UNIVERSIDADE FEDERAL DO MARANHÃO – CIDADE UNIVERSITÁRIA

BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

SISTEMAS DISTRIBUÍDOS

LUIS FELIPE FERREIRA SOARES

# MONITORAMENTO E CONTROLE COM O ESP8266 E MQTT

Este documento descreve a execução do trabalho proposto.

## Descrição do projeto

Este trabalho se destina a projetar um sistema embarcado no microcontrolador ESP8266 que executa as tarefas de monitoramento e controle da rede elétrica de uma dependência qualquer. Em particular, deseja-se monitorar com o intuito de otimizar o consumo de energia elétrica, utilizando um sensor de presença PIR no monitoramento e na automação da tomada de decisão, quando assim for desejado. O monitoramento e controle é feito através de uma aplicação externa, que recebe e transmite mensagens utilizando o protocolo MQTT.

No projeto, foram consideradas algumas premissas:

**- Acesso à internet via WiFi**: A troca de mensagens é feita através da internet. O micro ESP8266 foi escolhido devido a facilidade de acesso à internet através da WiFi;

**- Dependência minimamente movimentada**: Devido à natureza do sensor PIR, assume-se que os presentes na dependência não estejam completamente estáticos em intervalos de tempo definidos, permitindo o pleno funcionamento do sensor.

**- Dependência de tamanho limitado**: Embora seja possível criar uma rede com mais de sensor para uma mesma dependência, a capacidade de um único sensor é relativamente limitada (seção esférica de raio 7m e ângulo 120°). Assim, para dependências maiores, a abordagem utilizando este sensor se torna imprecisa.

## Implementação

Inicialmente, o microcontrolador estabelece a conexão com a internet e com o broker MQTT, sendo estas ações prioritárias em relação a qualquer outra função, ou seja, antes do envio/recebimento de mensagens remotas, o micro deve estabelecer, e também reestabelecer, a comunicação com o broker.

O microcontrolador é responsável pela detecção e controle físico dos estados pertinentes a dependência (estados das luzes, ar condicionado, presença de pessoas, etc.), sendo estas informações obtidas através de sensores (de luz, de presença, etc.) e controladas por chaves (digitais ou relés). Estas informações são publicadas em um tópico destinado este propósito (DATA) quando ocorre a mudança de leitura dos estados dos sensores ou quando solicitado por uma aplicação externa (MASTER), um dashboard, por exemplo. De modo análogo, os estados dos dispositivos eletroeletrônicos associados podem ser controlados por aplicações externas. As aplicações externas devem assinar o tópico de dados (CONTROL) e publicar no tópico relacionado ao microcontrolador de uma determinada dependência.

### DA COMUNICAÇÃO

Uma vez configurado com os dados de autenticação da rede WiFi, o ESP8266 gerencia automaticamente a conexão e a reconexão com a internet. Para a utilização do protocolo MQTT, um broker é necessário. Para tanto, foi utilizado o broker público (iot.eclipse.org), que também suporta websockets. Antes de qualquer comunicação, é verificada a conexão com o broker, tentando a reestabelecer quando perdida.

### DOS ESTADOS DA DEPENDÊNCIA

Cada dispositivo eletroeletrônico tem um estado associado a variável monitorada. Estas variáveis são capturadas/controladas através de sensores/atuadores:

**- Lâmpadas**: Atuador/Relé;

- **Ar Condicionado**: Atuador/Relé;

- **Detector de Presença**: Sensor/PIR.

Na detecção/controle dos estados, são publicadas mensagens que contém as seguintes informações: ID da estação (0-255), comando (GET ou SET), dispositivo/variável (0-255), estado da variável (0-255, mas tipicamente, 0 ou 1). Além destas variáveis, existe uma variável determinada ao controle automático da dependência, que também pode ser controlada por aplicações externas. Comandos válidos são respondidos com um eco, validando sua execução, enquanto que comandos que não estejam dentro da faixa de possibilidades são rejeitados e uma mensagem de erro é enviada para o remetente.

### CÓDIGO DO PROJETO

O código do microcontrolador foi escrito em linguagem C, utilizando os recursos da Arduino IDE. Um exemplo de aplicação WEB (dashboard) foi escrita em Javascript, utilizando websockets. Link: <https://github.com/luisfelipe-fs/MONITORAMENTO_MQTT>.