



INSTITUTO FEDERAL
CIÊNCIA DA COMPUTAÇÃO

FELIPE DOS SANTOS NEVES
PROF. DR. RICARDO CESAR CÂMARA FERRARI

ANÁLISE DE IMPACTO DA SDN NA COMUNICAÇÃO DE DISPOSITIVOS
IOT EM NUVEM

Presidente Epitacio - SP
20/01/2024

SUMÁRIO

1. Resumo	3
2. Introdução	3
3. Infraestrutura de Rede	3
4. Desenvolvimento da Aplicação	3
4.1. Protótipo	3
4.2. Implementação	4
4.3. Bibliotecas	5
4.4. Testes	6
5. Coleta de Dados	6
6. Discussão Análise dos resultados	6
7. Conclusão	6

1. Resumo

2. Introdução

A presente pesquisa concentra-se na análise do impacto da tecnologia SDN (Software Defined Network) na comunicação de dispositivos IoT (Internet of Things) em ambientes de nuvem. O desenvolvimento desse estudo surge da crescente presença da Internet das Coisas em nosso cotidiano, conectando uma variedade de dispositivos à internet. A ideia central nasceu da necessidade de compreender como a SDN pode influenciar a eficiência e confiabilidade da comunicação entre dispositivos IoT e serviços em nuvem.

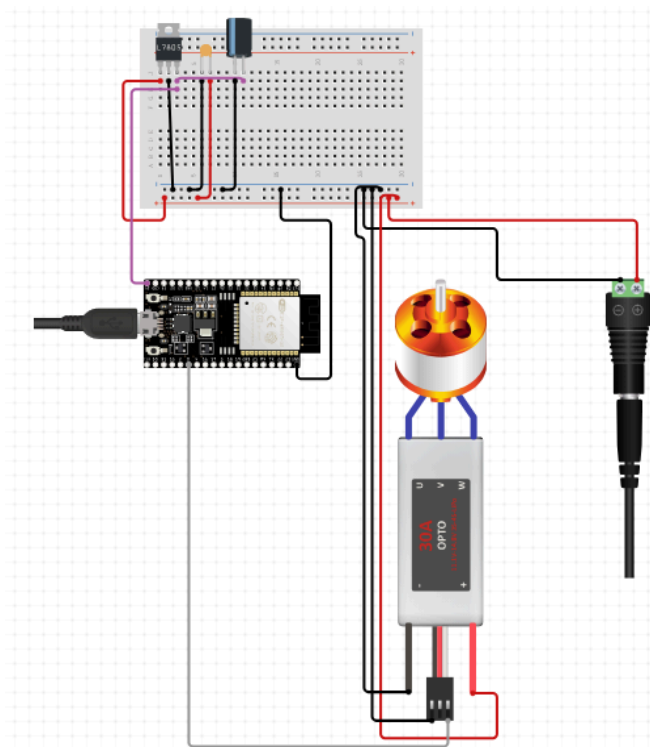
3. Infraestrutura de Rede

4. Desenvolvimento da Aplicação

4.1. Protótipo

Antes da implementação completa do sistema, foi desenvolvido um protótipo do circuito utilizando, como mostra a figura 1:

Figura 1: Protótipo.



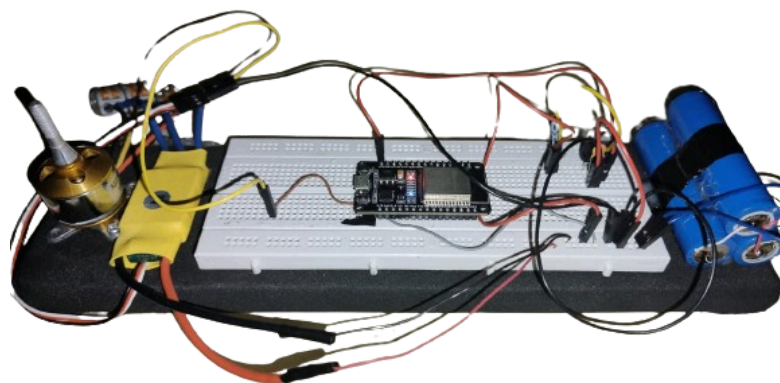
Para isto os seguintes componentes foram utilizados:

- BreadBoard: Utilizada para facilitar a montagem e conexão dos componentes.
- ESP32 - DevKitV1: Placa de desenvolvimento que integra o módulo ESP32, responsável pela comunicação e controle do sistema.
- A2212 Brushless Motor 1000KV (with 30A ESC): Motor brushless utilizado como componente central no projeto.
- Fonte de alimentação para o sistema.
- Voltage Regulator 5v: Regulador de tensão para fornecer uma alimentação estável ao ESP32.
- Capacitor Ceramic 100nF e Electrolytic Capacitor - 1uF/50V: Capacitores para estabilização da alimentação.
- Baterias (Opcional): Bateria como uma fonte de energia alternativa a fonte de alimentação.

4.2. Implementação

Com base no protótipo, o sistema foi implementado de acordo com o diagrama inicial, como mostra na figura 2.

Figura 2: Equipamento montado.



Pinos utilizados no Esp32 DevkitV1:

- VIN (Pino 5V): O uso do pino VIN é adequado para conectar a fonte de alimentação de 12VDC diretamente ao ESP32 DevKit V1. Isso é necessário para garantir uma alimentação estável e suficiente para todos os componentes, especialmente para o motor brushless, que pode demandar uma corrente significativa.
- GND (Pino de Terra): A conexão ao pino GND é essencial para criar uma referência comum para todos os componentes do circuito. Isso assegura uma operação confiável e evita potenciais problemas de referência entre os dispositivos.
- D2 (Pino Digital 2): A escolha do Pino Digital 2 para o controle do motor brushless se deve à sua capacidade de gerar sinais PWM. A modulação por largura de pulso é fundamental para controlar a velocidade do motor de forma eficaz. Ao utilizar esse pino, conseguimos ajustar a largura do pulso enviado ao motor, regulando assim a velocidade de rotação.

O ESP32 foi programado para controlar o motor brushless, estabelecer a conexão com a plataforma Adafruit IO e receber comandos remotos.

4.3. Bibliotecas

As bibliotecas essenciais para o funcionamento do código foram:

"`Arduino.h`": Biblioteca padrão do Arduino, necessária para a programação do ESP32.

"`<ESP32Servo.h>`": Biblioteca para controlar servomotores, essencial para o controle do motor brushless.

"`<AdafruitIO_WiFi.h>`": Biblioteca que permite a comunicação do ESP32 com a plataforma Adafruit IO.

O código segue no seguinte repositório: [código](#)

4.4. Testes

Durante os testes, o motor brushless foi conectado com sucesso à feed criada no Adafruit IO, denominada "potenciômetro". Essa feed incorpora um componente slider, simulando um potenciômetro virtual, cujo valor varia de 0 a 1023. Essa faixa de valores é escolhida para refletir a amplitude de leitura típica de um potenciômetro analógico, sendo 0 a posição mínima e 1023 a posição máxima. Essa configuração permite uma correspondência direta entre os valores do potenciômetro virtual e a velocidade do motor brushless.

Após a configuração bem-sucedida, o motor foi acionado remotamente com sucesso, respondendo aos comandos enviados pela plataforma Adafruit IO. Esse teste confirma a eficácia da comunicação entre o sistema IoT e a nuvem, estabelecendo uma base sólida para futuras análises de desempenho em ambientes SDN.

5. Coleta de Dados

6. Discussão Análise dos resultados

7. Conclusão