BLE GATEWAY NETWORK

Dec 20, 2023

Name: Divyam Chandak
Email: divyam.22110804@viit.ac.in
College: VIIT, Pune

Name: Sudarshan Ingale
Email: sudarshan.22110852@viit.ac.in
College: VIIT, Pune

Name: Sakshi Aru
Email: sakshi.22110521@viit.ac.in
College: VIIT, Pune

Tables Of Contents

INTRODUCTION

- **Project aim:** Exploring potential of BLE technology by establishing a connection between: a **Raspberry Pi** and a **Mobile Phone.**

- Bluetooth Low Energy (BLE) technology provides wireless connections between devices, while minimizing power consumption.

- Transfering of data, such as files, sensor readings, or control commands.

- The project leverages the benefits of BLE, including low energy consumption and ease of use, making it suitable for applications in various domains, including IoT (Internet of Things), home automation, healthcare, and more.

OBJECTIVES

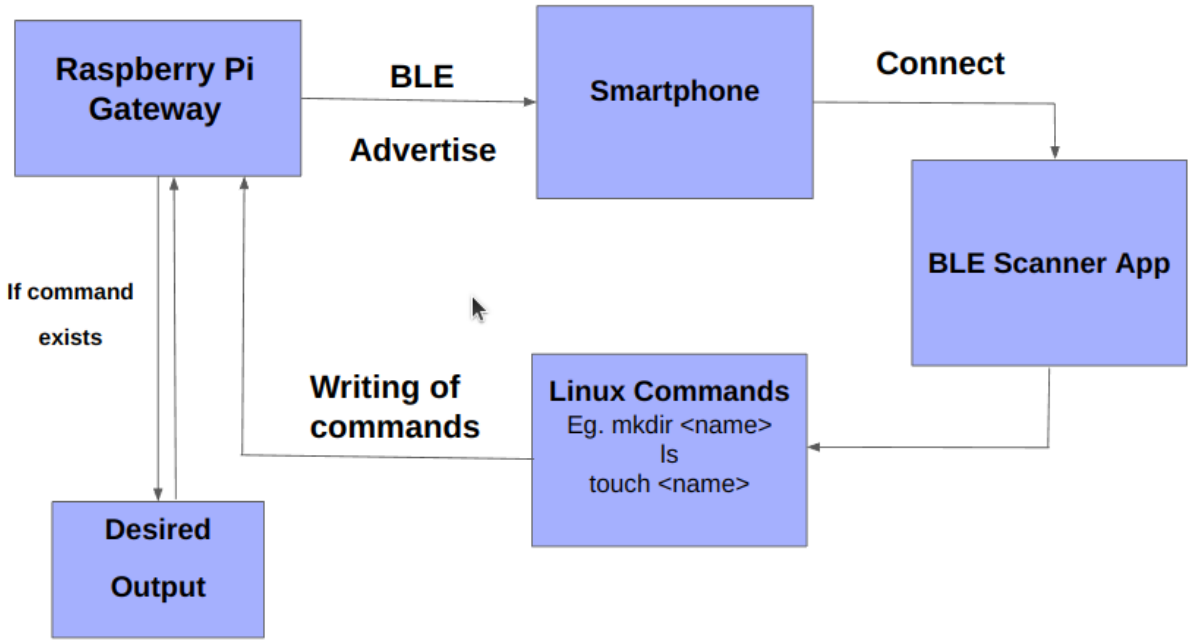
- **Understanding BLE Technology**: Understanding of Bluetooth Low Energy (BLE) technology, its principles, protocols, and advantages over classic Bluetooth.

- **Hardware and Software Specification:** Getting Familiarize with the hardware components (Raspberry Pi with BLE capabilities) and software tools (programming languages, libraries, frameworks) needed for BLE communication.

- **Implementing Device Communication:** Establish a functional communication link between the Raspberry Pi and a mobile phone using BLE. This involves configuring the Raspberry Pi as a BLE peripheral and the mobile phone as a central device or vice versa.

- **Data Exchange and Interaction:** Enable data exchange between the devices.

BLOCK DIAGRAM



HARDWARE SPECIFICATION

Raspberry Pi 4 B+

- **Bluetooth and BLE Support:** Ensure the Raspberry Pi's Bluetooth chipset supports BLE. Verify compatibility with the BlueZ stack (commonly used for Bluetooth communication on Linux-based systems) or other suitable BLE libraries.

- **Processor and RAM:** Adequate processing power and RAM are crucial for smooth operation and multitasking. A more powerful CPU and sufficient RAM can handle communication tasks effectively.

- **Wireless Connectivity:** Besides BLE, consider Wi-Fi capabilities for internet connectivity if the gateway needs access to online services or cloud platforms.

SOFTWARE SPECIFICATIONS

- **Network Connectivity:**
  Ensure the Raspberry Pi is connected to the network, either through Ethernet or Wi-Fi, for remote access.

- **SSH Server:**
  Ensure that the OpenSSH server is installed and enabled on the Raspberry Pi to allow remote SSH connections.

- **Remote Development Extension:**
  Install the "Remote - SSH" extension in Visual Studio Code on the Raspberry Pi. This extension allows you to work on a remote machine using SSH.

- **Coding and Development:**
  Edit, create, and debug code in VS Code on your local machine while the changes are applied and executed on the Raspberry Pi remotely.

- **Libraries used in Python Code:**
  - Bluez Peripheral
    (sub libraries- bluez_peripheral -gatt.service, bluez_peripheral.gatt.characteristic, bluez_peripheral.advert, bluez_peripheral..agent),
  - subprocess
  - import asyncio

LIBRARIES USED

1. bluez_peripheral:

Description: BlueZ is the official Linux Bluetooth stack. The bluez_peripheral module likely contains functionalities or classes related to creating a Bluetooth peripheral device using the BlueZ stack in Python.

Functions:

- *gatt.service*: Handles Bluetooth GATT (Generic Attribute Profile) services, allowing you to define services offered by the Bluetooth peripheral.
- *gatt.characteristic*: Deals with characteristics within GATT services, specifying the data attributes and properties.
- *advert*: Involves advertising functionality, enabling the peripheral to broadcast its presence to other devices.
- *agent*: Likely handles agent functionalities for interactions, such as authentication and pairing, between devices.

2. subprocess:

Description: The subprocess module in Python provides functionalities for spawning new processes, connecting to their input/output/error pipes, and obtaining their return codes.

Functions:

- *subprocess.run()*: Executes a command in a subprocess.
- *subprocess.Popen()*: Opens a process and returns a Popen object,allowing manipulation of the subprocess.

3. asyncio:

Description: The asyncio module in Python provides infrastructure for writing single-threaded concurrent code using coroutines, multiplexing I/O access, and handling asynchronous I/O operations.

Functions:

- *asyncio.create_task()*: Initiates a coroutine and schedules its execution as a task.
- *asyncio.run()*: Runs the top-level entry point for an asyncio application.

SOURCE CODE

Git: https://github.com/Divyam1202/BLE_Gateway_Network.git

REFERENCES LINK

- https://en.m.wikipedia.org/wiki/Bluetooth
- https://devzone.nordicsemi.com/guides/short-range-guides/b/bluetooth-low-energy/posts/ble-characteristics-a-beginners-tutorial
- https://learn.adafruit.com/introduction-to-bluetooth-low-energy/gatt
- https://www.datacamp.com/tutorial/python-subprocess