International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Enhancing IOT Connectivity: Leveraging Modern Web Development with Serial Port Communication

[1]Krishna Adhikari, [2]Dr. Kamalraj R

[1],[2] Department of CS & IT, JAIN(Deemed-To-Be-University), Bangalore, India
[1] adh.krishna23@gmail.com, [2]r.kamalraj@jainuniversity.acin

ABSTRACT:

Integrating web development with IoT is of great significance as web development is used in creating a wide variety of applications and services that enhance the functionality, accessibility, and control of IoT devices. This paper presents a modern web development approach for IoT through Serial Port Communication. The combination of web development and IoT creates new opportunities to connect and oversee smart devices online, allowing users to remotely monitor, control, and engage with their IoT setups. This paper delves into how we can build a user-friendly web interface for IoT systems using contemporary programming languages and frameworks. It aims to provide practical insights and solutions for making IoT accessible through the web. The proposed methodology uses modern languages like TypeScript and frameworks such as Angular and NestJS to create our web interface for IoT. RESTful APIs facilitate communication between the frontend and backend. Additionally, the use of a PostgreSQL database and TypeORM for seamless connectivity is proposed. Serial port communication is recommended for establishing connectivity with IoT devices. This method facilitates direct and efficient data exchange, serving as a reliable bridge between the web interface and IoT devices, ensuring seamless communication and control. Within this research, the advantages and disadvantages of employing these contemporary methods, in contrast to older approaches are also discussed, to gain valuable insights.

Keywords—Web Development, Typescript, Angular,Internet of Things(IoT)

I. INTRODUCTION

The Internet of Things (IoT) is the network of connected devices on the internet, sharing data and services with each other. It is a combination of sensors and actuators which provides informations in digitized form and is able to transmit all the data over a network which can be further used by a lot of services and final users.[1]

There is communication between the devices connected in the IoT and the data acquired from it is stored in a database. If the data is of large size the storage is done in the cloud. In general, IoT relies on sensors and actuators. Sensors gather information from the surroundings, like measuring temperature or detecting motion, while actuators carry out actions based on that data, such as adjusting a thermostat or turning on lights. Some of the examples of IoT devices are smart home systems, wearable fitness trackers and smart watches, etc.

Web development is the development of fast and efficient web apps that can be accessed over the web. There are two aspects of web development: Frontend and the backend.

Frontend is the user interface of a website or app where the user interacts with the screen when visiting a website or an app. Backend on the other hand is the behind the scenes part where the data is stored, accessed and processed. Server, database and server side logic are present here which make sure everything runs smoothly.

In IoT, web development is the development of web apps that serve various purposes like backend integration, data storage in a database, user interface, data visualization, etc. Web development and iot are intertwined closely as the developers create web apps for IoT devices making it accessible and user friendly for different purposes. In modern web development, JavaScript is the primary language of choice due to its versatility and widespread adoption. The use of different frameworks like angular have also evolved as these frameworks provide structured methodologies and powerful tools to streamline development processes. So use of such powerful tools and frameworks for development of IoT interfaces will have a lot of benefits.

Typescript and Angular:

Typescript and Angular are two of the most important tools in Modern web development, frontend development in particular. Typescript is a superset of javascript that is developed and maintained by Microsoft. Angular on the other hand is a popular open-source front-end framework which facilitates building modern web applications. Together these can be used for building modern, scalable, and maintainable web applications.

NestJS:

NestJs is an up-and-coming framework for building server-side applications which usesTypescript and Node.js. It is particularly used for building web applications and APIs and is designed to provide a solid architectural foundation and a set of powerful tools to streamline the development process. It has some key features and concepts which has made it popular in the web development community like: Modular and Decorator-Based Architecture, Dependency Injection, Controllers and Routes, Middleware, etc.

TypeOrm and PostGreSQL

TypeOrm is an Object-Relational Mapping(ORM) library for TypeScript. TypeORM makes it easier to interact with databases using objects instead of writing SQL. Objects are used to represent data so there is no worry about the underlying database structure. PostgreSQL is a powerful database that is often used with TypeORM. It is a relational database management system which is used with TypeOrm which provides several benefits, such as Type-Safe Database Interactions, Object-Oriented Approach, Database Portability, Productivity and query building in a programmatic and type-safe way.

RESTFUL API

RESTful APIs stands for Representational State Transfer which is used for designing web services. It is based on the concept of resources, which are identified by unique URLs, and interactions with these resources are performed using standard HTTP methods.[3] Some of the standard HTTP methods are GET, POST, PUT, PATCH, etc.RESTful APIs offer a number of benefits such as simplicity, flexibility, scalability, performance, interoperability, etc.

Serial Port Communication

Serial port communication was invented in the 1960s by the Electronic Industries Association (EIA). The first serial port standard, RS-232, was published in 1962. RS-232 is still widely used today, but there are also other serial port standards, such

as RS-485 and RS-422.[1] It is a method used for data exchange between devices and is widely used in embedded systems and IoT applications. The communication can be made directly without the use of a network when using a wired connection like a serial port.The data in the serial port is transmitted in a continuous stream of bits, and devices interact using standardized protocols and baud rates. Serial port communication is still popular and extensively used due to the advantages such as reliability, low overhead, and ease of implementation.

II. Methodology:

Using TypeScript, NestJS, and Angular together is a powerful stack for building full-stack web applications. Typeorm and Postgresql are used for the database. The detailed process is shown in the figure below:

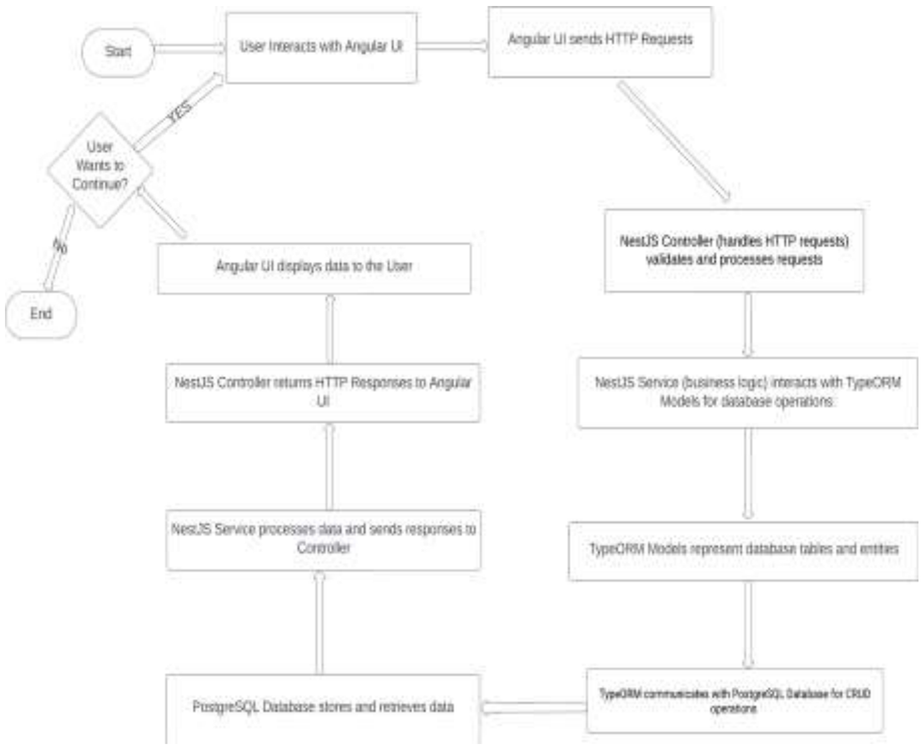
Start
User Interacts with Angular UI
Angular UI sends HTTP Requests
YES
User Wants to Continue?
No
End
Angular UI displays data to the User
NestJS Controller (handles HTTP requests) validates and processes requests
NestJS Controller returns HTTP Responses to Angular UI
NestJS Service (business logic) interacts with TypeORM Models for database operations
NestJS Service processes data and sends responses to Controller
TypeORM Models represent database tables and entities
PostgreSQL Database stores and retrieves data
TypeORM communicates with PostgreSQL Database for CRUD operations

fig: Flowchart of proposed methodology

- Angular is used for creating user interfaces.
- HTTP request is sent to the backend from the frontend.
- Now the backend server built upon NestJS will receive the request.
- The RestAPI methods are used for
- The data is accessed from the backend database.
- The service and controller of NestJS works to return the data.
- CRUD operations are performed.
- The output is sent back to the user when the user sends a request.

Iot Architecture:

Architects and engineers are always trying to optimize frameworks to meet the demands of the modern Internet of `Things (IoT) environment. Even if they were once efficient, traditional three-layer architectures are no longer able to keep up with the ever-increasing demands of today's dynamic IoT scenarios. A four-layer architecture is therefore suggested as a full solution to these problems. This creative method separates the Internet of Things ecosystem into four distinct layers, each of which is essential to the smooth operation of IoT systems: application, middleware, network, and perception.

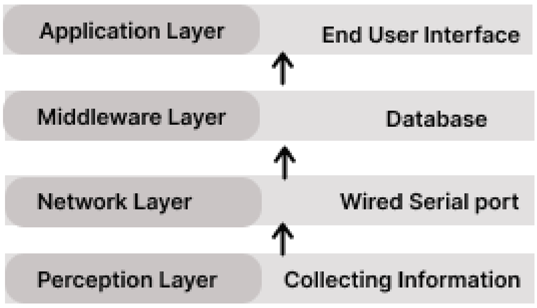

fig: Proposed Iot Architecture [7]

1. Application Layer: End User Interface:

The Application layer, which acts as the interface between end users and the extensive network of networked devices, is at the forefront of the suggested architecture. By offering user-friendly interfaces, guaranteeing accessibility, and enabling smooth interaction with IoT applications, this layer places a high priority on the user experience. Individuals can easily monitor, control, and interpret data provided by the Internet of Things ecosystem thanks to advanced user interfaces.

2. Middleware Layer: Database Integration:

The middleware layer, which manages data processing, storage, and connection between various IoT devices, serves as the framework of the architecture. The use of strong databases, which facilitate effective data administration and analysis, is essential to this layer. The architecture guarantees the smooth flow of information by putting sophisticated middleware solutions into place, which enables real-time decision-making and improved overall system performance.

3. Network Layer: Wired Serial Port Integration

The network layer emphasizes the incorporation of wired serial ports for improved security and dependability while concentrating on data transmission between Internet of Things devices. This layer recognizes that in order to serve the IoT ecosystem's communication needs, a reliable and effective network infrastructure is essential. Using wired serial connections lowers latency, increases data transfer rates, and strengthens the architecture against potential security flaws.

4. Perception Layer: Collecting Information

The central component of the design, the Perception layer, is responsible for obtaining raw data from sensors and other sources. Since it is necessary to collect data about the physical world, this layer serves as the cornerstone of Internet of Things activities. Using advanced sensors and data gathering techniques, the Perception layer makes precise and timely data available, laying the groundwork for higher layers of the architecture to make well-informed decisions.

III. Serial connection:

For serial connection the function 'SerialPort' is used. The path and baudRate which is the transform rate is also specified. After defining the serial port, the ‘setupSerialCommunication’ function is called. The code snippet shown below:

```
 const serialPortPath = "COM8";
  this.serialPort = new SerialPort({ path: serialPortPath, baudRate: 9600 },(error) => {
    if (error) {
      console.error("Error opening serial port:", error);
    } else {
      console.log("Serial port opened successfully.");
      console.log("Time:", new Date());
    }})
    this.setupSerialCommunication();
  }
```

fig: Code snippet for serial port setup

IV. Conclusion:

In conclusion, this research paper has explored the adoption of a technology stack comprising Angular, NestJS, TypeORM, PostgreSQL, and serial port connections for Internet of Things (IoT) development. This stack offers a range of advantages, including end-to-end TypeScript consistency for enhanced code quality, modularity for maintainability, and access to strong ecosystems supporting rapid development. The use of PostgreSQL ensures data integrity in IoT applications, and the integration of serial port connections allows seamless hardware interaction. However, it is vital to acknowledge the challenges inherent in this stack, such as complexity and resource intensiveness, particularly in resource-constrained IoT devices. Furthermore, ongoing maintenance and scalability must be carefully managed to sustain a robust IoT system. Ultimately, the choice to implement this technology stack should be guided by the specific needs and constraints of the IoT project. When utilized judiciously, this stack holds the potential to empower developers in building efficient and reliable IoT solutions

References:

[1] Beal, Vangie. "Serial Port." Webopedia. Webopedia, September 1, 1996. Updated May 24, 2021

[2] Zia, Ahmad Bilal, and Ms Kshamta Chauhan. "A research paper on internet of things based upon smart homes with security risk assessment using OCTAVE Allegro." *Int J Eng Res Technol* 9.06 (2020): 940-948.

[3] Jyoti L. Kanchane, Latika R. Desai. “Web Services in IoT”.(2014)

[4] Salazar Soler, Jorge, and Santiago Silvestre Bergés. "Internet of things." (2017).

[5] Song, Linxuan, and Marisol García-Valls. "Improving security of web servers in critical IoT systems through self-monitoring of vulnerabilities." *Sensors* 22.13 (2022): 5004.

[6] Kumar, Sachin, Prayag Tiwari, and Mikhail Zymbler. "Internet of Things is a revolutionary approach for future technology enhancement: a review." *Journal of Big data* 6.1 (2019): 1-21.

[7] Senthilkumar, S. Poorana, and B. Subramani. "Study on iot architecture, application protocol and energy needs." International Journal of Scientific Research in Network Security and Communication 8.5 (2020): 7-12.

[8] P. Sethi and S. R. Sarangi, “Internet of Things: Architectures, Protocols, and Applications,” J. Electr. Comput. Eng., vol. 2017,2017.

[9] Bogner, Justus, and Manuel Merkel. "To type or not to type? a systematic comparison of the software quality of javascript and typescriptapplications on github." Proceedings of the 19th International Conference on Mining Software Repositories. 2022.

[10] Pham, Huy Quang. "Online Banking Web Application." (2022).

[11] Sabo, Mario. NestJS. Diss. Josip Juraj Strossmayer University of Osijek. Department of Mathematics. Chair of Applied Mathematics. Computer Science Research Group, 2020.