

RELATÓRIO TÉCNICO

Gerenciamento Remoto de Temperatura

Jhonatan R. Souza - si.jhonatan.souza@gmail.com
Marcelo B. Feltran - marcelobenign@gmail.com
Pablo E. G. Nunes - pabloedug7@gmail.com
Rafael A. C. Pires - rafael_augusto1910@hotmail.com
Sérgio H. Nonomura - shnonomura@gmail.com

Novembro de 2019

Resumo

O objetivo deste projeto visa promover o gerenciamento remoto de temperatura de um ambiente residencial. Com o advento em voga da IoT-Internet of Things onde tudo deve estar conectado por meio da internet, este projeto visa demonstrar por meio de dispositivos e sensores tal fato. Foram encontradas diversas dificuldades, desde a concepção do projeto, o escopo, os dispositivos a serem utilizados, bem como, a tecnologia a ser empregada. A princípio, decidiu-se a utilização somente de sensor de temperatura para realização do gerenciamento, mas a posteriori foi utilizado o sensor de chuva. O que levou maior esforço e desgaste durante o projeto para a equipe. Foram utilizados para realização, integração e concepção do projeto: o microcontrolador ESP32, sensor de umidade DHT11, sensor de chuva YL-83, motor server, ventoinha de PC, lâmpada incandescente, Firebase real database, dispositivo/aplicativo Android, internet e uma maquete. O funcionamento realizou-se da seguinte forma: procedeu-se a configuração da rede do ESP32 via WifiManager que capturou e gravou na memória Flash a rede, login e senha possibilitando a conexão do mesmo a Internet. O aplicativo Android captura a temperatura atual do ambiente e o usuário estabelece os limites máximo e mínimo de temperatura (intervalo em que se encontra a temperatura desejada) que é enviada ao Firebase Realtime Database. O ESP32 envia para o Firebase a temperatura atual e captura o faixa da temperatura desejada em intervalos de tempo. Ele conectado aos dispositivos controladores das janelas, ventiladores e aquecedores promove o controle de temperatura até o atingimento da temperatura desejada (entre os limites mínimo e máximo) de forma automática.

Sumário

1	Introdução	3
2	Requisitos.....	4
3	Tecnologias	4
3.1	ESP32	5
3.2	Relê	7
3.3	DHT11	9
3.4	Servo-Motor	10
3.5	Sensor de chuva	12
3.6	Wi-fi	13
3.7	Firebase	13
4	Desenvolvimento	15
5	Testes.....	15
6	Dificuldades.....	15
7	Conclusão	16
8	Referências	17

1 Introdução

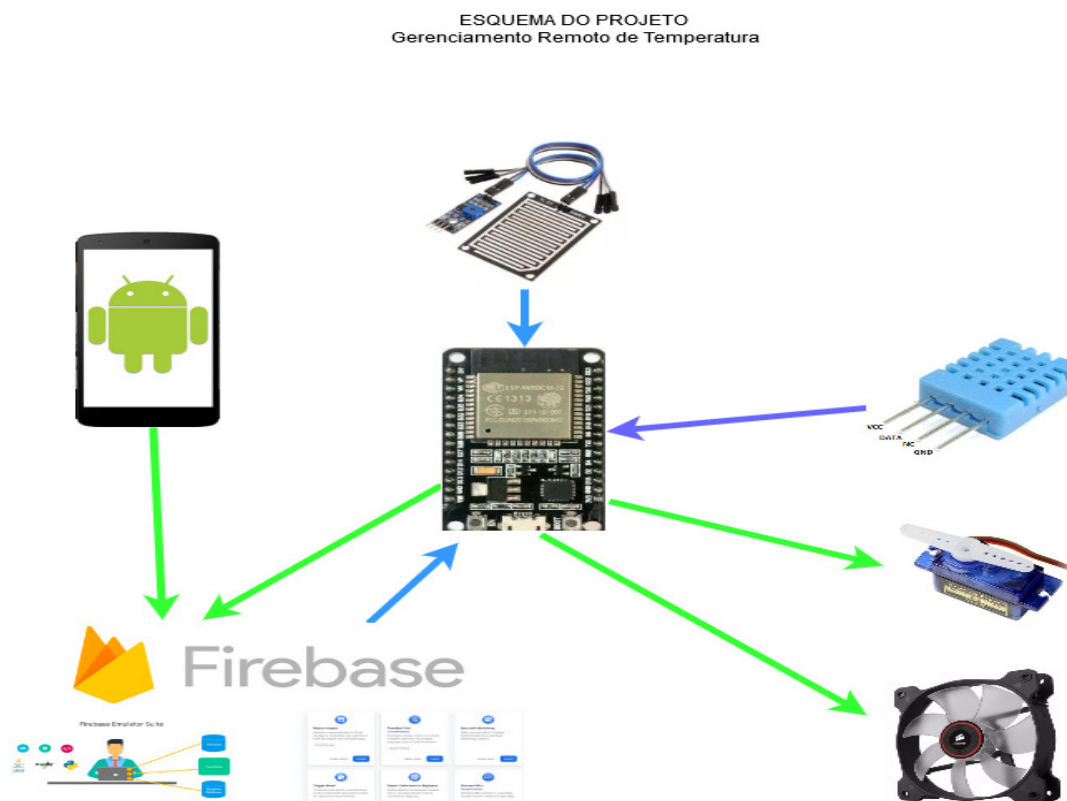


Figura 1-diagrama esquemático do projeto

O presente projeto visa desenvolver um sistema de gerenciamento remoto de temperatura para que possa controlar a temperatura de um residência remotamente. O sistema através de sensores de temperatura, dispositivos/equipamentos acionadores (ventiladores, aquecedores e motores acionam abertura/fechamento janelas), microcontroladores realizarão a comunicação com dispositivos móveis via internet.

O usuário remotamente poderá receber e enviar dados de temperatura permitindo e, assim, intervir no sistema definindo a temperatura desejada.

Foram utilizados para realização, integração e concepção do projeto: o microcontrolador ESP32, sensor de umidade DHT11, sensor de chuva YL-83, motor servo, ventoinha de PC, lâmpada incandescente, Firebase real database, dispositivo/aplicativo Android, internet e uma maquete.

O funcionamento ocorre da seguinte forma: procede-se a configuração da rede do ESP32 via WifiManager que captura e grava na memória Flash a rede, login e senha possibilitando a conexão do mesmo a Internet. O aplicativo Android captura a temperatura atual do ambiente e o usuário estabelece os limites máximo e mínimo de temperatura (intervalo em que se encontra a temperatura desejada) que é enviada ao

Firebase Realtime Database. O ESP32 envia para o Firebase a temperatura atual e captura o faixa da temperatura desejada em intervalos de tempo. Ele conectado aos dispositivos controladores das janelas, ventiladores e aquecedores promove o controle de temperatura até o atingimento da temperatura desejada (entre os limites mínimo e máximo) de forma automática.

2 Requisitos

Os requisitos apresentados na tabela abaixo foram divididos em duas partes. A parte 1 e 2 apresentam, respectivamente, os principais requisitos funcionais e não funcionais do sistema de gerenciamento remoto de temperatura.

Requisitos funcionais	
Requisito	Descrição
RF01	configurar a rede de acesso wi-fi do microcontrolador ESP32
RF02	criar o database no firebase
RF03	abrir e fechar automaticamente as janelas
RF04	ligar e desligar automaticamente os ventiladores
RF05	ligar e desligar automaticamente os aquecedores
RF06	mudança do controle automático para manual e vice-versa
Requisitos não-funcionais	
Requisito	Descrição
NF001	usabilidade
NF002	desempenho
NF003	hardware e software

Tabela 1 - requisitos funcionais e não-funcionais do projeto de gerenciamento remoto de temperatura.

3 Tecnologias

A seguir encontram-se breves descrições das tecnologias utilizadas no projeto.

3.1 ESP32



O ESP32 possui comunicação Bluetooth e também acesso a rede via WiFi, com isso é possível desenvolver a partir dele uma gama maior de variedade de projetos IoT. O WiFi suporta uma taxa de 150Mbps e potencia de 20,5 dBm na antena, permitindo e conexão direta à internet. O Bluetooth BLE permite que qualquer celular se conecte a ele enviando diversos tipos de informação com baixo consumo de energia.

Possui um sistema operacional FreeRTOS (sistema operacional de tempo real), e capacidade de atualização de firmware pelo WiFi (Ota – Over The Air) assim permitindo atualizações constantes em seus projetos.

É um módulo muito poderoso tanto em capacidade de processamento quanto em memória (RAM e Flash), e dispensa o uso de qualquer Arduino, pois fonalém de tradicionalmente possuir interface de comunicação, portas GPIO, canais de conversores A-D e sensores internos, é uma placa que integra comunicação Bluetooth e WiFi. É uma plataforma completa para desenvolvimento de projetos tanto simples quanto mais complexos voltados a automações e a IoT.

Características principais

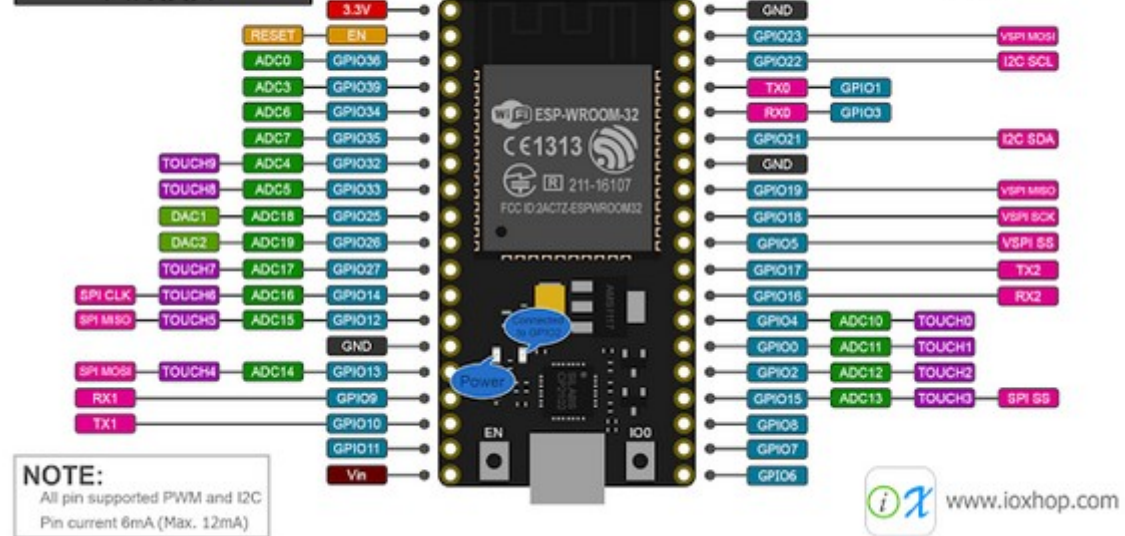


- Chip com WiFi embutido: padrão 802.11 B/G/N, operando na faixa de 2.4 a 2.5GHz
- Modos de operação: Client, Access Point, Station+Access Point
- Microprocessador dual core Tensilica Xtensa 32-bit LX6
- Clock ajustável de 80MHz até 240MHz
- Tensão de operação: 3.3 VDC
- Possui SRAM de 512KB
- Possui ROM de 448KB
- Possui memória flash externa de 32Mb (4 megabytes)
- Corrente máxima por pino é de 12mA (recomenda-se usar 6mA)
- Possui 36 GPIOs
- GPIOs com função PWM / I2C e SPI
- Possui Bluetooth v4.2 BR / EDR e BLE (Bluetooth Low Energy)

Comparativo entre ESP32, ESP8266 e Arduino R3

	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
Clock	160MHz	80MHz	16MHz
WiFi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

NodeMCU-32S PINOUT



3.2 Relê

O **relê** é um [componente](#) eletromecânico que faz a comutação dos contatos quando sua bobina é excitada.

Ele também é utilizado em **dispositivos eletrônicos**, em estágio de saída. Neste caso, é conhecido como relê eletrônico, um dispositivo misto, ou seja, eletromecânico e [eletrônico](#).

Uma característica importante é que ele pode ser energizado com correntes muito pequenas quando comparada com a corrente que o circuito controlado requer para funcionar.

Assim, podemos **controlar** circuitos com **altas correntes** (por exemplo, motores, lâmpadas, máquinas industriais) utilizando dispositivos eletrônicos como transistores, circuitos integrados e fotoresistores, que **fornecem baixas correntes**.

A corrente direta fornecida por um transistor de pequena potência da ordem de 0,1 A não é o bastante para [controlar um motor](#) ou uma lâmpada, porém esta corrente consegue ativar um relê para através deste controlar a carga que possui maior potência.

Este componente isola o circuito de controle do circuito que está sendo controlado, o que dá segurança. **Não há contato elétrico** entre o circuito da bobina e os contatos.

Assim, não existe passagem de corrente elétrica do circuito que ativa o relé para o [circuito](#) que ele controla. Para um circuito controlado de alta tensão, o isolamento é importante, por razões óbvias.

- Composição do relé
- Bobina

Gera o campo magnético indispensável para acionar o circuito magnético. É constituída por um enrolamento de fio de cobre esmaltado.

Suas características são definidas em função do número de espiras, da resistência do fio e do seu diâmetro. A temperatura da [bobina](#) **não** deve ultrapassar:

- 110°C para fios esmaltados com vernizes diarresinos;
- 150°C para fios esmaltados com vernizes sintéticos de acetato de polivinil.

Normalmente a potência da bobina varia entre **0,5 W e 2 W**, atingindo **6 W** nos casos de relés de **potência**. Existem relés sensíveis, com consumo muito inferior.

- Circuito magnético

O circuito magnético é formado por:

- Núcleo no interior da bobina;
- Armadura fixa, que atua como suporte do relé;
- Armadura móvel, que se move pela atração do campo magnético induzido no núcleo. Este movimento causa os movimentos dos contatos.

Estas peças são feitas de ferro doce ou com ligas especiais de alta permeabilidade, para evitar uma magnetização permanente.

- Contatos

Os contatos são os elementos que efetuam a [abertura ou fechamento](#) do circuito.

Modelo Relé 5V- SRD-05VDC-SL-C (utilizado no projeto)

- Tensão de operação : 5 VDC
- Modelo Relé : SRD-05VDC-SL-C ([Datasheet](#))
- Permite controlar cargas de 220V AC
- Nível de sinal dos pinos IN1 e IN2 : 5 VDC
- Corrente de operação : 15 ~ 20 mA
- Tempo de resposta : 5 ~ 10 ms
- 4 furos de 3mm para fixação, nas extremidades da placa
- Dimensões reduzidas : 51 x 38 x 20 mm

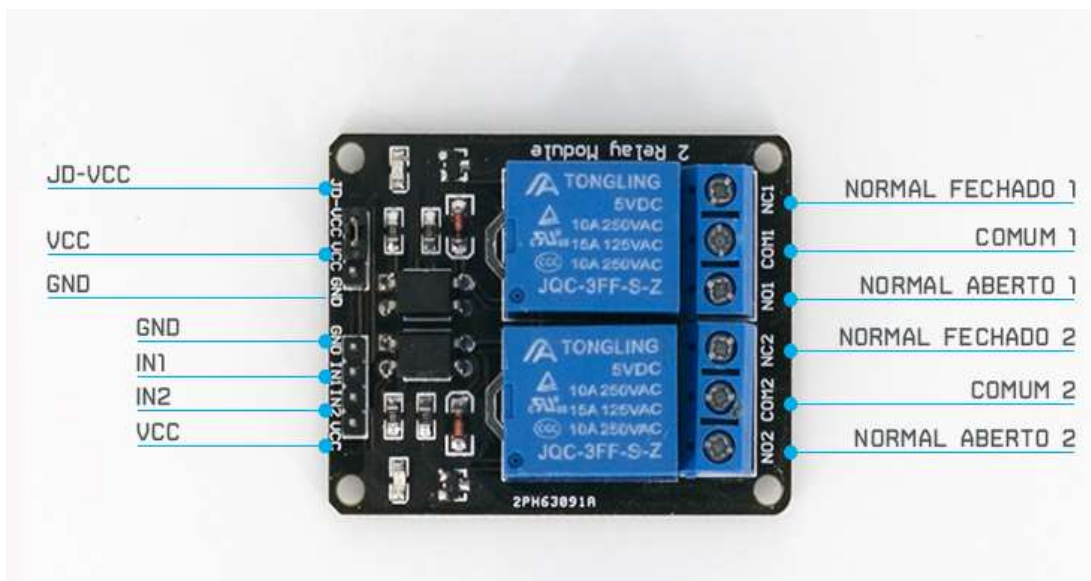


Figura 2-Módulo relê 5V

3.3 DHT11

Os **Sensores DHT** são sensores de baixo custo para aferição de temperatura e umidade de ambientes, que apesar de sua simplicidade, atendem muito bem a necessidade de muitos hobbistas e iniciantes do mundo da eletrônica em projetos de monitoramento das condições do ambiente.

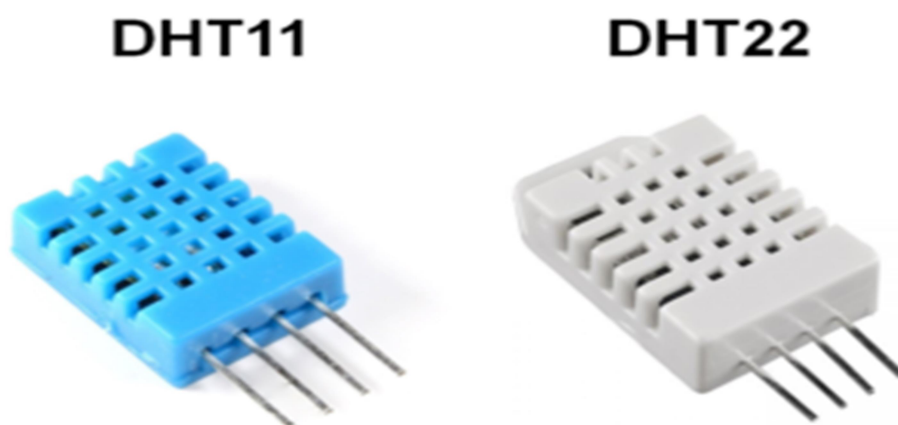


Figura 3- Modelos mais encontrados: Sensor DHT11 e DHT22

	↕ DHT11	↕ DHT22	↕
Alimentação	3V ~ 5V	3V ~ 5V	
Temperatura (Mín. / Máx.)	0°C / +50°C	-40°C / +125°C	
Precisão de temperatura	±2°C	±0,5°C	
Umidade (Mín. / Máx.)	20% / 80%	0% / 100%	
Precisão de umidade	0.05	2% ~ 5%	
Razão de leitura	1 Hz	2 Hz	
Dimensões (C x L x A)	5,5mm x 12mm x 15,5mm	7,7mm x 15,1mm x 25,1mm	
Quantidade de pinos	4	4	

Tabela 2 - Características técnicas dos sensores DHT11 e DHT22.

Os sensores DHTs são constituídos de duas partes principais: um sensor de umidade capacitivo e um termistor. As conexões são compostas por 4 pinos identificados de 1 a 4 e possuem as funções identificadas conforme a imagem abaixo:

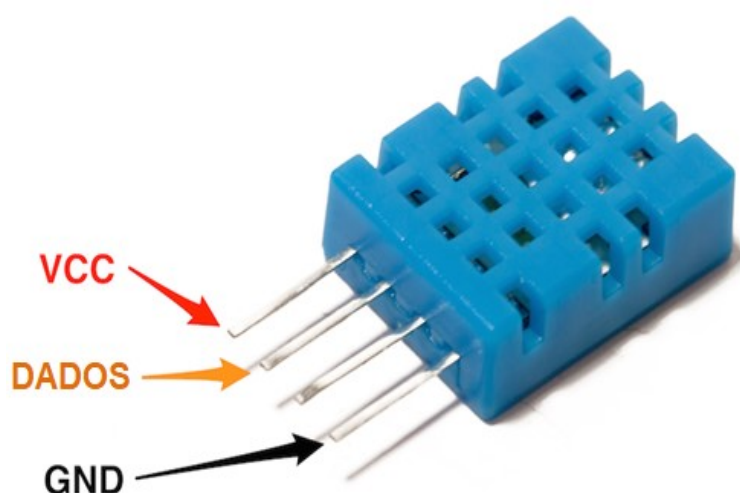


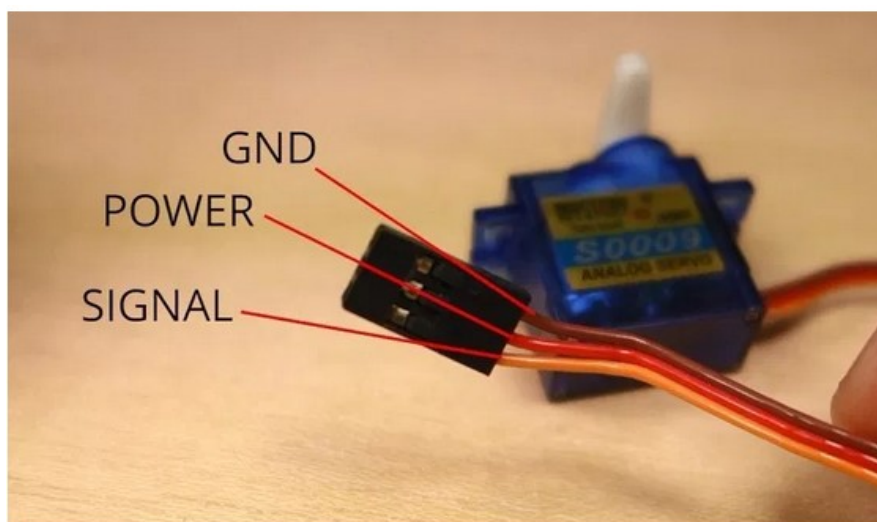
Figura 4-identificação das funções dos pinos do sensor de temperatura DHT11 utilizado no projeto

3.4 Servo-Motor

O eixo do servo-motor pode variar o seu ângulo de 0 a 180°. Ele é controlado utilizando um sinal PWM (pulse width modulation) enviado pelo ESP32. Isto significa que o sinal PWM enviado ao motor determinará a posição do eixo.



Figura 5-servo-motor



Wire	Color
Power	Red
GND	Black, or brown
Signal	Yellow, orange, or white

Figura 6-pinagem do servo-motor.

3.5 Sensor de chuva

Foi utilizado um sensor de chuva para determinar o fechamento da janela caso seja detectado uma ocorrência de chuva.

Utiliza-se interrupções externas para detectar quando a chuva cai sobre o sensor, sem haver a necessidade de polarizá-lo constantemente para verificar o seu estado. Dessa forma, possibilita deixar o processamento do ESP32 livre para outras atividades e somente interagir quando necessário.

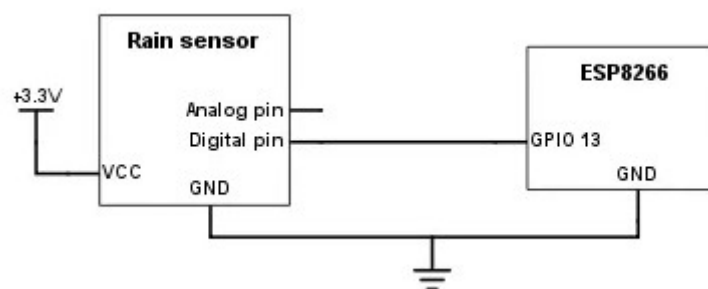


Figura 7-diagrama de conexão entre o sensor de chuva e o ESP32.

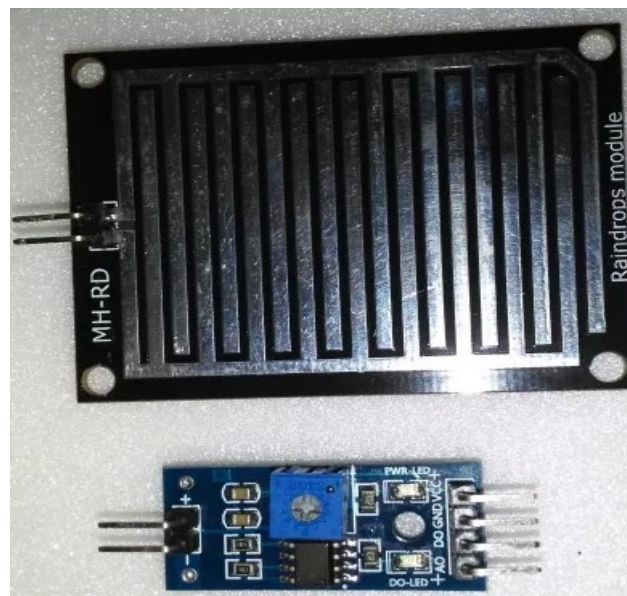


Figura 8-sensor de chuva MH-RD

Basicamente, necessita-se ligar o DC do sensor de chuva e conectar o pino de saída digital para o GPIO do ESP32 para o qual a interrupção será recebida; O pino analógico deverá ser mantido desconectado.

3.6 Wi-fi



Wi-Fi (pronúncia em inglês [/'waɪfaɪ/](#)) é uma [marca registrada](#) da *Wi-Fi Alliance*. É utilizada por produtos certificados que pertencem à classe de dispositivos de rede local sem fios ([WLAN](#)) baseados no padrão [IEEE 802.11](#). O padrão *Wi-Fi* opera em faixas de frequências que não necessitam de licença para instalação e/ou operação. Este fato as torna atrativas. No entanto, para uso comercial no [Brasil](#), é necessária o equipamento ser homologado pela [Agência Nacional de Telecomunicações](#).

[Hotspot Wi-Fi](#) existe para estabelecer ponto de acesso para conexão à internet. O ponto de acesso transmite o sinal sem fios numa pequena distância, geralmente de até 100 metros, mas se a rede for do padrão IEEE 802.11n a distância pode chegar até 300 metros. Quando um periférico que permite *Wi-Fi*, como um *Pocket PC*, encontra um *hotspot*, o periférico pode, na mesma hora, conectar-se à rede sem fio.

3.7 Firebase

O Firebase Realtime Database é uma database NoSQL hospedado na nuvem e que permite armazenamento e sincronização de dados entre os seus usuários em tempo real.

O sincronismo em tempo real permite que todos os usuários acessem seus dados a partir de qualquer dispositivo, web ou mobile, permitem colaboração mútua entre todos.

Permite o desenvolvimento de aplicações sem a necessidade de gerenciamento do servidor o que possibilita a criação de códigos backend que responde aos eventos vinculados aos dados postados na nuvem utilizando as funções cloud para Firebase.

Mesmo quando o usuário está off-line, utiliza-se o cache do dispositivo para armazenar possíveis mudanças e quando o mesmo se conecta à rede, os dados são sincronizados automaticamente.

O Realtime Database integra um sistema de autenticação simples e intuitivo que pode ser utilizado pelo desenvolvedores.

3.8 Android

O Android é uma stack com base em Linux de código aberto criada para diversos dispositivos e fatores de forma. O diagrama abaixo mostra a maioria dos componentes da plataforma Android.



Figura 1. A pilha de software do Android.

A fundação da plataforma Android é o kernel do Linux. Usar o kernel do Linux permite que o Android aproveite os recursos de segurança principais e permitir que os fabricantes dos dispositivos desenvolvam drivers de hardware para um kernel maduro.

Para dispositivos com Android versão superior ao 5.0 (API nível 21) ou mais recente, cada aplicativo executa o próprio processo com uma instância própria do ART-Android Runtime. O ART é projetado para executar várias máquinas virtuais em dispositivos de baixa memória executando arquivos DEX, um formato de bytecode projetado especialmente para o Android, otimizado para utilizar o menor quantidade de memória.

4 Desenvolvimento

Nesta seção são abordados os processos de desenvolvimento do sistema de gerenciamento remoto de temperatura de um ambiente residencial. São eles:

- **Acesso à internet pelo ESP32** - configuração do ESP32 para possibilitar a conexão à internet por meio de uma rede definida pelo usuário
- **Acesso ao Firebase Realtime Database** - desenvolvimento de aplicativo Android a ser utilizado pelo usuário para enviar ao Firebase as temperaturas mínimo e máximo. Tal intervalo que estabelece a faixa em que a temperatura desejada deve-se encontrar.
- **Captação da leitura do sensor de temperatura pelo ESP32** - consiste na montagem de circuito onde são empregados sensor de temperatura DHT11, ESP32, resistor, Arduino IDE, hotspot de forma a estabelecer efetiva comunicação e recebimento de dados pelo ESP32 a partir do sensor de temperatura.
- **Captação da leitura do sensor de chuva pelo ESP32** - consiste na montagem de circuito onde são empregados sensor de chuva, ESP32, Arduino IDE para estabelecer comunicação com a porta digital e informar se está chovendo ou não. Com tal informação, atuar-se-á no fechamento das janelas no situação em que estiver chovendo através de um servo motor.
- **Acionamento de dispositivos por meio de relês** - consiste na conexão de dispositivos, tais como ventoinha ou aquecedores, por meio de relês comandados pelo ESP32. Estes dispositivos serão os intervenientes responsáveis pelo aumento e diminuição da temperatura juntamente com a abertura e fechamento de janela.

5 Dificuldades

As dificuldades encontradas durante o desenvolvimento do projeto

- escolha de site para postar o blog que seja mais flexível e com mais recursos de apresentação.
- prazo restrito haja visto que todos apresentam outras atividades durante o dia.

- falta de conhecimento de como utilizar a comunicação do aplicativo com os sensores por bluetooth.
- indefinição processo de comunicação com a nuvem.
- indefinição do protocolo a ser adotado.
- indefinição do escopo e, conseqüentemente, dos casos de uso dada a diversidade de fatores envolvidos.
- desenvolver solução de conexão webservice, clientservice e cloud MQTT.
- desenvolver solução de conexão bluetooth para o aplicativo Android a ser utilizado.
- reaquadação dos casos de uso e diagramas.
- definir as telas do aplicativo Android e qual software de projeto a ser utilizado.
- como realizar a conexão com o firebase - ESP32 e este acessar as temperaturas mínimas e máximas.
- gravar na memória flash dos ESP32, a rede e senha de acesso a internet.
- informar aos ESP32 da casa, a rede e senha para permitir o acesso a internet.
- preocupação em diminuir o consumo de memória para gravação do código no EPROM/FLASH do ESP32.
- dificuldade no compartilhamento de informações durante o desenvolvimento do projeto.
- dificuldade em reunir todos os integrantes do grupo e conseguir a participação de todos com foco no escopo do projeto.
- indefinição de utilizar um site como webserver ou utilização do firebase.
- após decisão em utilizar o firebase, dificuldade para configuração e acesso ao firebase.
- superado o acesso ao firebase, necessidade do desenvolvimento do aplicativo android com acesso ao firebase.
- dificuldade em gravar na flash memory do ESP32, a rede de acesso a internet, login e senha.
- dificuldade de implementação dos relês controladores de acionamento da ventiladores e aquecedores.
- dificuldade de leitura do sensor de chuva e o seu correto acionamento do servo motor responsável pela abertura e fechamento da janela.
- dificuldade na integração de todos os sistemas desenvolvidos em um único sistema.

6 Conclusão

Após várias dificuldades encontradas durante o desenvolvimento do projeto conseguiu-se atingir com sucesso um sistema de gerenciamento remoto de temperatura de ambiente residencial.

7 Referências

- [1] Baú da Eletrônica-componentes eletrônicos. Disponível : <<https://www.baudaeletronica.com.br/placa-doit-esp32-bluetooth-e-wifi.html>> Acesso em: 15/10/2019.
- [2] Baú da Eletrônica-componentes eletrônicos. Disponível : < <http://blog.baudaeletronica.com.br/dht11-com-arduino/>> Acesso em 10/11/2019
- [3] Eletrônica Aqui. Disponível : <<https://eletronicaqui.com/2016/11/rele/>> Acesso em: 15/10/2019.
- [4] ESP8266 e ESP32 com wifiManager - Disponível : < <https://www.fernandok.com/2018/03/esp8266-e-esp32-com-wifimanager.html/>> Acesso em: 16/10/2019.
- [5] Wi-fi. Disponível : <<https://pt.wikipedia.org/wiki/Wi-Fi>> Acesso em: 15/10/2019.
- [6] FILIPEFLOP. Disponível : <<https://www.filipeflop.com/blog/controle-modulo-rele-arduino/>> Acesso em: 10/11/2019.
- [7] Firebase Realtime Database - Disponível: < <https://firebase.google.com/products/realtime-database/>>. Acesso em 10/11/2019
- [8] ESP32 Servo Motor Web Server with Arduino IDE - Disponível < <https://randomnerdtutorials.com/esp32-servo-motor-web-server-arduino-ide/>>. Acesso em 10/11/2019/
- [9] TECHTUTORIALSX - Disponível < <https://techtutorialsx.com/2016/12/23/esp8266-detecting-rain-drops/>> . Acesso em 10/11/2019.
- [10] Como Programar o Sensor de Chuva - Vídeo #26 - Disponível <<https://www.youtube.com/watch?v=mnPhQ9e0T7E>>. Acesso em 10/11/2019.
- [11] - Disponível <>. Acesso em 10/11/2019.
- [12] - Disponível <>. Acesso em 10/11/2019.
- [13] - Disponível <>. Acesso em 10/11/2019.
- [14] - Disponível <>. Acesso em 10/11/2019.