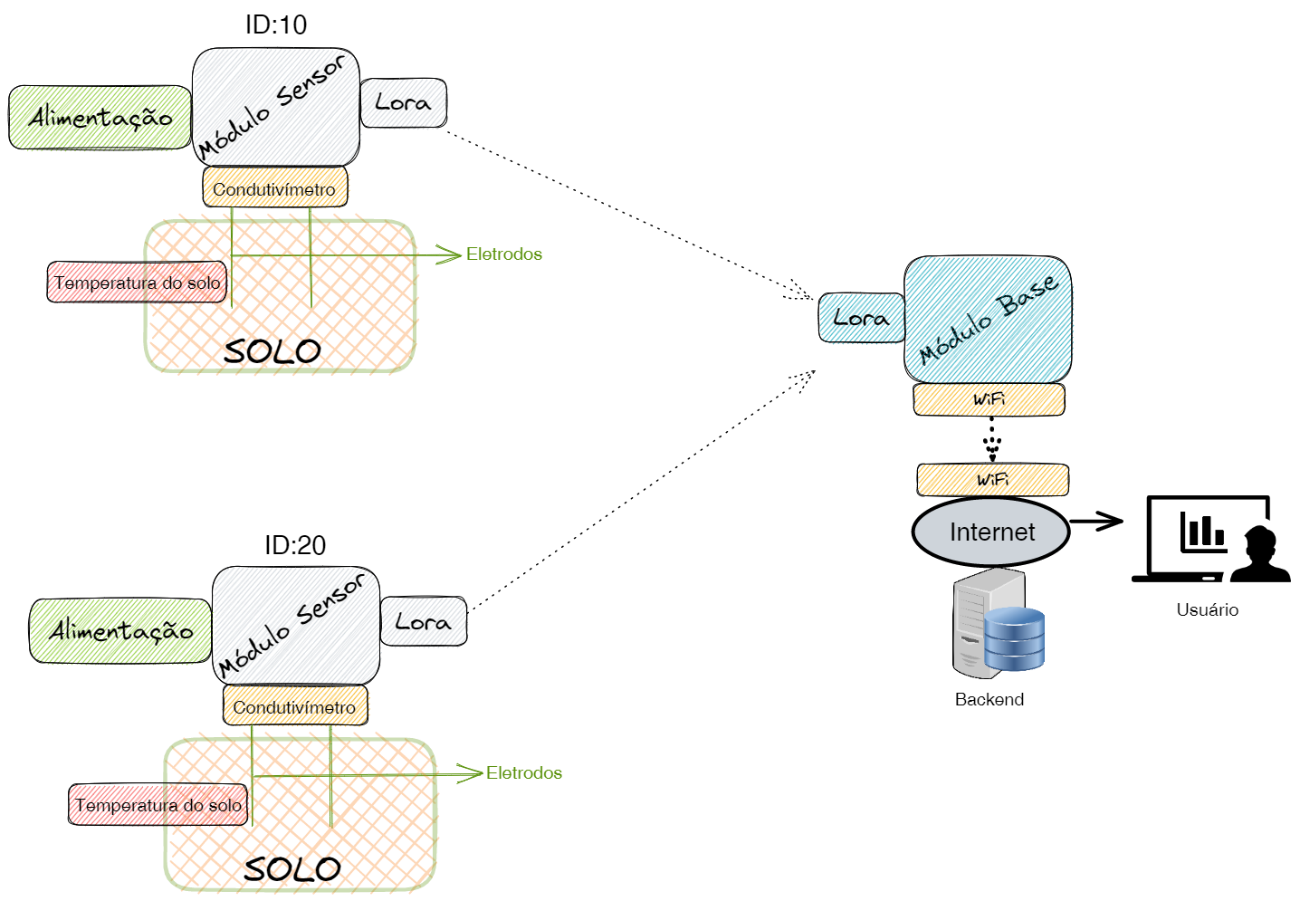
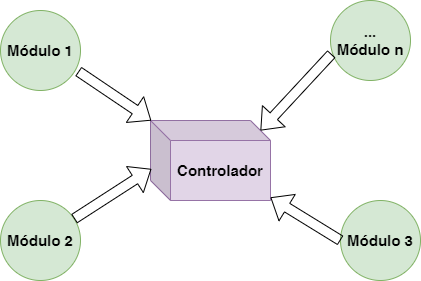
* **Motivos que levaram a substituição de placas baseadas em Arduino (UNO, MEGA, NodeMCU... para Lora).**
  + Maior alcance de comunicação: O Heltec LoRa Wifi utiliza a tecnologia LoRa para comunicação sem fio, que possui um alcance muito maior do que o WiFi do NodeMCU (ESP8266). Isso significa que o Heltec pode ser usado em projetos que precisam se comunicar a longas distâncias, como por exemplo em projetos de monitoramento remoto ou de IoT para áreas rurais.
  + Maior eficiência energética: O Heltec LoRa Wifi também é conhecido por ter um consumo de energia muito mais eficiente do que o NodeMCU (ESP8266). Isso significa que é possível usar o Heltec em projetos que precisam funcionar por longos períodos de tempo sem precisar trocar as baterias ou conectá-lo a uma fonte de energia externa.
  + Melhor suporte a LoRa: Como o próprio nome sugere, o Heltec LoRa Wifi é projetado especificamente para suportar a tecnologia LoRa. Isso significa que ele possui recursos e funcionalidades que são específicas para a tecnologia LoRa, tornando-o uma escolha melhor para projetos que se baseiam nessa tecnologia.
  + Recursos adicionais: Além de suportar LoRa, o Heltec LoRa Wifi também possui recursos adicionais, como um display OLED integrado, um módulo WiFi e uma antena LoRa integrada. Esses recursos adicionais podem ser muito úteis para certos tipos de projetos, permitindo que você economize tempo e dinheiro ao não precisar comprar e integrar esses componentes separadamente.
  + Facilidade de uso: Por fim, muitas pessoas acham que o Heltec LoRa Wifi é mais fácil de usar do que o NodeMCU (ESP8266). Isso pode ser devido à sua interface de programação amigável ou ao fato de que ele já possui muitos recursos integrados, tornando-o uma escolha mais simples e direta para projetos que precisam de comunicação sem fio de longo alcance.

Arquitetura desenvolvida:



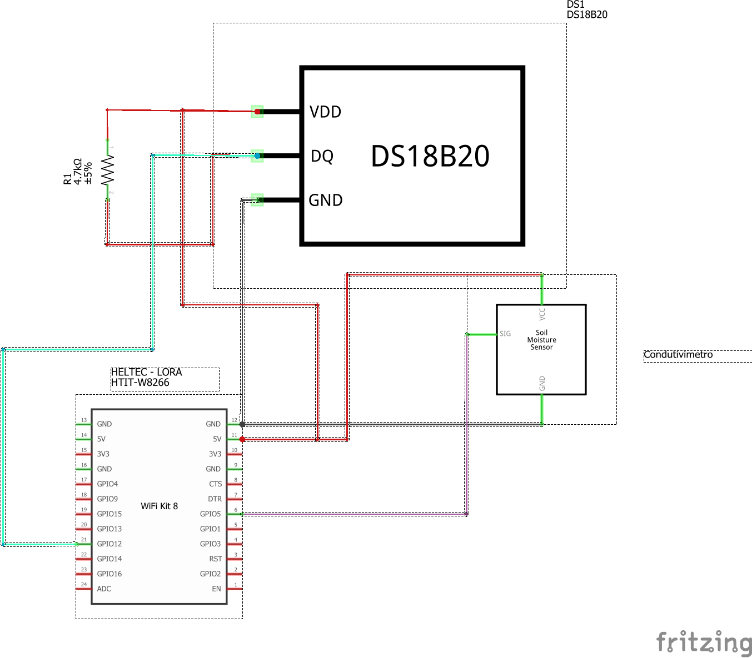
O sistema foi planejado e construído utilizando tecnologias de última geração e eficiência energética (Lora Wifi). Além disso, o processo de ligação entre os componentes é muito importante para o funcionamento do sistema. Para construção do mesmo, foi escolhida uma topologia similar a Estrela. um sistema com topologia em estrela, controlador centralizado e módulos descentralizados oferece maior confiabilidade, facilidade de manutenção, flexibilidade, desempenho e segurança em relação a outras topologias de rede. Essas vantagens tornam esse tipo de sistema uma escolha ideal para muitas aplicações, como sistemas de automação industrial, monitoramento remoto e IoT.



**Figura 1 - Topologia utilizada no sistema (Estrela)**

Os sistemas com topologia em estrela, controlador centralizado e módulos descentralizados possuem diversas vantagens em relação a outras topologias de rede. Abaixo, destacamos algumas das principais vantagens desse tipo de sistema:

* Maior confiabilidade: Com um controlador centralizado, é possível monitorar e controlar todos os módulos do sistema a partir de um único ponto. Isso significa que, em caso de falha em um dos módulos, é possível isolar e resolver o problema rapidamente, sem afetar o restante do sistema.
* Facilidade de manutenção: A topologia em estrela permite que cada módulo seja facilmente conectado e desconectado do controlador central. Isso torna a manutenção e a substituição de módulos muito mais simples e rápidas, reduzindo o tempo de inatividade do sistema.
* Maior flexibilidade: Com a descentralização dos módulos, é possível adicionar ou remover módulos conforme necessário, sem afetar o restante do sistema. Isso permite que o sistema seja facilmente expandido ou adaptado a novas necessidades.
* Melhor desempenho: Com um controlador centralizado, é possível otimizar o desempenho do sistema e maximizar a eficiência de cada módulo. Além disso, a descentralização dos módulos pode reduzir a carga de processamento no controlador central, melhorando ainda mais o desempenho geral do sistema.
* Melhor segurança: A topologia em estrela com controlador centralizado permite um maior controle de acesso aos módulos do sistema. Isso pode ajudar a prevenir acessos não autorizados e garantir a segurança das informações e dos dados armazenados no sistema.



Códigos

# **Controlador**:

#include "heltec.h"

#include <WiFi.h>

#include "ThingSpeak.h"

#define BAND 868E6 //you can set band here directly,e.g. 868E6,915E6

const char**\*** ssid **=** "IFMS-NA-CP"**;**

const char**\*** password **=** "adm@IFmscpna!#"**;**

WiFiClient client**;**

unsigned long myChannelNumber **=** 1548634**;**

const char **\*** myWriteAPIKey **=** "9IWG1KSKXYAPLZG9"**;**

void setup**()** **{**

//WIFI Kit series V1 not support Vext control

Heltec**.**begin**(true** /\*DisplayEnable Enable\*/**,** **true** /\*Heltec.LoRa Disable\*/**,** **true** /\*Serial Enable\*/**,** **true** /\*PABOOST Enable\*/**,** BAND /\*long BAND\*/**);**

WiFi**.**mode**(**WIFI\_STA**);**

**if** **(**WiFi**.**status**()** **!=** WL\_CONNECTED**)** **{**

Serial**.**print**(**"Attempting to connect"**);**

**while** **(**WiFi**.**status**()** **!=** WL\_CONNECTED**)** **{**

WiFi**.**begin**(**ssid**,** password**);**

delay**(**5000**);**

**}**

Serial**.**println**(**"\nConnected."**);**

Serial**.**print**(**"ESP Board MAC Address: "**);**

Serial**.**println**(**WiFi**.**macAddress**());**

**}**

Heltec**.**display**->**init**();**

Heltec**.**display**->**flipScreenVertically**();**

Heltec**.**display**->**setFont**(**ArialMT\_Plain\_16**);**

Heltec**.**display**->**clear**();**

Heltec**.**display**->**drawString**(**33**,** 5**,** "GEPSO"**);**

Heltec**.**display**->**drawString**(**10**,** 30**,** "OK"**);**

Heltec**.**display**->**display**();**

delay**(**1000**);**

ThingSpeak**.**begin**(**client**);** // Initialize ThingSpeak

**}**

void loop**()** **{**

// try to parse packet

int packetSize **=** LoRa**.**parsePacket**();**

**if** **(**packetSize**)** **{**

// received a packet

Serial**.**print**(**"Received packet '"**);**

// read packet

String req **=** ""**;**

**while** **(**LoRa**.**available**())** **{**

req **+=** **(**char**)**LoRa**.**read**();**

//Serial.print((char)LoRa.read());

//req+=(char)LoRa.read();

**}**

// print RSSI of packet

Serial**.**print**(**"' with RSSI "**);**

Serial**.**println**(**LoRa**.**packetRssi**());**

Serial**.**println**(**"Valor:" **+** req**);**

String temperatura **=** req**.**substring**(**req**.**indexOf**(**"-"**)** **+** 1**,** req**.**indexOf**(**":"**));**

String condutividade **=** req**.**substring**(**req**.**indexOf**(**":"**)** **+** 1**,** req**.**indexOf**(**"+"**));**

String id **=** req**.**substring**(**req**.**indexOf**(**"+"**)** **+** 1**);**

Serial**.**println**(**temperatura**);**

Serial**.**println**(**condutividade**);**

Serial**.**println**(**id**);**

ThingSpeak**.**setField**(**1**,** temperatura**);**

ThingSpeak**.**setField**(**2**,** condutividade**);**

ThingSpeak**.**setField**(**3**,** id**);**

int x **=** ThingSpeak**.**writeFields**(**myChannelNumber**,** myWriteAPIKey**);**

Heltec**.**display**->**clear**();**

Heltec**.**display**->**setTextAlignment**(**TEXT\_ALIGN\_LEFT**);**

String lido**=**"H:"**+**String**(**x**)+**"-ID:"**+**String**(**id**)+**"/"**+**String**(**condutividade**)+**"/"**+**String**(**temperatura**);**

Heltec**.**display**->**drawString**(**0**,** 5**,** lido**);**

Heltec**.**display**->**display**();**

**if** **(**x **==** 200**)** **{**

Serial**.**println**(**"Channel update successful."**);**

**}**

**else** **{**

Serial**.**println**(**"Problem updating channel. HTTP error code " **+** String**(**x**));**

**}**

**}**

delay**(**1**);**

**}**

# **Sensores**

#include "heltec.h"

#include <OneWire.h>

#include <DallasTemperature.h>

const int oneWireBus **=** 13**;**

const int ID **=** 20**;**//ID?

OneWire oneWire**(**oneWireBus**);**

DallasTemperature sensors**(&**oneWire**);**

#define BAND 868E6 //you can set band here directly,e.g. 868E6,915E6

unsigned int counter **=** 0**;**

String rssi **=** "RSSI --"**;**

String packSize **=** "--"**;**

String packet **;**

void setup**()**

**{**

//WIFI Kit series V1 not support Vext control

Heltec**.**begin**(false** /\*DisplayEnable Enable\*/**,** **true** /\*Heltec.Heltec.Heltec.LoRa Disable\*/**,** **true** /\*Serial Enable\*/**,** **true** /\*PABOOST Enable\*/**,** BAND /\*long BAND\*/**);**

Heltec**.**display**->**init**();**

Heltec**.**display**->**flipScreenVertically**();**

Heltec**.**display**->**setFont**(**ArialMT\_Plain\_10**);**

delay**(**1500**);**

Heltec**.**display**->**clear**();**

Heltec**.**display**->**drawString**(**0**,** 0**,** "Heltec.LoRa Initial success!"**);**

Heltec**.**display**->**display**();**

delay**(**1000**);**

pinMode**(**25**,**OUTPUT**);**

sensors**.**begin**();**

Serial**.**begin**(**9600**);**

**}**

void loop**()**

**{**

Heltec**.**display**->**clear**();**

Heltec**.**display**->**setTextAlignment**(**TEXT\_ALIGN\_LEFT**);**

Heltec**.**display**->**setFont**(**ArialMT\_Plain\_10**);**

Heltec**.**display**->**drawString**(**0**,** 0**,** "Sending packet: "**);**

Heltec**.**display**->**drawString**(**90**,** 0**,** String**(**analogRead**(**36**)));**

Heltec**.**display**->**display**();**

LoRa**.**beginPacket**();**

LoRa**.**setTxPower**(**14**,**RF\_PACONFIG\_PASELECT\_PABOOST**);**

sensors**.**requestTemperatures**();**

float temperatureC **=** sensors**.**getTempCByIndex**(**0**);**

// Serial.println(temperatureC);

analogWrite**(**25**,**256**);**

delay**(**500**);**

LoRa**.**print**(**"-"**+**String**(**temperatureC**)+**":"**);**

LoRa**.**print**(**analogRead**(**36**));**

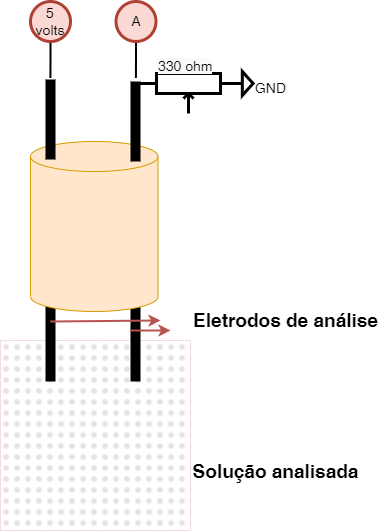
LoRa**.**print**(**"+"**+**String**(**ID**));**

LoRa**.**endPacket**();**

analogWrite**(**25**,**0**);**

delay**(**60000**);**//Verificar tempo

**}**



**Tabela 1 - Código Sensores**