NEW HORIZON
COLLEGE OF ENGINEERING

Autonomous College Permanently affiliated with VTU, approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, accredited by NBA

DEPARTMENT OF ARTIFICIAL ENGINEERING AND MACHINE LEARNING

"Smart Parking Management System"

COURSE NAME : MINI PROJECT

COURSE CODE:22AIM48

Presentation by,

Anu Dsilva- 1NH22AI020

Deekshitha Ganiga-1NH22AI037

Harshitha P Shetty-1NH22AI052

Under the Guidance of ,

Ms. Rajashree

Asst professor

NHCE

OVERVIEW

- Introduction
- Project planning and Research
- Existing Systems
- Required Components
- Hardware setup
- Software setup
- Prototype Development
- Initial Code
- Data Communication and Analysis
- Conclusion

Introduction

“Parking Jam!!? ”
PARK-PILOT got the plan!

The **Park-Pilot** system revolutionizes parking management using IoT technology to automate and optimize the process. By detecting vehicle entry and exit, Park-Pilot accurately tracks available parking slots in real-time and controls the entry and exit gates based on slot availability, ensuring efficient space utilization.

Drivers receive immediate information on available parking spaces, reducing search time and enhancing convenience. Park-Pilot opens the entry gate if slots are available and updates the slot count, while the exit gate opens and adjusts the count when a vehicle leaves. This system significantly improves parking efficiency and user experience through smart automation.

Park-Pilot project aims to develop a smart, automated parking management system designed to optimize parking space usage in urban environments

Scope of the Domain

- Design and Implementation
- System Components
- User Interaction
- Scalability

Objectives

- Optimize Parking Space Utilization
- Provide Real-Time Information
- Enhance User Experience

Application

- Urban Parking Facilities
- Shopping Malls
- Event Venues and ETC

EXISTING SYSTEM

Current parking systems can be quite hectic and inefficient. Manual systems depend on attendants to manage parking, which is labor-intensive, error-prone, and often leads to congestion. Basic parking meters and ticket systems offer limited convenience and lack real-time updates on available spaces leading to a frustrating and inefficient parking experience for users

Limitations

- Labor-Intensive Operations
- Limited Convenience
- No Real-Time Updates
- Errors and User Frustration

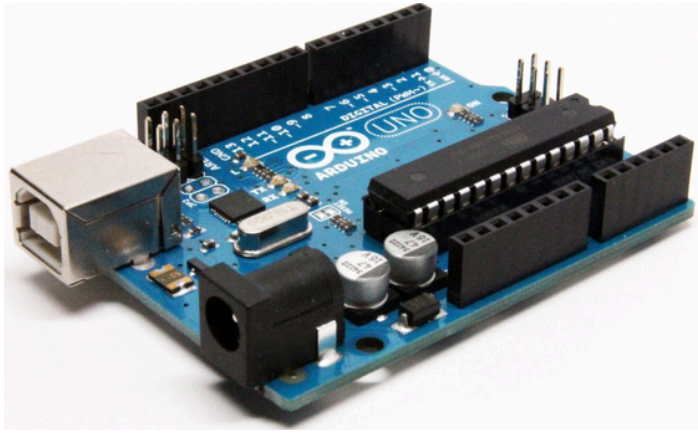
Scope of Improvement

- Advanced Technology Integration
- Cost-Effective Solutions
- Data Collection and Analytics
- Efficienct and user friendly

Hardware setup

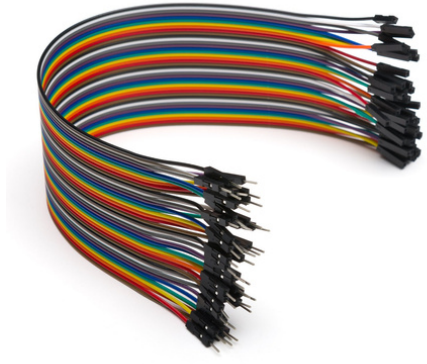
SERVO MOTOR

IR SENSOR

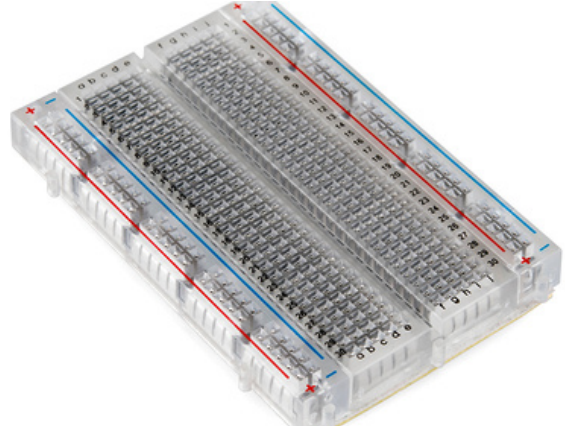


ARDUINO UNO

LCD MODULE



JUMPER WIRES



BREADBOARD

Component	Uses	Function
Arduino Uno	Central control unit	Processes sensor inputs, controls servos and display, executes system algorithms.
IR Sensors(2)	Vehicle detection	Entry Sensor: Detects vehicles entering. Exit Sensor: Detects vehicles exiting
Servo Motors (2)	Gate control for entry and exit	Entry Gate Servo: Manages vehicle entry. Exit Gate Servo: Manages vehicle exit.
16x2 LCD Display	Displays available parking slots	Shows the number of available parking slots to drivers.
Breadboard	Component prototyping	Facilitates the setup and testing of circuit connections.
Jumper Wires	Connects components	Establishes electrical connections between sensors, servos, and the display

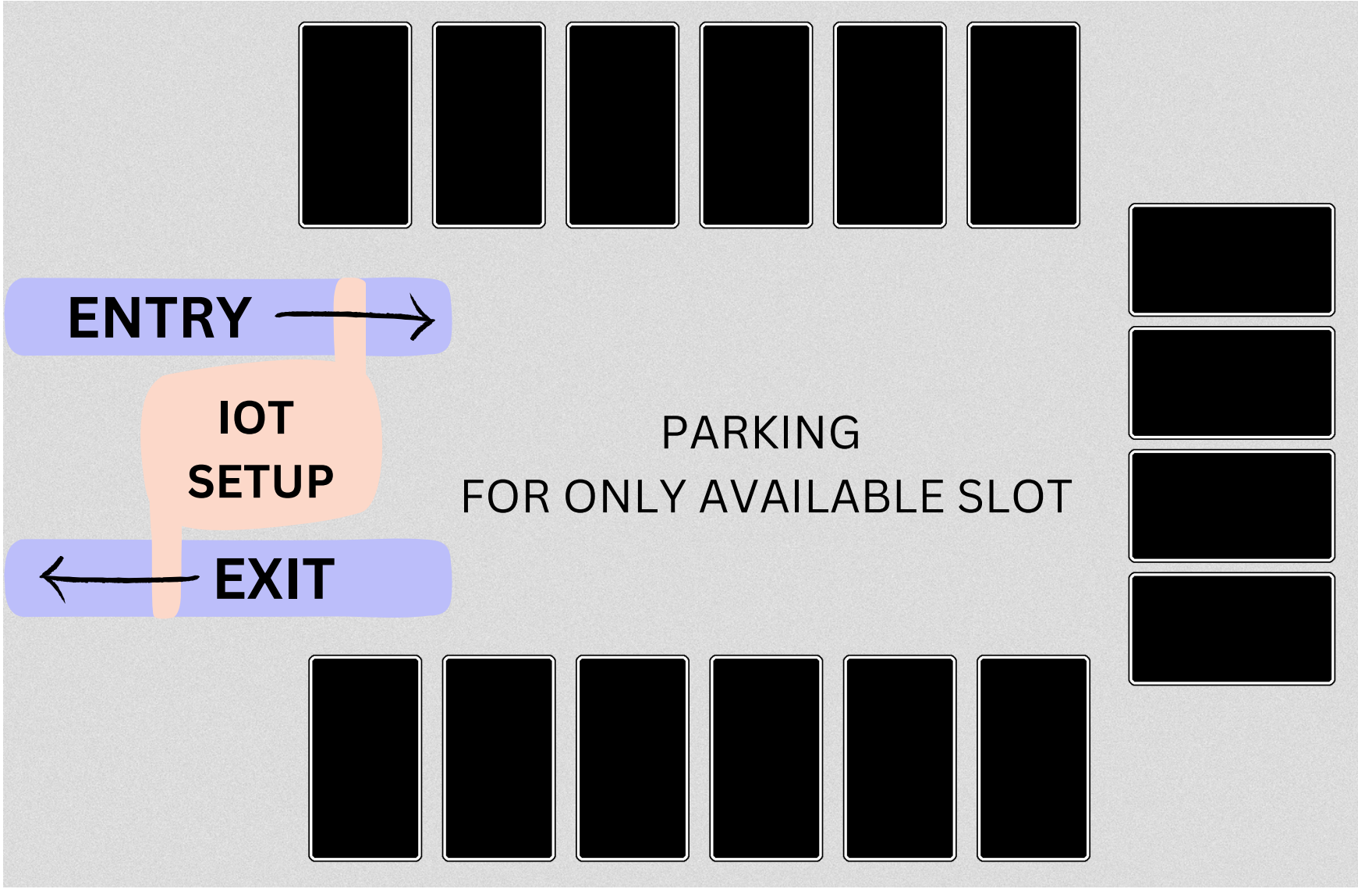
Software setup

The Arduino IDE is the development environment used for the Park-Pilot project. It provides a platform to write, compile, and upload code to the Arduino Uno.

- Code Editor: Write and edit the program code.
- Compiler: Converts code into machine language.
- Uploader: Transfers the code to the Arduino board.
- Serial Monitor: Views real-time data for debugging.

Role in Park-Pilot: It enables the creation and management of the code that controls sensors, servos, and the LCD display, ensuring efficient parking management.

Prototype Development



INITIAL CODES

Servo Motor	IR Sensor	16x2 LCD Display

#include <Servo.h>
 	
Servo servoEntry;
Servo servoExit;
 	
void setup() {
servoEntry.attach(3);
servoExit.attach(5);
servoEntry.write(100);
servoExit.write(100);
}
	const int IR_Entry = 2;
	const int IR_Exit = 4;
	void setup() {
	pinMode(IR_Entry, INPUT);
	pinMode(IR_Exit, INPUT);
	}
	void loop() {
	int sensorEntry
	= digitalRead(IR_Entry);
	int sensorExit
	= digitalRead(IR_Exit);
		#include
		<LiquidCrystal.h>
		
		LiquidCrystal lcd(12, 11, 10, 9, 8, 7);
		
		void setup() {
		lcd.begin(16, 2);
		lcd.print("Slots Available:"); }

Data Communication and Analysis

Aspect	Details
Communication Type	Digital Communication
Data Flow	Sensors → Arduino → LCD Display
IR Sensors	**Detect Vehicle Presence**: Sends a HIGH or LOW signal to the Arduino based on whether a vehicle is present at the entry or exit
Arduino UNO	**Data Processing:** Processes sensor data to manage parking slots and control the entry and exit gates. Updates the LCD display with the number of available slots.
LCD Display	**Information Display:** Shows the number of available parking slots to users
Servo Motors	**Gate Control:** Opens or closes the entry and exit gates based on parking availability and vehicle presence

Conclusion

Park-Pilot offers a smart solution for parking management, combining IR sensors for vehicle detection with servo motors for gate control and an LCD display for slot updates. This system simplifies parking processes and provides real-time availability information.

By leveraging IoT technologies, Park-Pilot showcases how innovative solutions can address everyday challenges in parking management, making it a practical and scalable tool for improving efficiency and user experience.

References

- Arduino IR Sensor Tutorial
- Arduino Servo Motor Library
- Arduino LCD Library Documentation

Thank you

Queries ?