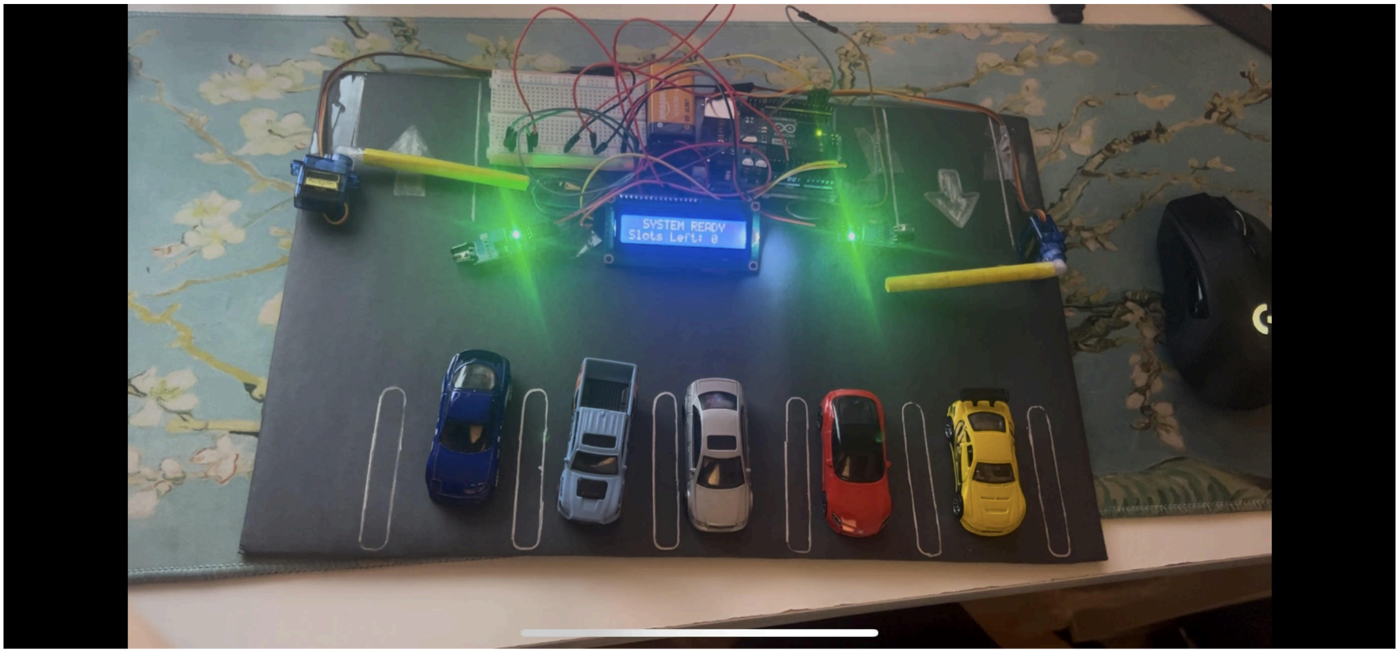
CS 578 Final Project

Name: Autonomous Parking Lot System

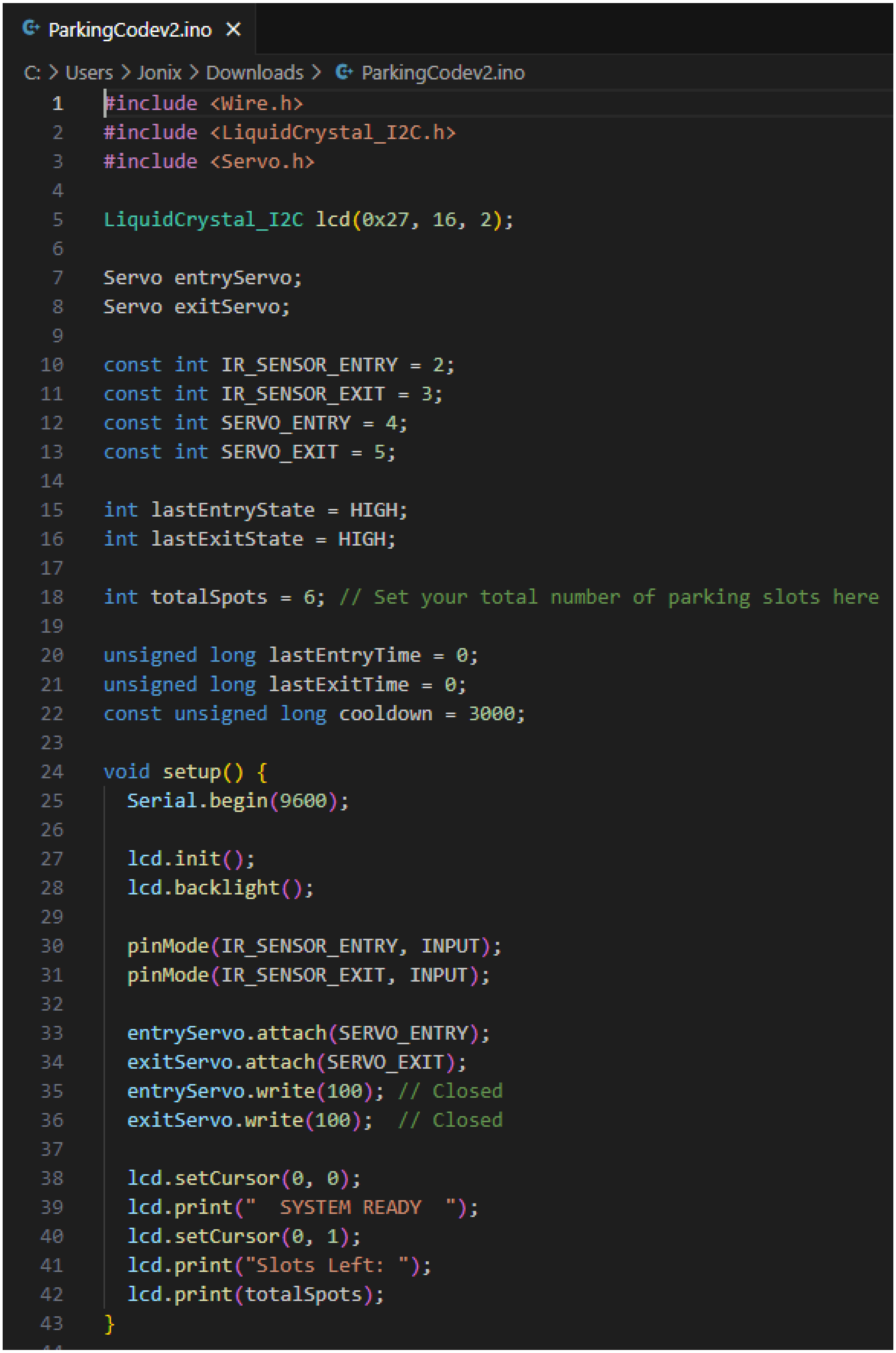
Description: Our Autonomous Parking Lot System uses an Arduino Uno as the main controller that detects vehicle presence via two IR sensors (one for entry and another for exit) and controls two micro servo motors to raise/lift barrier gates accordingly. A small LCD screen is installed as well to provide feedback such as confirmation that a car has entered the parking lot, displaying a counter of how many spots are left, and vice versa for when a car leaves the parking lot. Once the parking lot is full, it will refuse entry to another car until a car has left, freeing up a spot on the parking lot. A D battery connected to the Arduino's AC adapter ensures portable power, maintaining system mobility. With this system, we hope to increase the speed and efficiency of finding a parking spot by only allowing cars to enter the parking lot, only if there are available spots, as opposed to allowing traffic to flow in and circle the lot without a chance of finding one as a result of it being completely full.

Picture of our Autonomous Parking Lot System:

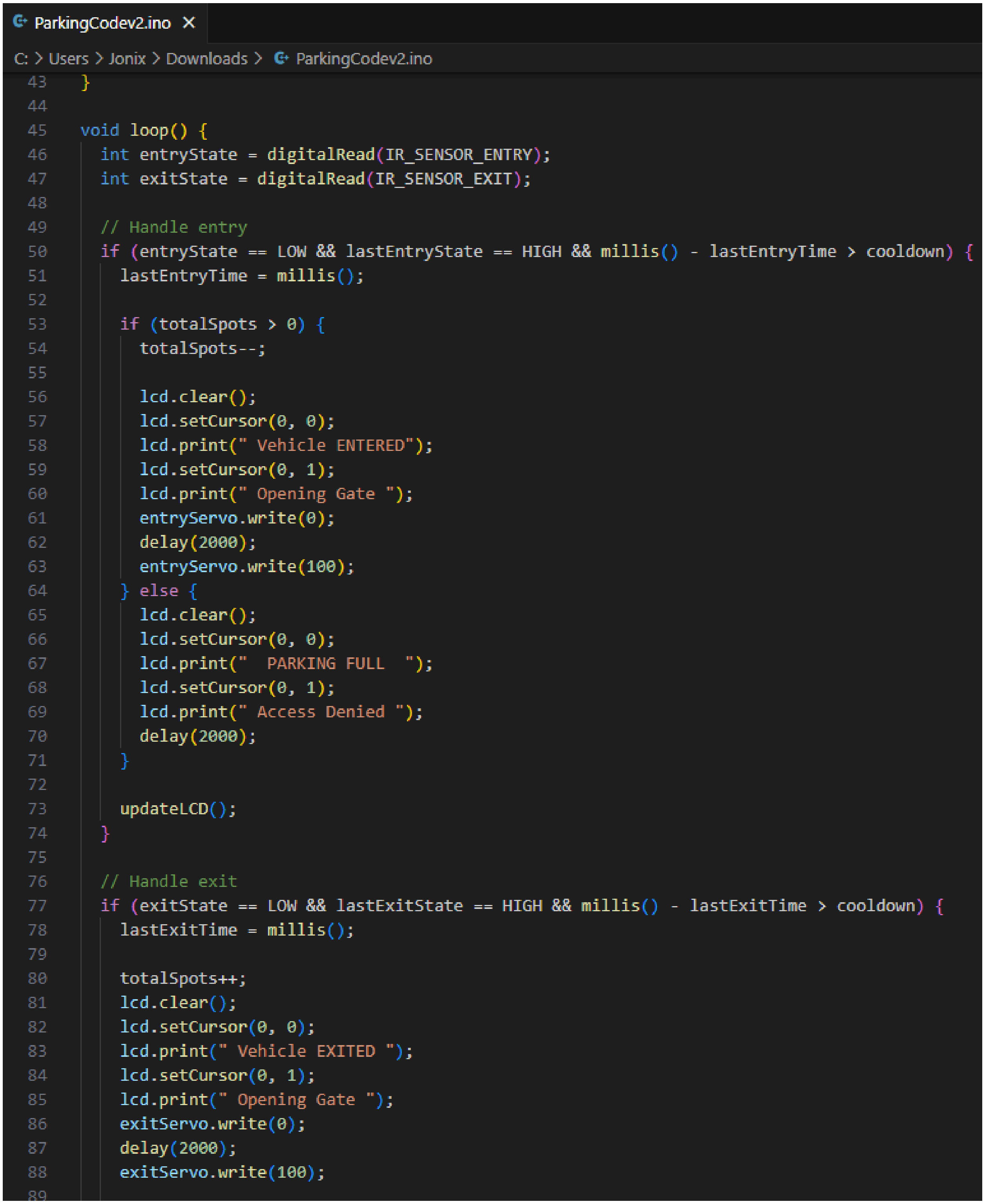


All parking spots are full, therefore no further cars will be allowed inside.

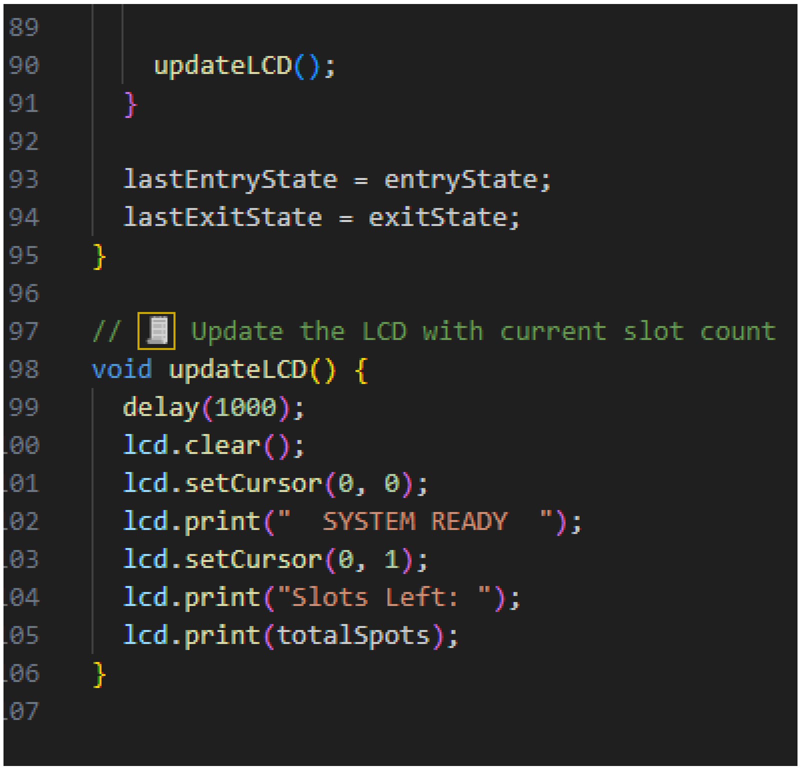
Source Code:



```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

Servo entryServo;
Servo exitServo;

const int IR_SENSOR_ENTRY = 2;
const int IR_SENSOR_EXIT = 3;
const int SERVO_ENTRY = 4;
const int SERVO_EXIT = 5;

int lastEntryState = HIGH;
int lastExitState = HIGH;

int totalSpots = 6; // Set your total number of parking slots here

unsigned long lastEntryTime = 0;
unsigned long lastExitTime = 0;
const unsigned long cooldown = 3000;

void setup() {
  Serial.begin(9600);

  lcd.init();
  lcd.backlight();

  pinMode(IR_SENSOR_ENTRY, INPUT);
  pinMode(IR_SENSOR_EXIT, INPUT);

  entryServo.attach(SERVO_ENTRY);
  exitServo.attach(SERVO_EXIT);
  entryServo.write(100); // Closed
  exitServo.write(100);  // Closed

  lcd.setCursor(0, 0);
  lcd.print("  SYSTEM READY  ");
  lcd.setCursor(0, 1);
  lcd.print("Slots Left: ");
  lcd.print(totalSpots);
}
```



```
}

void loop() {
  int entryState = digitalRead(IR_SENSOR_ENTRY);
  int exitState = digitalRead(IR_SENSOR_EXIT);

  // Handle entry
  if (entryState == LOW && lastEntryState == HIGH && millis() - lastEntryTime > cooldown) {
    lastEntryTime = millis();

    if (totalSpots > 0) {
      totalSpots--;

      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print(" Vehicle ENTERED");
      lcd.setCursor(0, 1);
      lcd.print(" Opening Gate ");
      entryServo.write(0);
      delay(2000);
      entryServo.write(100);
    } else {
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("  PARKING FULL  ");
      lcd.setCursor(0, 1);
      lcd.print(" Access Denied ");
      delay(2000);
    }

    updateLCD();
  }

  // Handle exit
  if (exitState == LOW && lastExitState == HIGH && millis() - lastExitTime > cooldown) {
    lastExitTime = millis();

    totalSpots++;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" Vehicle EXITED ");
    lcd.setCursor(0, 1);
    lcd.print(" Opening Gate ");
    exitServo.write(0);
    delay(2000);
    exitServo.write(100);
```



```
    updateLCD();
  }

  lastEntryState = entryState;
  lastExitState = exitState;
}

// 📟 Update the LCD with current slot count
void updateLCD() {
  delay(1000);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("  SYSTEM READY  ");
  lcd.setCursor(0, 1);
  lcd.print("Slots Left: ");
  lcd.print(totalSpots);
}
```