

Localizador de Objetos em Ambientes Internos Utilizando BLE e ESP32

Lucas de Andrade Martins
Luiza Farias
Otávio da Silva Gonçalves

Redes de Computadores
Professora Janine Kniess

01/07/2024

Sumário

1	Introdução.....	3
2	Motivação.....	4
3	Trabalhos Relacionados.....	5
4	Ferramentas e Linguagem Utilizadas.....	6
5	Abordagem Proposta.....	7
5.1	Cálculo da Distância e Filtragem.....	7
5.2	Integração de Hardware e Software.....	8
5.2.1	Figura 1.....	9
5.2.2	Figura 2.....	10
5.2.3	Figura 3.....	11
5.2.4	Figura 4.....	12
5.2.5	Figuras 5 e 6.....	13
5.3	Testes em Ambientes Reais.....	14
6	Resultados Finais.....	15
6.1	Tabela 1.....	15
6.1.1	Figura 7.....	16
6.1.2	Figura 8.....	17
6.1.3	Figura 9.....	18
6.1.4	Figura 10.....	19
7	Considerações Finais.....	20
8	Referências Bibliográficas.....	21

■ Introdução

- **Problema:** Perda de objetos em ambientes internos.
- **Solução:** Sistema de localização usando Bluetooth Low Energy (BLE).
- **BLE:** Tecnologia de comunicação sem fio de baixo consumo.
- **Beacons BLE:** Dispositivos que transmitem sinais para estimar a distância.
- **Objetivo:** Criar um sistema acessível e eficiente para localizar objetos.

Motivação

- Combinação de BLE com MQTT para localização precisa e confiável.
- Integração com dispositivos móveis e plataformas IoT.
- Contribuição para pesquisas na área de localização interna.
- Aplicação em diversos cenários, como logística, saúde, varejo e segurança.

■ Trabalhos Relacionados

- **Mekki et al. (2019):** Sistema de posicionamento interno baseado em BLE e MQTT.
- **Wang et al. (2015):** Sistema de localização baseado em proximidade usando RSSI do BLE.
- **Lira et al. (2019):** Localizador de objetos em curtas distâncias baseado em BLE e MQTT.
- **Venkatesh et al. (2021):** Uso de filtros para melhorar a precisão da localização baseada em RSSI.

Ferramentas e Linguagem Utilizadas

- Microcontrolador ESP32 com Wi-Fi e Bluetooth integrados.
- Protocolo MQTT para comunicação e monitoramento em tempo real.
- Fones de ouvido Galaxy Buds+ como tags BLE.
- Pulseira inteligente Mi Band 3 como receptor.
- Linguagem C/C++ para programação do ESP32.
- Arduino IDE para desenvolvimento.

■ Abordagem

Abordagem Proposta: Cálculo da Distância e Filtragem

- **Métrica Chave:** RSSI (*Received Signal Strength Indicator*)
- **Unidade de Medida:** Decibéis (dBm)
- **Modelo de Cálculo:**
 - Distância (d) = $10^{\left(\frac{\text{rssi_ref} - \text{RSSI}}{10 * N}\right)}$
 - N: Constante ambiental (2 a 4)
 - rssi_ref: RSSI a 1 metro do transmissor (média do RSSI)
- **Filtro de Média:**
 - Suavização dos valores de RSSI
 - Remoção de ruídos
 - Melhora da precisão na estimativa de distância

Abordagem

Abordagem Proposta: Integração de Hardware e Software

- **Configuração do Hardware:** Dispositivos Bluetooth BLE (Galaxy Buds+, Mi Band 3)
- **Servidor MQTT:**
 - Recebe dados de RSSI
 - Calcula a localização dos objetos
 - Envia resposta ao usuário
- **Implantação do Software:**
 - Dispositivos móveis
 - Servidor central
- **Código:** Base para comunicação e processamento de dados
- **Componentes do sistema:**
 - **Número da primeira porta do cliente:** 54745
 - **IP do ESP32:** 192.168.170.193.
 - **IP do servidor que está enviando a resposta MQTT:** 192.168.170.239.
 - **Número da porta programado no ESP32 para enviar ao MQTT:** 1883.

Abordagem

Figura 1: Configuração da Conexão Wi-Fi.

```
void connectWiFi() {  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  Serial.println("");  
  Serial.println("WiFi conectado");  
  Serial.println("Endereço IP: ");  
  Serial.println(WiFi.localIP());  
}
```

Fonte: Os Autores, 2024.

Abordagem

Figura 2: Configuração do Cliente MQTT

```
void connectMQTT() {  
  while (!mqttClient.connected()) {  
    if (mqttClient.connect(mqttClientId, mqttUsername, mqttPassword)) {  
      Serial.println("Conectado ao servidor MQTT");  
    } else {  
      Serial.print("Falha na conexão MQTT, rc=");  
      Serial.print(mqttClient.state());  
      Serial.println(" tentando novamente em 5 segundos");  
      delay(5000);  
    }  
  }  
}
```

Fonte: Os Autores, 2024.

Abordagem

Figura 3: Configuração do Scanner Bluetooth

```
void scanBLE() {  
    BLEScan* pBLEScan = BLEDevice::getScan();  
    pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());  
    pBLEScan->setActiveScan(true);  
    BLEScanResults foundDevices = pBLEScan->start(scanTime);  
    Serial.print("Dispositivos encontrados: ");  
    Serial.println(foundDevices.getCount());  
    Serial.println("Escaneamento concluído");  
}
```

Fonte: Os Autores, 2024.

Abordagem

Figura 4: Classe *MyAdvertisedDeviceCallbacks*

```
class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
    void onResult(BLEAdvertisedDevice advertisedDevice) {
        String foundDevices = advertisedDevice.getAddress().toString().c_str();
        if (foundDevices == "dd:f8:44:df:fa:ad") { // Se for o dispositivo autorizado
            Serial.println("Dispositivo IDENTIFICADO!");
            Serial.print("Nome do dispositivo: ");
            Serial.println(advertisedDevice.getName().c_str());
            int rssi = advertisedDevice.getRSSI();
            Serial.print("RSSI: ");
            Serial.println(rssi);

            // Aplicar filtro de média no RSSI
            int filteredRSSI = meanFilter(rssi);

            // Cálculo da distância usando a fórmula
            double distance = pow(10, ((levelRSSI - filteredRSSI) / (10.0 * N)));

            String payload = "{\"Nome do dispositivo\": \"" + String(advertisedDevice.getName().c_str()) +
                            "\", \"MAC\": \"" + foundDevices +
                            "\", \"RSSI\": " + String(rssi) +
                            "\", \"Distancia\": " + String(distance, 2) + " metros\"}"; // Distância com 2 casas decimais e unidade de medida

            // Enviar mensagem MQTT apenas se o RSSI filtrado não for zero
            if (filteredRSSI != 0) {
                mqttClient.publish(mqttTopic, payload.c_str(), true);
            }
        }
    }
};
```

Fonte: Os Autores, 2024.

Abordagem

Figura 5: Filtro de Média.

```
int meanFilter(int newValue) {  
    // Subtrai o valor mais antigo do buffer da soma  
    rssiSum = rssiSum - rssiBuffer[rssiIndex] + newValue;  
  
    // Adiciona o novo valor ao buffer  
    rssiBuffer[rssiIndex] = newValue;  
  
    // Avança para o próximo índice circular  
    rssiIndex = (rssiIndex + 1) % windowSize;  
  
    // Retorna a média  
    return rssiSum / windowSize;  
}
```

Fonte: Os Autores, 2024.

Figura 6: Cálculo da distância com base no RSSI filtrado.

```
double distance = pow(10, ((levelRSSI - filteredRSSI) / (10.0 * N)));
```

Fonte: Os Autores, 2024.

Abordagem

Abordagem Proposta: Testes em Ambientes Reais

- **Teste de Conectividade Wi-Fi:** Verificar se o ESP32 se conecta corretamente à rede Wi-Fi e mantém a conexão durante a operação.
- **Teste de Conectividade MQTT:** Garantir que o cliente MQTT se conecte ao broker e publique os dados corretamente no tópico especificado.
- **Teste de Varredura Bluetooth:** Validar a capacidade do scanner BLE de detectar dispositivos autorizados e medir os valores de RSSI.
- **Validação dos Filtros de RSSI:** Testar a eficácia do filtro de média em diferentes condições para assegurar a precisão dos valores de RSSI e, consequentemente, das estimativas de distância.
- **Teste de Publicação de Dados:** Verificar se os dados de RSSI filtrados e as distâncias calculadas são publicados corretamente no tópico MQTT.

Resultados Finais

Tabela 1: Resultados Finais

Distância	RSSI Médio Filtrado (dBm)	Distância Calculada	Observações
50 cm	-65	50cm \pm 5cm	Detecção rápida e precisa, variações mínimas no RSSI
3 m	-88	3m \pm 70cm	Precisão aceitável, erro dentro dos limites toleráveis
4 m	-95	4m \pm 2cm	Excelente precisão, erro mínimo
6 m	-88	6m \pm 10cm	Maior variação no RSSI, menor precisão, mas dados utilizáveis

Fonte: Os Autores, 2024.

Resultados Finais

Figura 7: Teste feito a aproximadamente 50 centímetros de distância do receptor ESP32



Fonte: Os Autores, 2024.

Resultados Finais

Figura 8: Teste feito a aproximadamente 3 metros de distância do receptor ESP32



Fonte: Os Autores, 2024.

Resultados Finais

Figura 9: Teste feito a aproximadamente 4 metros de distância do receptor ESP32.



Fonte: Os Autores, 2024.

Resultados Finais

Figura 10: Teste feito a aproximadamente 6 metros de distância do receptor ESP32.

```
{"Nome do dispositivo": "Mi Band 3", "MAC": "dd:f8:44:df:fa:ad", "RSSI": -88, "Distancia": 6.31 metros}
topico/scan

{"Nome do dispositivo": "Mi Band 3", "MAC": "dd:f8:44:df:fa:ad", "RSSI": -88, "Distancia": 6.31 metros}
topico/scan

{"Nome do dispositivo": "Mi Band 3", "MAC": "dd:f8:44:df:fa:ad", "RSSI": -88, "Distancia": 6.31 metros}
topico/scan

{"Nome do dispositivo": "Mi Band 3", "MAC": "dd:f8:44:df:fa:ad", "RSSI": -89, "Distancia": 5.62 metros}
topico/scan
```

Fonte: Os Autores, 2024.

■ Considerações Finais

- Resultados promissores: viabilidade e eficiência da solução demonstradas.
- Desempenho variável em precisão devido a fatores externos.
- Viabilidade do ESP32 destacada pela versatilidade.
- Futuras melhorias: otimização de código e implementação de bancos de dados locais.
- Expansão das funcionalidades: integração com assistentes virtuais e uso de IA.

Referências

ESPRESSIF. Documentação ESP32 Series. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 17 jun. 2024.

IoT MQTT Panel. Site Oficial. Disponível em: <https://snrlab.in/>. Acesso em: 26 jun. 2024.

K. MEKKI, E. BAJIC AND F. MEYER. **Indoor Positioning System for IoT Device based on BLE Technology and MQTT Protocol**. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, 2019, pp. 787-792, doi: 10.1109/WF-IoT.2019.8767287.

LIRA, Filipe Almeida; C. JUNIOR, Francisco L.; DO NASCIMENTO, Erik J. F.; JUCA, Sandro C. S.; M. JÚNIOR, Jose N. **Localizador de objetos em curtas distâncias baseado em Bluetooth BLE com monitoramento IoT via MQTT**. In: ESCOLA REGIONAL DE COMPUTAÇÃO APLICADA À SAÚDE (ERCAS), 7., 2019, Teresina. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 109-114.

MOSQUITTO. Documentação. Disponível em: <https://mosquitto.org/documentation/>. Acesso em: 10 jun. 2024.

SILVA, L. R. B. Da. (2014) **Método para aferição de distância entre nós sensores baseados em RSSI**. Luiz Rodolfo Barreto da Silva- Campinas: PUC-Campinas. 121p. 22.ed.CDD – t621.385.

VENKATESH R., MITTAL V., TAMMANA H. **Indoor Localization in BLE using Mean and Median Filtered RSSI Values**. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2021, pp. 227-234, doi: 10.1109/ICOEI51242.2021.9453000.

WANG, Y.; Ye, Q.; CHENG, J.; and WANG, L. (2015). **Rssi-based bluetooth indoor localization**. In 2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN), pages 165–171. IEEE. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7420939/> . Acesso em: 30 jun. 2024.



Obrigado

**UDESC – Universidade do Estado de
Santa Catarina**

contato.udesc@gmail.com

www.udesc.br

www.facebook.com/udesc

(48) 3664-8000

Rua Madre Benvenuta, 2007, Itacorubi
Florianópolis - SC
CEP 88035-901