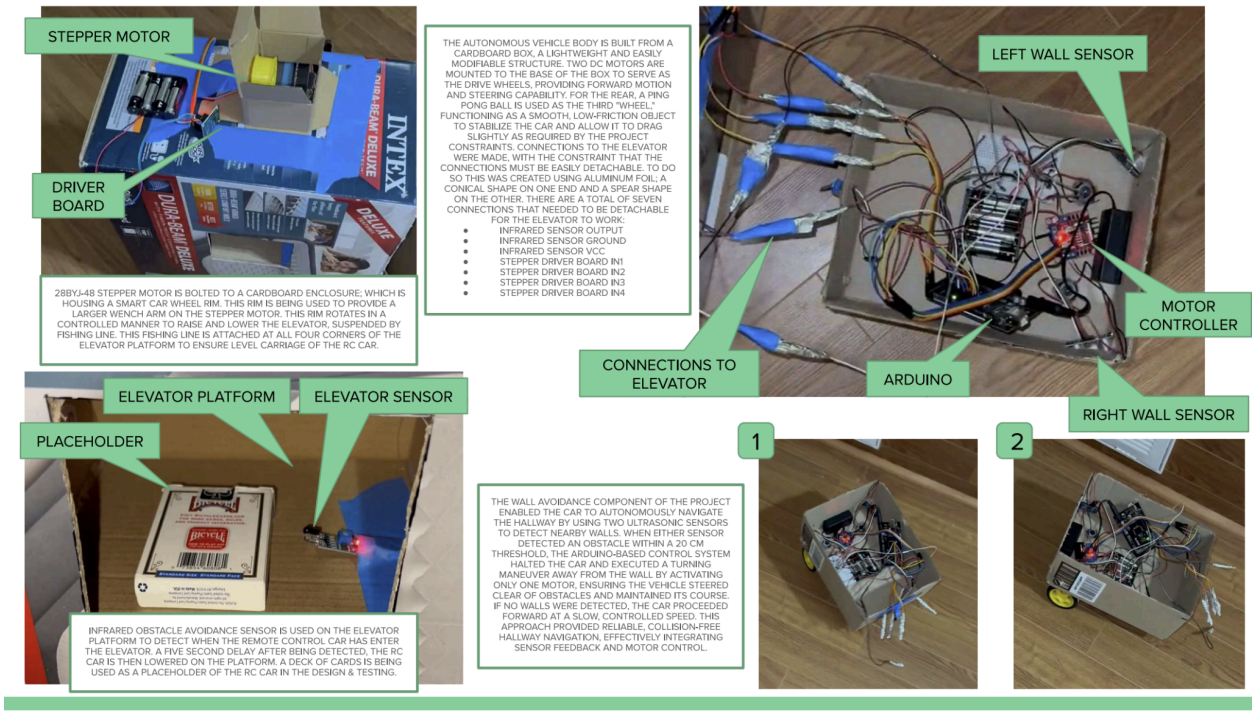
ZACHERY TAYLOR
EML4804L
07/17/2025

Project 2 Report



Project Summary:

Automated Elevator and Wall-Avoiding Car System

This project focused on designing and constructing an automated vehicle and elevator system using only the components provided in the kit and original materials for the vehicle and elevator. The objective was to create an elevator that could automatically lower a remote control car (not created in project) from a height of 10 inches to 1 inch, then have an autonomously navigable vehicle waiting at the bottom of the elevator that can then drive down a 2-meter-wide hallway, avoiding all walls.

The car body was built from a cardboard box, a lightweight and easily modifiable structure. The two DC motors from the kit were mounted at the front to serve as the drive wheels, providing forward motion and steering capability. For the rear, a ping pong ball was used as the third "wheel," functioning as a smooth, low-friction object to stabilize the car and allow it to drag slightly as required by the project constraints. This was necessary for two reasons; to tilt the weight in the vehicle more towards the front, and to reduce the surface area of the friction from

dragging. All components, including the Arduino board, L298N motor controller, battery holders, and sensors, were securely attached to the cardboard body using tape and zip ties.

The elevator was constructed from original materials and powered by a stepper motor from the kit. An IR sensor detected when the car arrived on the elevator, triggering a programmed sequence that lowered the platform after a 5-second delay. The elevator was designed to accommodate the car's dimensions (6.5" x 2.5") and weight (under 200g), ensuring a controlled descent from the starting height to the desired stop height.

Upon reaching the lower position, the car is able to extend out of the elevator and trigger a limit switch on the back of the autonomous vehicle. This initiated the autonomous driving mode, detached from the elevator, where the car used two ultrasonic sensors to detect and avoid hallway walls. The code enabled the car to move forward slowly and make corrective turns when obstacles were detected.

The main technical challenge encountered involved the detachable connections between the car and the elevator as the car began driving mode. These connections sometimes required adjustment to ensure the car could drive away smoothly without snagging or losing contact with the elevator platform. The result was a fully functional automated car and elevator system that met the project’s objectives and constraints.

See code below.

ZACHERY TAYLOR
EML4804L
07/17/2025

Arduino Code:

```
// Sequence:
   // RC car enters ->
   // elevator lowers ->
   // RC car exits ->
   // limit switch triggers car - >
   // car runs wil wall avoidance.


// ===== INPUT DEVICES =====
int IR_SENSOR = 2;       // IR sensor for RC car detection
int LIMIT_SWITCH = 3;    // Micro limit switch

// ===== LEFT MOTOR CONTROL =====
int PWM_L = 5;           // PWM pin for left motor speed
int IN_L_F = 7;          // Left motor forward
int IN_L_R = 6;          // Left motor reverse

// ===== RIGHT MOTOR CONTROL =====
int PWM_R = 8;           // PWM pin for right motor speed
int IN_R_F = 10;         // Right motor forward
int IN_R_R = 9;          // Right motor reverse

// ===== LEFT ULTRASONIC SENSOR =====
int LEFT_TRIG = 11;      // Left sensor trigger
int LEFT_ECHO = 12;      // Left sensor echo

// ===== RIGHT ULTRASONIC SENSOR =====
int RIGHT_TRIG = 14;     // Right sensor trigger
int RIGHT_ECHO = 15;     // Right sensor echo

// ===== GLOBAL VARIABLES =====
int d = 8;                   // delay between stepper steps
int u = 750;                 // stepper run duration
int motor_speed = 180;           // Base motor speed
int wall_threshold = 20;        // Distance to trigger avoidance (cm)
int turn_away_duration = 1000;   // Time to turn away from wall
int turn_center_duration = 500; // Time to center after turn

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

  // SETUP STEPPER MOTOR (PORTA = A0, A1, A2, A3)
```

```
  DDRA = B00001111;

  // SETUP INPUT DEVICES
  pinMode(IR_SENSOR, INPUT);
  pinMode(LIMIT_SWITCH, INPUT);

  // SETUP ULTRASONIC SENSORS
  pinMode(LEFT_TRIG, OUTPUT);
  pinMode(LEFT_ECHO, INPUT);
  pinMode(RIGHT_TRIG, OUTPUT);
  pinMode(RIGHT_ECHO, INPUT);

  // SETUP MOTOR CONTROLS
  pinMode(PWM_L, OUTPUT);
  pinMode(IN_L_F, OUTPUT);
  pinMode(IN_L_R, OUTPUT);
  pinMode(PWM_R, OUTPUT);
  pinMode(IN_R_F, OUTPUT);
  pinMode(IN_R_R, OUTPUT);

  // Initialize system
  stop_motors();
  PORTA = B00000000;  // STEPPER NEEDS TO START ALL TO ZERO

  delay(2000);
  Serial.println("System Initialized - Waiting for RC car");
}

void loop() {
  // Step 1: Wait for RC car to enter elevator
  //raise_elevator();  //RESET TO LIFT HEIGHT WHEN NEEDED
  wait_for_rc_car_entry();


  // Step 2: Lower elevator
  //raise_elevator();  //RESET TO LIFT HEIGHT WHEN NEEDED
  lower_elevator();


  // DON'T NEED
  // Step 3: Wait for RC car to exit elevator
  wait_for_rc_car_exit();
```

```
  // Step 4: Wait for limit switch trigger, then start wall avoidance
  wait_for_limit_switch_and_start_driving();

}

// ===== MAIN FUNCTIONS =====
void wait_for_rc_car_entry() {
  Serial.println("Waiting for RC car to enter elevator");

  while (digitalRead(IR_SENSOR) == 1) {
    delay(1000);
  }

  Serial.println("RC car detected in elevator");
}

void lower_elevator() {
  Serial.println("Lowering elevator");
  delay(5000);
  for (int step = 0; step < u; step++) {
    COUNTERCLOCKWISE(d);
  }

  Serial.println("Elevator 1 inch above ground level");
  delay(1000);
}

void wait_for_rc_car_exit() {  //CREATED BEFORE LEARNING WE DON'T NEED THIS
  Serial.println("Waiting for RC car to exit elevator...");

  while (digitalRead(IR_SENSOR) == 0) {
    delay(100);
  }

  Serial.println("RC car has exited elevator");
  delay(5000);
}

void raise_elevator() {  //WILL JUST BE USED TO RESET

  for (int step = 0; step < u; step++) {
    CLOCKWISE(d);
  }
```

```
  delay(1000);
}

void wait_for_limit_switch_and_start_driving() {
  Serial.println("Waiting for RC car to bump limit switch...");

  while (digitalRead(LIMIT_SWITCH) == 0) {
    delay(10);
  }

  wall_avoidance_mode();
}

void wall_avoidance_mode() {
  while (1) {
    long left_dist = read_ultrasonic(LEFT_TRIG, LEFT_ECHO);
    long right_dist = read_ultrasonic(RIGHT_TRIG, RIGHT_ECHO);

    if (left_dist < wall_threshold || right_dist < wall_threshold) {
      stop_motors();

      if (left_dist < wall_threshold) {
        left_only_forward();
        delay(turn_away_duration);
        stop_motors();
        right_only_forward();
        delay(turn_center_duration);
      }
      else if (right_dist < wall_threshold) {
        right_only_forward();
        delay(turn_away_duration);
        stop_motors();
        left_only_forward();
        delay(turn_center_duration);
      }
      stop_motors();
    }
    else {
      move_forward_slow();
    }
    delay(100);
  }
}
```

ZACHERY TAYLOR
EML4804L
07/17/2025

```
// ===== ELEVATOR STEPPER FUNCTIONS =====
void CLOCKWISE(int w) {
  PORTA = B00000001;    // STEP 1
  delay(w);
  PORTA = B00000010;    // STEP 2
  delay(w);
  PORTA = B00000100;    // STEP 3
  delay(w);
  PORTA = B00001000;    // STEP 4
  delay(w);
}

void COUNTERCLOCKWISE(int w) {
  PORTA = B00000001;    // STEP 1
  delay(w);
  PORTA = B00001000;    // STEP 2
  delay(w);
  PORTA = B00000100;    // STEP 3
  delay(w);
  PORTA = B00000010;    // STEP 4
  delay(w);
}

// ===== SENSOR FUNCTIONS =====
long read_ultrasonic(int trig_pin, int echo_pin) {
  digitalWrite(trig_pin, 0);
  delayMicroseconds(2);
  digitalWrite(trig_pin, 1);
  delayMicroseconds(10);
  digitalWrite(trig_pin, 0);
  long duration = pulseIn(echo_pin, 1);
  return (duration * 0.034 / 2);
}

// ===== MOTOR CONTROL FUNCTIONS =====
void move_forward_slow() {
  digitalWrite(IN_L_F, 1);
  digitalWrite(IN_L_R, 0);
  digitalWrite(IN_R_F, 1);
  digitalWrite(IN_R_R, 0);
  analogWrite(PWM_L, motor_speed);
  analogWrite(PWM_R, motor_speed);
}
```

```
void left_only_forward() {
  digitalWrite(IN_L_F, 1);
  digitalWrite(IN_L_R, 0);
  digitalWrite(IN_R_F, 0);
  digitalWrite(IN_R_R, 0);
  analogWrite(PWM_L, motor_speed);
  analogWrite(PWM_R, 0);
}

void right_only_forward() {
  digitalWrite(IN_L_F, 0);
  digitalWrite(IN_L_R, 0);
  digitalWrite(IN_R_F, 1);
  digitalWrite(IN_R_R, 0);
  analogWrite(PWM_L, 0);
  analogWrite(PWM_R, motor_speed);
}

void stop_motors() {
  digitalWrite(IN_L_F, 0);
  digitalWrite(IN_L_R, 0);
  digitalWrite(IN_R_F, 0);
  digitalWrite(IN_R_R, 0);
  analogWrite(PWM_L, 0);
  analogWrite(PWM_R, 0);
}
```

ZACHERY TAYLOR
EML4804L
07/17/2025

Circuit Diagram:

