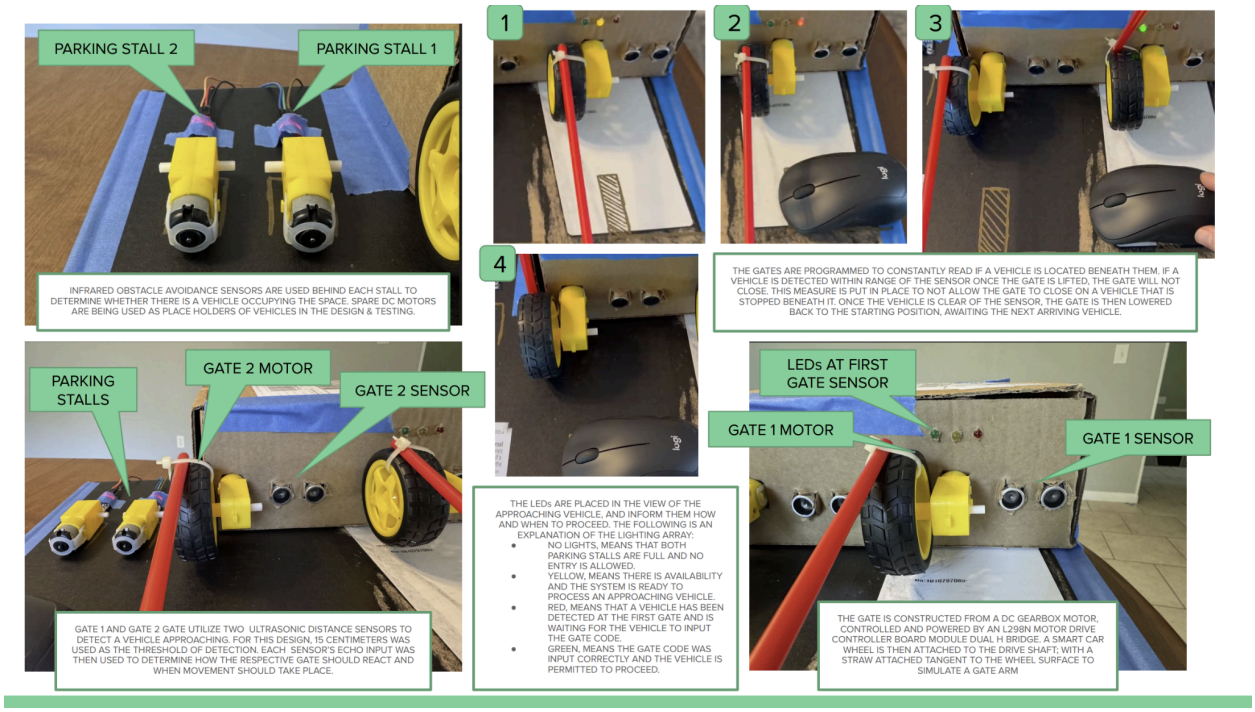
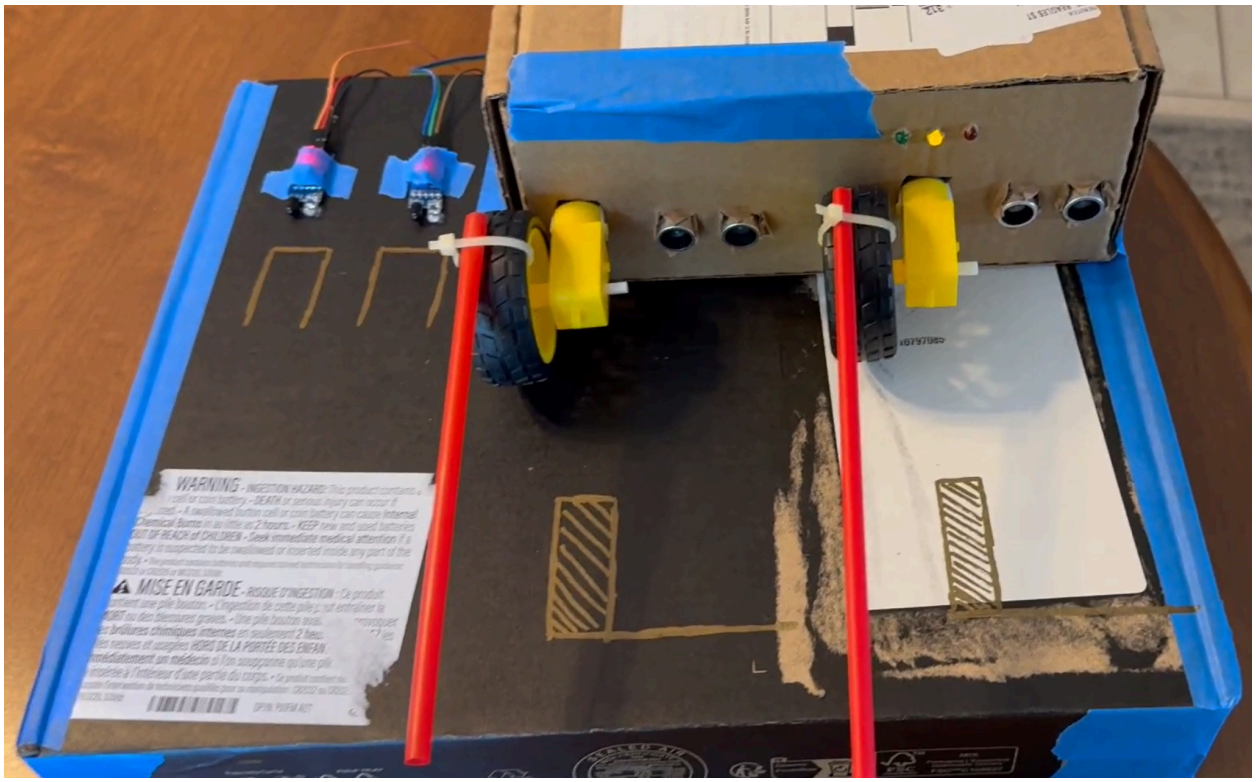
ZACHERY TAYLOR
EML4804L
06/14/2025

Project 1 Report





ZACHERY TAYLOR
EML4804L
06/14/2025

Project Summary:

Automated Parking Gate System: Design and Implementation

Building a simulated parking access gate and garage with an Arduino Mega 2560 led to practical exploration of sensor integration and mechanical design. The project used two ultrasonic sensors to detect vehicles at the entry gates, two infrared sensors to monitor parking stall occupancy, and DC motors controlled by an L298N motor drive controller to operate the gates. A set of six input pins allow the cars looking to enter input a binary code, enabling access if correct. The design also includes three LEDs to visually signal to the user about the system's status and availability.

One of the main challenges was mounting the DC motors securely so that they could act as gate arms. Without access to a 3D printer, I improvised by cutting X-shaped slots into layered cardboard from shipping boxes. This allowed the motor shafts to be held parallel to the ground, with a wheel and straw attached as the gate arm. The approach worked well enough for demonstration, though it required careful adjustment to keep the gates from touching the "ground" surface.

The design operates smoothly: Infrared sensors at each parking stall check for occupancy, and if both stalls are full, the system prevents any further entry. When a vehicle approaches the first gate, the ultrasonic sensor detects its presence and a red LED lights up. The user then enters a six-bit binary code (from zero to sixty-three using toggle wires. If the code matches the preset correct value when converted from binary to decimal, a green LED lights up and the first gate opens for ten seconds. A while loop checks the sensor continuously during this time, preventing the gate from closing if a vehicle is still present. After the first gate closes, the second gate opens following the same logic. The L298N motor controller manages gate movement, with analogWrite() to PWMs used to control speed and direction.

All project requirements were met, and there were no unresolved technical issues. The cardboard motor mounts could be more durable, but they held up well enough for the demonstration. In future versions, it might be interesting to explore wireless access codes or use servo motors for more precise gate control.

Overall, this project reinforced the value of modular code and creative problem-solving with available materials. The system demonstrates that Arduino-based automation is a practical approach for access control and parking management applications.

See code below.

ZACHERY TAYLOR
EML4804L
06/14/2025

Arduino Code:

```
int gate1trig = 28; // RED WIRE
int gate1echo = 29; // BLUE WIRE
int gate2trig = 24; // PURPLE WIRE
int gate2echo = 25; // YELLOW WIRE

int distance1 = 0;
long duration1 = 0;
int distance2 = 0;
long duration2 = 0;

int gate1yellow = 35;
int gate1red = 34;
int gate1green = 33;

int gatecodedecimal = 63; // CORRECT CODE AT GATE IN DECIMAL

int n0 = 0; // USER INPUT AT GATE CONVERTED TO DECIMAL
int a0 = 0;  //HELPER INPUT VARIABLE PINS
int a1 = 0;
int a2 = 0;
int a3 = 0;
int a4 = 0;
int a5 = 0;

int stall1sensor = 39;
int stall2sensor = 38;

int in1 = 42;
int in2 = 43;
int in3 = 44;
int in4 = 45;
int enable1 = 3;
int enable2 = 2;

void setup() {
  pinMode(gate1trig, OUTPUT);
  pinMode(gate1echo, INPUT);
  pinMode(gate2trig, OUTPUT);
  pinMode(gate2echo, INPUT);

  pinMode(gate1yellow, OUTPUT);
  pinMode(gate1red, OUTPUT);
```

```
  pinMode(gate1green, OUTPUT);

  pinMode(stall1sensor, INPUT);
  pinMode(stall2sensor, INPUT);

  pinMode(49, INPUT);
  pinMode(51, INPUT);
  pinMode(53, INPUT);
  pinMode(48, INPUT);
  pinMode(50, INPUT);
  pinMode(52, INPUT);

  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  Serial.begin(9600);

}

void loop() {
  int n0 = 0; // GATE CODE VARIABLE
  int stall1fill = 0; // INITIALIZE FULLNESS AS ZERO BEFORE EVERY LOOP
  int stall2fill = 0;
  int stallsfill = 0;

  // DETECT EACH OF THE 2 STALLS INDIVIDUALLY WITH IR AND ASSIGN TO INTS

  stall1fill = digitalRead(stall1sensor);
  stall2fill = digitalRead(stall2sensor);

  stallsfill = stall1fill + stall2fill;

  if (stallsfill == 0){
    digitalWrite(gate1yellow, 0);
    digitalWrite(gate1red, 0);
    digitalWrite(gate1green, 0);
    Serial.println("PARKING SPACES ARE FULL!");
  }

else if (stallsfill > 0) {
  // GATE 1 CHECK FOR EMPTY
  digitalWrite(gate1trig, 0);
```

```
delayMicroseconds(2); // TRIG OFF FOR 2 MICROSECONDS
digitalWrite(gate1trig, 1);
delayMicroseconds(10); // TRIG ON FOR 10 MICROSECONDS
digitalWrite(gate1trig, 0);
duration1 = pulseIn(gate1echo, 1);
distance1 = (duration1 * 0.034 / 2);

// GATE 2 CHECK FOR EMPTY
digitalWrite(gate2trig, 0);
delayMicroseconds(2); // TRIG OFF FOR 2 MICROSECONDS
digitalWrite(gate2trig, 1);
delayMicroseconds(10); // TRIG ON FOR 10 MICROSECONDS
digitalWrite(gate2trig, 0);
duration2 = pulseIn(gate2echo, 1);
distance2 = (duration2 * 0.034 / 2);

digitalWrite(gate1yellow, 1);
digitalWrite(gate1red, 0);
digitalWrite(gate1green, 0);


if (distance1 < 15){
  digitalWrite(gate1yellow, 0);
  digitalWrite(gate1red, 1);
  delay(3000);

  n0 = 0;
  a0 = digitalRead(49);
  a1 = digitalRead(51);
  a2 = digitalRead(53);
  a3 = digitalRead(48);
  a4 = digitalRead(50);
  a5 = digitalRead(52);

  n0 = (a0 * 1) + (a1 * 2) + (a2 * 4) + (a3 * 8) + (a4 * 16) + (a5 * 32);
  Serial.print("Your pins in deciaml is: ");
  Serial.println(n0);

  if (n0 == gatecodedecimal){
    digitalWrite(gate1yellow, 0);
    digitalWrite(gate1red, 0);
    digitalWrite(gate1green, 1);
    Serial.println("gate and input are correct");
    delay(5000);
```

```
// COMMAND GATE1MOTOR TO LIFT
analogWrite(enable2, 150);
digitalWrite(in4, 1);
digitalWrite(in3, 0);
delay(150);
digitalWrite(in4, 0);
digitalWrite(in3, 0);
delay(10000);
digitalWrite(gate1trig, 0);

// CHECKING FOR CAR AT GATE 1
digitalWrite(gate1trig, 0);
delayMicroseconds(2); // TRIG OFF FOR 2 MICROSECONDS
digitalWrite(gate1trig, 1);
delayMicroseconds(10); // TRIG ON FOR 10 MICROSECONDS
digitalWrite(gate1trig, 0);
duration1 = pulseIn(gate1echo, 1);
distance1 = (duration1 * 0.034 / 2);

while (distance1 < 15){
  delay(5000);
  digitalWrite(gate1trig, 0);
  delayMicroseconds(2); // TRIG OFF FOR 2 MICROSECONDS
  digitalWrite(gate1trig, 1);
  delayMicroseconds(10); // TRIG ON FOR 10 MICROSECONDS
  digitalWrite(gate1trig, 0);
  duration1 = pulseIn(gate1echo, 1);
  distance1 = (duration1 * 0.034 / 2);

}

if (distance1 > 15){
  // COMMAND GATE1MOTOR TO RETURN BACK.. REVERSE COMMAND ABOVE
  analogWrite(enable2, 150);
  digitalWrite(in4, 0);
  digitalWrite(in3, 1);
  delay(150);
  digitalWrite(in4, 0);
  digitalWrite(in3, 0);
  // COMMAND GATE2MOTOR TO LIFT
  analogWrite(enable1, 150);
  digitalWrite(in1, 1);
  digitalWrite(in2, 0);
```

```
      delay(150);
      digitalWrite(in1, 0);
      digitalWrite(in2, 0);
      delay(10000);

      // CHECK AT GATE 2 FOR CAR
      digitalWrite(gate2trig, 0);
      delayMicroseconds(2); // TRIG OFF FOR 2 MICROSECONDS
      digitalWrite(gate2trig, 1);
      delayMicroseconds(10); // TRIG ON FOR 10 MICROSECONDS
      digitalWrite(gate2trig, 0);
      duration2 = pulseIn(gate2echo, 1);
      distance2 = (duration2 * 0.034 / 2);

      while (distance2 < 15){
        delay(5000);
        digitalWrite(gate2trig, 0);
        delayMicroseconds(2); // TRIG OFF FOR 2 MICROSECONDS
        digitalWrite(gate2trig, 1);
        delayMicroseconds(10); // TRIG ON FOR 10 MICROSECONDS
        digitalWrite(gate2trig, 0);
        duration2 = pulseIn(gate2echo, 1);
        distance2 = (duration2 * 0.034 / 2);

      }
     }
     if (distance2 > 15){
       // COMMAND GATE2MOTOR TO RETURN BACK .. REVERSE GATE 2 COMMAND
ABOVE
      analogWrite(enable1, 150);
      digitalWrite(in1, 0);
      digitalWrite(in2, 1);
      delay(150);
      digitalWrite(in1, 0);
      digitalWrite(in2, 0);
     }

   }

   else {
     Serial.println("GATE CODE ENTERED IS INCORRECT.");
   }

 }
```

```
}

  // PRINTING DISTANCES
  Serial.print("Distance at Gate 1: ");
  Serial.println(distance1);
  Serial.print("Distance at Gate 2: ");
  Serial.println(distance2);


  delay(2000);


}
```

ZACHERY TAYLOR
EML4804L
06/14/2025

Circuit Diagram: