GNANAMANI COLLEGE OF TECHNOLOGY

DEPARTMENT: BIO MEDICAL ENGINEERING

YEAR: THIRD YEAR

TOPIC: SMART PARKING

TEAM MEMBERS

1.ATCHAYA.J (620821121012)

2.JOY LOURDU PREETHI.M (620821121043)

3.GAYATHRI.R (620821121024)

4.MAHALAKSHMI.S (620821121060)

5.LOKESHWARI.R (620821121057)

_ By Joy Lourdu preethi

SMART PARKING

INTRODUCTION:

Smart parking is an IOT based parking system is a centralized management that enables drivers to such for and reserve a parking spot remotely through their smart phones .it offers a convenient arrangements for drivers to park their cars when they are looking to avoid potential traffic congestion.

PROBLEM STATEMENT:

In recent research in metropolitan cities the parking management problem can be viewed from various angles such as high vehicle density on roads.

This result in annoying issues for the drivers to park their vehicles as it very difficult to find a parking slots.

The drivers usually waste time and effort in finding parking space and end up parking their vehicles a space on the street which further leads to space on congestion. In worst case, people fail to find any parking space especially during peak hours and festive season.

HARDWARE USED IN SMART PARKING :

- Enode MCU (ESP8266)
- Jumper wires
- Infrared sensors
- 16*2 Led display
- DC motor

SOFTWARED USED IN SMART PARKING :

- ARDUINO IDE

INITIAL SETUP :

The initial case of the system when we turn on our project, which indicates the number of vacant and filled spots on a 16*2 display LCD and similarly on the blink app.

PARKING VEHICLES :

Once when the user enters the parking detect sensor he would receive a parking slot number on his mobile application which he is supposed to park is vehicle.IR sensor successfully detecting the vehicle it show a notification on the app the start time of the vehicle.

UNPARKING VEHICLES:

Unparking your vehicle from the parking slot would pop a notification on the application app starting the start time and end time user has parked the vehicle in the parking slot.

STEPS FOR PARKING :

Step 1: Install the smart parking application on your mobile device.

Step 2: On the 16*2 display the number of vacant and filled spots are displayed so that the user can see the status of parking zone.

Step 3: Once the user logs into the app he would see the parking architecture with the cars filled at which position and positions which are empty .

Step 4: When the user is near to the parking IR detect sensor , he would receive a notification on his app on which slot he can park his vehicle if there is a empty slot.

Step 5: If there is no empty slot the user will be displayed with an appropriate message on the mobile application .

Step 6: On availability of parking area and user parking into the respective slot he/she would receive a message which states the start time of the parking and the slot in which he/she has parked.

Step 7: On successfully un-parking your vehicle from the parking slot the user will receive a message which states the start time and end time of his parking time and an amount which he needs to pay for the parking duration.

SYSTEM ANALYSIS AND DESIGN:

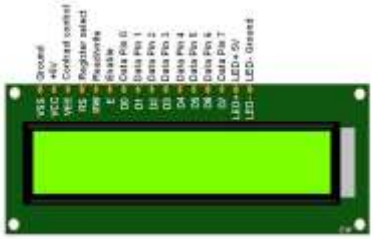
Node MCU :

NodeMCU is capable of either connecting to an existing wireless connection or hosting An application over http protocol.

Node MCU module

16*2 LCD Display

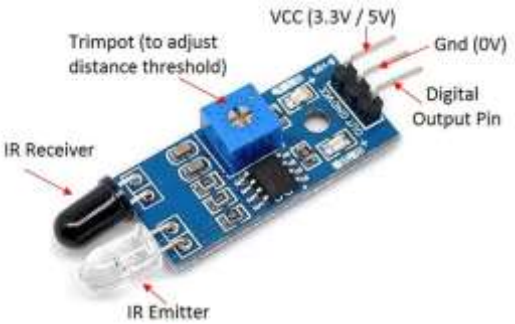
An LCD is an electronic display module which uses liquid crystal to produce a visible Image. The 16×2 LCD display is a very basic module commonly used in DIY's and Circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix. The 16*2 display is used to Display the number of vacant and spilled spot . It also gets updated on the display LCD when a vehicle parks or unparks the vehicle .



16*2 LCD display

IR SENSOR :

An infrared (IR) sensor is an electronic device that measures and detects infrared Radiation in its surrounding environment. Infrared radiation was accidentally Discovered by an astronomer named William Herchel in 1800. While measuring the Temperature of each color of light (separated by a prism), he noticed that the Temperature just beyond the red light was highest. IR is invisible to the human eye,



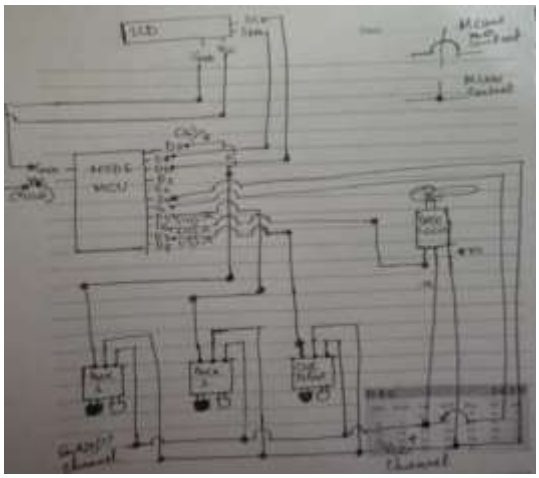
as Its wavelength is longer than that of visible light (though it is still on the same Electromagnetic spectrum).

IR SENSOR

OBJECTIVES OF SMART PARKING:

- Optimized parking.
- Reduced traffic.
- Reduced pollution.
- Increased Safety.
- Decreased Management Costs.
- Enhanced User Experience.

CIRCUIT DIAGRAM:



EXPERIMENTAL SETUP:

FEATURE ENGINEERING:

1. **Occupancy Detection**: Use IR sensors to detect vehicle presence in parking spots.
2. **LCD Display**: Implement a clear 16x2 LCD display to show real-time parking information.
3. **Light Indicators**: Include LED indicators to signal occupied or available parking spaces.
4. **Wireless Connectivity**: Enable communication between nodes for a networked parking system.
5. **Data Logging**: Keep a log of parking occupancy data for analysis or future reference.
6. **Energy Efficiency**: Implement sleep modes for the MCU to conserve power when not actively sensing.
7. **User Interface**: Develop a user-friendly interface on the LCD for easy interactions.
    - .

MODEL TRAINING:

1. **Data Collection**: Gather data on parking spot occupancy using IR sensors. Record when a car is present or when a spot is empty.

2. **Data Labeling**: Label the collected data to create a dataset indicating occupied and unoccupied parking spots.

3. **Data Preprocessing**: Prepare the dataset by cleaning, normalizing, and splitting it into training and testing sets.

4. **Model Selection**: Choose a machine learning model suitable for binary classification, such as logistic regression or a simple neural network.

5. **Feature Engineering**: Extract relevant features from the sensor data, such as time of day, to enhance the model's performance.

6. **Model Training**: Train the selected model using the training dataset. Adjust parameters for optimal performance.

7. **Model Evaluation**: Evaluate the model on the testing dataset to ensure it generalizes well to new data.

8. **Integration with Node MCU**: Once satisfied with the model, integrate it with the Node MCU platform. Ensure compatibility and establish communication between the model and MCU.

9. **LCD Display Integration**: Develop code to display the model's predictions on the 16x2 LCD display, indicating parking spot availability.

10. **Testing and Optimization**: Test the complete system in a real-world environment, gather feedback, and optimize as needed.

EVALUATION:

1. **Accuracy**: Measure how accurately the system detects parking spot occupancy. Use metrics like True Positive, True Negative, False Positive, and False Negative rates.

2. **Real-time Performance**: Evaluate the responsiveness of the system in real-time. Check if the LCD display updates promptly based on changes in parking spot occupancy.

3. **User Interface Interaction**: Assess the user interaction with the LCD display. Ensure it provides clear and understandable information about parking availability.

4. **Reliability**: Test the system's reliability over an extended period. Verify that it consistently provides accurate information and doesn't produce false positives or negatives.

5. **Energy Efficiency**: Evaluate the power consumption of the Node MCU module. Ensure it's optimized for efficient use of energy, especially if deployed in battery-powered scenarios.

6. **Scalability**: Consider how well the system scales to a larger parking area with multiple nodes. Ensure that communication and coordination between nodes function seamlessly.

7. **Data Logging and Analysis**: If applicable, analyze the data logs generated by the system. Check if it provides useful insights into parking patterns, which can be valuable for future optimizations.

8. **User Feedback**: Gather feedback from users interacting with the smart parking system. Use their input to identify areas for improvement and refinement.

9. **Integration Challenges**: Assess any challenges encountered during the integration of the machine learning model with the Node MCU and LCD display. Ensure that the components work harmoniously.

10. **Cost-effectiveness**: Consider the cost of the hardware components and development efforts versus the benefits provided by the smart parking system. Evaluate its cost-effectiveness in comparison to alternative solutions.

OBJECTIVES:

The project objectives of a Smart Parking System in IoT typically include optimizing parking space utilization, reducing traffic congestion, enhancing user convenience through real-time information, and integrating IoT technologies for efficient monitoring and management of parking spaces.

IOT SENSOR SETUP:

In a Smart Parking System with IoT, sensor setup involves deploying smart sensors in parking spaces. These sensors can be ultrasonic, infrared, or other types, depending on the system design. They detect the presence or absence of vehicles and communicate this data to a centralized system through IoT connectivity, enabling real-time monitoring and management of parking spaces.

MOBILE APP:

Developing a mobile app for a Smart Parking System involves creating a user-friendly interface for drivers. The app should include features such as real-time parking space availability, navigation to available spots, online payments, and notifications. Integration with the IoT sensors enables users to access accurate and up-to-date information, enhancing their overall parking experience.

PROGRAM:

```
From gpiozero import DistanceSensor
```

```
From time import sleep


# Define GPIO pin numbers for trigger and echo pins
GPIO_TRIGGER1 = 17
GPIO_ECHO1 = 18
GPIO_TRIGGER2 = 23
GPIO_ECHO2 = 24


Sensor1 = DistanceSensor(trigger=GPIO_TRIGGER1, echo=GPIO_ECHO1)
Sensor2 = DistanceSensor(trigger=GPIO_TRIGGER2, echo=GPIO_ECHO2)
# Add more sensors if needed


Def check_parking_spaces(distance1, distance2):
   # You can define your parking space logic here
   # For example, check if a car is within a certain distance threshold
   If distance1 < 0.2:
      Print(“Parking Space 1 is occupied”)
   Else:
      Print(“Parking Space 1 is available”)


   If distance2 < 0.2:
      Print(“Parking Space 2 is occupied”)
   Else:
      Print(“Parking Space 2 is available”)
```

```
While True:
    Distance1 = sensor1.distance
    Distance2 = sensor2.distance
    # Read distances from more sensors if needed

    # Process distance data and manage parking spaces here
    Check_parking_spaces(distance1, distance2)

    Sleep(1)  # Delay for better readability
```

CONCLUSION:

- The concept of smart cities has always been a dream for humanity . The growth of Internet of Things and cloud technologies have given rise to new possibilities in terms of smart cities.

- Smart parking facilities and traffic management system have always been at the core of constructing smart cities.

- In this project ,we address the issue of parking and present IOT based cloud integrated smart parking system