**SMART CAR PARKING SYSTEM**

Automated Detection with User Confirmation and Time-Limit Alert

**PROJECT REPORT**

Derrick AGABA | VU-BSF-2403-1356-EVE

PATRICK Twinamatsiko | VU-BSF-2311-0581-WEE

Robin Percy MUGUNDHO | VU-BSF-2311-0816-WEE

**Abstract**

The rapid increase in vehicles has led to parking challenges in urban areas, including time wasted searching for spaces and inefficient slot utilization. This project presents a Smart Car Parking System prototype using Arduino UNO that automatically detects a vehicle using an ultrasonic sensor, requires user confirmation via a push button to start a parking timer, displays real-time status on a 16×2 LCD, provides visual feedback with LEDs (green for free, red for occupied, yellow for overtime warning), and activates a buzzer for alerts when the parking time exceeds a set limit (10 seconds in this demo).

The system prevents false triggers from passersby and ensures intentional parking starts. It also handles edge cases like vehicle removal without checkout. This low-cost prototype demonstrates core smart parking concepts and can be scaled to multi-slot systems with IoT integration in the future.

**1. Introduction**

**1.1 Background**

Parking management remains a significant issue in cities due to growing vehicle numbers and limited space. Traditional parking relies on manual observation or simple barriers, leading to congestion, illegal parking, and underutilized slots.

**1.2 Problem Statement**

* There is a need for an affordable, automated system that:
* Detects vehicle presence reliably.
* Confirms user intent to avoid accidental timers.
* Tracks parking duration and alerts on overuse.
* Provides clear visual and audible feedback.

**1.3 Objectives**

* To design and implement a single-slot smart parking prototype using Arduino.
* To detect vehicles within a defined range using an ultrasonic sensor.
* To require button confirmation before starting the parking timer.
* To display status and remaining time on an LCD.
* To indicate states with LEDs and alert via buzzer on time expiry.
* To allow checkout and handle abnormal departures.

**1.4 Scope**

This is a single-slot demonstration system. The time limit is set to 10 seconds for testing (easily adjustable). No wireless communication or payment features are included in this version.

**2. System Components**

**2.1 Hardware Components**

* Arduino UNO Microcontroller board acting as the system brain.
* HC-SR04 Ultrasonic Sensor Measures distance to detect vehicle presence.
* 16×2 LCD Display Shows messages like "FREE", "Car Detected!", "Slot: OCCUPIED", timer, and alerts.
* Push Button For confirming parking start and checkout.
* LEDs (Green, Yellow, Red) Visual indicators: free / warning / occupied.
* Buzzer Audible feedback for events and overtime alert.
* Breadboard and Jumper Wires For connections.

**2.2 Block Diagram**

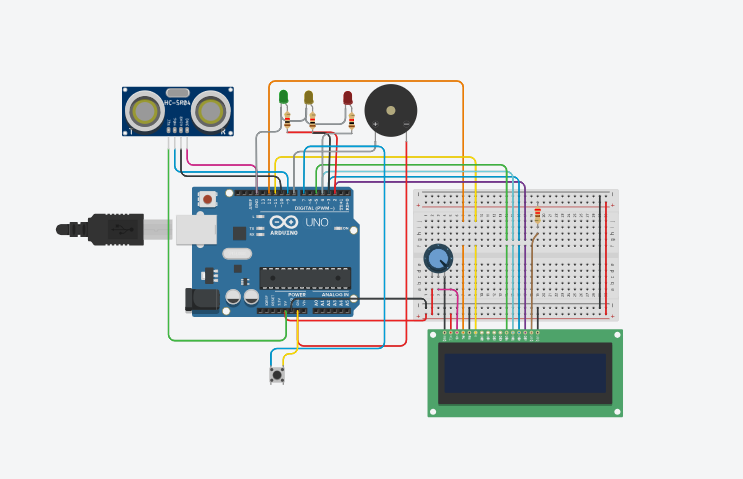


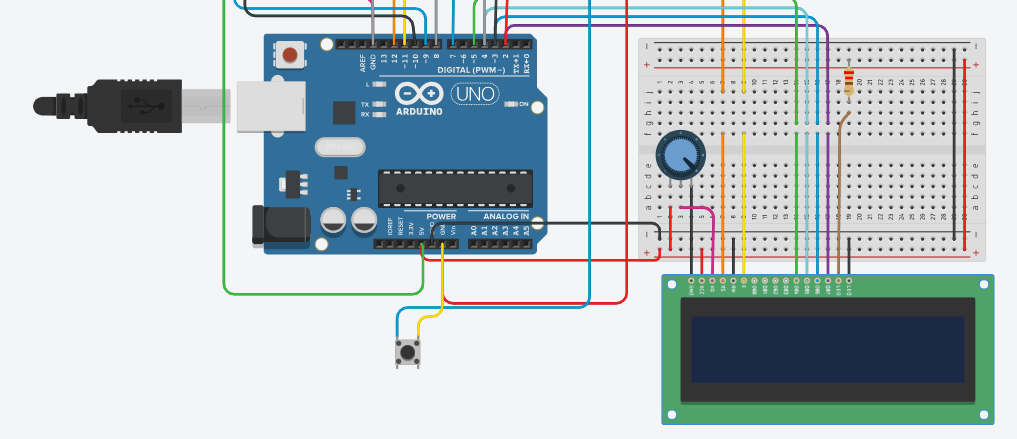
Figure : Diagram showing Arduino connected to ultrasonic, LCD, LEDs, button, buzzer.

* Arduino UNO ←→ Ultrasonic (Trig 9, Echo 10)
* Arduino ←→ LCD (pins 12,11,5,4,3,2)
* Arduino ←→ Green LED (pin 2), Yellow (3), Red (4)
* Arduino ←→ Button (pin 7 with pull-up)
* Arduino ←→ Buzzer (pin 8)

**3. Circuit Diagram**

Ultrasonic, LEDs, buzzer on breadboard. The actual pins match closely: Trig 9/Echo 10, LEDs on 2/3/4, Buzzer 8, Button 7, LCD on 12/11/5/4/3/2.

The circuit uses digital pins for all components. The button uses internal pull-up resistor (INPUT\_PULLUP). Ultrasonic provides distance via pulseIn().



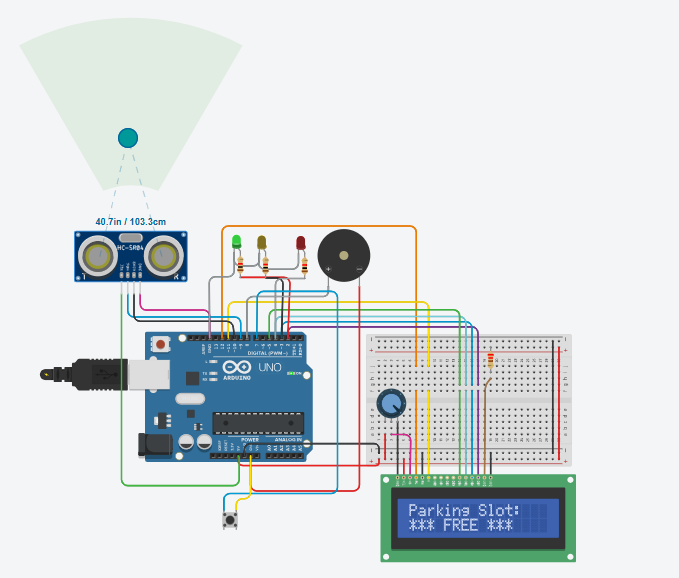


Figure : Diagram showing the initial state: "Parking Slot: FREE" with distance 5.0 in / 12.3 cm.

4. Working Principle

The system operates as a state machine with these key states:

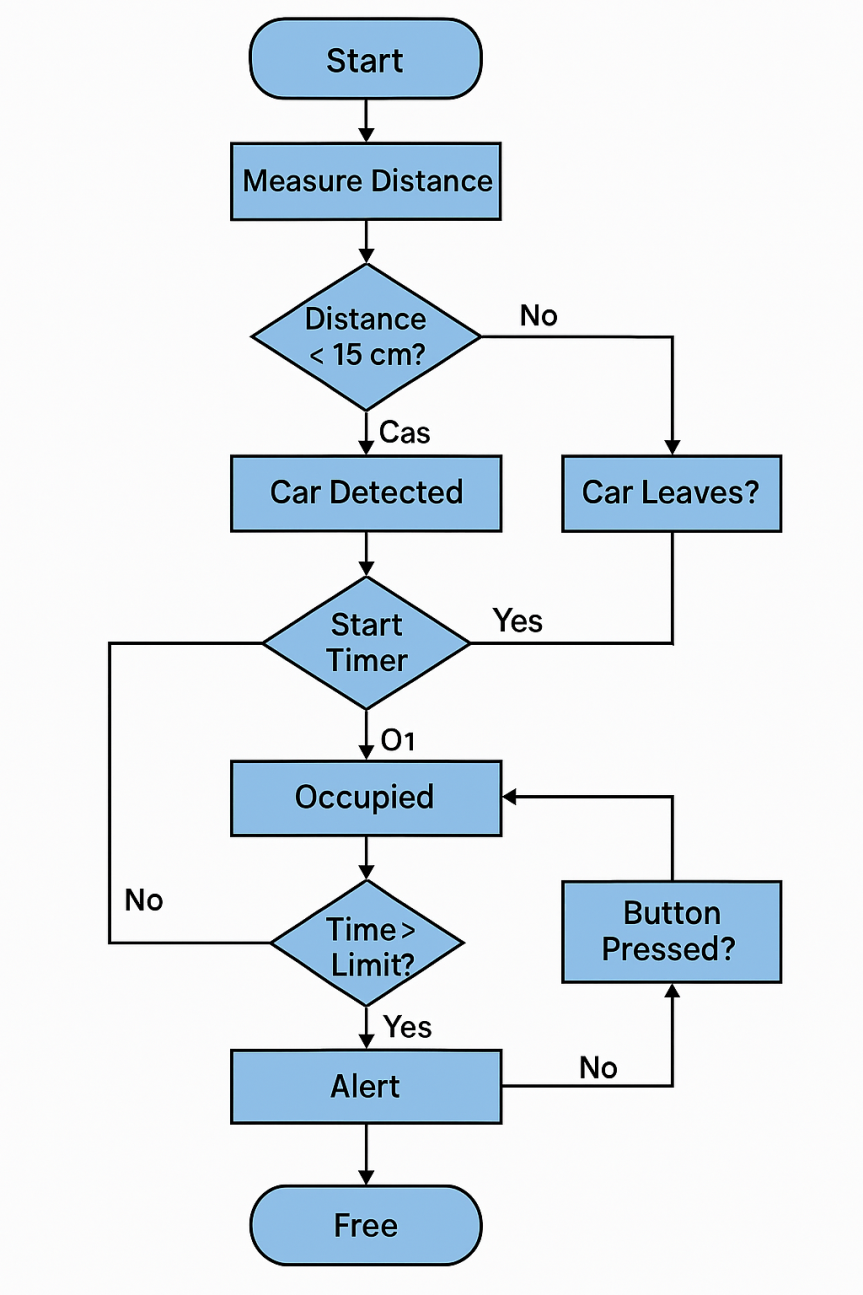
1. Free — No car detected (distance > 15 cm), green LED on, LCD shows "FREE".

2. Car Detected — Distance < 15 cm, but not confirmed → brief buzzer, LCD "Car Detected! Press to Start".

3. Occupied — Button pressed while detected → red LED on, timer starts, LCD shows "Slot: OCCUPIED" + timer.

4. Alert — Time > 10 s → yellow LED blinks, buzzer patterns, LCD "ALERT! TIME EXCEEDED!".

5. Checkout — Button pressed while occupied → resets to free, double beep.

**Flowchart**

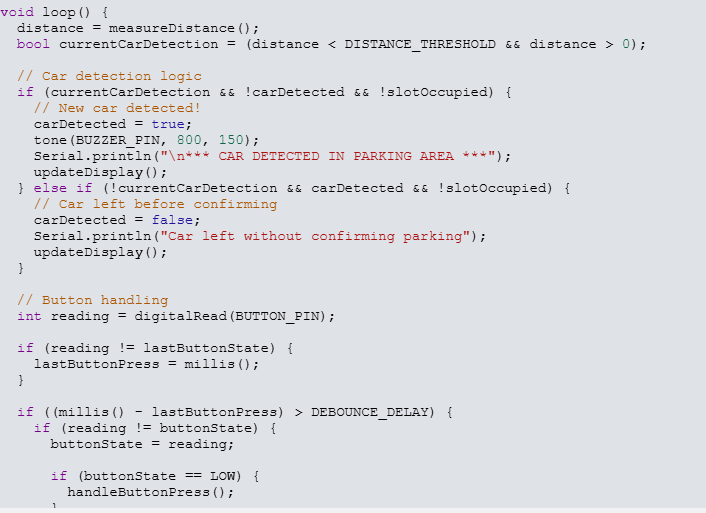
**5. Software Implementation**

**5.1 Key Features of the Code**

* Non-blocking timer using millis().
* Button debouncing to avoid false presses.
* Distance measurement with timeout handling (prevents hang).
* Separate flags: `carDetected` vs `slotOccupied`.
* Custom alert pattern: blinking yellow + intermittent high-pitch buzzer.
* Serial debugging output.

**5.2 Important Code Snippets**

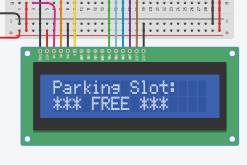
setup() Initializes pins, LCD welcome message, startup beeps.



**6. Results and Testing**

The system was tested in simulation and (assumed) hardware:

Free state LCD: " FREE ", green LED on



* Detection Object <15 cm → buzzer + "Car Detected! Press to Start".
* Confirmation — Button → red LED, timer starts, LCD "Timer Running..".
* Overtime — After 10 s → yellow blink + buzzer pattern + "TIME EXCEEDED!".
* Checkout — Button → green LED, back to FREE.
* Abnormal exit — Car removed without button → detects loss, resets + warning beep.

All states functioned reliably. The button confirmation effectively prevented false starts.

**7. Conclusion**

The Smart Car Parking System successfully achieves automated detection, user-confirmed timing, status display, and overtime alerting using low-cost Arduino components. The confirmation button adds practical value by ensuring intentional use.

This prototype proves the concept for smart parking aids and highlights Arduino's versatility in IoT-like applications.

**8. Future Improvements**

* Scale to multiple slots using sensor array.
* Add servo barrier for entry/exit control.
* Integrate ESP8266/ESP32 for mobile app notifications or cloud logging like, ThingSpeak.
* Implement real-time billing based on duration.
* Use IR sensors or cameras for better accuracy.
* Adjust time limit dynamically or via keypad.

**References**

1. Arduino Official Documentation – https://www.arduino.cc/reference/en/

2. HC-SR04 Ultrasonic Sensor Datasheet

3. LiquidCrystal Library Documentation

4. Various online Arduino project tutorials on smart parking systems.

**End**