

Documentação do Código: ESP32 como Servidor BLE

Este documento descreve o código de exemplo para configurar o ESP32 como um servidor BLE (Bluetooth Low Energy).

O servidor BLE no ESP32 oferece um serviço e uma característica, permitindo que dispositivos clientes (como smartphones ou outros dispositivos BLE) interajam com ele.

1. Bibliotecas Importadas:

O código utiliza as seguintes bibliotecas para configurar o servidor BLE no ESP32:

- `BLEDevice.h`: Fornece as funções principais para o funcionamento do BLE.
- `BLEServer.h`: Utilizada para criar e gerenciar o servidor BLE.
- `BLEUtils.h`: Contém funções úteis para trabalhar com BLE.
- `BLE2902.h`: Usada para adicionar descritores à característica BLE.

2. Definição dos UUIDs:

O código utiliza dois UUIDs:

- `**SERVICE_UUID**`: O UUID do serviço BLE que o ESP32 oferece. Este é um identificador único para o serviço.
- `**CHARACTERISTIC_UUID**`: O UUID da característica que o cliente pode ler ou escrever.

Exemplo:

- `SERVICE_UUID = "12345678-1234-5678-1234-56789abcdef0"`
- `CHARACTERISTIC_UUID = "abcdef01-1234-5678-1234-56789abcdef0"`

3. Funções Importantes:

a. `MyServerCallbacks`:

Esta classe contém os callbacks para o servidor BLE. Ela define o que acontece quando um cliente se conecta ou se desconecta:

- `onConnect`: Acionado quando um cliente se conecta ao servidor BLE.
- `onDisconnect`: Acionado quando um cliente se desconecta.

b. `MyCharacteristicCallbacks`:

Esta classe contém os callbacks para a característica BLE. Ela define o que acontece quando o cliente escreve na característica:

- `onWrite`: Acionado quando o cliente escreve um valor na característica. O valor é lido e exibido no monitor serial como um array de bytes.

4. Configuração do Servidor BLE:

a. Inicialização do BLE:

A função `BLEDevice::init("ESP32-Server")` inicializa o dispositivo BLE com o nome "ESP32-Server".

b. Criação do Servidor:

Um servidor BLE é criado com `BLEDevice::createServer()`.

c. Criação do Serviço:

O serviço BLE é criado com o UUID especificado em `SERVICE_UUID` através de `pServer->createService(SERVICE_UUID)`.

d. Criação da Característica:

A característica é criada com o UUID `CHARACTERISTIC_UUID`. A característica é configurada

para leitura e escrita, o que permite que o cliente leia e escreva nela.

e. Configuração Inicial:

Um valor inicial (um array de bytes) é configurado para a característica.

f. Publicidade:

A publicidade do dispositivo é iniciada com ``pAdvertising->start()``, permitindo que dispositivos clientes descubram o servidor BLE.

5. Fluxo do Código:

1. O servidor BLE aguarda a conexão de um cliente.
2. Quando um cliente se conecta, ele pode ler e escrever na característica.
3. A característica envia ou recebe dados do cliente conforme os comandos enviados.
4. O código também exibe no monitor serial os dados recebidos.

Exemplo de Valores Configurados:

- Valor inicial de ``pCharacteristic``: "Hello" (em hexadecimal: 0x48, 0x65, 0x6C, 0x6C, 0x6F).

Conclusão:

Este código configura o ESP32 como um servidor BLE simples. Ele é capaz de enviar dados para um cliente, bem como receber e processar dados enviados pelo cliente. É um ponto de partida para projetos mais complexos que envolvem comunicação BLE.