



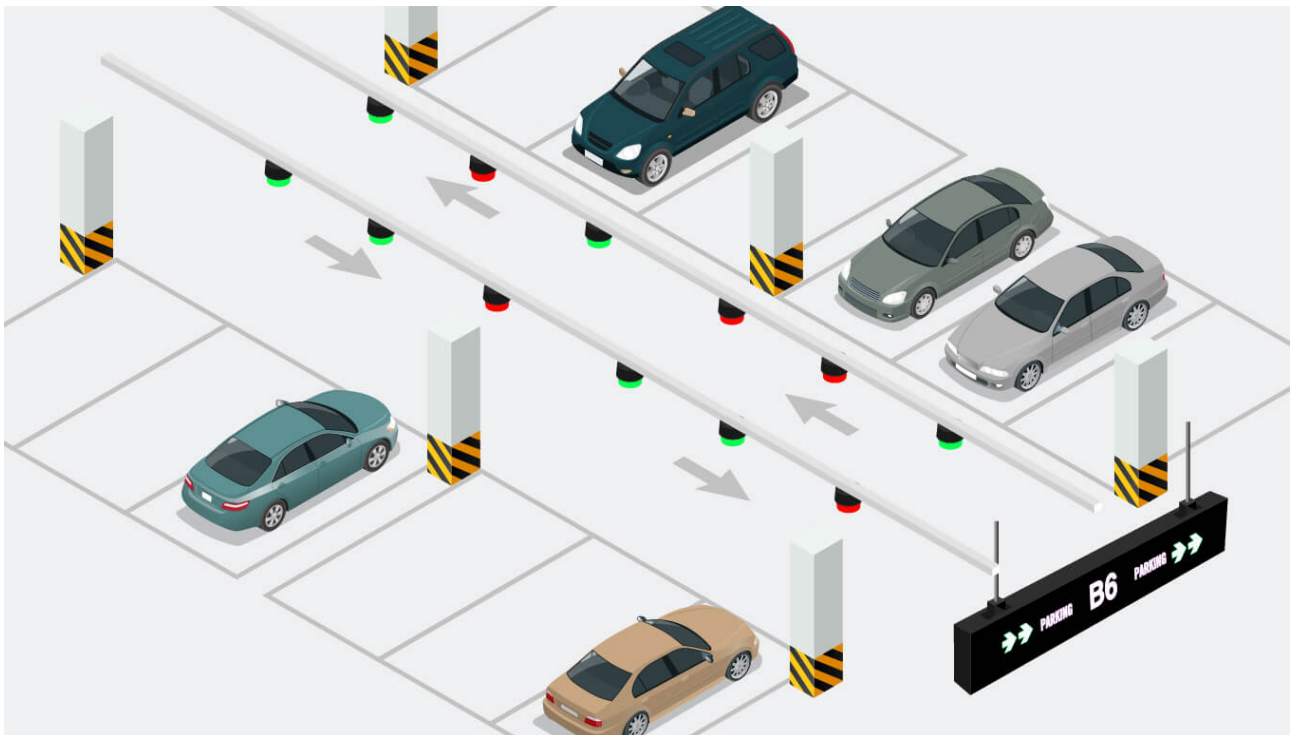
Smart Automatic Car Parking System Using Arduino and Proteus Simulation

Professor: **FILIPPO D'IPPOLITO**

Group composed by:

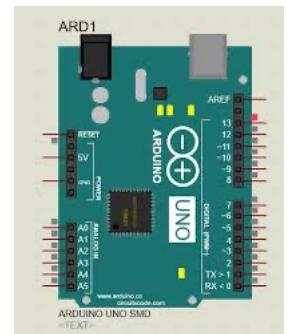
YASIN SHADROUH - 0780715

HAMED NAHVI - 0780707



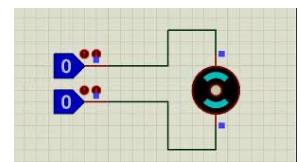
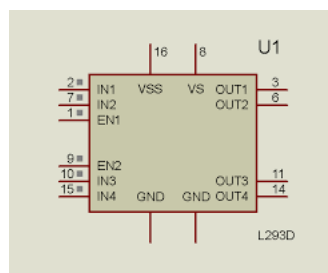
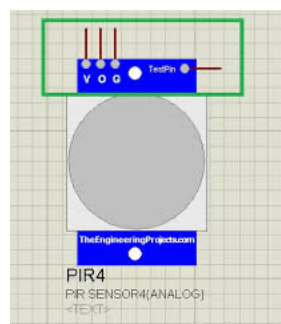
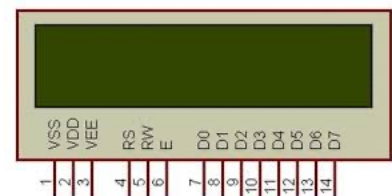
Devices used

Arduino UNO Microcontroller
PIR Sensor
Simple DC Motor
LM016 Display
L293d Motor Driver



Software used

Type of software → *Proteus + Arduino IDE*





Project idea:

An Arduino-based automatic parking system project simulated in Proteus. It uses sensors and control logic to detect vehicles and manage parking slots efficiently. The project showcases how embedded systems can automate parking operations with real-time feedback and smart control.

Project Description:

This project demonstrates a smart parking management system for a four-slot parking area, designed and simulated using Arduino Uno and Proteus. The system provides an automated solution to control vehicle entry and manage limited parking spaces using sensors, DC motors, and an LCD interface.

The main objective is to **monitor parking space availability** and manage the **entry gate** based on current occupancy. The project is based on logic to meet small-scale parking infrastructure needs.

System Features

1. Parking Space Detection

- The system constantly monitors the status of four individual parking spaces using IR sensors.
- Occupied or vacant states are tracked and updated in real time.

2. Entry Gate Control with DC Motor

- A DC motor simulates a gate barrier.
- When all parking spaces are full, the gate automatically closes to prevent further entry.
- Once at least one car leaves and space becomes available, the gate reopens.



3. Real-Time Monitoring via LCD

- A 16x2 LCD display shows:
 - Number of available parking slots.
 - Entry gate status (OPEN or CLOSED).
 - Live parking state updates, offering feedback for both users and operators.

4. Operator-Friendly Interface

- Operators can visually monitor the number of cars in the lot and the current state of the gate.
- Ensures effective control of traffic inflow, especially when full occupancy is reached.

Working Principle Summary

- When a car approaches the entrance, IR sensors detect its presence.
- If slots are available, the gate opens automatically via DC motor control.
- Once the car enters, the system updates the LCD and decreases the free slot count.
- If the parking is full, the gate remains closed until a car exits.
- Exit detection is also done via sensors, and upon detection, the free slot count increases and the gate may reopen.

Applications

- Shopping malls
- Residential complexes
- Office buildings
- Airports and public smart parking systems



Code Overview:

1. Sensors (PIR)

- Four digital pins (8–11) are connected to parking spot detectors (PIR/IR sensors).
- If a sensor reads **LOW**, it means a car is present (**Full**); if **HIGH**, the spot is **Open**.

2. Gate Motor Control

- Pins 6 and 7 control the **DC motor** for gate operation:
 - motorOpen (pin 6) activates the motor to open the gate.
 - motorClose (pin 7) activates it to close the gate.
- The system **automatically closes the gate** when all spots are full and **reopens it** when at least one becomes free.

3. LCD Display

- A 16x2 LCD connected via pins 0–5 displays:
 - Welcome message on startup.
 - Current status of each parking slot: F (Full) or O (Free).
 - Messages like “**Parking is Full!**”, “**Gate Closing...**”, and “**Gate Opening...**” to indicate real-time actions.

4. Gate Status Tracking

- A Boolean variable GateCloseStatus tracks whether the gate is currently open or closed to **prevent redundant motor actions**.



Summary:

This project presents a smart automatic car parking system designed using Arduino Uno and simulated with Proteus software. The system automates a small-scale four-slot parking facility by employing IR sensors to detect vehicle presence, a DC motor to control an entry gate, and a 16x2 LCD display for real-time feedback. When a car approaches the entrance, sensors check for available space; if at least one slot is free, the gate opens. The LCD updates the number of free slots and gate status dynamically. If all spots are full, the system keeps the gate closed until a car exits, ensuring efficient traffic control.

The system's core components include an Arduino Uno, PIR/IR sensors for slot detection, a DC motor managed by an L293D driver for gate operation, and an LCD for monitoring. It provides a reliable embedded solution for places like malls, residential complexes, or airports, where space management is crucial. The logic-driven design ensures minimal manual intervention and smooth operation. The code handles sensor input, gate motor control, and real-time display, with gate status tracking to avoid unnecessary motor actions, making it an efficient and scalable model for smart parking solutions.