

Localizador de Objetos com ESP32 Utilizando Bluetooth Low Energy (BLE) e Monitoramento IoT via MQTT

Bruno G. Viero¹, Filipe Z. da Silva¹, João V. Spezia¹

¹Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC)
Centro de Ciências Tecnológicas
Rua Paulo Malschitzki, nº 200, Zona Industrial Norte, Joinville/SC - Brasil. CEP: 89219-710

bruno.viero@edu.udesc.br, filipe.silva01@edu.udesc.br, joao.spezia@edu.udesc.br

Abstract. This article presents the development of an object locator based on the ESP32 microcontroller using Bluetooth Low Energy (BLE) and IoT monitoring via the MQTT protocol. The system detects nearby BLE devices, estimates their distance through RSSI analysis, and transmits the data to an online MQTT broker for real-time visualization. Results demonstrate the approach's effectiveness for short-range localization and highlight its low-cost and energy-efficient characteristics.

Keywords: BLE, ESP32, RSSI, MQTT, IoT

Resumo. Este artigo apresenta o desenvolvimento de um localizador de objetos baseado no microcontrolador ESP32 utilizando Bluetooth Low Energy (BLE) e monitoramento IoT via MQTT. O sistema identifica dispositivos BLE próximos, estima a distância por meio do RSSI e envia os dados a um broker MQTT para visualização em tempo real. Os resultados demonstram a eficácia da solução para localização em curtas distâncias, destacando seu baixo custo e eficiência energética.

Palavras-chave: BLE, ESP32, RSSI, MQTT, IoT

1. Introdução e Motivação

Tecnologias de comunicação sem fio de baixo consumo energético, como o Bluetooth Low Energy (BLE), são cruciais em aplicações de Internet das Coisas (IoT) devido à sua eficiência, baixo custo e baixo consumo de bateria. O BLE é amplamente adotado em wearables, sensores e dispositivos portáteis. A necessidade de localizar objetos em curtas distâncias é crescente, abrangendo contextos como segurança pessoal, rastreamento de equipamentos e organização indoor.

Neste cenário, o microcontrolador ESP32 surge como uma solução acessível e eficiente para a captura de sinais BLE. O presente trabalho se propõe a desenvolver um sistema que:

- Identifique objetos dentro do alcance do ESP32;
- Estime suas distâncias;
- Disponibilize essas informações pela Internet utilizando o protocolo MQTT.

O projeto também cumpre a demanda da disciplina de Redes de Computadores, integrando comunicação sem fio, monitoramento IoT e tratamento de sinais RSSI (Received Signal Strength Indicator).

2. Trabalhos Relacionados

A literatura demonstra diversas iniciativas que empregam o BLE para localização em curtas distâncias. Estudos como o de [Lira et al. 2019] propõem localizadores baseados em BLE com monitoramento via MQTT, confirmando o uso eficiente do RSSI para estimar distâncias em ambientes indoor.

Apesar de fatores como obstáculos físicos, multipercorso e interferência eletromagnética serem limitações conhecidas do uso do RSSI, a simplicidade e a disponibilidade da técnica a tornam relevante para aplicações de proximidade em sistemas IoT. O diferencial do presente trabalho reside na integração completa do processo: captura BLE, processamento local no ESP32, publicação MQTT, exibição dos dados em dashboard online e monitoramento energético do dispositivo.

3. Abordagem Proposta

A solução desenvolvida é um sistema de localização de objetos de baixo custo, implementado em um microcontrolador ESP32 programado com a linguagem C++ (via framework Arduino). O sistema opera em três etapas principais, além do monitoramento de bateria.

3.1. Hardware e Configuração Inicial

O sistema utiliza um ESP32 com conectividade Wi-Fi e BLE. A aplicação é codificada em C++ para Arduino.

Código de Configuração (Setup) O setup do código realiza as seguintes ações:

1. Inicializa a comunicação serial;
2. Configura o LED onboard (pino 2) para sinalização;
3. Conecta-se à rede Wi-Fi (ssid e senha definidos);
 - SSID: Teste ESP32
 - Senha: teste123
4. Configura o cliente MQTT para o servidor test.mosquitto.org na porta 1883
5. Inicializa o BLE (`BLEDevice::init()`).
6. Obtém uma instância do scanner BLE (`pBLEScan`) e configura a classe `MyAdvertisedDeviceCallbacks` para lidar com os resultados das varreduras.
7. Define o scanner como ativo (`pBLEScan->setActiveScan(true)`).

3.2. Captura dos Sinais BLE e Processamento de Dados

O ESP32 realiza varreduras contínuas de dispositivos BLE no ambiente, com duração de 5 segundos (`scanTime`) por ciclo. A classe `MyAdvertisedDeviceCallbacks` é o coração da captura, sendo acionada para cada dispositivo que está anunciando (`advertisedDevice`) e que é detectado.

Para cada dispositivo identificado, o callback executa os seguintes passos:

- Identificação: Coleta o endereço MAC e o nome do dispositivo (quando disponível);
- Captura o valor do RSSI (Received Signal Strength Indicator);

- Estima a distância aproximada (d) em metros utilizando o RSSI (R) em dBm e um modelo empírico de perda de sinal (Lei de Potência), com base no valor de TxPower (P_0) a 1 metro e o fator de propagação (n).

$$d = 10^{\frac{P_0 - R}{10 \cdot n}}$$

O código utiliza $P_0 = -70$ e $n = 4.0$, conforme a função `calcularDistancia(int rssi)`:

```

1 float calcularDistancia(int rssi) {
2     int txPower = -70; // RSSI esperado a 1 metro
3     float n = 4.0; // fator de propagação
4
5     if (rssi == 0) return -1.0;
6     float exponent = (txPower - rssi) / (10.0 * n);
7     return pow(10.0, exponent);
8 }
```

Estruturação JSON: Os dados coletados são organizados em uma estrutura JSON:

- nome, mac, rssi, distancia_m (dados do objeto BLE).
- ipv4_esp32, porta_esp32 (dados de rede do ESP32).
- consumo_W (monitoramento de energia).

3.3. Envio das Informações via MQTT

A cada detecção de um dispositivo BLE, a string JSON é publicada no broker público (`test.mosquitto.org`) no tópico `esp32/localizadorBLE_FILIPE`.

```

1 client.publish(mqtt_topic, buffer);
```

Uma aplicação frontend (web ou desktop) deve se inscrever neste tópico para receber e exibir os dados em tempo real. A função `reconnect()` garante que o ESP32 tente restabelecer a conexão MQTT caso ela caia.

3.4. Monitoramento de Bateria e Conectividade

O sistema implementa a função `calcularConsumo()` para estimar o consumo de energia em Watts (W) do ESP32, baseando-se em tempos e correntes de operação estimados:

- Corrente BLE Ativo: 40 mA
- Corrente IDLE: 10 mA
- Tensão (Estimada): 3.3 V
- Duração Scanner (30% do tempo): 1.5 s
- Duração IDLE (70% do tempo): 3.5 s

O consumo de energia total é calculado como:

$$\text{Consumo}(W) = \text{Tensão} \cdot ((\text{Corrente}_{BLE} \cdot \text{Duração}_{Scanner}) + (\text{Corrente}_{IDLE} \cdot \text{Duração}_{IDLE}))$$

Além disso, a função `reconectarWiFi()` verifica e tenta restabelecer a conexão Wi-Fi, garantindo a continuidade do envio dos dados via MQTT.

4. Resultados Finais

Os testes iniciais foram conduzidos utilizando fones de ouvido Bluetooth como dispositivos-alvo, simulando um cenário de localização de objetos de uso pessoal. Durante os experimentos:

- O ESP32 identificou corretamente os fones de ouvido Bluetooth e outros dispositivos BLE em seu alcance;
- A estimativa de distância baseada em RSSI apresentou variação esperada devido a reflexões e interferências, mas permaneceu dentro de um intervalo funcional para localização aproximada;
- As mensagens MQTT foram recebidas em tempo real sem perdas significativas;
- O broker MQTT exibiu corretamente número de dispositivos, MAC, nome, RSSI, distância e dados de rede.

A aplicação atendeu integralmente aos requisitos do projeto.

5. Conclusões

Este trabalho resultou no desenvolvimento de um localizador de objetos eficiente, utilizando a combinação de ESP32, BLE e MQTT. O sistema provou ser capaz de identificar dispositivos BLE próximos e transmitir as informações de forma simples e confiável.

O uso do RSSI para estimativa de distância, apesar das limitações conhecidas, foi adequado para as estimativas aproximadas necessárias ao projeto. A integração com o protocolo MQTT permitiu o monitoramento remoto em tempo real, destacando o potencial do sistema em aplicações IoT de baixo custo.

Como trabalhos futuros, sugere-se:

- Aplicação de filtros de suavização do RSSI (como filtros de Kalman ou média móvel) para melhorar a precisão da distância;
- Uso de múltiplos ESP32 para implementar técnicas de triangulação, oferecendo maior precisão na localização;
- Criação de um dashboard web mais completo.

Referências

- Lira, F. A., Junior, F. L. C., do Nascimento, E. J. F., Juca, S. C. S., and Junior, J. N. M. (2019). Localizador de objetos em curtas distâncias baseado em bluetooth ble com monitoramento iot via mqtt. In *Anais da 7ª Escola Regional de Computação Aplicada à Saúde (ERCAS)*, pages 109–114, Porto Alegre. Sociedade Brasileira de Computação.