PES UNIVERSITY, Bengaluru
Department of Computer Science and Engineering
B. Tech (CSE) – 5th Semester – Aug-Dec 2024

B.TECH. (CSE)
V Semester
UE22CS342AA3 –Internet of Things

PROJECT REPORT
on

Project Title:

SMART PARKING SYSTEM

Submitted by :

Name 1: Soumya Ranjan Mishra	SRN 1: PES2UG22CS571
Name 2: Shreyas S	SRN 2: PES2UG22CS540

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BENGALURU – 560100, KARNATAKA, INDIA

Introduction:

The Smart Parking System is an IoT-based project designed to efficiently manage vehicle parking in real time. The system uses a Raspberry Pi Pico W microcontroller to control an entry gate and exit gate, each equipped with a servo motor for automated access control. Ultrasonic sensors monitor the occupancy status of individual parking slots, providing real-time updates on availability. The system is integrated with the ThingSpeak IoT cloud for data storage and processing, enabling basic machine learning analytics to predict usage patterns and optimize parking management. This cost-effective and scalable solution addresses the growing need for automated parking systems in urban areas, reducing manual effort and enhancing user convenience.

Hardware Components Used

1. **Raspberry Pi Pico W**
2. **Servo Motor (SG90)**
3. **IR object Avoidance sensor**
4. **Ultrasonic Sensor (HC-SR04)**
5. **Bread-Board and Connecting Wires**

DEVELOPMENT BOARD SPECIFICATION

Raspberry Pi Pico W

The Raspberry Pi Pico W is a compact and versatile development board designed for embedded systems and IoT applications. Based on the RP2040 microcontroller, it features dual ARM Cortex-M0+ processors, making it capable of handling real-time operations efficiently. The inclusion of built-in Wi-Fi connectivity makes the Pico W an excellent choice for IoT projects requiring wireless communication.

Key Specifications:

Microcontroller: RP2040 (Dual ARM Cortex-M0+ processors, up to 133 MHz)

Memory: 264 KB SRAM

Storage: Supports up to 16 MB of external QSPI Flash (2 MB Flash onboard)

Wireless Connectivity: 802.11n Wi-Fi (2.4 GHz)

GPIO Pins: 26 multifunctional pins, including I2C, SPI, UART, ADC, and PWM

Power Supply: 1.8–5.5V input via USB or external supply

Operating Voltage: 3.3V for GPIO

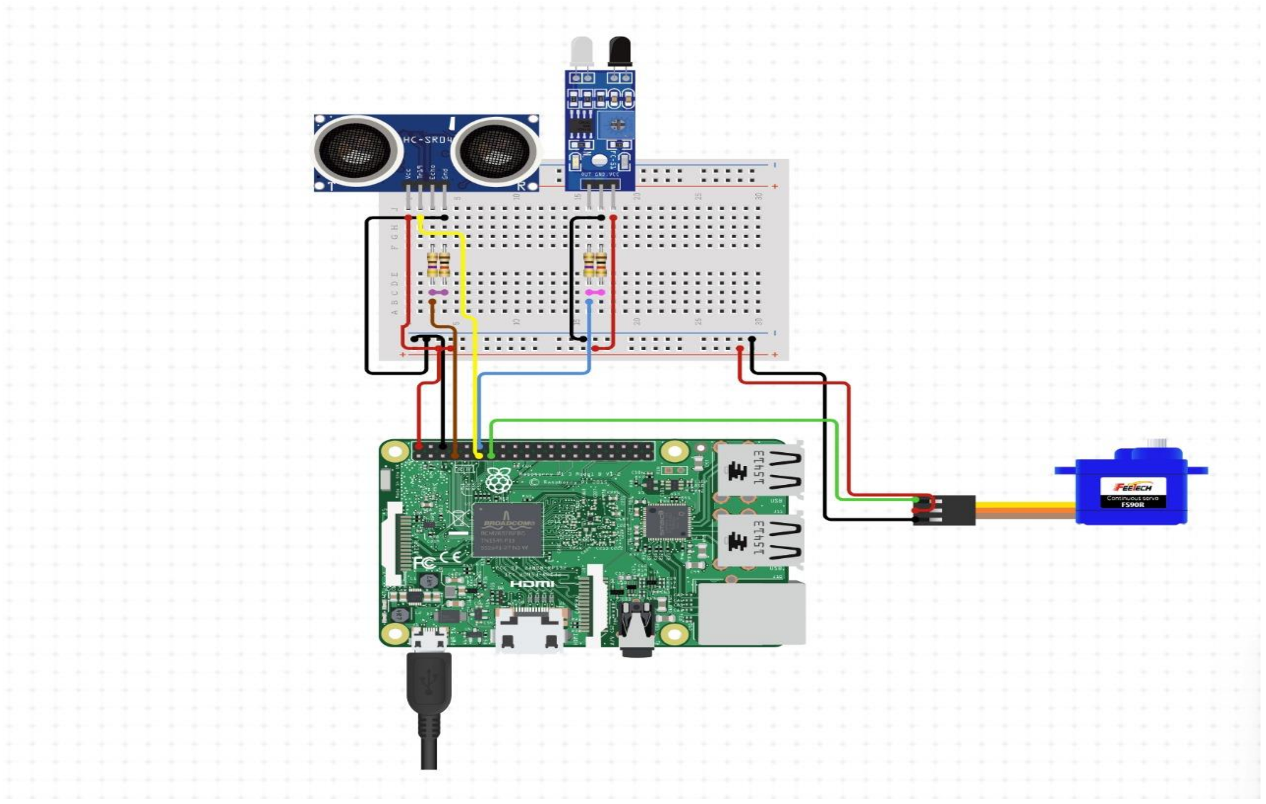
Dimensions: 51 x 21 mm

Temperature Range: -20°C to +85°C

USB Interface: Micro-USB for power and programming

Programming Languages: Supports C/C++ and MicroPython

CIRCUIT DIAGRAM



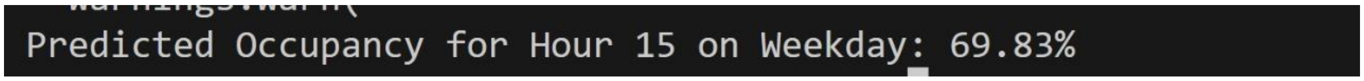
PREDICTIVE ANALYSIS

The smart parking system incorporates machine learning to perform occupancy prediction, enhancing its functionality and efficiency. By analyzing data collected from ultrasonic sensors, the system forecasts parking slot availability based on real-time and historical data patterns. The occupancy prediction model helps anticipate slot usage during different times of the day, ensuring efficient space utilization and minimizing user search time.

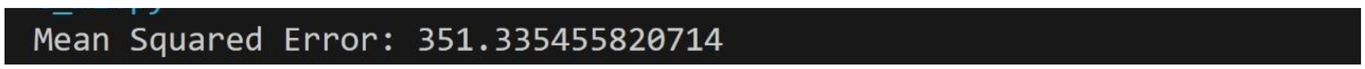
CODE (for temperature data collection and sending to cloud using esp8266):

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Load the dataset from CSV
file_path = "parking_data.csv"  # Replace with your file path
data = pd.read_csv(file_path)

# Clean up column names to remove any extra spaces
data.columns = data.columns.str.strip()

# Define features (X) and target (y)
X = data[["Hour of Day", "Day Type"]]
y = data["Current Occupancy"]  # Updated to remove extra space

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)

# Example prediction
hour = 15  # Example hour
day_type = 0  # Example day type (0=Weekday)
predicted_occupancy = model.predict([[hour, day_type]])
print(f"Predicted Occupancy for Hour {hour} on {'Weekday' if day_type == 0 else
'Weekend'}: {predicted_occupancy[0]:.2f}%")
```

OUTPUT:



```
warnings.warn(
Predicted Occupancy for Hour 15 on Weekday: 69.83%
```


```
Mean Squared Error: 351.335455820714
```

MODEL AND METHODOLOGY

To predict parking slot occupancy, we used a Linear Regression model, which is suitable for predicting a continuous variable based on numerical and categorical input features. The dataset included three columns: Hour of Day, Day Type (Weekday or Weekend), and Current Occupancy (target variable representing the parking occupancy percentage).

The Day Type was encoded as a numerical feature (0 for Weekday, 1 for Weekend), and the data was split into input features (Hour of Day and Day Type) and the target variable. The model was trained to minimize Mean Squared Error (MSE), which measures the difference between predicted and actual occupancy.

The model's performance was evaluated with an MSE of 351.34, indicating the model's accuracy in predicting occupancy. For example, the model predicted 69.83% occupancy for Hour 15 on a Weekday. While the model is functional, improvements could be made by incorporating more data and additional features to enhance prediction accuracy.

Cloud Platform Used:

The project utilizes ThingSpeak, a cloud-based IoT analytics platform, to store, process, and visualize timestamp data collected during entry/exit of a car . ThingSpeak provides a robust framework for real-time data monitoring.

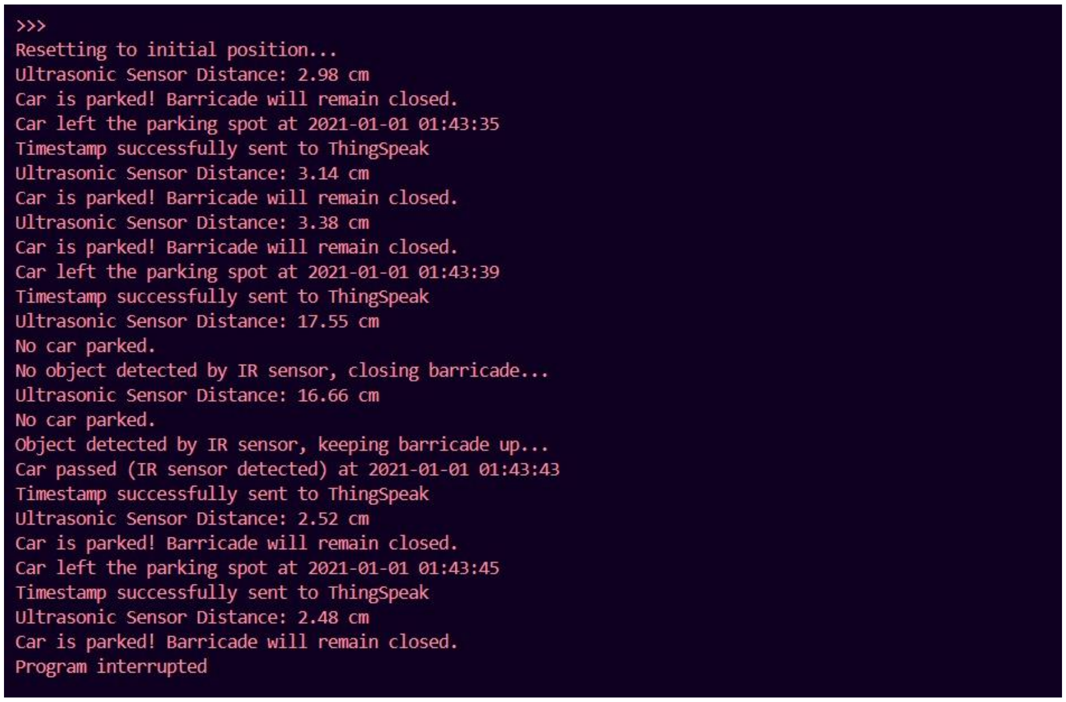
Reason for using ThingSpeak:

ThingSpeak is specially designed for IoT applications and o ers the following advantages:

- Real-Time Data Logging: Allows for the collection and storage of sensor data with timestamps.
- Data Visualization: Provides built-in widgets to graph temperature data and analyse trends.

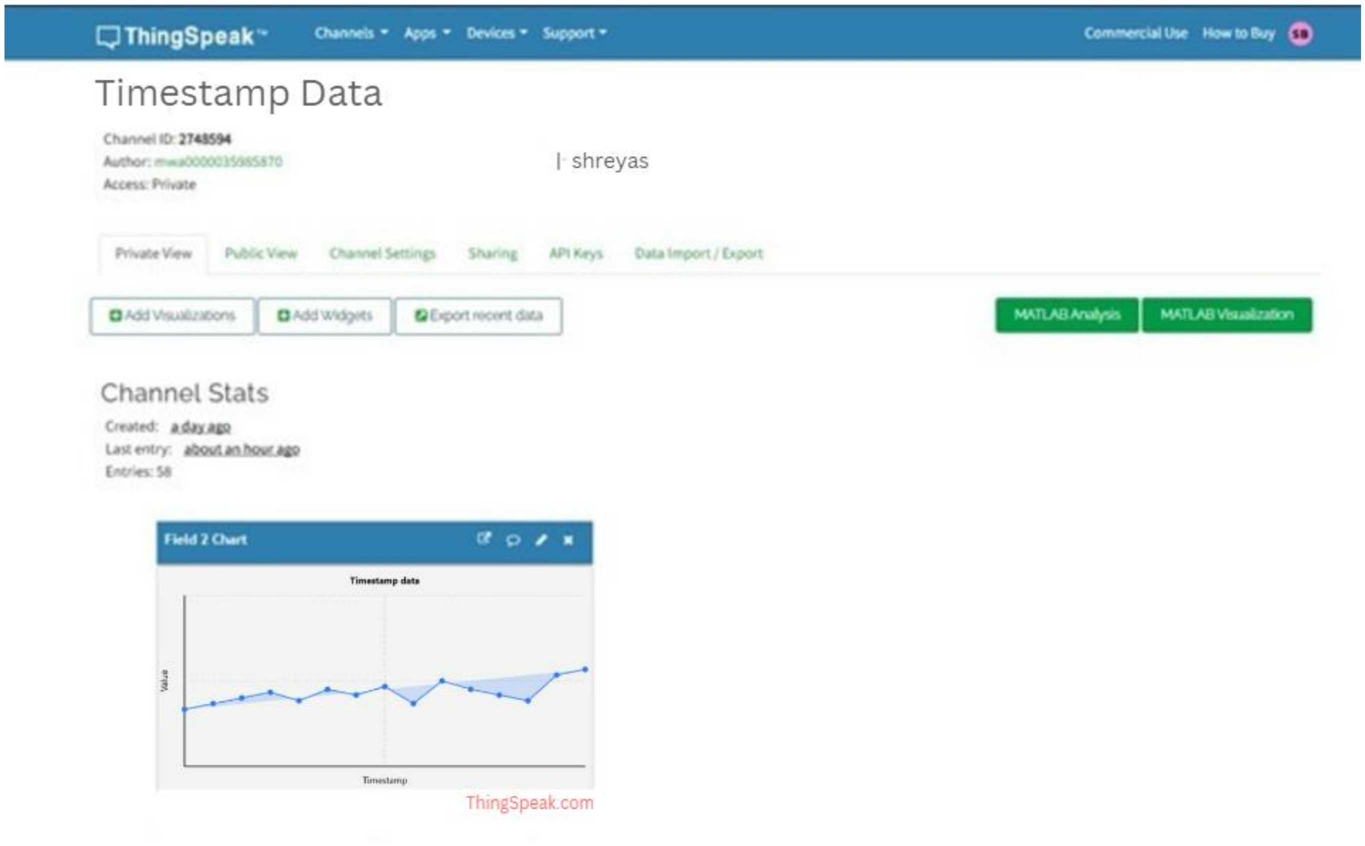
- RESTful API: Uses a simple HTTP-based API for secure data transmission from IoT devices.
- Integration with MATLAB: Enables advanced analytics and predictive modelling.
- CSV Export: Supports easy data export for one analysis and machine learning model training

<u>TERMINAL OUTPUT (for sending data to thingspeak):</u>



```
>>>
Resetting to initial position...
Ultrasonic Sensor Distance: 2.98 cm
Car is parked! Barricade will remain closed.
Car left the parking spot at 2021-01-01 01:43:35
Timestamp successfully sent to ThingSpeak
Ultrasonic Sensor Distance: 3.14 cm
Car is parked! Barricade will remain closed.
Ultrasonic Sensor Distance: 3.38 cm
Car is parked! Barricade will remain closed.
Car left the parking spot at 2021-01-01 01:43:39
Timestamp successfully sent to ThingSpeak
Ultrasonic Sensor Distance: 17.55 cm
No car parked.
No object detected by IR sensor, closing barricade...
Ultrasonic Sensor Distance: 16.66 cm
No car parked.
Object detected by IR sensor, keeping barricade up...
Car passed (IR sensor detected) at 2021-01-01 01:43:43
Timestamp successfully sent to ThingSpeak
Ultrasonic Sensor Distance: 2.52 cm
Car is parked! Barricade will remain closed.
Car left the parking spot at 2021-01-01 01:43:45
Timestamp successfully sent to ThingSpeak
Ultrasonic Sensor Distance: 2.48 cm
Car is parked! Barricade will remain closed.
Program interrupted
```

<u>DATA VISUALIZATION IN THINGSPEAK:</u>



ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy

Timestamp Data

Channel ID: 2748594
Author: mwa0000035985870
Access: Private

shreyas

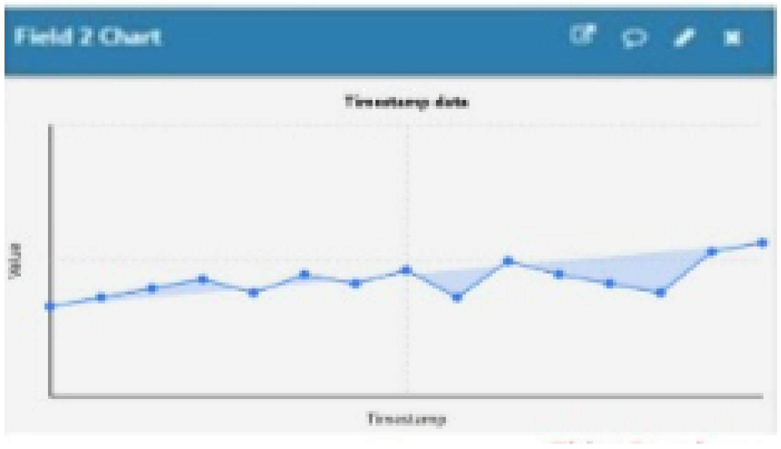
Private View | Public View | Channel Settings | Sharing | API Keys | Data Import / Export

Add Visualizations | Add Widgets | Export recent data | MATLAB Analysis | MATLAB Visualization

Channel Stats

Created: a day ago
Last entry: about an hour ago
Entries: 58

GRAPH:



PICTURES OF HARDWARE:

zon.in
KT8
KT8