LAB REPORT

IRE 212 : IoT Architecture and Technologies Sessional

PREPARED BY

Mehrin Farzana
ID: 2101013
Session: 2021-2022
Date: 22/11/2024

SUPERVISED BY

Suman Saha
Assistant Professor & Chairman
Department of IRE, BDU

Bangabandhu Sheikh Mujibur Rahman
Digital University

BANGABANDHU SHEIKH MUJIBUR

RAHMAN DIGITAL UNIVERSITY

(BDU)

List of Experiments

a) Smart Car Parking Monitoring System Using Arduino

Experiment No.: 01
Experiment Statement: Smart Car Parking Monitoring System Using Arduino

Theory:
Smart car parking systems are an integral part of modern infrastructure, designed to optimize parking space utilization, reduce manual effort, and enhance convenience. These systems use sensors to monitor parking availability and automate gate operations.

In this experiment, we demonstrate a simplified smart parking system with a single parking slot. The system uses an IR sensor to detect a car at the entry and another IR sensor to monitor the parking slot. A servo motor is employed to control the gate based on sensor inputs. LEDs provide visual feedback about the gate's status, and messages are displayed on the Serial Monitor for system monitoring.

Due to limited hardware, the system does not include an exit sensor, meaning it cannot automatically update the slot status when a car leaves. Despite this limitation, the experiment showcases the core functionality of an automated parking system, highlighting its potential for scalability and real-world application.

Components:
1.Arduino Uno – Microcontroller for system logic.
2.Servo Motor – Controls the parking gate.
3.IR Sensors – Two sensors used:
->One at the entry to detect cars approaching the parking lot.
->One to monitor the single parking slot.
4.LEDs –
->Red LED: Indicates that the gate is closed.
->Green LED: Indicates that the gate is open.
5.Resistors – 220Ω for LEDs.
6.Connecting Wires – For circuit connections.
7.Power Supply – To power the Arduino.

Circuit:

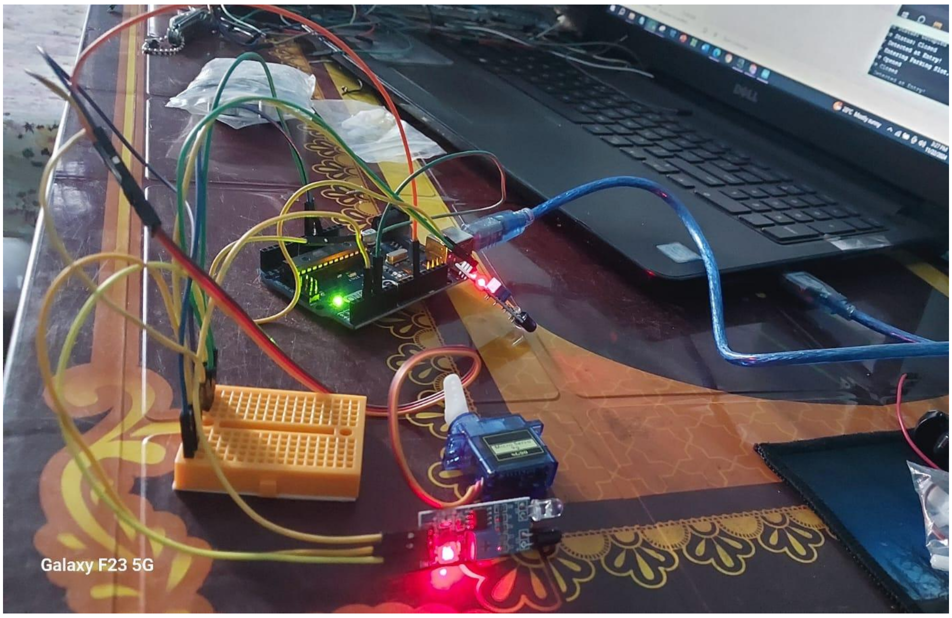


Fig 1: Circuit diagram of Smart Car Parking System

Code:

```
#include <Servo.h>
// Initialize Servo motor
Servo entryServo;
// Define pins for IR sensors
int irEntrySensor = 2; // IR sensor to detect car at entry
int irSlotSensor = 4;  // IR sensor for the single parking slot
// Define pins for indicator LEDs
int redLED = 6;       // Red LED pin for gate closed
int greenLED = 7;     // Green LED pin for gate open
bool isSlotOccupied = false; // Track parking slot status
// Function prototypes
void openGate();
void closeGate();
void setup() {
    Serial.begin(9600);
    // Initialize LEDs and IR sensors
    pinMode(redLED, OUTPUT);
    pinMode(greenLED, OUTPUT);
    pinMode(irEntrySensor, INPUT);
    pinMode(irSlotSensor, INPUT);

    // Attach the servo
    entryServo.attach(11);
    entryServo.write(180); // Initial position: Gate closed

    // Display initial status on Serial Monitor
    Serial.println("System Initialized.");
    Serial.println("Slot Status: Empty");
    Serial.println("Gate Status: Closed");
    digitalWrite(redLED, HIGH);  // Indicate gate closed
    digitalWrite(greenLED, LOW);
}

void loop() {
    // Read sensor states
    int entrySensorState = digitalRead(irEntrySensor);
    int slotSensorState = digitalRead(irSlotSensor);

    // Check the status of the parking slot
    isSlotOccupied = (slotSensorState == LOW);
    // Print car detection status
    if (entrySensorState == LOW) {
        Serial.println("Car Detected at Entry!");
    }
    if (!isSlotOccupied && entrySensorState == LOW) {
        Serial.println("Car Entering Parking Slot...");
        openGate();  // Open the gate if the slot is vacant
        delay(5000); // Simulate the car entering
        closeGate(); // Close the gate after the car enters
    } else if (isSlotOccupied) {
        Serial.println("Slot Status: Occupied");
        Serial.println("Gate Status: Closed");
    } else {
        Serial.println("Slot Status: Empty");
        Serial.println("Gate Status: Closed");
    }
    delay(1000); // Delay for readability
}
// Function to open the gate
void openGate() {
    entryServo.write(0);                // Open the gate
    digitalWrite(greenLED, HIGH);       // Turn on green LED
    digitalWrite(redLED, LOW);          // Turn off red LED
    Serial.println("Gate Opened");
}
// Function to close the gate
void closeGate() {
    entryServo.write(180);              // Close the gate
    digitalWrite(greenLED, LOW);        // Turn off green LED
    digitalWrite(redLED, HIGH);         // Turn on red LED
    Serial.println("Gate Closed");
}
```

Explanation of Code:

- **Setup Function**:

  - Initializes the servo motor and sets its initial position (closed).
  - Configures input pins for the IR sensors and output pins for the LEDs.
  - Prints an initial system status to the Serial Monitor.

- **Loop Function**:

  - **Reads the IR Sensors**:

    - **Entry Sensor**: Detects a car approaching the parking gate.
    - **Slot Sensor**: Checks if the parking slot is vacant or occupied.

  - **Gate Control Logic**:
    - If a car is detected at the entry and the slot is vacant, the gate opens for 5 seconds to allow the car to enter, then closes.
    - If the slot is occupied, the gate remains closed.

  - **Status Messages**:
    - Prints the gate and slot status to the Serial Monitor for debugging and monitoring purposes.

- **Gate Operations**:
  - **OpenGate()**:
    - Moves the servo motor to the open position.
    - Activates the green LED.
    - Displays `"Gate Opened"` on the Serial Monitor.

  - **CloseGate()**:

    - Moves the servo motor to the closed position.

- Activates the red LED.
- Displays `"Gate Closed"` on the Serial Monitor.

Output:

Slot Vacant, Car Detected:

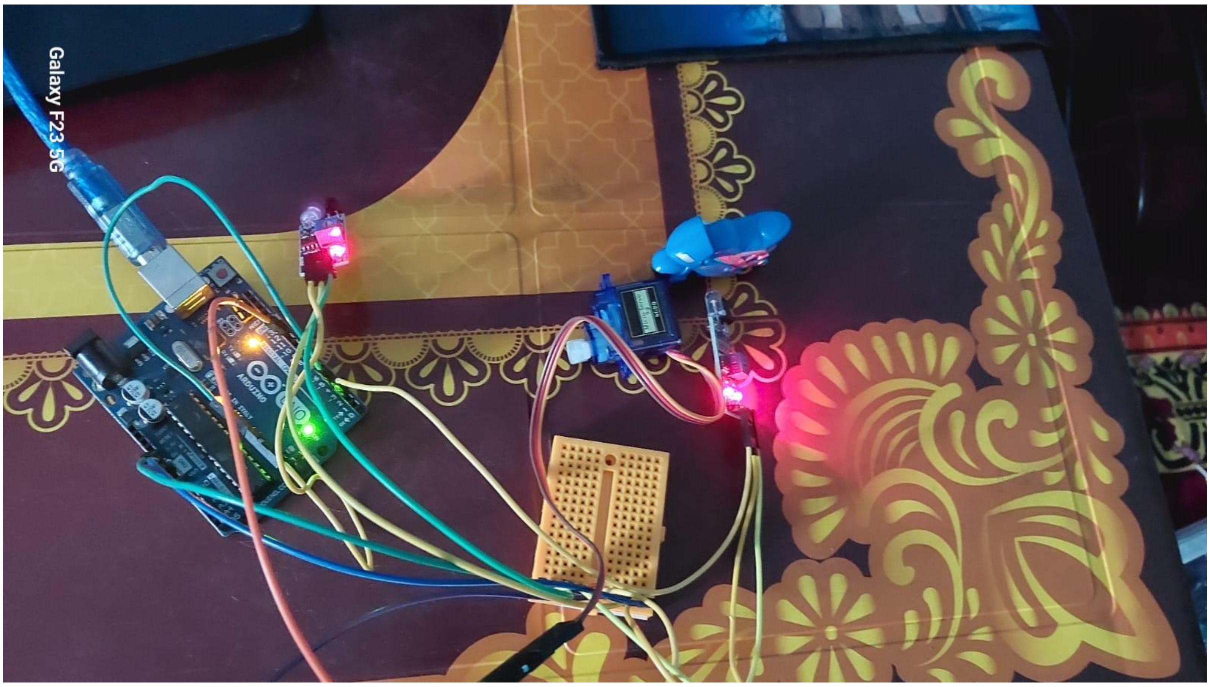
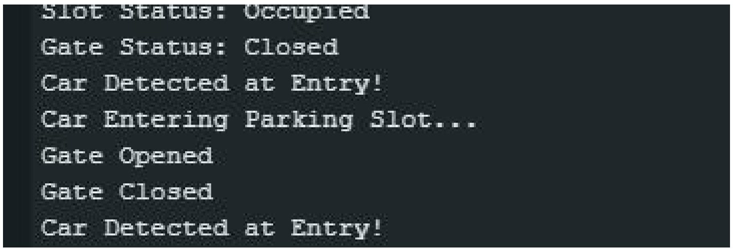


Fig 02: Slot Vacant, Car Detected

- Serial Monitor:



```
Slot Status: Occupied
Gate Status: Closed
Car Detected at Entry!
Car Entering Parking Slot...
Gate Opened
Gate Closed
Car Detected at Entry!
```

Fig 03: Serial Monitor for Slot Vacant, Car Detected

Slot Occupied:

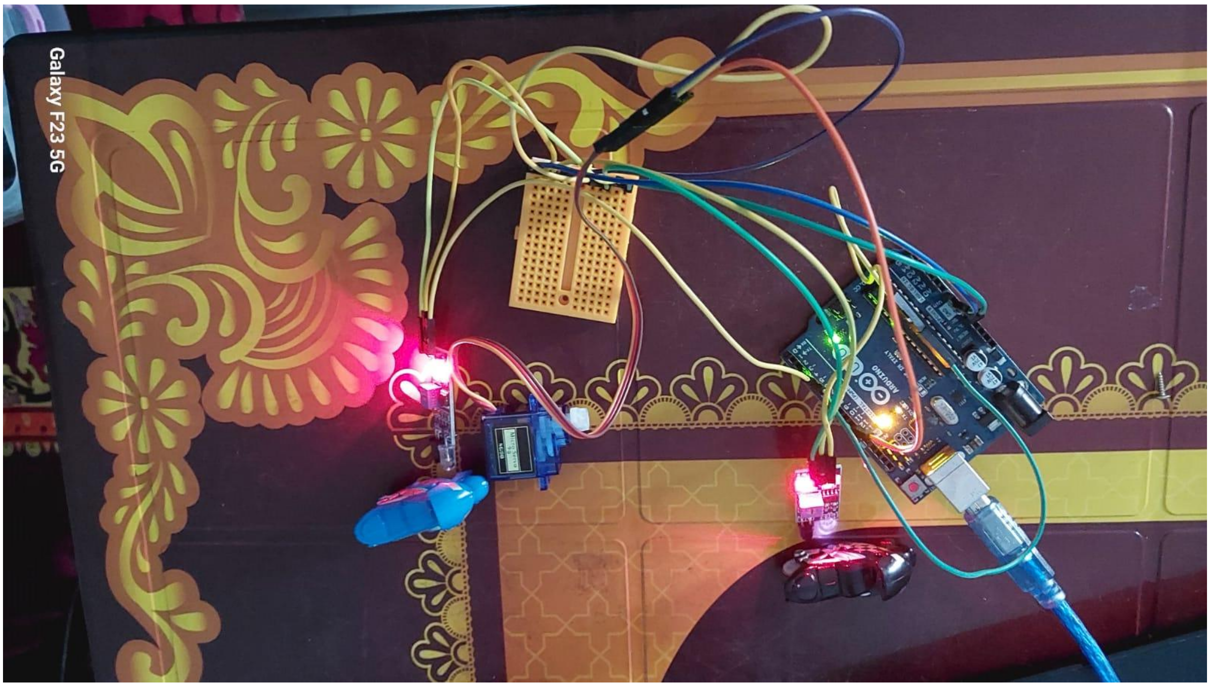
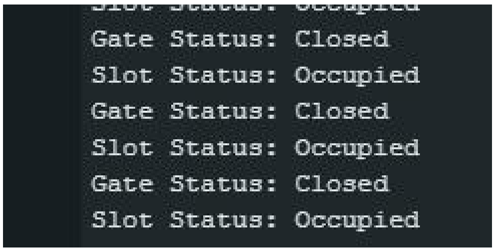


Fig 04: **Slot Occupied**

Serial Monitor:



```
Slot Status: Occupied
Gate Status: Closed
Slot Status: Occupied
Gate Status: Closed
Slot Status: Occupied
Gate Status: Closed
Slot Status: Occupied
```

Fig 05: Serial monitor for Slot Occupied

Additional Notes:

1. **Limitations**:

- The system has one parking slot due to limited IR sensors.
- It lacks an exit car detector, so cars must exit manually, and the slot status will not update automatically after a car leaves.

2. **Future Improvements**:

- Adding an exit sensor to detect cars leaving the slot.
- Expanding the system for multiple slots and implementing dynamic parking allocation.

Conclusion:

This experiment successfully demonstrates the implementation of a basic smart parking monitoring system using Arduino. The system automates gate control based on parking slot availability, ensuring efficient space utilization. Despite having only one slot and lacking an exit sensor due to hardware limitations, the setup effectively highlights the core principles of smart parking technology. The experiment provides a foundation for scaling up and incorporating additional features like multiple slots and exit detection in future iterations.