

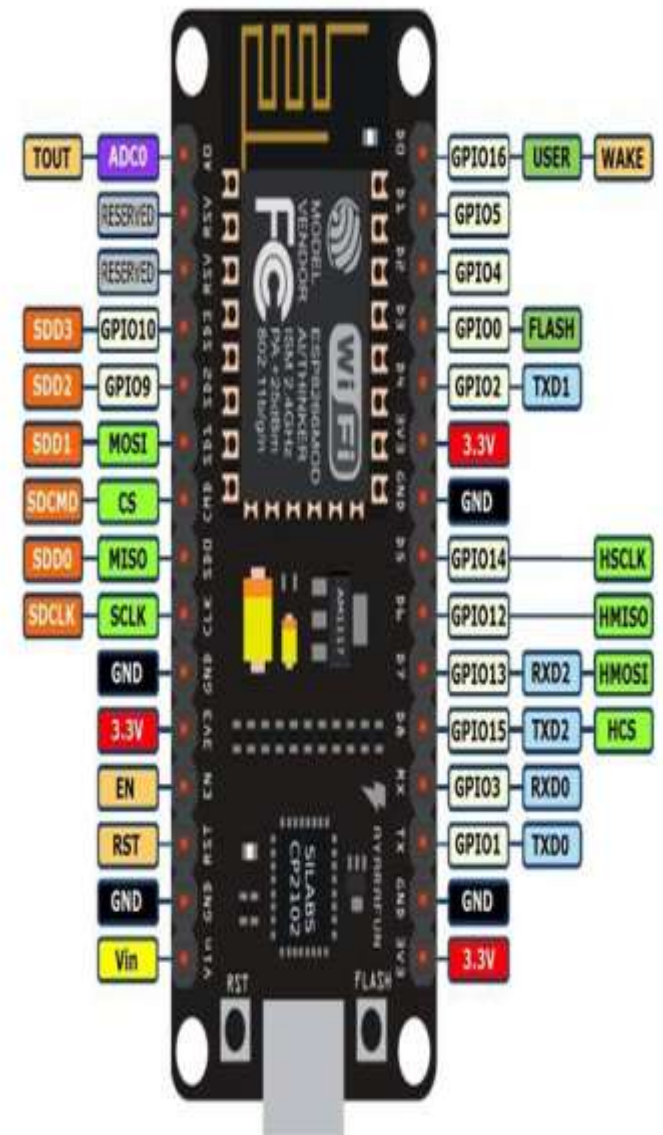
# PHASE-4 (DESIGN PART-2)

- **INTRODUCTION**

- In this phase we are going to design the last 4 levels of the innovation which is described in the phase 2
- This level includes
  - 1.control unit
  - 2.LoRaWAN
  - 3.Database server
  - 4.user interface
- **1.CONTROL UNIT**
  - This contains the combination of arduino sensors,Raspberry pi,ESP8266 and ESP32
  -

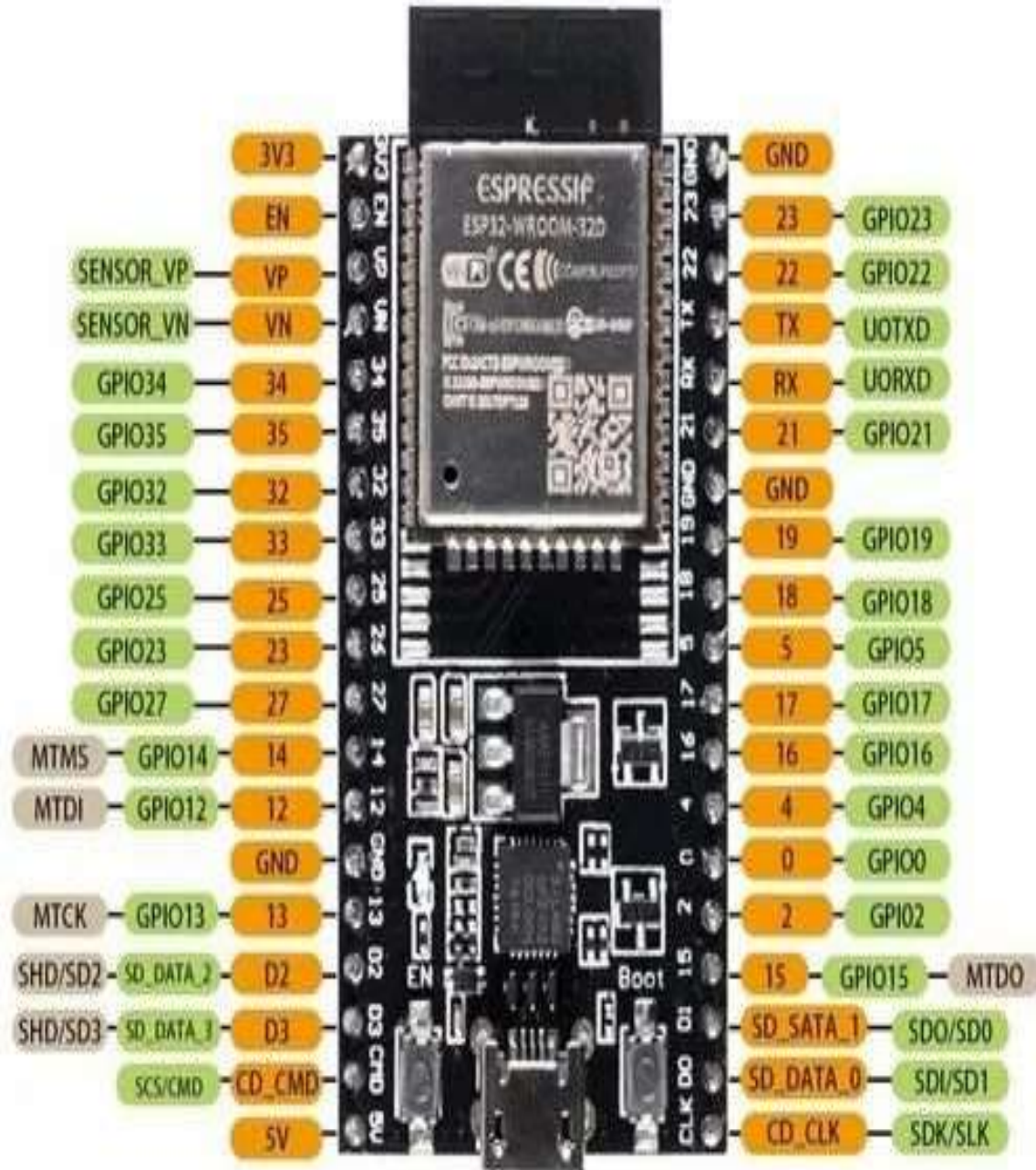
## ESP8266

The ESP8266 is a highly popular and versatile microcontroller module renowned for its low cost and built-in Wi-Fi capabilities. At its core, it features a Tensilica L106 32-bit microcontroller unit (MCU), which, while not the most powerful MCU available, is well-suited for a wide array of Internet of Things (IoT) applications. Its most prominent feature is its Wi-Fi connectivity, supporting 802.11 b/g/n standards, enabling it to seamlessly connect to Wi-Fi networks and the internet. Additionally, it typically includes onboard flash memory for storing firmware and data, making it a self-contained solution. The ESP8266 offers a limited number of general-purpose input/output (GPIO) pins for interfacing with sensors, actuators, and other external devices. It's programmable using various development environments, such as the Arduino IDE, PlatformIO, MicroPython, and the native Espressif SDK, offering flexibility in development options. The ESP8266 boasts a vibrant and supportive community of developers, which means you can readily find a plethora of resources, tutorials, and libraries online. With its relatively low power consumption and affordability, it's a popular choice for hobbyists and makers, as well as for projects involving remote monitoring, DIY IoT devices, and cost-conscious home automation. Different variants of the ESP8266, like the ESP-01 and ESP-12E, provide various form factors and pin configurations to cater to diverse project requirements, making it a staple in the world of IoT development.



## ESP32

The ESP32 is a highly versatile microcontroller and communication module that has gained widespread popularity in the field of electronics and IoT (Internet of Things). Developed by Espressif Systems, the ESP32 is celebrated for its powerful capabilities. It features a dual-core processor, integrated Wi-Fi, and dual-mode Bluetooth, making it a robust choice for a wide range of applications. With an ample number of GPIO (General-Purpose Input/Output) pins, it can interface with various sensors, actuators, and devices, making it suitable for both simple and complex projects. The ESP32's affordability and extensive support from a dedicated community of developers and enthusiasts have further contributed to its acclaim. Its open-source nature and the availability of numerous libraries and resources make it an excellent choice for those looking to create IoT devices, embedded systems, and more. Whether you're a hobbyist, maker, or professional, the ESP32 offers a powerful and cost-effective solution for your microcontroller and wireless communication needs.



**ESP32-WROOM-32D**

# Raspberry Pi

- The Raspberry Pi is a revolutionary single-board computer that has made a significant impact on the world of technology, education, and DIY (Do It Yourself) electronics. Developed by the Raspberry Pi Foundation, this compact and affordable computer has captured the imaginations of students, hobbyists, and professionals alike. Despite its small size, the Raspberry Pi boasts impressive processing power, memory, and versatile connectivity options.
- What makes the Raspberry Pi truly remarkable is its commitment to accessibility and open-source principles. It was created with the goal of making computing affordable and approachable for people of all ages and backgrounds. The Raspberry Pi has found its place in a wide range of applications, from building retro gaming consoles and home automation systems to serving as a foundation for creative and educational projects. Its expansive ecosystem of software, add-ons, and a dedicated community of enthusiasts has further solidified its reputation as a versatile and powerful tool.
- Notably, the Raspberry Pi has played a pivotal role in fostering innovation in areas like IoT (Internet of Things) and digital making, empowering individuals to explore and experiment with technology. With its influence extending from classrooms to remote regions where affordable computing solutions are essential, the Raspberry Pi has become an iconic symbol of accessible and affordable computing, inspiring countless individuals to learn, create, and innovate in the world of technology.

# FUNCTIONS OF CONTROL UNIT

- The main function of control unit is to control all the activities. It is directly connected to the data base server .It reacts according to the responds given by the database server.it is pre programmed by set of instructions and its response is according to the set of instructions .simply it act like a mini computer where it can able to process the commands the very quick and fast .Functions of the control unit is already discussed in the previous phases

## • 2. LoRaWAN

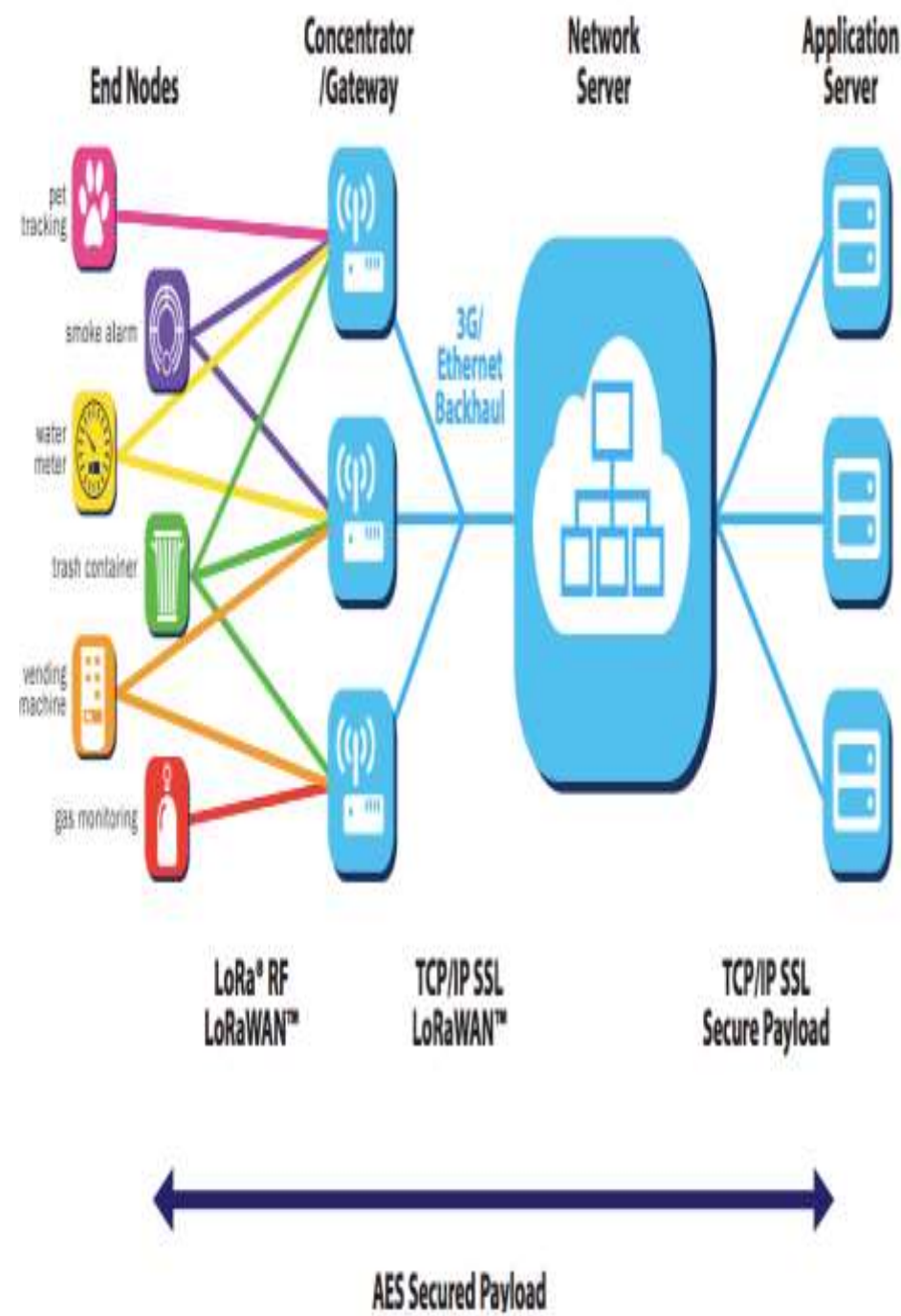
- LoRaWAN (Long Range Wide Area Network) is a wireless communication technology tailored for low-power, long-range IoT connectivity. It employs the LoRa (Long Range) modulation method, enabling data transmission over considerable distances while conserving energy in battery-powered devices. LoRaWAN's strengths lie in its wide area coverage, scalability for numerous devices, and bi-directional communication capabilities. While it's not suitable for high-bandwidth applications, it excels in scenarios requiring long-range coverage for transmitting small amounts of data. Operating in unlicensed ISM bands, LoRaWAN has found applications in smart agriculture, smart cities, asset tracking, and environmental monitoring, among others.



## 2.LoRaWAN

The LoRaWAN (Long Range Wide Area Network) is a wireless communication technology tailored for low-power, long-range IoT connectivity. It employs the LoRa (Long Range) modulation method, enabling data transmission over considerable distances while conserving energy in battery-powered devices. LoRaWAN's strengths lie in its wide area coverage, scalability for numerous devices, and bi-directional communication capabilities. While it's not suitable for high-bandwidth applications, it excels in scenarios requiring long-range coverage for transmitting small amounts of data. Operating in unlicensed ISM bands, LoRaWAN has found applications in smart agriculture, smart cities, asset tracking, and environmental monitoring, among others.

LoRaWAN (Long Range Wide Area Network) is a wireless communication technology designed for low-power, long-range connectivity in the Internet of Things (IoT). It relies on LoRa (Long Range) modulation, enabling the transmission of data over extensive distances while maintaining energy efficiency in battery-powered IoT devices. LoRaWAN's standout features include its wide area coverage, scalability to accommodate a large number of devices, and support for bi-directional communication. This makes it well-suited for applications such as smart agriculture, smart cities, asset tracking, and environmental monitoring. While it is not suitable for high-bandwidth applications, it excels in scenarios where long-range data transmission at low data rates is essential. Operating in unlicensed ISM bands, LoRaWAN eliminates the need for specific licenses, which has contributed to its popularity in various IoT deployments.



# Functions of LoRaWAN

- **Long-Range Communication:** LoRaWAN is specifically designed for long-range communication, making it ideal for applications that require connectivity over several kilometers. This makes it well-suited for scenarios such as agriculture, where sensors need to communicate across large fields, or smart city applications that cover wide areas.
- **Low Power Consumption:** LoRaWAN devices are engineered for low power consumption, enabling battery-powered devices to operate for extended periods without the need for frequent battery replacement or recharging. This is particularly important for remote or hard-to-reach locations.
- **Scalability:** LoRaWAN supports a large number of devices within a single network, making it suitable for applications with thousands or even millions of connected sensors and devices. It can handle massive deployments with ease.
- **Bi-Directional Communication:** Unlike some other IoT technologies, LoRaWAN allows for two-way communication. Devices can both transmit data to a central server and receive commands or data from that server. This bidirectional capability enables more interactive and responsive applications.
- **Low Data Rate:** LoRaWAN is not designed for high-bandwidth applications but rather for scenarios where small amounts of data are periodically transmitted. This low data rate is well-suited for monitoring and control applications in areas like agriculture, environmental sensing, and asset tracking.

It also includes the following

- **Wide Area Coverage:** LoRaWAN is capable of providing coverage over a wide area, and it can penetrate obstacles like buildings and vegetation, which can be advantageous in urban and rural settings.
- **License-Free Spectrum:** LoRaWAN operates in unlicensed ISM bands, which means you don't need to acquire specific licenses to use the technology, reducing the regulatory burden for deploying IoT devices.
- **Security:** LoRaWAN networks typically include security features to protect the integrity and confidentiality of the data being transmitted. Encryption and authentication mechanisms are often implemented to secure communications.
- Where In our application it is needed to provide a fast connection between the device (iot devices) and the data base server with the high data rate (speed of the connection)



### 3.Data Base Server

- A NoSQL (Not Only SQL) database is a type of database management system known for its flexibility and scalability. It diverges from traditional relational databases by accommodating unstructured, semi-structured, and structured data, making it an ideal choice for applications demanding adaptability. NoSQL databases often shine in scenarios that require high data throughput and rapid scalability. They feature schema flexibility, distributed architecture, high availability, and data partitioning capabilities. Common types of NoSQL databases include document stores (e.g., MongoDB), key-value stores (e.g., Redis), column-family stores (e.g., Apache Cassandra), graph databases (e.g., Neo4j), and wide-column stores (e.g., ScyllaDB). They find applications in various use cases, from real-time big data analytics and content management systems to IoT data storage, social networks, and time-series data management. The choice of NoSQL database depends on the specific needs of the application, such as data model and scalability requirements.

- oSQL databases, standing for "Not Only SQL," represent a category of database management systems designed to address the limitations of traditional relational databases. These databases offer several distinctive features and are well-suited for various use cases. They are prized for their schema flexibility, accommodating unstructured, semi-structured, and structured data. Scalability is a key advantage, as NoSQL databases are built to scale horizontally across multiple servers or nodes, making them ideal for high-velocity data workloads and applications requiring flexibility in data models. Many NoSQL databases are distributed, ensuring high availability and fault tolerance, crucial for mission-critical applications

A NoSQL (Not Only SQL) database is a type of database management system that provides a mechanism for storage and retrieval of data that doesn't adhere to the traditional relational database model. NoSQL databases are designed to handle a variety of data types, including unstructured, semi-structured, and structured data. These databases are commonly used in applications and use cases where scalability, flexibility, and high data throughput are essential. Here are some key features and use cases of NoSQL databases:

### **Key Features of NoSQL Databases:**

**Schema Flexibility:** NoSQL databases allow for flexible and dynamic data models, which is useful when dealing with rapidly evolving or semi-structured data.

**Scalability:** NoSQL databases are typically designed to scale out horizontally, making them well-suited for high-velocity and high-volume data workloads.

**High Availability:** Many NoSQL databases are built with features to ensure high availability and fault tolerance, important for mission-critical applications.

**Distributed Architecture:** NoSQL databases are often distributed systems, designed to run across multiple servers or nodes, which helps in achieving scalability and fault tolerance.

**Data Variety:** They can handle a variety of data types, including text, JSON, XML, binary, and more. This makes them suitable for applications with diverse data requirements.

**Data Partitioning:** NoSQL databases can distribute data across multiple servers or clusters, which allows for efficient data partitioning and retrieval.

## 4. User interface

- A user interface (UI) serves as the bridge between users and digital systems, be it computer software, websites, or applications. Its primary objective is to facilitate effective communication and interaction, enabling users to perform tasks, access information, and control the system. UIs can take various forms, including Graphical User Interfaces (GUI) with visual elements like icons and buttons, Text-Based User Interfaces (TUI) relying on text and keyboard input, Web User Interfaces for online experiences, Mobile User Interfaces for smartphones and tablets, Voice User Interfaces (VUI) for voice-activated systems, and even Augmented Reality (AR) and Virtual Reality (VR) interfaces for immersive experiences. Key components encompass menus, buttons, forms, icons, layout, and accessibility features, all designed with the goal of enhancing user satisfaction and efficiency. A well-crafted UI can greatly improve the user experience, while a poorly designed one can lead to frustration and inefficiency, emphasizing the importance of user interface design in software and web development.
- A user interface (UI) is a critical element in the world of digital interactions, serving as the vital point of contact between users and computer systems, software applications, or websites. Its fundamental purpose is to facilitate efficient and intuitive communication, empowering users to access information, execute tasks, and govern the functionality of the system.

Diverse forms of UIs exist, ranging from the familiar Graphical User Interfaces (GUIs), replete with visual components such as icons and buttons, to Text-Based User Interfaces (TUIs), which rely on textual input. Web User Interfaces are designed to enhance online experiences, while Mobile User Interfaces cater to the unique demands of smartphones and tablets. Voice User Interfaces (VUIs) enable hands-free interaction, and Augmented Reality (AR) and Virtual Reality (VR) interfaces create immersive, three-dimensional worlds.

Within these UIs, a multitude of elements and principles come into play. Menu structure options, buttons and controls empower actions, forms and input fields facilitate data entry, and icons and graphics visually convey information and actions. The layout and design of UIs are meticulously crafted, employing color, typography, and spacing to achieve both aesthetic appeal and functionality. Feedback mechanisms, including error messages, ensure that users are informed and guided in their interactions. Navigation features allow for seamless movement through the system, and a focus on accessibility ensures inclusivity for users with disabilities.

. Ultimately, user interface design is a dynamic and multidisciplinary field, where the careful consideration of user behavior and needs can transform a mere interface into a powerful tool for enhancing user satisfaction, productivity, and the overall digital experience.

## Python program

```
import tkinter as tk

# Function to display parking space information
def display_parking_info():
    space_name =
space_name_entry.get()
    availability = availability_var.get()

# In a real app, you would retrieve parking space information here
# from a database or another data source based on user input.
result_label.config(text=f"Space Name: {space_name}\nAvailability: {availability}")

# Create the main application window
app = tk.Tk()
app.title("Parking Space Information App")

# Create and pack widgets
label = tk.Label(app, text="Parking Space Information")
label.pack(pady=10)

space_name_label = tk.Label(app, text="Space Name:")
space_name_label.pack()

space_name_entry = tk.Entry(app)
space_name_entry.pack()
```

```
availability_label = tk.Label(app, text="Availability:")
availability_label.pack()
availability_var = tk.StringVar()
availability_var.set("Available")
availability_dropdown = tk.OptionMenu(app, availability_var, "Available", "Occupied")
availability_dropdown.pack()
submit_button = tk.Button(app, text="Submit", command=display_parking_info)
submit_button.pack(pady=10)
result_label = tk.Label(app, text="") result_label.pack(pady=10)
# Start the application
app.mainloop()
```