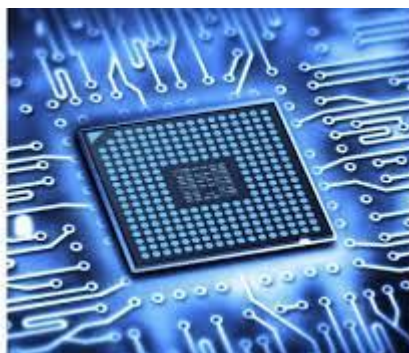


IFMT
DAEE
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

André Martins Pommot Maia



MICROCONTROLADORES
RELATÓRIO DE PROJETO
“CONTROLE DE TEMPERATURA DE AMBIENTES REMOTOS
Node MCU ESP32”

Cuiabá-MT
10/10/2021

Sumário

Microcontroladores – Projeto controle temperatura ambientes remotos

1 – Definição de Microcontroladores.	3
2 – Aplicações e funcionalidades	3
3 – Proposta do projeto	3
4 – Objetivos do projeto.	3
5 - Descrição do projeto	4
5.1 – Descrição dos materiais	4
5.2 – Conhecendo os materiais e componentes utilizados.	4
5.3 – Montando circuito físico.	10
5.4 – Principio de funcionamento.	11
5.5 - Montagem do projeto no Blynk App.	11
6 – Resultados finais.	15
7 – Código fonte do projeto.	17
8 – Conclusão.	17
9 – Bibliografia.	17

1 - Definição de Microcontroladores

Um microcontrolador pode ser definido como um single-chip computer (computador em um único chip), microcomputer, ou ainda como embedded controller. O termo MCU (Micro Controller Unit) também é bastante utilizado para se designar esse dispositivo. No mesmo chip estão integrados uma CPU, também chamada de core (núcleo), e circuitos auxiliares (periféricos) como memória de programa, memória de dados, circuito de clock, interface de comunicação serial, temporizadores, contadores, portas de I/O, etc. Esses diferentes recursos embutidos em um microcontrolador variam em função do modelo e do fabricante.

2 - Aplicações e funcionalidades

Microcontroladores são usados em produtos e dispositivos automatizados, como os sistemas de controle de automóvel, dispositivos médicos implantáveis, controles remotos, máquinas de escritório, eletrodomésticos, ferramentas elétricas, brinquedos e outros sistemas embarcados. Ao reduzir o tamanho e o custo em comparação a um projeto que usa um dispositivo microprocessado, microcontroladores tornam-se econômicos para controlar digitalmente dispositivos e processos. Microcontroladores de sinal misto são comuns, integrando componentes analógicos necessários para controlar sistemas eletrônicos não digitais.

O seu consumo de energia é relativamente baixo, normalmente, na casa dos miliwatts e possui habilidade para entrar em modo de espera (Sleep ou Wait) aguardando por uma interrupção ou evento externo, como, por exemplo, o acionamento de uma tecla, ou um sinal que chega via uma interface de dados. O consumo destes microcontroladores em modo de espera pode chegar na casa dos nanowatts, tornando-os ideais para aplicações onde a exigência de baixo consumo de energia é um fator decisivo para o sucesso do projeto.

De forma diferente da programação para microprocessadores, que em geral contam com um sistema operacional e um BIOS, o programador ou projetista que desenvolve sistemas com microcontroladores, geralmente, cria todo programa que será executado pelo sistema ou pode usar um sistema operacional próprio para microcontroladores chamado de RTOS.

3 - Proposta do projeto

A proposta deste projeto baseia-se na necessidade de controlar ambientes extremamente sensíveis a variação de temperatura. Podemos citar por exemplo, ambientes cujo a característica é abrigar e manter equipamentos eletrônicos cuja sensibilidade à variação de temperatura pode danificar tais dispositivos. Como tecnico e utilizando experiência própria em manutenção de equipamentos eletrônicos de telecomunicações, pude constatar falhas e defeitos em muitos sítios repetidores de telecomunicações cujo ambientes e temperaturas não eram adequados aos tipos de dispositivos implantados, o que gerava manutenções desnecessárias e custos à operação destes sistemas, pois havia a necessidade continua de deslocamentos para atender falhas relacionadas a climatização e refrigeração destes ambientes.

Pensando nisso optei por desenvolver um protótipo que atenda essa demanda, utilizado um microcontrolador IOT (esp32) e alguns periféricos de baixo custo operacional que tem a rotina de controlar e monitorar a temperatura ambiente de um local remoto, acionar equipamentos de reforço/alarme e enviar notificações caso haja mudanças na variável de controle, tudo através de uma plataforma Web.

4 - Objetivos do projeto

Os objetivos desse projeto são:

- Desenvolvimento das habilidades praticas na elaboração de protótipo de microcontroladores;
- Aplicar o conhecimento teórico dos fundamentos da microeletrônica e programação obtido através das aulas online remotas;
- Criar soluções e inovações praticas que auxiliam e desenvolvem o aprendizado acadêmico;
- Compreender a operação e o funcionamento de novos dispositivos de controle (IOT) integrados a WEB.
- Interagir com colegas de classe e professores afim de estar desenvolvendo as boas práticas de aprendizado pratico/teórico.

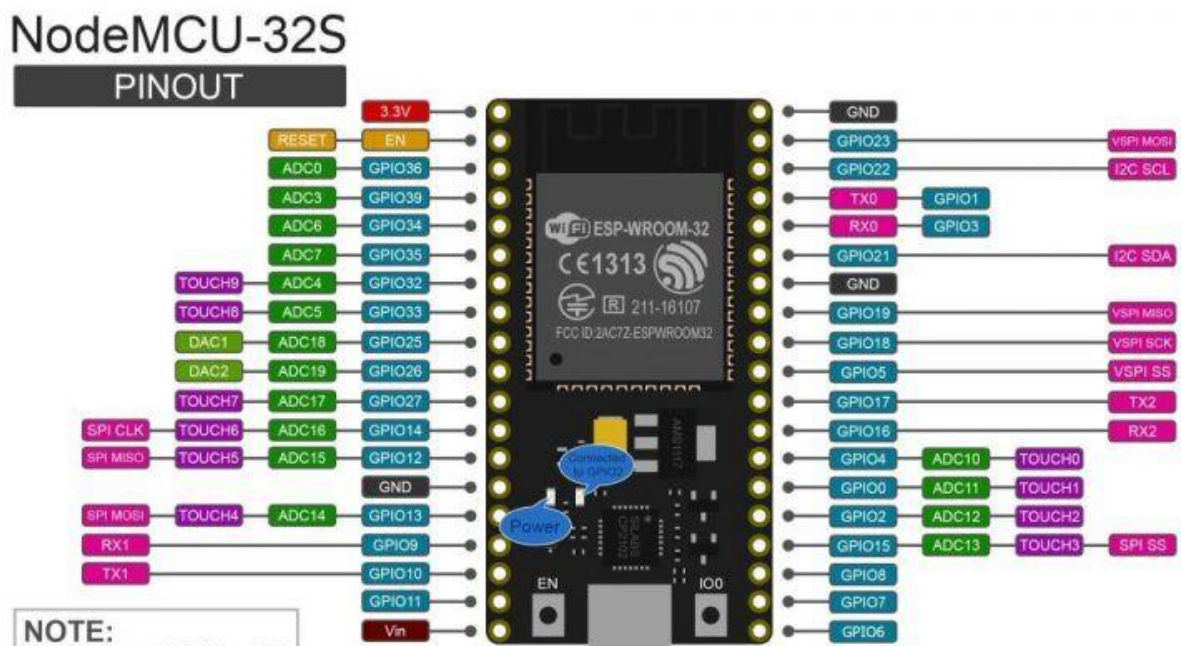
5 - Descrição do projeto

5.1 - Materiais Utilizados

- 1 Placa de desenvolvimento modulo ESP NodeMCU-32S;
- 1 Protoboard (placa de ensaios de circuitos elétricos);
- 1 Sensor de temperatura DS18B20 5V;
- 1 modulo rele 2 canais 5V;
- 1 resistor de 4K7;
- 1 potenciômetro 10K;
- 1 display lcd 16x2;
- 1 fonte externa 5V – carregador de celular;
- 1 PC com Arduino IDE instalado;
- 20 Jumpers macho/fêmea;
- Plataforma IOT Blynk;
- Roteador wi-fi com acesso à internet;

5.2 - Conhecendo os componentes e materiais utilizados

- Placa de desenvolvimento modulo ESP NodeMCU-32S;



Principal característica é o seu poder de duplo processamento (dual core), comunicação wireless, WiFi ou Bluetooth, já integrado à placa. Ela pode ser utilizada em diversos projetos robóticos e, principalmente, em projetos IOT.

Portas GPIO - acrônimo de General Purpose Input/Output, são pinos responsáveis pela entrada e saída de sinais digitais. Elas são capazes, individualmente, de fornecer um máximo de 12 mA de corrente.

Portas Touch - Esses pinos funcionam como sensores capacitivos, ou seja, reagem quando há uma alteração em seu estado. Um simples exemplo é tocá-lo: isso iniciará um processo de troca de calor entre dois corpos distintos, onde o sensor perceberá e emitirá sinais para a ESP. Essa placa possui 10 pinos touch.

ADC - (Analog Digital Converter), são famosos na área da eletrônica. Eles convertem grandezas analógicas (velocímetro de ponteiro do carro, por exemplo) em grandezas digitais (um velocímetro digital, por exemplo). Segundo o diagrama do ESP, ele possui 16 pinos com essa capacidade.

Demais pinos - O NodeMCU ESP32 conta com 3 portas GND, 1 porta de 3.3V e 1 porta de 5V (que no diagrama está identificada por "V in", visto que pode ser usada para alimentação da placa caso ela não esteja conectada pelo cabo USB). Há também outros pinos, como o RX e TX, CLK, MISO e outros, note que alguns deles acumulam diversas funções em um pino só.

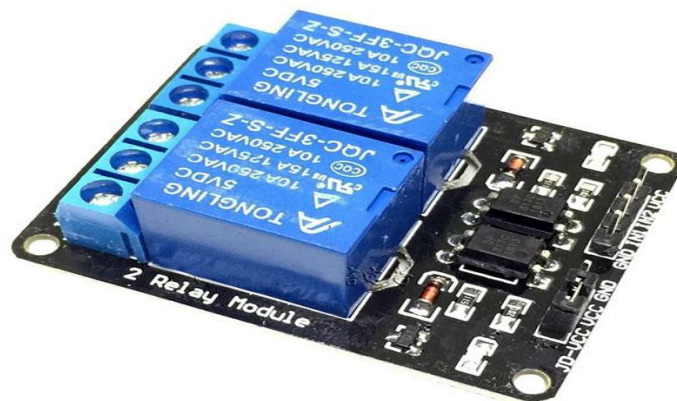
- **Sensor de temperatura DS18B20 5V;**



- Tensão de alimentação: 3.0 – 5.5VDC;
- Leitura de temperatura entre: -55°C a +125°C;
- Tempo de atualização: menor que 750 ms;
- Precisão: $\pm 0.5^{\circ}\text{C}$;
- Resolução: 9 ou 12 bits;
- Interface 1 fio (1-Wire).

Podendo ser alimentado de forma convencional (VCC – GND) ou pela própria linha de dados (chamado de “modo parasita”). Além disso cada sensor possui um ID exclusivo de 64bits, permitido que outros sensores DS18B20 possam ser conectados no barramento utilizando apenas uma porta do microcontrolador. O DS18B20 também pode ser encontrado na versão à prova d’água, que permite que seja utilizado submerso em líquidos.

- **Modulo controle rele 2 canais 5V;**

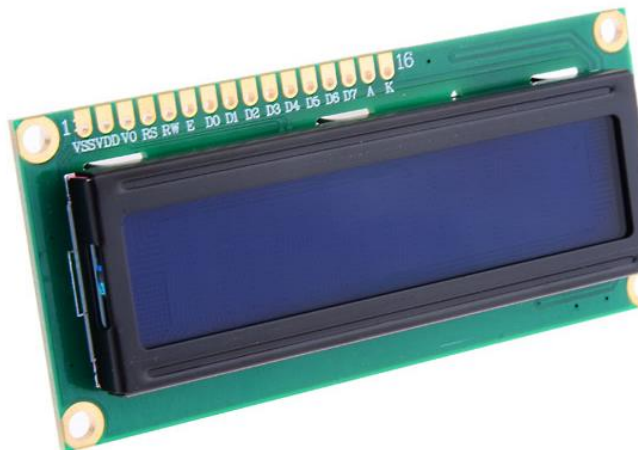


Módulo para acionamento de cargas de até 220V AC, como lâmpadas, equipamentos eletrônicos, motores, ou usá-lo para fazer um isolamento entre um circuito e outro. O módulo é equipado com transistores, conectores, leds, diodos e relés de alta qualidade. Cada canal possui um LED para indicar o estado da saída do relé.

Especificações:

- Modelo: JQC-3FF-S-Z
- Tensão de operação: 5 VDC
- Permite controlar cargas de até 220V AC
- Corrente nominal: 71,4 mA
- LED indicador de status
- Pinagem: Normal Aberto, Normal Fechado e Comum
- Tensão de saída: (28 VDC a 10A) ou (250VAC a 10A) ou (125VAC a 15A)
- Furos de 3mm para fixação nas extremidades da placa
- Tempo de resposta: 5~10ms
- Dimensões: 50 mm x 37 mm x 18 mm
- Peso: 30g

- **Display LCD 16x2 5V**



Especificações:

- Cor backlight: Azul
- Cor escrita: Branca
- Dimensão Total: 80mm X 36mm X 12mm
- Dimensão Área visível: 64,5mm X 14mm
- Dimensão Caracter: 3mm X 5,02mm
- Dimensão Ponto: 0,52mm X 0,54mm

Lcd 5 Volts com 16 colunas por 2 linhas, backlight azul e escrita branca. Possui o controlador HD44780 usados em toda indústria de LCD's como base de interface. Será utilizado para visualização da temperatura do local e também como indicativo de estado de temperatura normal/alarme.

- **Plataforma Blynk IOT**

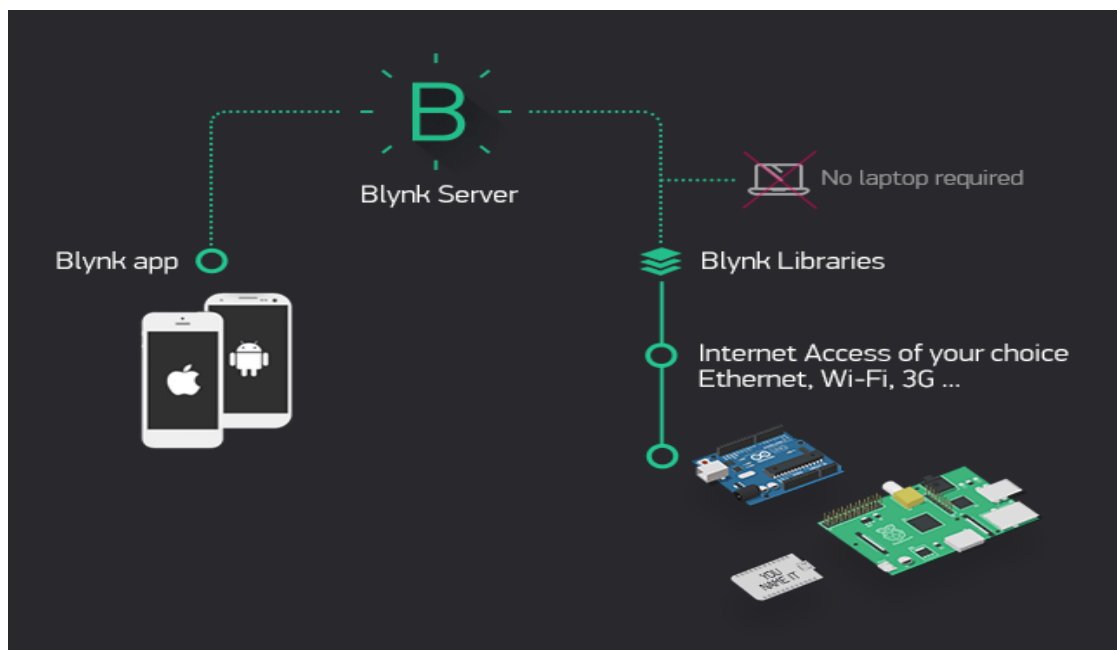
O Blynk é uma plataforma IoT que permite realizar comunicação entre um smartphone e diferentes tipos de soluções microcontroladas como Arduino, ESP32, Raspberry Pi de forma simples. Com o Blynk é possível monitorar e controlar dispositivos pelo celular, realizar a comunicação entre dispositivos utilizando o Blynk Cloud sem a necessidade de app, permitindo o envio de e-mails, tweets, notificações push e etc.

O Blynk possui uma arquitetura simples composta por 3 elementos principais:

Aplicativo Blynk: Com o aplicativo podemos adicionar e configurar novos dispositivos, adicionar widgets para visualizar dados, adicionar botões de controle e etc. Todos os ajustes são feitos de forma extremamente simples e com poucas interações é possível realizar comunicação com nossa solução microcontrolada. O aplicativo Blynk está disponível para sistemas Android e IOS.

Servidor Blynk: Esse elemento gerencia toda a comunicação entre o app e os dispositivos microcontrolados. Geralmente utiliza-se o Blynk Cloud, mas é possível executar um servidor Blynk localmente, num Raspberry Pi por exemplo.

Bibliotecas Blynk: São bibliotecas que trazem rotinas e funções úteis desde controle de GPIO até conexão com o servidor Blynk. Existem Bibliotecas para as plataformas mais populares como Arduino, ESP32/8266 e Raspberry Pi.



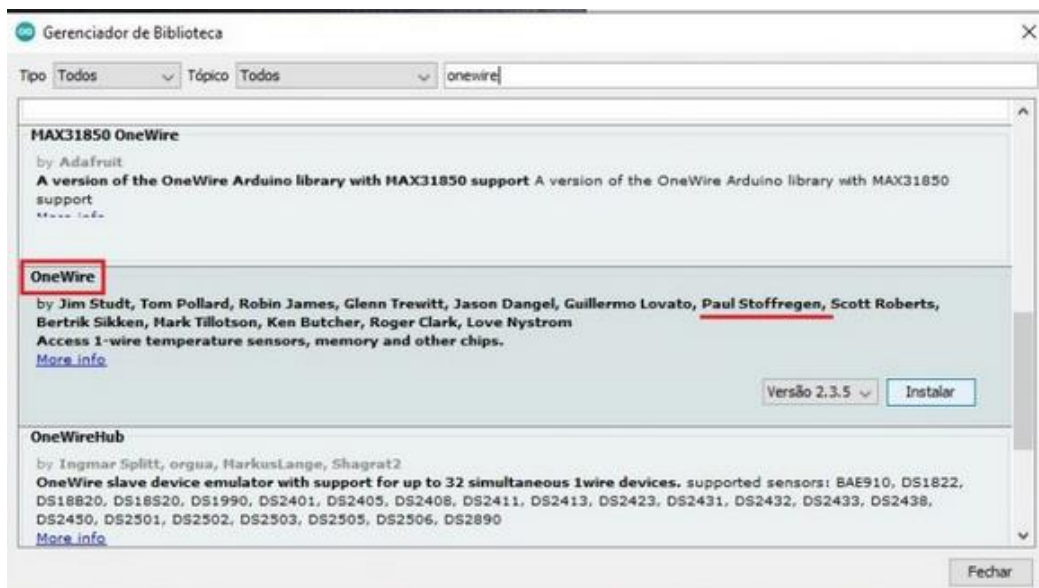
Para conectar o hardware à Internet, você pode escolher qualquer módulo embutido ou externo via:

WiFi; Ethernet; Celular (GSM, 2g, 3g, 4g, LTE); Serial; USB através do seu PC.

- **IDE Arduino – Bibliotecas**

Neste projeto precisamos instalar bibliotecas para utilizar o sensor de temperatura DS18B20, o LCD e a biblioteca de acesso ao Blynk IOT. Podemos encontrar todas estas bibliotecas no gerenciador de bibliotecas da Arduino IDE.

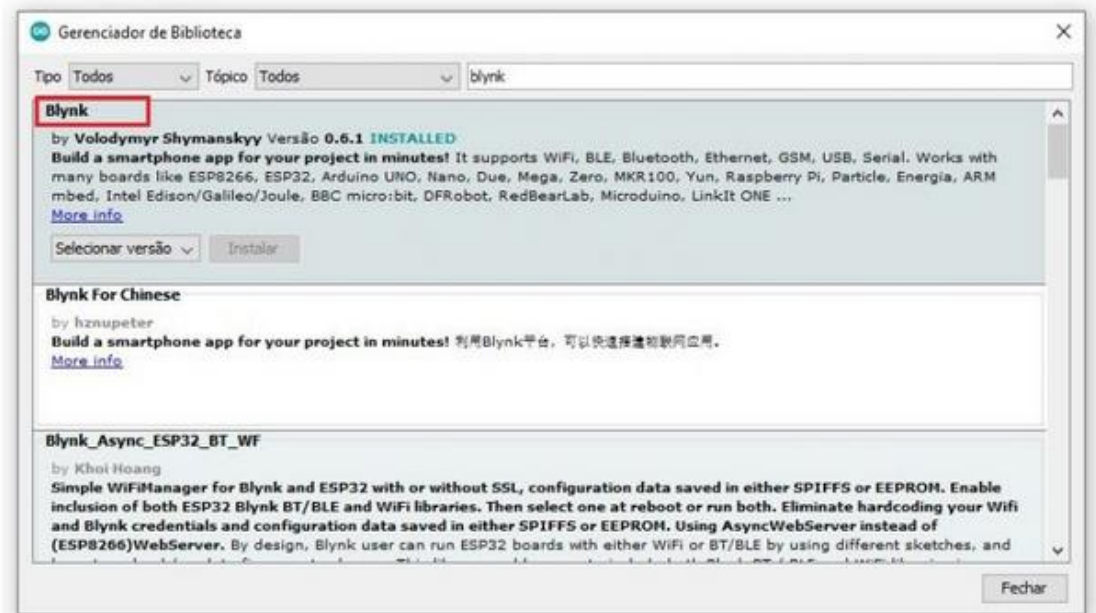
Biblioteca OneWire – Para troca de dados no sensor de temperatura:



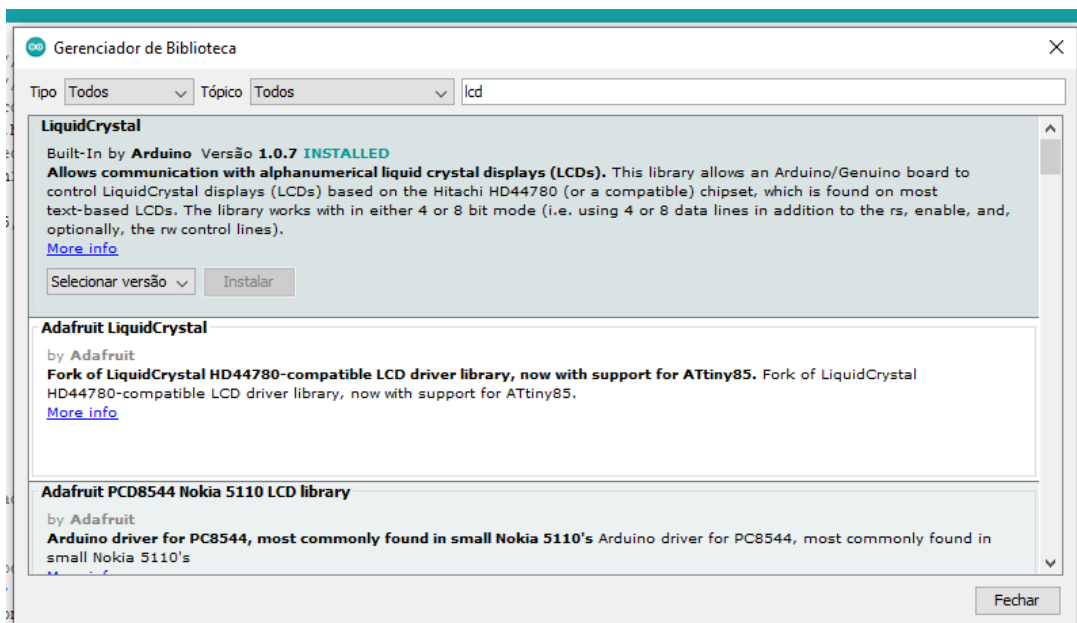
Biblioteca DallasTemperature para controle de dados de temperatura



Biblioteca Blynk para comunicação com servidor e acesso remoto.



Biblioteca LiquidCrystal para acesso e controle do LCD



5.3 – Montando circuito físico

O projeto foi desenvolvido em forma de protótipo e utiliza uma protoboard simples para interligação dos circuitos e periféricos às interfaces I/O do MCU ESP32, abaixo será mostrado as conexões físicas destas interfaces.

O sensor de temperatura **DS18B20** a prova d'água possui 3 fios que serão ligados na seguinte sequência ao ESP32.

- **Fio Amarelo** – Conectado ao pino GPIO23 do ESP32;
 - **Fio Preto** – ligado ao GND do barramento negativo da protoboard pino GND ESP32;
 - **Fio Vermelho** – ligado ao barramento positivo da protoboard pino 3.3 V do ESP32;
 - **Resistor de 4k7** – Ligado entre o pino GPIO23 e o pino 3.3V do ESP32
- Precisamos ligar o fio de dados (amarelo) do sensor em resistor de **Pullup**. O ESP 32 fornece essa ligação automaticamente, porem para este projeto foi escolhido a forma manual para o **Pullup**;

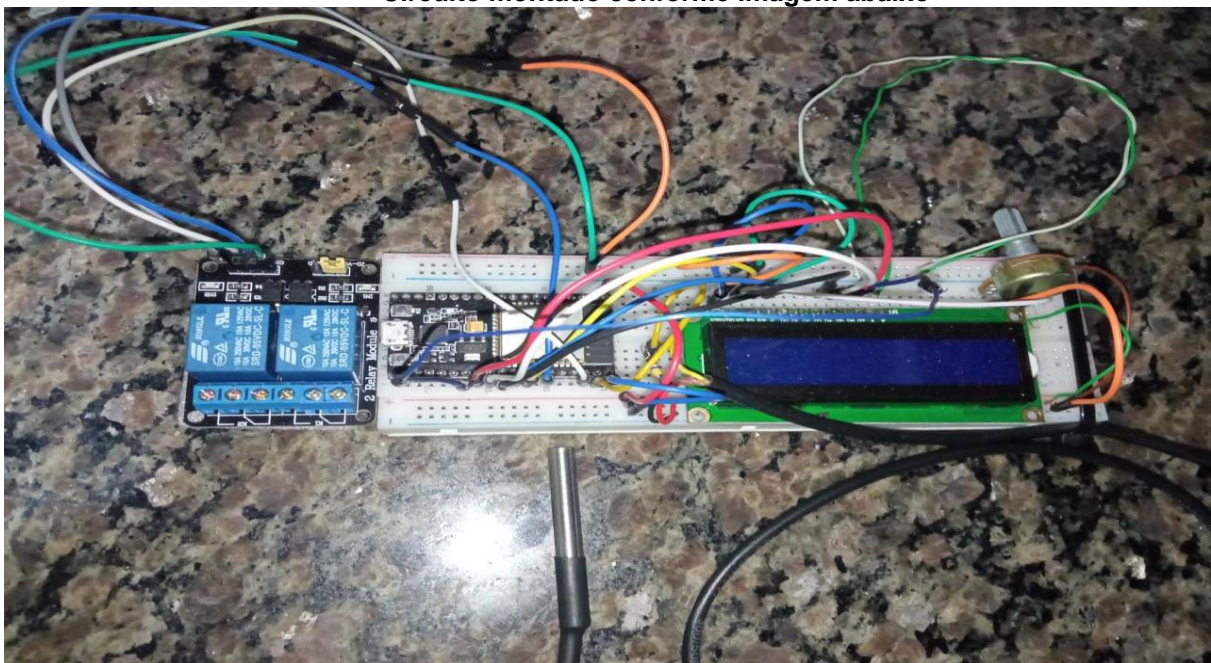
Para o LCD configuramos as seguintes pinagens:

- Pino 1 LCD - **GND** – Ligado ao barramento negativo GND da fonte externa;
- Pino 2 LCD - **5V** – Ligado ao barramento positivo da fonte externa 5V;
- Pino 3 LCD - **V0** - ligado ao pino central do potenciômetro (contraste do LED interno);
- Pino 4 LCD - **RS** – Ligado ao pino GPIO19 do ESP32;
- Pino 5 LCD - **R/W** – Ligado ao barramento negativo GND da fonte;
- Pino 6 LCD - **E** – Ligado ao pino GPIO18 do ESP32;
- Pino 11 LCD - **DB4** – Ligado ao pino GPIO05 do ESP32;
- Pino 12 LCD - **DB5** – Ligado ao pino GPIO17 do ESP32;
- Pino 13 LCD - **DB6** – Ligado ao pino GPIO16 do ESP32;
- Pino 14 LCD - **DB7** – Ligado ao pino GPIO04 do ESP32;
- Pino 15 LCD - **BPL 5V** – Ligado ao barramento positivo da fonte externa;
- Pino 16 LCD - **GND** - Ligado ao barramento negativo GND da fonte externa
- Potenciômetro ligado ao barramento positivo e negativo da fonte externa e pino central conectado ao pino 3 do LCD (Para controle de contraste do led LCD);

Para o modulo RELE 2 canais configuramos as seguintes pinagens:

- Pino 1 modulo rele - **GND** – Ligado ao barramento negativo GND da fonte externa;
- Pino 2 modulo rele - **Canal 0** – Ligado ao pino GPIO22 ESP32;
- Pino 3 modulo rele - **Canal 1** – Ligado ao pino GPIO21 ESP32;
- Pino 4 modulo rele – **5V** – Ligado ao barramento positivo +5V fonte externa;

Circuito montado conforme imagem abaixo



5.4 – Princípio de funcionamento

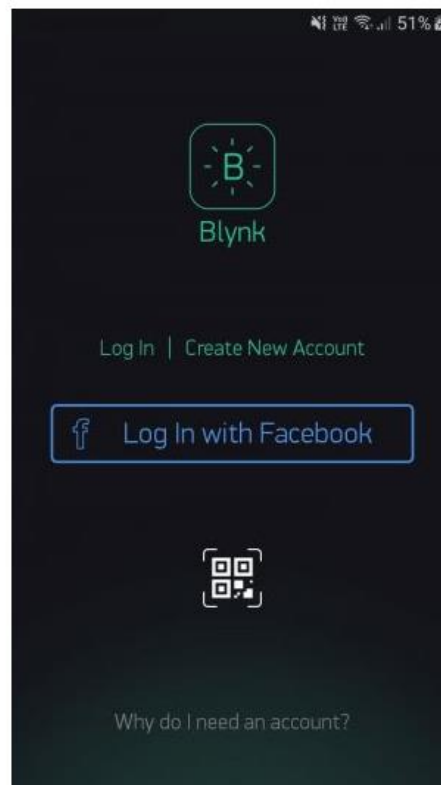
O princípio de funcionamento deste protótipo baseia-se no controle de temperatura de um ambiente remoto (iremos simular um sitio repetidor de telecomunicações). Supomos que foi realizado uma pesquisa de campo para instalar equipamentos de rádio frequência RF em um sitio de telecomunicações, cujo uma das características para o bom funcionamento e garantia de estabilidade /disponibilidade acima de 98% dos serviços dispostos, dependem da climatização deste ambiente, o qual foi projetado para não ultrapassar 32 graus Celsius. Caso isso ocorra poderá haver indisponibilidade do serviço devido ao superaquecimento dos transmissores.

No levantamento realizado foi informado que será utilizado duas maquinas refrigeradoras de ar, devendo estas, alternar o ciclo de funcionamento a cada 12 horas afim de manter a temperatura abaixo dos 32° C e caso ocorra o aquecimento, as duas maquinas refrigeradoras deverão atuar de forma conjunta para diminuir a temperatura e estabilizar a climatização do ambiente abaixo do nível projetado.

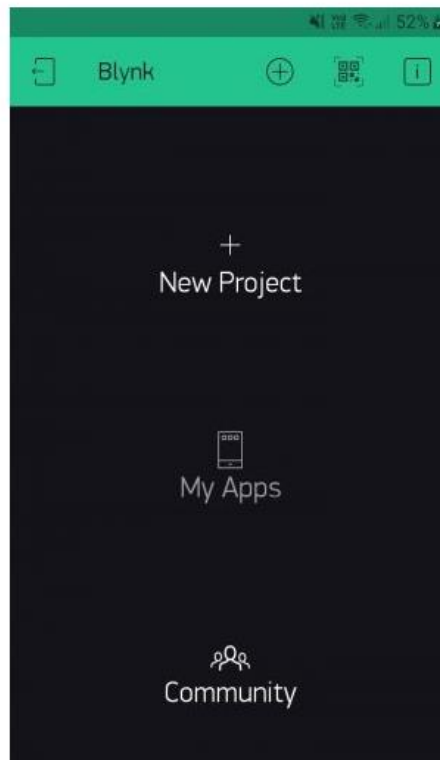
Este protótipo desenvolvido com MCU ESP32 realizara a cada 2 segundos a varredura da temperatura ambiente através do sensor **DS18B20** (variável de controle). Através dos dados dispostos pelo sensor, programamos o **modulo rele dois canais** para trabalharem ciclos alternados enquanto a temperatura estiver abaixo do nível desejado. Se a temperatura ultrapassar o nível desejado, o MCU imediatamente enviara sinais para os **dois canais do modulo rele**, atuarem e estabilizarem a climatização. Para a visualização da temperatura no local, será utilizado um display **LCD 16x2** que fornecerá as informações de temperatura ambiente, bem como o estado temperatura normal/ temperatura alta. Para o controle e monitoramento remoto, será utilizado a **plataforma Blynk**, que fara o controle (**via Widgets**) e enviara notificações de alarme via **e-mail** para o centro de gerencia, alertando a mudança no estado da climatização, o qual tomara as devidas providencias de manutenção e operação do sistema.

5.5 – Montagem do projeto no Blynk App

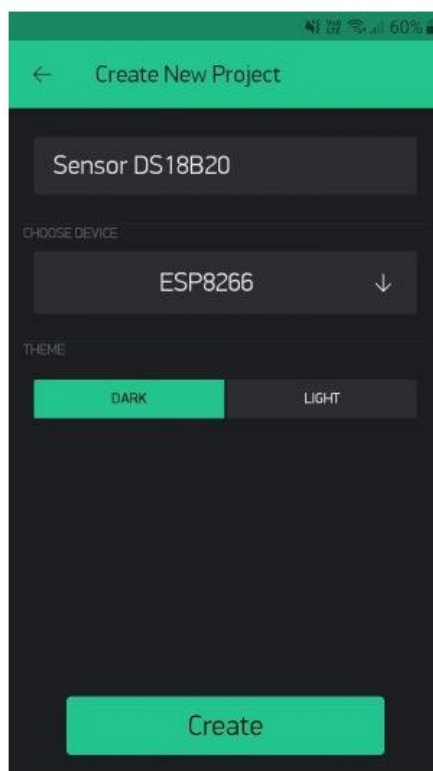
Antes de tudo instalei o Blynk no meu dispositivo móvel. Para este projeto utilizei um smartphone com sistema Android. Após abrir pela primeira vez o aplicativo ele nos pedirá para criar uma conta. Utilize um e-mail válido pois será por ele que você receberá os tokens necessários para cada novo dispositivo.



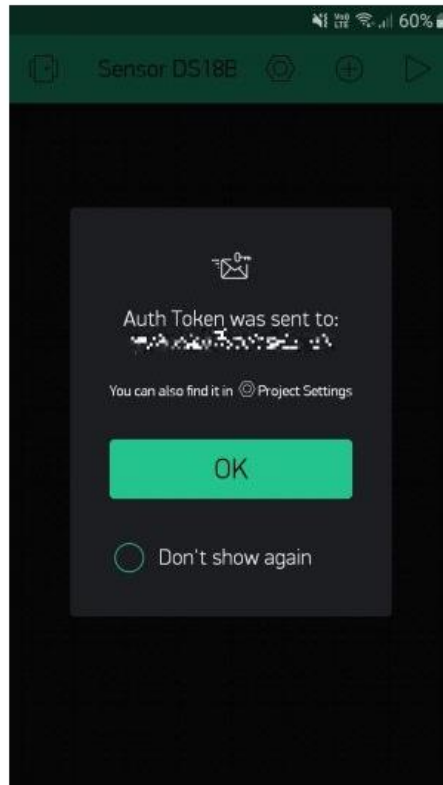
Após concluir o login estamos prontos para criar um novo projeto.



Escolha a opção **New Project** e em seguida temos a opção de escolher o nome do projeto e qual dispositivo iremos utilizar. Para esse exemplo usaremos o ESP32 como dispositivo, e nomearemos o projeto de "Sensor DS18B20". Também temos a opção do tema escuro ou claro, mas não interfere no funcionamento do nosso sistema.



Clicando em **Create** o Blynk irá emitir um aviso informando que seu token foi enviado por e-mail.

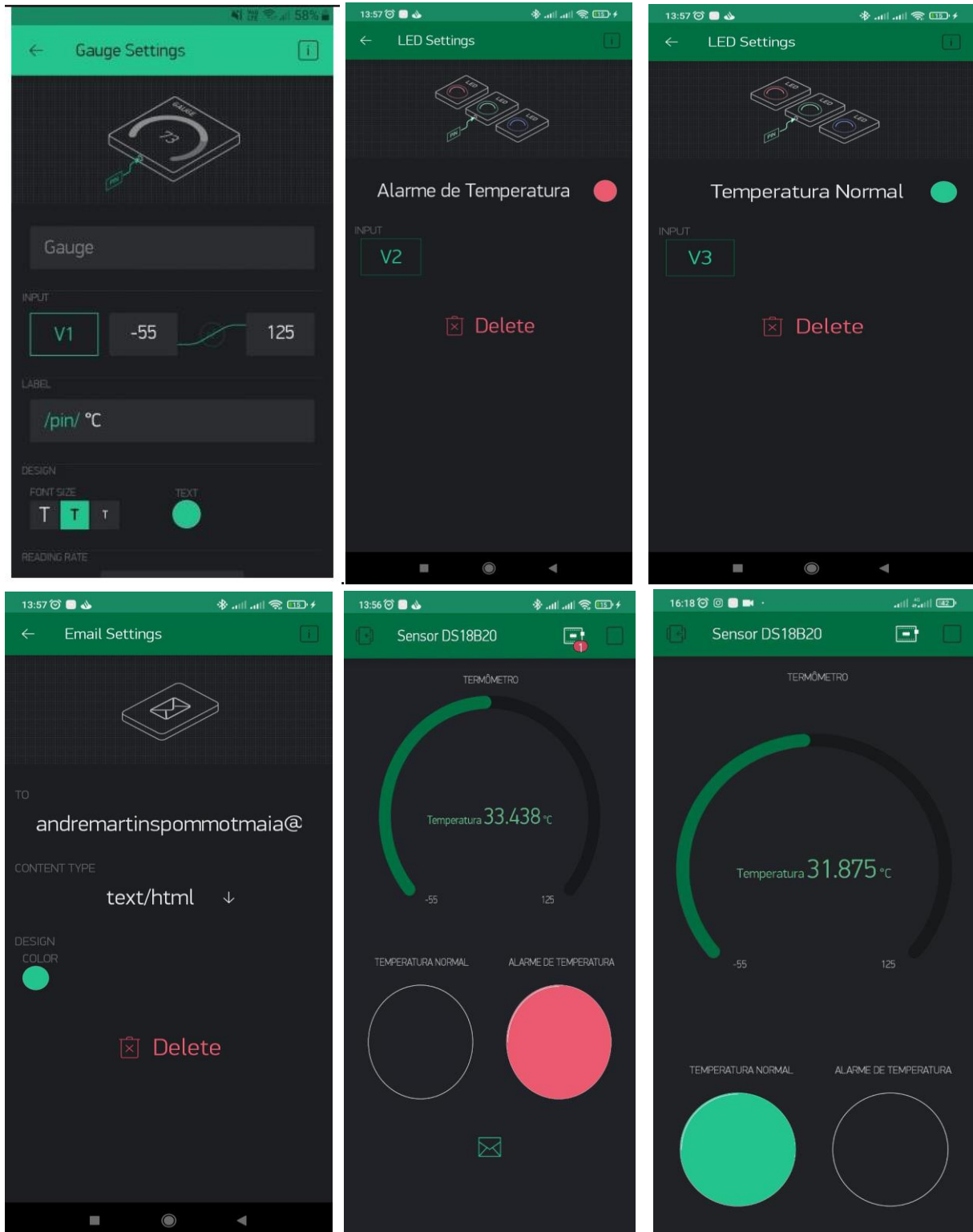


Verifique a caixa de entrada do seu e-mail e guarde o token, ele será utilizado para autenticar o MCU ESP32.

Adicionando Widget ao projeto

Vamos adicionar no projeto um Widget do tipo **Gauge**, dois widget do tipo **Led** e um do tipo notificação. Primeiramente, o widget **Gauge** é semelhante a um relógio onde existe uma barra em forma de arco com 2 limites, essa barra será preenchida de acordo com a variável de temperatura. Os dois led adicionados servirão como parâmetro de estado Normal/Alarme onde o led verde ficará aceso na condição normal com temperatura abaixo de 32° C e o led vermelho indica alarme com temperatura acima dos 32° C e por último o Widget de notificação que emitira um e-mail de alerta de alarme de temperatura alta Vale destacar que cada widget consome uma quantidade de “energia” do app. Temos 1200 pontos de energia para trabalhar na versão gratuita do app, para o nosso exemplo será mais do que o suficiente

Configurando os Widgets no Blynk app conforme mostra as figuras abaixo:



6 – Resultados finais

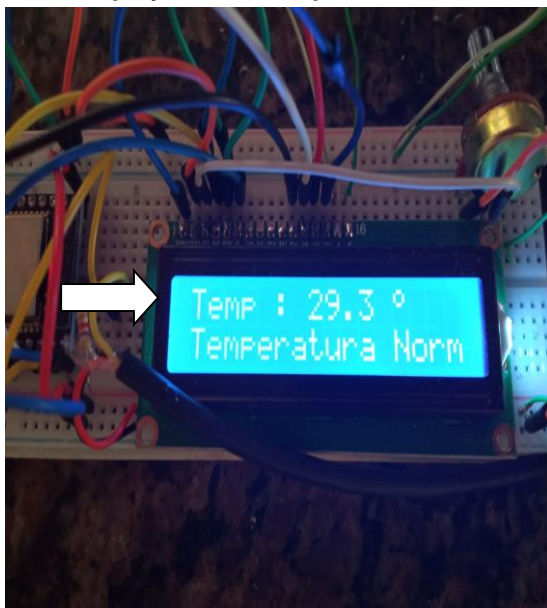
Após a montagem do circuito físico, programação do MCU e integração com aplicativo de gerenciamento remoto Blynk App, foi iniciada a fase de testes contendo as seguintes premissas:

- ✓ Checagem da calibração do sensor DS18B20 (temperatura ambiente);
- ✓ Checagem do LED interno na condição de alarme do sistema;
- ✓ Checagem da temporização para comutação dos canais RELE 0/RELE 1;
- ✓ Checagem da comutação dos canais RELE 0 /RELE 1;
- ✓ Checagem do contraste de brilho no display LCD;
- ✓ Checagem das informações apresentadas no display LCD;
- ✓ Checagem das informações de leitura de temperatura apresentadas no Blynk App;
- ✓ Checagem dos LED de estado Normal/Alarme Blynk App;
- ✓ Checagem das notificações via E-mail nas condições de alarme de temperatura alta;
- ✓ Testes de temperatura sob condição normal de funcionamento (temp. menor que 32°C);
- ✓ Testes de temperatura sob condições de alarme de funcionamento; (temp, maior que 32°C);
- ✓ Testes finais de simulação com acionamento de dois ventiladores;

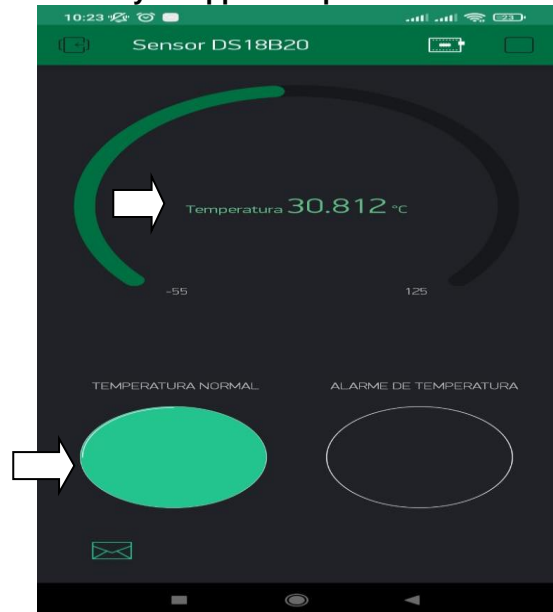
1º Teste – Condições normais de funcionamento

Para o primeiro teste, obtive resultados satisfatórios de acordo com o projetado, ou seja, sob temperatura abaixo de 32°C o MCU ESP32 realizou a temporização dos canais 0 e 1 do modulo rele, alternando em ciclos a ligação dos refrigeradores. O sensor DS18B respondeu exatamente como especificado pelo fabricante, pois comparando com parâmetros externos de temperatura (Google Clima), constatamos que praticamente não houve diferença entre ambas. A visualização local de temperatura pelo display LCD apresentou-se conforme programado, informando a temperatura do ambiente em tempo real e indicando estado **Normal** de Temperatura. Para a visualização e controle remoto, o Blynk App se comportou-se conforme programado, pois realizei o monitoramento remoto em tempo real da temperatura do ambiente local, informando no LED verde estado **Normal** de funcionamento conforme imagens abaixo:

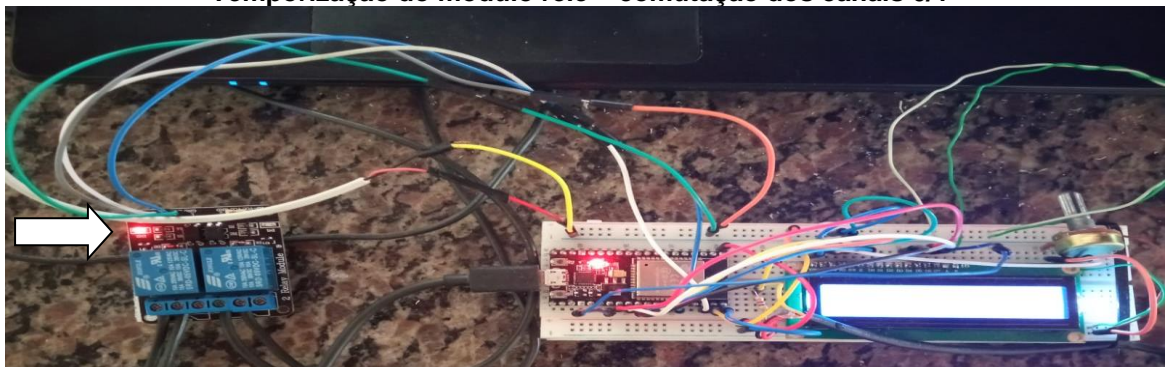
Display LCD – Temperatura normal



Blynk App – Temperatura normal



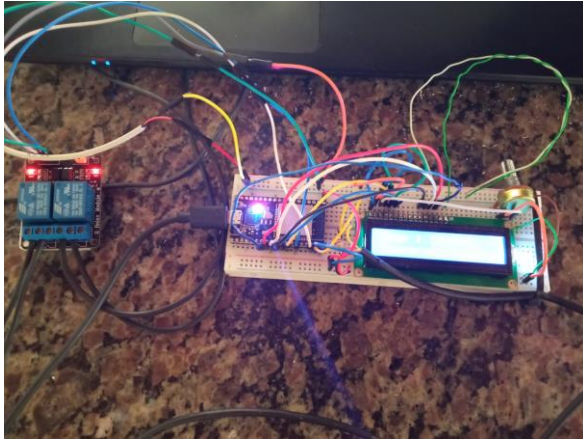
Temporização do modulo rele – comutação dos canais 0/1



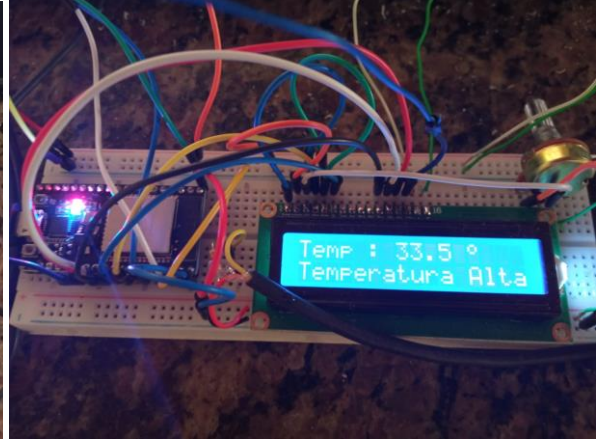
2º Teste – Condições alarme de funcionamento

Para o segundo teste, os resultados obtidos também foram satisfatórios conforme o projetado, ou seja, para temperaturas maiores que 32°C o node MCU iniciou a rotina de resfriamento do ambiente, ligando os dois canais do modulo RELE e consequentemente os dois refrigeradores começaram a atuar de forma conjunta afim de manter/restabelecer a temperatura abaixo dos 32°C. No display LCD foi verificado que houve alteração para o estado de alarme de temperatura alta conforme o programado. Para o monitoramento remoto, o Blynk app apresentou funcionamento conforme programado, ou seja, informando a temperatura em tempo real, acendendo LED indicador de alarme em vermelho e enviando notificações via E-mail com as informações de alarme de temperatura alta, conforme imagens abaixo:

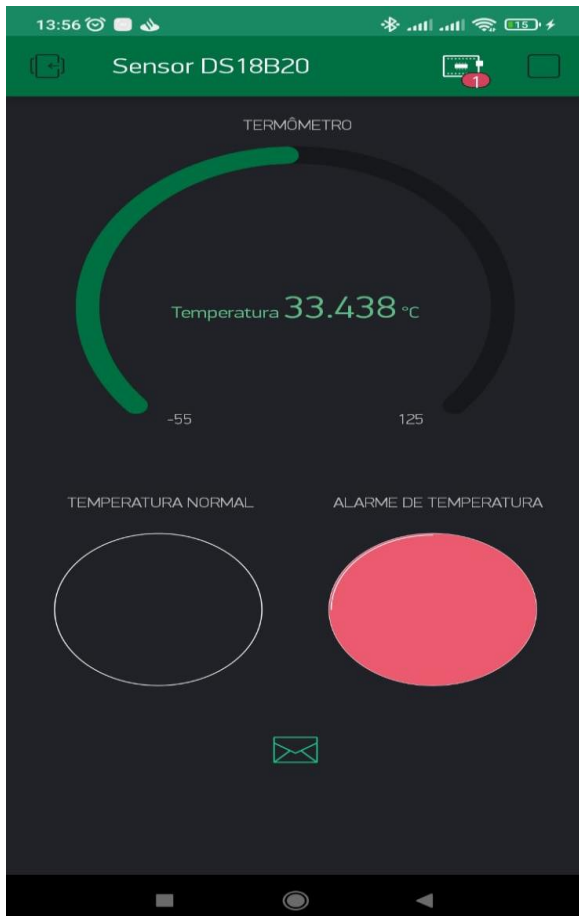
Modulo rele 2 canais acionado



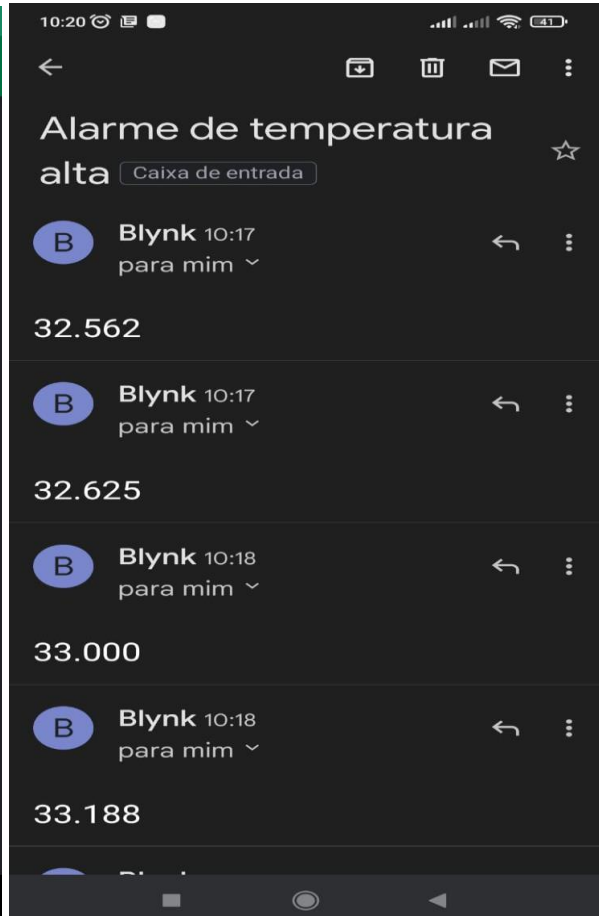
LCD – Temperatura alta



Blynk App – Alarme Temperatura



Notificação via E-mail – Temperatura alta



7 – Código fonte do projeto

O código fonte deste projeto pode ser verificado e baixado através do seguinte link:

https://github.com/pommot/Microcontroladores_ESP32_Projeto_Control_Temperatura.git

8 - Conclusão Final

As aulas experimentais e os projetos práticos são de fundamental importância para o aprendizado pedagógico uma vez que tendo contato com a prática, com os instrumentos e com as dinâmicas aplicadas, as dúvidas teóricas vão sendo sanadas, as teorias podem ser devidamente comprovadas, acrescentando valor ao conhecimento e a produção intelectual.

9 - Referencia Bibliográfica

- Prof. Dr. Alberto Willian Mascarenhas - IFMT <https://sites.google.com/site/awmascarenhas>
- Blynk App <https://blynk.io/>
- Arduino IDE - <https://www.arduino.cc/en/software>
- Node MCU ESP 32 - <https://www.espressif.com/>
- Sensor Dallas DS18B20 – Datasheet:
https://www.alldatasheet.com/view.jsp?Searchword=Datasheet%20ds18b20&gclid=EAIaIQobChMI4svIm6Hj8wIVBQeRCh0wUwFFEAAAYAiAAEgLxdfD_BwE
- LCD Display 16x2 - Datasheet:
https://category.alldatasheet.com/index.jsp?sSearchword=16x2%20lcd%20datasheet&gclid=EAIaIQobChMI67X4t6Hj8wIV3MizCh3sagoZEAAYASAAEgLcMPD_BwE
- Modulo Rele 2 Canais – Datasheet:
<https://www.arduinolandia.com.br/produto/modulo-rele-5v-2-canais.html>