Título do Tutorial

Neste tutorial, vamos desenvolver um sistema de localização em tempo real para equipamentos médicos, como cadeiras de rodas e desfibriladores, usando RFID ou Bluetooth Low Energy (BLE) com gateways IoT e uma interface de mapa interativo. Esse projeto é ideal para hospitais e clínicas que buscam otimizar a logística interna, melhorando o tempo de resposta em emergências e assegurando a disponibilidade de equipamentos essenciais.

Índice

1. Introdução
2. Requisitos
3. Configuração do Ambiente
4. Montagem do Circuito
5. Programação
6. Teste e Validação
7. Expansões e Melhorias
8. Referências
9.

Introdução

Este projeto foca na criação de um sistema de rastreamento para equipamentos médicos dentro de unidades hospitalares, aumentando a eficiência na gestão de dispositivos críticos e agilizando o atendimento a pacientes em situações de emergência. Equipamentos como desfibriladores e cadeiras de rodas serão etiquetados com dispositivos RFID ou BLE, permitindo que suas localizações sejam monitoradas em tempo real e visualizadas em uma interface de mapa interativo.

Requisitos

Hardware

- **Gateways IoT**: Raspberry Pi ou ESP32 com conectividade Wi-Fi ou Ethernet.
- **Etiquetas de Rastreamento**: Dispositivos RFID ou BLE (como Estimote beacons).
- **Leitores RFID (se aplicável)**: Antenas instaladas em locais estratégicos.

Software

- **Python** (para programar o gateway e comunicação).
- **Node.js ou Flask** (para criar a API de comunicação).
- **DynamoDB ou MySQL** (para o armazenamento de dados).
- **Mapbox ou Leaflet.js** (para o mapa interativo).

Configuração do Ambiente

Passo 1: Instalação do Software

1. **Configurar a Placa do Gateway**: Instale o sistema operacional no Raspberry Pi e configure os pacotes necessários para comunicação (Python, MQTT, ou HTTP).
2. **Instalar Software e Dependências**:

- `pip install paho-mqtt requests` (para Python, caso use MQTT para enviar dados).

Passo 2: Configuração das Placas

1. **Instalação de Etiquetas nos Equipamentos**: Fixe etiquetas RFID ou BLE nos equipamentos médicos.
2. **Distribuição de Gateways**: Instale os gateways nos pontos estratégicos do ambiente, garantindo que todas as áreas cobertas pelos dispositivos estejam conectadas e capazes de enviar dados.

Montagem do Circuito

Conecte o leitor RFID ao Arduino:

- **SDA** → Pino Digital 10
- **SCK** → Pino Digital 13
- **MOSI** → Pino Digital 11
- **MISO** → Pino Digital 12
- **IRQ** → Não conectado
- **GND** → GND
- **RST** → Pino Digital 9
- **VCC** → 3.3V

Programação

Passo 1: Configuração do Gateway:

Programe o Raspberry Pi para escutar etiquetas RFID/BLE e enviar as leituras ao servidor.

Passo 2: Servidor em Nuvem:

Configure a plataforma de IoT para armazenar dados recebidos e fornecê-los à interface do mapa interativo.
Exemplo do código em C no Arduino:

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN 9
#define SS_PIN 10

MFRC522 rfid(SS_PIN, RST_PIN);

void setup() {
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  Serial.println("RFID Scanner pronto.");
}

void loop() {
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
    return;

  Serial.print("UID: ");
  for (byte i = 0; i < rfid.uid.size; i++) {
    Serial.print(rfid.uid.uidByte[i], HEX);
    Serial.print(" ");
  }
  Serial.println();
  delay(1000);
}
```

Exemplo do código em C no ESP32:

```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>

BLEScan* pBLEScan;

void setup() {
  Serial.begin(115200);
  BLEDevice::init("");
  pBLEScan = BLEDevice::getScan();
  pBLEScan->setActiveScan(true);
}

void loop() {
```

```
  BLEScanResults scanResults = pBLEScan->start(5);
  for (int i = 0; i < scanResults.getCount(); i++) {
    Serial.println(scanResults.getDevice(i).toString().c_str());
  }
  delay(5000);
}
```

Teste e Validação

- **Teste de Cobertura e Alcance**: Teste os gateways em várias áreas para garantir a cobertura e a precisão da localização.
- **Teste de Interface**: Confirme que a interface de mapa reflete as localizações em tempo real.

Expansões e Melhorias

- **Notificações em Tempo Real**: Configurar notificações via SMS ou email caso um dispositivo crítico se mova para uma área inesperada.
- **Análise de Uso de Equipamentos**: Use análises para entender a frequência de uso dos dispositivos e otimizar a distribuição de equipamentos.

Referências

- Documentação de BLE Beacons e RFID
- Exemplos de uso de Leaflet.js e Mapbox para mapas interativos
- Tutoriais de configuração MQTT para Raspberry Pi
