



# AN IOT BASED SMART PARKING SYSTEM

<sup>1</sup>Damarla Nigham Sai, <sup>2</sup>Meghanath Someswar Singh, <sup>3</sup>Maguluri Sessa Sai Babu, <sup>4</sup>Bade Triveni Gopi Charan

<sup>1,2,3</sup>UG Student

<sup>1</sup>Electrical and Electronics Engineering, St. Ann's College of Engineering and Technology, Chirala, India

**Abstract:** The IoT (Internet of Things) based smart parking system is a modern solution designed to address the challenges of urban parking management. Leveraging the power of interconnected sensors, communication technologies, and data analytics, this system offers real-time monitoring and optimization of parking spaces, enhancing efficiency, convenience, and user experience.

The IoT based smart parking system is a comprehensive solution aimed at revolutionizing urban parking management. Traditional parking systems often suffer from inefficiencies, such as limited visibility into parking space availability, long search times for drivers, and inefficient space utilization. To address these challenges, the smart parking system integrates various IoT technologies to provide real-time monitoring, analysis, and optimization of parking spaces.

Overall, the IoT based smart parking system represents a paradigm shift in urban parking management, offering a scalable, efficient, and user-centric solution to address the challenges of modern parking infrastructure. By harnessing the power of IoT technologies, data analytics, and seamless connectivity, this system aims to create smarter, more sustainable cities for the future.

At the core of the system are IoT sensors deployed in parking spaces, capable of detecting the presence or absence of vehicles. These sensors communicate wirelessly with a centralized control system, which aggregates the data and provides actionable insights. Through web or mobile applications, drivers can access real-time information about available parking spaces, navigate to the nearest available spot, and even reserve parking in advance.

**Keywords - Parking Slot, Arduino UNO, NodeMCU, Blynk application, SPS - Smart Parking System**

## I. INTRODUCTION

The arrival of urbanization has brought about numerous challenges, and one of the most intense issues faced by cities worldwide is parking management systems. With the ever-growing number of vehicles on the roads, finding parking spaces has become a time consuming and frustrating task for drivers, leading to congestion, pollution, and wasted fuel. Traditional parking systems often suffered from inefficiencies, such as limited visibility into parking space availability, lack of real-time information, and inefficient space utilization.

To address all these challenges, smart parking systems have emerged as an optimistic solution, leveraging the power of technology to revolutionize parking management in urban areas. A smart parking system utilizes a combination of sensors, communication technologies, data analytics and provides real time monitoring, optimization for users.

At its core, a smart parking system focuses on:

- **Improve User Experience** by providing drivers with real-time information about parking space availability, smart parking systems reduce search times, reduce congestion, and enhance overall user experience.
- **Optimize Space Utilization** by analyzing parking occupancy data and predicting demand patterns, smart parking systems enable efficient allocation of parking spaces, reducing underutilization and maximizing revenue generation.
- **Enhance Operational Efficiency** by providing real-time information for users will reduce the time taking process to search an empty slot in a parking area.

In conclusion, smart parking systems represent a pattern shift in metropolitan parking management, offering a scalable, efficient, and user-centric solution to address the challenges of modern parking infrastructure. By harnessing the power of technology, data analytics, and seamless connectivity, smart parking systems pave the way for smarter, more sustainable cities for the future.

## II.SMART PARKING SYSTEM

The concept of Smart Parking System has emerged as a promising solution for modernizing the existing parking problems around the world. In order to achieve smart parking, it includes sensing the information from installed sensors, placing Wi-Fi module board and Arduino board.

### 2.1 Objectives

The objectives of a smart parking system are multifaceted, aiming to address various challenges and improve overall parking management in urban areas. Some of the key objectives include:

- **Reducing Search Time:** Reducing the time and majorly effort spent by the drivers searching for parking spaces by providing real-time information about available spots and guiding them to the nearest available space.
- **Promoting Sustainability:** Encouraging sustainable transportation practices by incentivizing the use of alternative modes of transportation, such as public transit, cycling, or carpooling, and reducing the environmental impact of parking-related activities.
- **Enhancing Accessibility:** Ensuring equitable access to parking facilities for all users, including individuals with disabilities, through the implementation of accessible parking spaces, signage, and amenities.
- **Streamlining Operations:** Optimizing parking facility operations by automating processes such as parking enforcement, maintenance scheduling, and facility management, leading to cost savings and operational efficiency.
- **Enhancing User Experience:** Improving the overall experience for drivers/users by offering convenient features such as mobile apps for parking reservation as well as providing seamless access to parking facilities.

### 2.2 Components

Smart Parking System consists of a variety of key components that work together to achieve the objectives mentioned above. These components include:

- **ARDUINO UNO:** The Arduino Uno is an adaptable and accessible platform for electronics prototyping and DIY projects, suitable for beginners and experienced makers. Its simplicity, affordability, and extensive ecosystem of hardware and software make it an ideal choice for a wide range of applications. From simple blinking LED projects to complex robotic systems, Arduino UNO is the primary selection for many of the users.
- **NODEMCU:** NodeMCU is a very powerful development board for IoT applications, offering built-in Wi-Fi modules which make easy connectivity, Lua-based firmware, Arduino compatibility, and a rich ecosystem of libraries and resources. NodeMCU provides the tools and capabilities to bring your

ideas to life. Whether you're a beginner exploring the world of IoT or an experienced developer building advanced IoT solutions/projects, NodeMCU provides the tools and capabilities to bring your ideas to life.

- **IR SENSOR:** IR sensors are electronic devices that detect infrared radiation emitted by objects, enabling a wide range of applications including motion detection, proximity sensing, object detection, and temperature measurement. With their ability to detect objects without physical contact and in various environmental conditions, IR sensors play a crucial role in modern technology and automation.
- **SERVO MOTOR:** Servo Motors are rotary actuators that provide precise control of angular position, making them ideal for a wide range of applications in robotics, automation, and electronics. With their compact size, high precision, and ease of use, servo motors are a popular choice for engineers, hobbyists, and makers alike.
- **BUCK CONVERTER:** Buck Converter is majorly used as a DC-DC converter that produces an output voltage that is less than input. It basically consists of LM2596 Chip, Power Inductor and an adjustable output voltage. It can allow 3 volts – 40 volts and can vary up to 1.5volts – 35 volts. In our project considerations we provided an input of 15 volts and varied up to 7 volts.

We all know very well about working of Buck Converter, it majorly comprises a combination of two switches (MOSFET & Freewheeling Diode) forms a connection with a low-pass LC filter to reduce the current/voltage ripples. This helps in generating a regulated dc output.

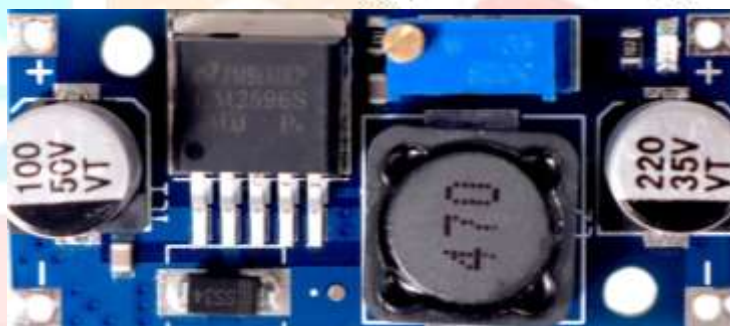


Fig 2.1 : Buck Converter

### 2.3 Challenges in Implementing Parking System

Smart Parking systems offer numerous benefits, they also present several challenges that need to be addressed during implementation. Here are some of the key challenges in implementing smart parking systems:

- **Infrastructure Requirements:** Implementing a smart parking system often requires significant infrastructure investments, including sensors, communication networks, data storage and processing systems, and user interfaces. Ensuring compatibility and interoperability among different components and technologies can be challenging, especially in existing parking facilities.
- **Sensor Accuracy and Reliability:** The accuracy and reliability of parking sensors play a crucial role in the effectiveness of smart parking systems. Challenges such as sensor calibration, environmental factors (e.g., weather conditions), and sensor malfunction can affect the accuracy of parking occupancy detection and lead to incorrect or inconsistent information.
- **Integration with Existing Systems:** Integrating smart parking systems with existing parking management systems, transportation networks, and urban infrastructure can be complex and require coordination among multiple stakeholders. Ensuring seamless interoperability and data exchange is crucial for maximizing the effectiveness of smart parking systems.
- **Regulatory and Legal Considerations:** Smart parking systems may be subject to regulatory requirements and legal constraints, including zoning regulations, parking policies, and data protection

laws. Ensuring compliance with relevant regulations and obtaining necessary permits and approvals can be time-consuming and complex.

By overcoming these challenges, smart parking systems can unlock their full potential to improve urban mobility, enhance user experience, and create more sustainable cities for the future.

### III. LITERATURE SURVEY

“The Smart Parking Management System” by Amira. A. Elsonbaty and Mahmoud Shams. This paper majorly addressed the parking issues facing by the people now-a-days. To overcome this problem, they introduced a Mobile Application which is more helpful to users to check the empty slots.

“Smart Parking Management System” by Vemula Krishna Chaitanya, Mamidi Sri Harsha, Lanka Maneesh Kumar, Vankadari Rahul. This project proposes a Mobile application with advanced features like Slot availability, database, security purposes and sends alerts to registered users.

“Smart Parking System using IoT” by Elakya, Juhi Seth, Pola Ashritha, R. Namith. In this project they used different technology to communicate the data with Arduino along with it they used RFID card which can be read by reader module.

“Smart Parking System” by R. Bharath, K. Deepak, B. Kokila, A. B. Madhurya and S. Shantha. They proposed the parking system by Mobile application and users can book a slot with the help of OTP (One Time Password) which is sent to a registered mobile number.

“Smart Parking Management” by Cybercinatics Private Limited. They proposed a highly advanced system and improved user experience facilities. They took an initiative for the user's security reasons. The technology used in this system will give each & every information about the Car owner and stored in a database for security purposes.

“Design and Development of Smart Parking System based on Fog Computing and Internet of Things” by Mohammed Balfaqih, Waheb Jabbar, Mashaeh Khayyat, Rosilah Hassan. It was implemented in a different perspective with help of Fog computing nodes, Cloud computing, a group of sensors, and parking management software to acquire and process the data related to parking booking.

“Arduino based Smart Car Parking System” by Shraddha Singh, Snehal More, Adesh Tarmale, Rohan Walimbe, Prof. Deepika Patil. In this system they provided a LCD display which is used to show the number of slots available and number of slots filled in a respective mall/area.

### IV. SYSTEM ARCHITECTURE

System Architecture plays a crucial role i.e., designing the connections. This includes,

#### 4.1 Conceptual Model

The below conceptual model describes the chronological Connections b/w Arduino board & NodeMCU.

- Usually 4 Lithium-ion batteries (15 volts) are used to provide supply.
- Initially we step-down the voltage to 7 volts by varying Potentiometer and distributed the power supply to three major devices in the circuit i.e., Arduino board, NodeMCU & Servo Motor.
- IR Sensors are powered-up by the help of Arduino and NodeMCU.
- The 4 IR Sensors are used at the Parking slot to detect the car, and the remaining two sensors are used at the servo motor, for the Automated gate system.

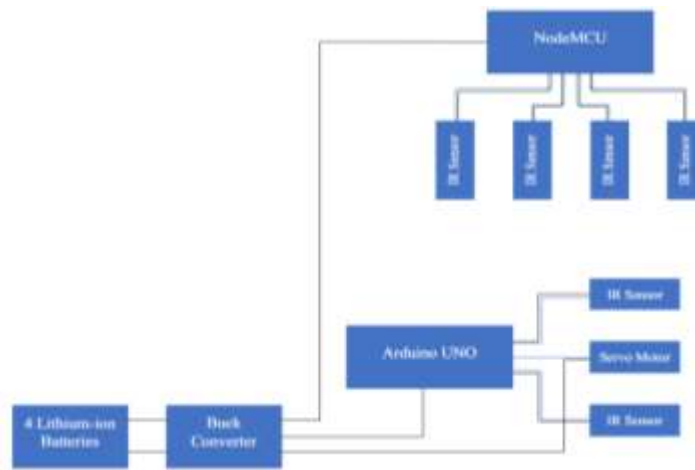


Fig 4.1 : Conceptual Model of Smart Parking System

## 4.2 Circuit Design

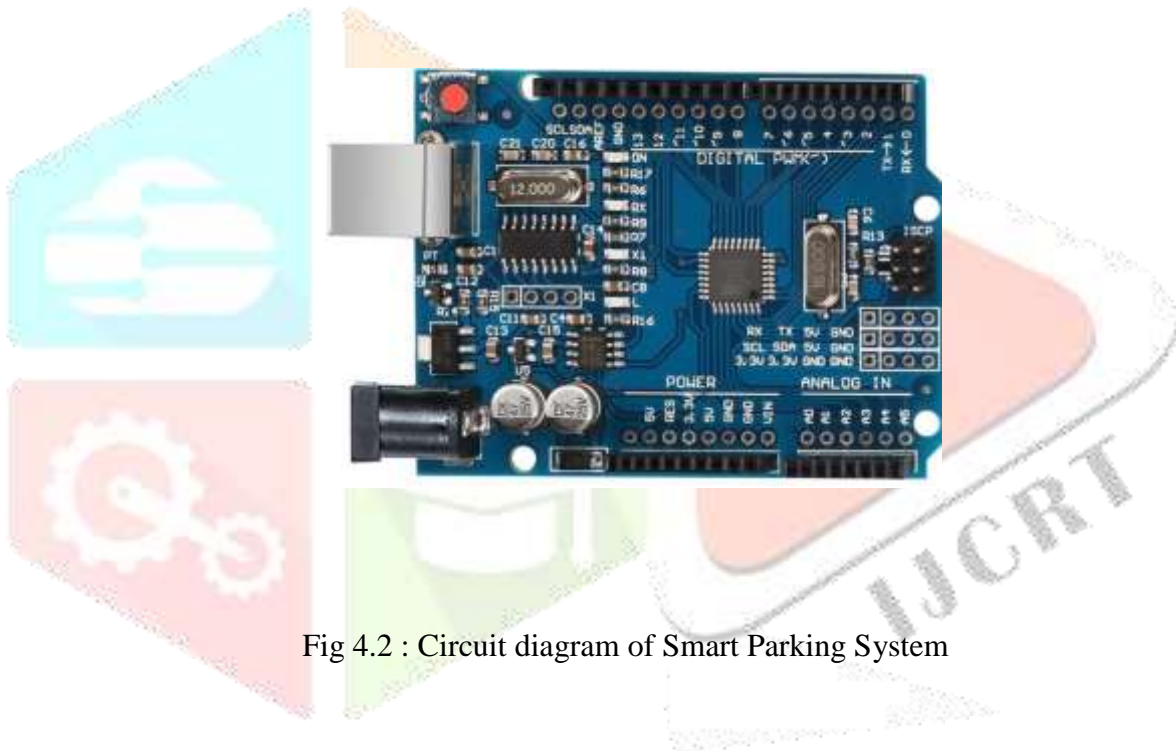


Fig 4.2 : Circuit diagram of Smart Parking System

## V. ARDUINO UNO

### 5.1 Introduction

The Arduino Uno is a widely-used open-source microcontroller board based on the ATmega328P microcontroller chip. It's designed for easy prototyping and development of interactive electronic projects. Here's an introduction to the Arduino Uno:

- **Microcontroller** is the heart of Arduino Uno. The ATmega328P microcontroller, which features 32KB of flash memory for storing code, 2KB of SRAM for variables, and 1KB of EEPROM for data storage. It runs at 16MHz and can be programmed using the Arduino Software (IDE).

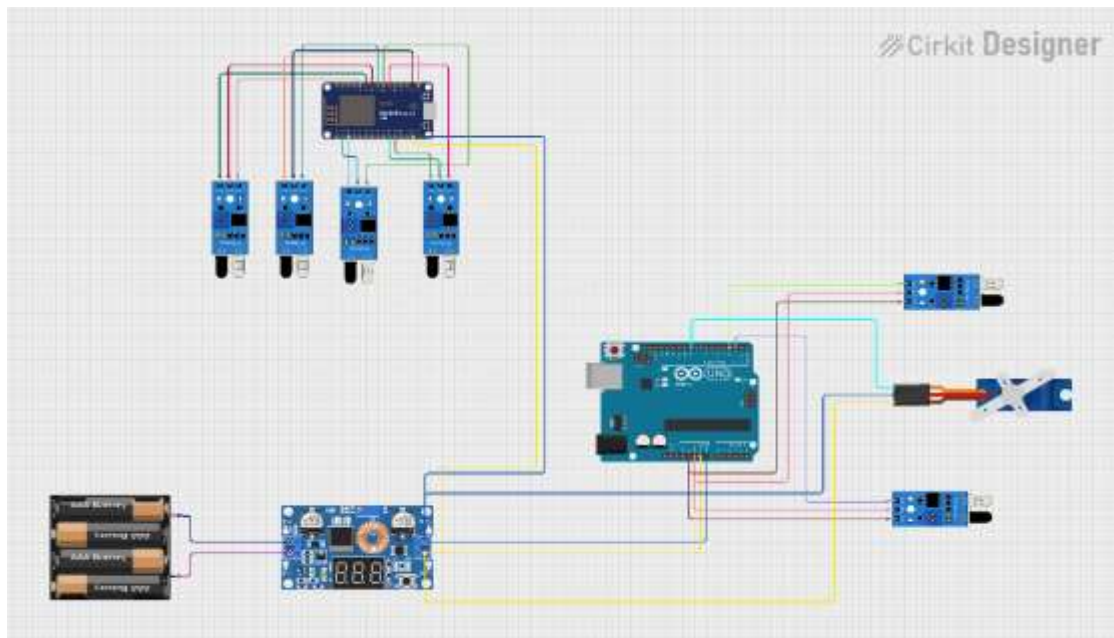


Fig 5.1 : Arduino UNO

- **Digital and Analog Pins** of Arduino Uno board features a total of 14 digital input/output (I/O) pins, of which 6 can be used as PWM (Pulse Width Modulation) outputs. Additionally, there are 6 analog input pins for reading analog sensors or voltage levels.
- **Power Supply** for Arduino Uno can be powered via USB connection from a computer or USB power adapter, or from an external power supply connected to the DC barrel jack. It operates at 5 volts, but it has a built-in voltage regulator that allows it to accept input voltages in the range of 7 to 20 volts.
- **Programming Interface** of Arduino Uno is done through the Arduino Integrated Development Environment (IDE), which provides a simple and intuitive platform for writing, compiling, and uploading code to the board. The programming language used is a variant of C/C++, with a set of libraries and functions specifically designed for Arduino development.
- **Expansion Options** for Arduino Uno is highly possible, with a variety of shields and modules available for adding additional functionality such as Ethernet connectivity, wireless communication (Wi-Fi, Bluetooth, etc.), motor control, display screens, and sensors for various applications.
- **Community Support** is one of the key strengths of the Arduino platform. It has a large and active community of users, makers, and developers. There are countless tutorials, guides, forums, and online resources available to help beginners get started and experienced users push the boundaries of what's possible with Arduino.

In summary, the Arduino Uno is a versatile and accessible platform for electronics prototyping and DIY projects, suitable for beginners and experienced makers alike. Its simplicity, affordability, and extensive ecosystem of hardware and software make it an ideal choice for a wide range of applications, from simple blinking LED projects to complex robotic systems.

## 5.2 Role of Arduino UNO in SPS

In a smart parking system, the Arduino Uno can play several important roles:

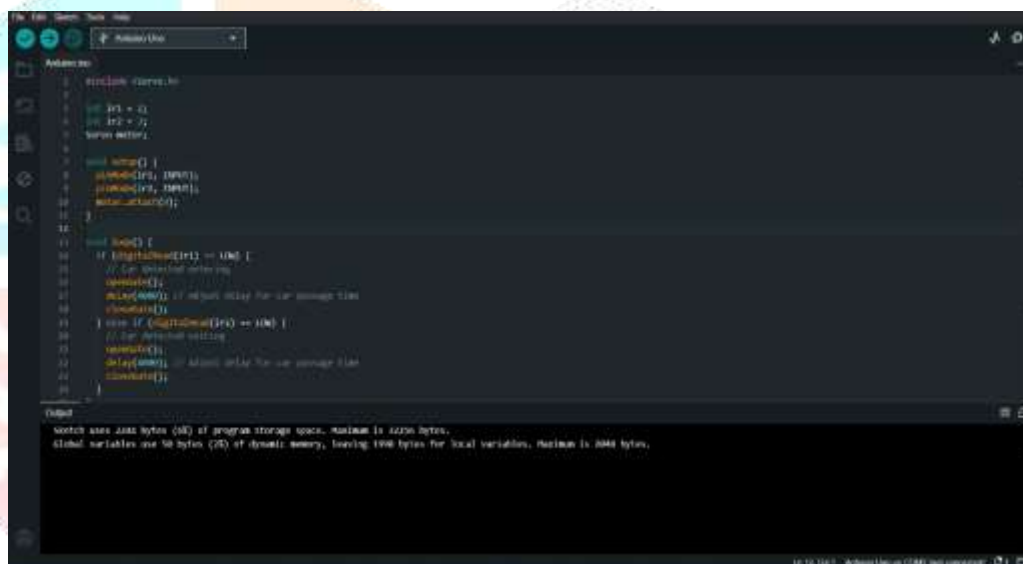
- **Sensor Interface:** Arduino Uno should interface with IR sensors, to detect the presence or absence of vehicles in parking spaces as well as the entry side. It can read sensor data and process it to determine parking space occupancy.
- **Data Processing:** Arduino Uno can process sensor data locally to detect changes in parking space occupancy and transmit this information to a central system. It can perform simple data filtering, thresholding, and decision-making algorithms to optimize sensor data processing.

- **Control Logic:** Arduino Uno control actuators such as servo motors to lift the gate when there is a presence of vehicle and closed after 4000ms.
- **Communication Interface:** Arduino Uno communicates with other components of the smart parking system, such as gateway devices, IR Sensors.
- **Customization & Expansion:** Arduino Uno provides a flexible and customizable platform for prototyping and developing smart parking solutions. Developers can write custom firmware and software for Arduino Uno to implement specific features or functionalities tailored to their requirements. Additionally, Arduino Uno supports expansion shields and modules, allowing for easy integration of additional sensors, actuators, or communication interfaces.

Overall, Arduino Uno plays a crucial role in smart parking systems by providing sensing, processing, control, communication, and customization capabilities. It enables the implementation of intelligent parking management solutions that improve efficiency, convenience, and user experience in parking facilities.

### 5.3 Software Implementation

Implementing software for the Arduino Uno involves several steps, including writing code, compiling it, and uploading it to the board. To accomplish the above tasks we used Arduino IDE, which is capable of writing code and dump into Arduino UNO board.



```

#include <Servo.h>

int pin1 = 0;
int pin2 = 2;
Servo motor;

void setup() {
  pinMode(pin1, INPUT);
  pinMode(pin2, INPUT);
  motor.attach(pin2);
}

void loop() {
  if (digitalRead(pin1) == LOW) {
    // Car detected entering
    digitalWrite(pin2, HIGH);
    delay(4000); // Adjust delay for car storage time
    digitalWrite(pin2, LOW);
  } else if (digitalRead(pin1) == HIGH) {
    // Car detected exiting
    digitalWrite(pin2, LOW);
    delay(4000); // Adjust delay for car storage time
    digitalWrite(pin2, HIGH);
  }
}

```

Sketch uses 4444 bytes (8%) of program storage space. Maximum is 54286 bytes.  
Global variables use 50 bytes (0%) of dynamic memory, leaving 1948 bytes for local variables. Maximum is 1998 bytes.

Fig 5.2 : Code in Arduino UNO

- The Arduino programming language is based on C/C++. The code typically consists of setup() and loop() functions, where you initialize variables and configure the board in the setup() function and implement the main logic in the loop() function.
- Once we write our code, we can verify it for any syntax errors or compilation errors. If there are any errors, we need to fix them before proceedings.
- Before uploading our code to the Arduino Uno, make sure to select the correct board and port in the Arduino IDE.
- Once we verified and selected the correct board and port, we can upload our code to the Arduino Uno. This will compile our code again (if necessary) and upload it to the Arduino Uno via the selected port. After uploading it starts iterating.

### 5.4 Advantages & Dis-advantages

The Arduino Uno offers several advantages and disadvantages, which are important to consider when choosing it for a project. Here's a breakdown of the advantages and disadvantages of the Arduino Uno:

### Advantages:

- **Ease of Use:** Arduino Uno is designed with beginners in mind, offering a user-friendly platform for learning and experimentation. It comes with a simple and intuitive integrated development environment (IDE) and a vast ecosystem of libraries and tutorials.
- **Open-Source:** Arduino Uno is an open-source platform, which means that its hardware and software designs are freely available for anyone to study, modify, and distribute. This fosters a collaborative and supportive community of makers, developers, and educators.
- **Versatility:** Arduino Uno can be used for a wide range of applications, from simple blinking LED projects to complex robotics and IoT (Internet of Things) applications. Its expandability and compatibility with various shields and modules make it suitable for prototyping and developing diverse projects.
- **Affordability:** Arduino Uno boards are relatively inexpensive compared to other microcontroller platforms, making them accessible to hobbyists, students, and enthusiasts with limited budgets. This affordability allows for experimentation and exploration without breaking the bank.
- **Cross-Platform Compatibility:** Arduino Uno IDE is available for Windows, Mac, and Linux operating systems, allowing users to develop code on their preferred platform. This cross-platform compatibility ensures that Arduino Uno is accessible to users regardless of their operating system.

### Dis-advantages

- **Limited Processing Power:** Arduino Uno is based on the ATmega328P microcontroller, which has limited processing power and memory compared to more powerful microcontrollers and single-board computers. This can be a limitation for projects that require complex computations or multitasking.
- **Limited Connectivity:** Arduino Uno has limited connectivity options compared to more advanced microcontroller boards. It lacks built-in Wi-Fi or Bluetooth capabilities, requiring additional modules or shields for wireless communication.
- **Not Suitable for High-Speed Applications:** Arduino Uno operates at a relatively low clock speed (16MHz) compared to other microcontrollers, making it less suitable for high-speed applications such as real-time signal processing or high-frequency communication.
- **Limited Input/Output Pins:** Arduino Uno has a limited number of digital and analog input/output pins, which can be a constraint for projects that require a large number of sensors, actuators, or peripherals. Using multiplexers or additional boards may be necessary to expand I/O capabilities.
- **Not Ideal for Mass Production:** While Arduino Uno is well-suited for prototyping and small scale projects, it may not be suitable for mass production due to its relatively high cost per unit and limited customization options. For large-scale production, custom-designed hardware may be more cost-effective.

Overall, Arduino Uno offers a versatile and accessible platform for electronics prototyping and DIY projects, but it also has some limitations that need to be considered when choosing it for a project. It's important to evaluate the specific requirements and constraints of your project to determine if Arduino Uno is the right choice.

### 5.5 Applications

Arduino Uno finds application in a wide range of projects and industries due to its versatility, ease of use, and affordability. Here are some common applications of Arduino Uno:



- **Home Automation:** Arduino Uno can be used to create DIY home automation systems for controlling lights, appliances, security cameras, and HVAC (heating, ventilation, and air conditioning) systems. It allows users to monitor and control their home remotely via the internet or smartphone.
- **IoT (Internet of Things) Projects:** Arduino Uno is widely used in IoT projects to collect data from sensors, communicate with cloud services, and control actuators. It enables the development of smart devices and systems for monitoring environmental conditions, tracking assets, and controlling home or industrial appliances.
- **Robotics:** Arduino Uno is popular in robotics projects for controlling motors, sensors, and other robotic components. It can be used to build robotic vehicles, robotic arms, drones, and educational robot kits for learning and experimentation.
- **Wearable Technology:** Arduino Uno can be used to create wearable devices such as fitness trackers, smartwatches, and health monitoring systems. It enables the integration of sensors, displays, and communication modules into wearable form factors for personal health and fitness monitoring.
- **Educational Projects:** Arduino Uno is widely used in educational settings, including schools, colleges, and makerspaces, for teaching electronics, programming, and STEM (science, technology, engineering, and mathematics) concepts. It provides a hands-on learning platform for students to explore electronics and programming principles.

These are just a few examples of the many applications of Arduino Uno. Its versatility and ease of use make it a popular choice for a wide range of projects and industries, from hobbyist tinkering to professional prototyping and development.

## VI. NODEMCU

### 6.1 Introduction

The NodeMCU is a popular open-source development board based on the ESP8266 microcontroller. It's designed for easy prototyping and development of IoT (Internet of Things) projects. Here's an introduction to the NodeMCU:

- **Microcontroller:** The NodeMCU is built around the ESP8266 microcontroller, which features a powerful 32-bit Tensilica Xtensa LX106 processor running at 80MHz (or overclockable to 160MHz). It comes with built-in Wi-Fi connectivity, making it ideal for IoT applications that require wireless communication.



Fig 6.1 : NodeMCU with in-built ESP8266 WiFi Module

- **Development Board:** The NodeMCU is a development board that integrates the ESP8266 microcontroller with additional components such as a USB-to-serial converter (CH340 or CP2102),

voltage regulator, and GPIO (General Purpose Input/Output) pins. It provides a convenient platform for prototyping and experimenting with IoT projects.

- **Built-in Wi-Fi:** One of the key features of the NodeMCU is its built-in Wi-Fi connectivity, which allows it to connect to Wi-Fi networks and communicate with other devices and services over the internet. This enables IoT applications such as remote monitoring, home automation, and data logging.
- **Programming Interface:** The NodeMCU can be programmed using the Arduino IDE, which provides a familiar environment for writing, compiling, and uploading code. It uses the Lua scripting language by default, but it can also be programmed using C/C++ with the Arduino core for ESP8266.
- **Expansion Options:** The NodeMCU features a row of GPIO pins that can be used for connecting sensors, actuators, displays, and other peripheral devices. It also supports the use of additional modules and shields for expanding its capabilities, such as sensors for temperature, humidity, motion, or light.
- **Community Support:** The NodeMCU has a large and active community of users, makers, and developers who contribute to its development and provide support through forums, tutorials, and online resources. This community-driven approach fosters collaboration and innovation in IoT projects.

Overall, the NodeMCU is a versatile and accessible platform for prototyping and developing IoT projects. Its built-in Wi-Fi connectivity, ease of programming, and expandability make it ideal for a wide range of applications, from home automation and environmental monitoring to industrial automation and smart devices.

## 6.2 Role of NodeMCU in SPS

The NodeMCU can play several important roles in a smart parking system:

- **Data Acquisition:** NodeMCU can interface with IR sensors to detect the presence or absence of vehicles in parking spaces. It can read sensor data and process it to determine parking space occupancy.
- **Wireless Communication:** NodeMCU has built-in Wi-Fi connectivity, allowing it to connect to Wi-Fi networks and communicate with other devices and services over the internet. It transmits sensor data to the Blynk application for further processing and analysis.
- **Data Processing and Decision Making:** NodeMCU can perform local data processing and decision-making tasks to optimize sensor data processing and reduce communication overhead. For example, it can aggregate sensor data from multiple nodes, perform simple data filtering or thresholding, and transmit only relevant information to the central system.
- **Integration with Blynk Application:** NodeMCU can integrate with mobile applications to exchange data, receive commands, and synchronize with other components of the smart parking system.
- **Edge Computing:** NodeMCU can perform edge computing tasks such as local data aggregation, pre-processing, or event detection to reduce latency and improve responsiveness. It can process sensor data locally and trigger actions or alerts based on predefined rules or thresholds.

Overall, NodeMCU plays a crucial role in smart parking systems by providing sensing, communication, data processing, and control capabilities at the edge of the network. It enables the implementation of intelligent parking management solutions that improve efficiency, convenience, and user experience in parking facilities.

## 6.3 Software Implementation

Implementing software on the NodeMCU involves several steps, including writing code, configuring settings, compiling, and uploading it to the board.



- **Versatility:** NodeMCU can be used for a wide range of IoT applications, from home automation and environmental monitoring to industrial automation and smart devices. Its expandability and compatibility with various sensors and actuators make it suitable for diverse projects.
- **Community Support:** The NodeMCU has a large and active community of users, makers, and developers who contribute to its development and provide support through forums, tutorials, and online resources. This community-driven approach fosters collaboration and innovation in IoT projects.

### Dis-advantages

- **Limited Processing Power:** While NodeMCU is capable of performing many IoT tasks, it has limited processing power compared to more powerful microcontroller platforms or single-board computers. This can be a limitation for projects that require complex computations or multitasking.
- **Limited Analog Pins:** NodeMCU has a limited number of analog input pins compared to digital input/output pins. This can be a constraint for projects that require a large number of analog sensors or inputs.
- **Voltage Limitations:** NodeMCU operates at 3.3 volts, which is lower than the 5 volts typically used by Arduino boards. This can cause compatibility issues with certain sensors or peripherals that require 5-volt operation.
- **Complexity of Wi-Fi Setup:** While built-in Wi-Fi connectivity is a major advantage of NodeMCU, setting up Wi-Fi connections and configuring network settings can be more complex compared to other microcontroller platforms. Beginners may encounter challenges in setting up Wi-Fi connections and troubleshooting network issues.
- **Flash Memory Limitations:** NodeMCU has limited flash memory for storing code and data, which can be a constraint for projects that require large codebases or data storage. Careful memory management is required to avoid running out of memory.

Overall, the NodeMCU offers a versatile and accessible platform for prototyping and developing IoT projects, but it also has some limitations that need to be considered when choosing it for a project. It's important to evaluate the specific requirements and constraints of your project to determine if NodeMCU is the right choice.

### 6.5 Applications

The NodeMCU finds application in a wide range of IoT (Internet of Things) projects due to its built-in Wi-Fi connectivity, ease of use, and affordability. Here are some common applications of the NodeMCU:

- **Home Automation:** NodeMCU is widely used in home automation projects to control lights, appliances, HVAC systems, security cameras, and door locks remotely via the internet. It enables users to monitor and control their home environment from anywhere using a smartphone or web browser.
- **Environmental Monitoring:** NodeMCU can be used to build environmental monitoring systems for measuring temperature, humidity, air quality, noise levels, and other environmental parameters. It allows users to track environmental conditions in homes, offices, industrial facilities, and outdoor environments.
- **Smart Agriculture:** NodeMCU is used in smart agriculture projects for monitoring soil moisture levels, temperature, humidity, and light levels in agricultural fields and greenhouses. It enables farmers to optimize irrigation, fertilization, and pest control practices for improved crop yield and quality.

- **Industrial Automation:** NodeMCU is employed in industrial automation applications for monitoring and controlling machinery, equipment, and production processes. It enables remote monitoring of equipment status, predictive maintenance, and real-time data analysis for optimizing production efficiency and reducing downtime.
- **Smart Energy Management:** NodeMCU can be used to monitor and control energy consumption in residential and commercial buildings. It enables the implementation of smart energy management systems for optimizing energy usage, reducing electricity bills, and promoting sustainability.

Overall, the NodeMCU is a versatile and accessible platform for prototyping and developing IoT projects across a wide range of applications. Its built-in Wi-Fi connectivity, ease of use, and affordability make it an ideal choice for both beginners and experienced makers alike.

## VII. SERVO MOTOR

### 7.1 Introduction

Servo Motor is an Electric device which can rotate with great accuracy. It consists of a motor, gearbox, control circuit, and feedback mechanism, which helps the servo motor to rotate with great exactness. Key points about Servo motor:

- **Precise Control:** Servo motors offer accurate control over angular position, speed, and acceleration. They can rotate to a specific angle and hold that position with great accuracy.
- **Feedback Mechanism:** Servo motors include a feedback mechanism that continuously monitors the motor's position and provides feedback to the control circuit. This feedback allows the motor to adjust its position in real-time, ensuring accurate positioning.
- **Closed-Loop Control:** Servo motors operate in a closed-loop control system, where the desired position is compared to the actual position measured by the feedback mechanism. The control circuit adjusts the motor's output to minimize the error between the desired and actual positions.
- **Torque:** Servo motors can provide high torque output relative to their size, making them suitable for applications that require both precision and power.
- **Variable Speed:** Servo motors can operate at different speeds, allowing for smooth and controlled motion over a wide range of speeds.
- **Types of Servo Motors:** There are various types of servo motors, including DC servo motors, AC servo motors, brushless DC (BLDC) servo motors, and linear servo motors. Each type has its own advantages and is suited to different applications.

Overall, servo motors are reliable components that play a crucial role in motion control systems, offering precise positioning, high torque, and smooth operation across a wide range of applications.

### 7.2 Role of Servo Motor in SPS

Well, coming to the working model of SPS, servo motors play a crucial role in order to allow/exit the drivers. In this case servo motors are used for barrier gates which should open when the IR sensor detects a car in entrance and closed with delay of 4000ms. And, same procedure and execution when a car wants to get out from parking areas.



Fig 7.1 Barrier Gate

Servo motor can bear a weight up to 2.5 kg when suspended at a distance of 1 cm. So, the longer this distance, the weaker the force will be. To accomplish the SPS task, servo motors need better communication with IR sensors. Unless it won't produce an accurate output.

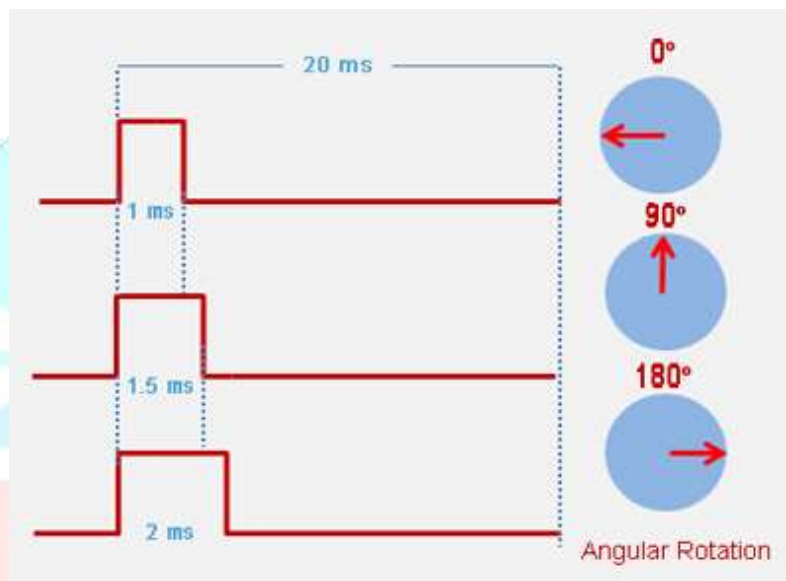


Fig 6.2 Servo motor module

Servo motor is involved with 3 different connections. One of the major connections is fed power supply, ground connection, and the other is Data pin, which is connected to Arduino UNO board and the rest of the connections are secured by Buck Converter for greater accuracy and working.

### 7.3 Working

Servo Motor works on the principle of PULSE WIDTH MODULATION (PWM), meaning its angle of rotation is controlled by a pulse applied to its control pin. It was developed and made by a DC motor which is controlled by a variable resistor (potentiometer) and some gears.

Servo Motor is able to rotate from 0 to 180 degrees, but it can achieve up to 210 degrees, depending on the manufacturing. The degree of rotation can be controlled by applying the Electrical Pulse properly to its control pin. Servo checks the pulse every 20 milliseconds. The pulse of 1ms (1 millisecond) width can rotate the servo motor to 0 degrees, 1.5ms can rotate to 90 degrees (neutral position) and 2ms pulse can rotate it to 180 degree.



Fig 6.3 Rotations with respect to milliseconds

All servo motors work on +5 volts supply, but we need to take care about how much current the motor will consume if you are planning to use it more. For a better and optimal performance servo needs an input of 4.8 volts – 6 volts (maximum).

#### 7.4 Advantages & Dis-advantages

Servo motors had various advantages and dis-advantage factors to use in this project. Few of them are:

##### Advantages

- **Precise Positioning:** One of the primary advantages of servo motors is their ability to achieve precise positioning with high accuracy. They can rotate to a specific angle and hold that position reliably, making them ideal for applications that require precise motion control.
- **High Torque:** Servo motors can deliver high torque output relative to their size, making them capable of exerting substantial force even in compact spaces. This high torque output enables servo motors to drive heavy loads and perform tasks that require significant force.
- **Variable Speed:** Servo motors can operate at variable speeds, allowing for smooth and controlled motion over a wide range of speeds. This versatility makes them suitable for applications that require dynamic speed control and acceleration.
- **Closed-Loop Control:** Servo motors operate in a closed-loop control system, where feedback from an encoder or potentiometer is used to continuously monitor and adjust the motor's position. This closed-loop control ensures accurate positioning and compensates for disturbances or changes in load conditions.
- **Fast Response Time:** Servo motors have fast response times, allowing them to quickly respond to changes in input signals and adjust their position accordingly. This rapid response time makes servo motors suitable for applications that require quick and precise motion control.
- **Wide Range of Applications:** Servo motors are used in a wide range of applications, including robotics, CNC machines, 3D printers, industrial automation, camera gimbals, and model aircraft. Their versatility and reliability make them indispensable components in motion control systems.

## Dis-advantages

- **Complex Control:** Servo motors require complex control algorithms to operate effectively, including PID (Proportional-Integral-Derivative) control loops and tuning parameters. Designing and implementing these control algorithms can be challenging and time-consuming, especially for inexperienced users.
- **Power Consumption:** Servo motors can consume significant amounts of power, especially when operating at high speeds or under heavy loads. This high-power consumption can lead to increased energy costs and may require additional cooling or heat dissipation measures.
- **Cost:** Servo motors tend to be more expensive than other types of motors, such as stepper motors or brushed DC motors. The cost of servo motors can be prohibitive for some applications, particularly those with strict budget constraints.
- **Limited Continuous Rotation:** Unlike stepper motors, which can rotate continuously without feedback, servo motors are limited in their continuous rotation capabilities. They typically have a limited range of motion, usually up to 180 degrees or less, which may not be suitable for all applications.
- **Feedback Mechanism Reliability:** Servo motors rely on feedback mechanisms, such as encoders or potentiometers, to determine their position. The reliability of these feedback mechanisms can be a concern, as they may be susceptible to wear and tear over time or environmental factors such as dust or moisture.

Overall, while servo motors offer precise positioning, high torque, and variable speed control, they also come with challenges such as complex control, cost, and power consumption. Understanding these advantages and disadvantages is crucial for selecting the right motor type for a given application.

## 7.5 Applications

Servo motors find application in a wide range of industries and fields due to their precise motion control capabilities and high torque output. Some common applications of servo motors include:

- **Robotics:** Servo motors are widely used in robotics for controlling the movement of robot arms, grippers, joints, and other mechanical components. They enable precise positioning and smooth motion control, making them essential for various robot applications, including industrial robots, robotic arms in manufacturing, and humanoid robots.
- **CNC Machines:** Servo motors are commonly used in computer numerical control (CNC) machines for controlling the movement of axes, such as the X, Y, and Z axes in milling machines, lathes, and routers. They provide accurate positioning, rapid response times, and high torque output, allowing for precise machining operations and complex part geometries.
- **3D Printing:** Servo motors play a critical role in 3D printing systems for controlling the movement of print heads, extruders, and build platforms. They enable precise layer-by-layer deposition of materials, accurate positioning of print nozzles, and smooth motion control, resulting in high quality printed parts with intricate details and complex geometries.
- **Industrial Automation:** Servo motors are extensively used in industrial automation systems for controlling conveyor belts, robotic arms, packaging machinery, pick-and-place systems, and other automated equipment. They enable high-speed operation, precise positioning, and dynamic motion control, enhancing productivity and efficiency in manufacturing and logistics operations.
- **Aerospace and Defence:** Servo motors are used in aerospace and defense applications for controlling aircraft flight surfaces, missile guidance systems, unmanned aerial vehicles (UAVs), and remotely operated vehicles (ROVs). They provide precise control over movement and stabilization, ensuring safe and accurate operation in challenging environments.



These are just a few examples of the diverse applications of servo motors across different industries and fields. Their versatility, precision, and reliability make them indispensable components in motion control systems and automation technologies.

## VIII. BLYNK APPLICATION

### 8.1 Introduction

Blynk is a popular platform for building IoT (Internet of Things) projects and applications. It consists of a mobile app, a cloud-based server, and a library of hardware components and libraries that enable developers to create connected devices and applications easily.

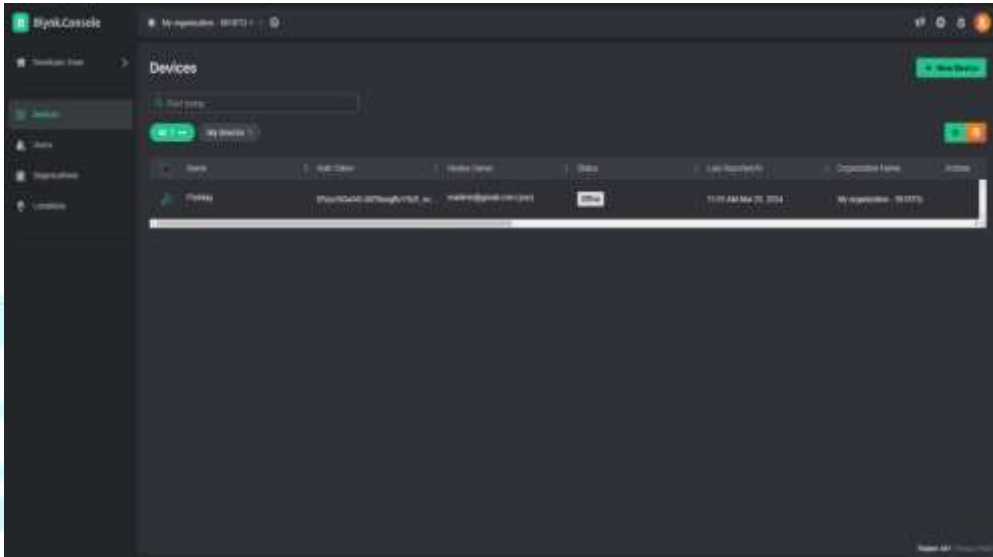


Fig 8.1 Interface of Blynk in Web

- **User-Friendly Interface:** The Blynk app features a user-friendly interface that allows users to create, configure, and control IoT projects with ease. It provides intuitive tools for adding widgets, configuring settings, and monitoring device status.
- **Widget Library:** Blynk offers a wide range of widgets that users can add to their projects to control devices, display sensor data, and interact with the physical world. Widgets include buttons, sliders, graphs, gauges, LEDs, displays, and more, allowing for versatile and customizable user interfaces.
- **Customizable Projects:** With Blynk, users can create custom IoT projects tailored to their specific needs and requirements. They can add, remove, and arrange widgets to create personalized dashboards and interfaces that suit their preferences.
- **Cloud Connectivity:** Blynk provides cloud connectivity that enables users to remotely monitor and control their IoT devices from anywhere with an internet connection. Users can access their projects and devices via the Blynk cloud server, making it easy to stay connected and in control.
- **Real-Time Data Visualization:** Blynk allows users to visualize real-time sensor data and device status using graphs, charts, and gauges. This enables users to monitor environmental conditions, track trends, and make informed decisions based on data insights.
- **Community and Documentation:** Blynk has a large and active community of users, makers, and developers who share projects, tutorials, and tips on the Blynk forum and social media channels. Additionally, Blynk provides comprehensive documentation, tutorials, and examples to help users get started and troubleshoot issues.

Overall, the Blynk app provides a powerful and versatile platform for creating IoT projects and applications, enabling users to build connected devices, monitor sensor data, and control their environment with ease. Whether you're a beginner or an experienced developer, Blynk offers the tools and resources you need to bring your IoT ideas to life.

## 8.2 Purpose of Using

Blynk application plays a major role in Smart Parking System. We approached the blynk mobile application from a different perspective, in order to show the available slots in the parking area, showing the presence of the car for user experience.

- **NEW DEVICE:** We created a new device and connected with the help of WiFi type. Initially in this first step we should select the respective Hardware and connection type manually. In our project we manually selected the ESP8266 hardware model and WiFi based connection.

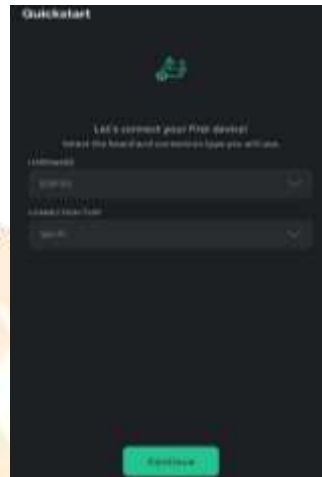


Fig 8.2 Creating a New Device



Fig 8.3 Device with 4 LED widget

- **DEVELOPMENT:** After creating a new device, we pick Developer Zone to make our user experience better. We simply created 4 LED widgets and connected them to respective DataStream's.



```

File Edit Sketch Tools Help
NodeMCU 1.0 (ESP-12E ...)
NodeMCU.ino
1 #include <BlynkEspClient.h>
2 #define BLYNK_PRINT Serial
3 #include <ESP8266WiFi.h>
4 #include <BlynkSimpleEsp8266.h>
5 char auth[] = "6TvpzT5Qx46LHZt9ungBuY0z6_vuT2P3";
6 char ssid[] = "Higham";
7 char pass[] = "12345678910";
8
9 widgetLED led1(V1);
10 widgetLED led2(V2);
11 widgetLED led3(V3);
12 widgetLED led4(V4);
13

```

Fig 8.4 Setting-up the DataStream's

- While connecting datastream's, once refer your NodeMCU code. You need to mention specific pins in your sketch.

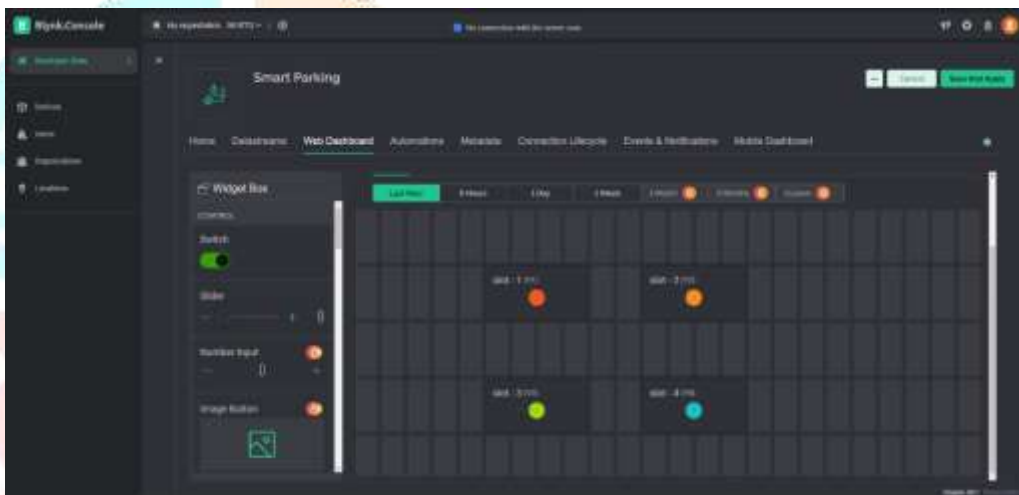


Fig 7.5 Datastream's mentioned in NodeMCU code

- Once we connected our datastream's to LED widgets we're now able to perform our SMART PARKING SYSTEM.

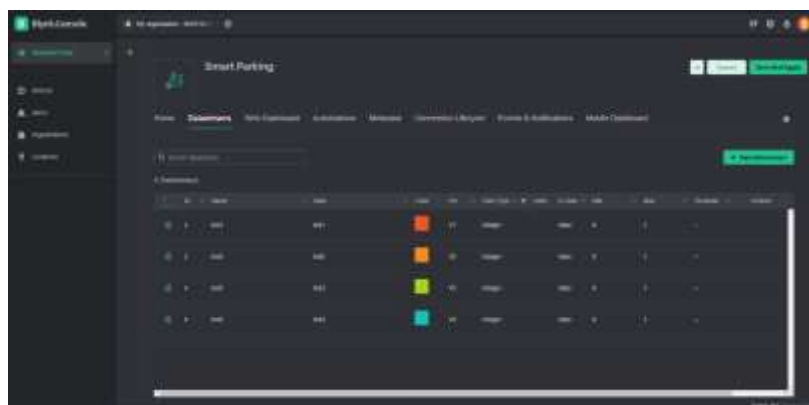


Fig 7.6 LED widgets showing the presence of car in slots

LED Widgets blink when there is a presence of a car in the respective slot. Well, it's all about our creativity to blink the sensor. This is how users can experience using Blynk Application.

## IX. DEVELOPED MODEL

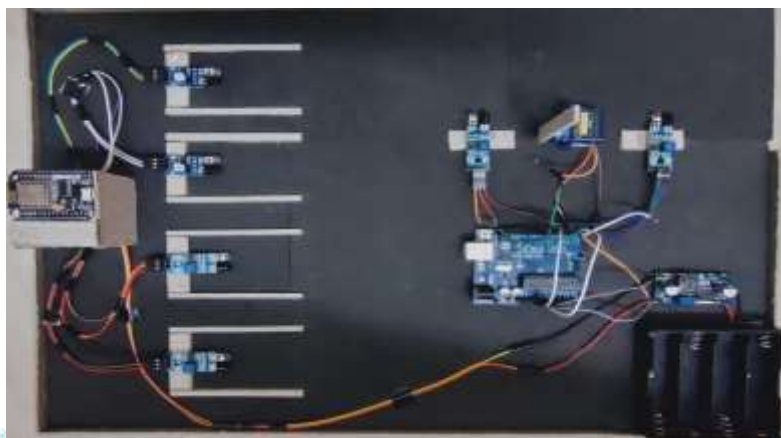


Fig 9.1 : Final Model design of SPS

## X. CONCLUSION

In conclusion, the smart parking system offers a comprehensive solution to the challenges faced in traditional parking management. By leveraging technologies such as IoT, sensors, and cloud computing, the smart parking system enhances efficiency, convenience, and user experience for both parking operators and drivers.

The system optimizes parking space utilization by providing real-time information on space availability, guiding drivers to vacant spots, and facilitating seamless entry and exit processes. It minimizes congestion, reduces vehicle emissions, and improves traffic flow by streamlining parking operations and reducing the time spent searching for parking.

Overall, the smart parking system represents a significant advancement in urban mobility management, offering a scalable, sustainable, and technology-driven solution to address the growing challenges of urbanization and transportation. As cities continue to evolve and grow, the adoption of smart parking systems will play a crucial role in shaping the future of urban mobility and improving the quality of life for residents and visitors alike.

## ACKNOWLEDGEMENT

It gives us immense pleasure to express my gratitude to everyone who helped us in the successful completion of our project, titled "AN IOT BASED SMART PARKING SYSTEM".

We would like to express our sincere thanks to the Department of Electrical and Electronics Engineering of St. Ann's College of Engineering and Technology, for providing me the opportunity to work in and extend all possible facilities.

## REFERENCES

- [1] "IoT based Smart Parking System" by Abhirup Khanna, University of Petroleum and Energy Studies, Dehradun
- [2] "Evaluation of Smart Parking Systems in the context of Smart Mobility" by Nasar Ali Razack and Parul Bansal, Samara University, Ethiopia.
- [3] "Artificial Intelligence-Enabled Smart Parking System" by Tanya Singh and Eshita Vijay.
- [4] "Smart Parking Systems and their Impact on the Efficiency of Parking officers" by Ahmad Alharbi and Ersin Dincelli, University of Colorado.

[5] "A review of Smart Parking Systems" by Sai Sneha Channamallu and Jay M. Rosenberger, University of Texas at Arlington

