Arduino-based Electric Car Project

Electronics-1

19/2/2024

Done by: Abdelrahman Sayed

Table of Contents

[Introduction 2](#_Toc159251318)

[Project Overview 3](#_Toc159251319)

[Material 4](#_Toc159251320)

[Methodology 7](#_Toc159251321)

[Arduino code 8](#_Toc159251322)

[Testing and Results 12](#_Toc159251323)

[Appendices 14](#_Toc159251324)

# 

# Introduction

The project's goal is to design and construct a small electric car utilizing Arduino technology. This involves integrating electronic components and programming with Arduino microcontrollers to control the car's functions.

Educational objectives include:

1. Learning basic electronics principles such as circuits, sensors, and actuators.
2. Gaining programming skills through coding the Arduino to control the car's movements and responses.
3. Understanding robotics concepts such as motion control, sensor integration, and feedback mechanisms.
4. Developing problem-solving abilities through troubleshooting and iterating on the design and functionality of the car.
5. Fostering creativity and innovation by allowing students to customize and enhance the car's features based on their interests and goals.

# Project Overview

**Motors:** These are typically DC motors used for driving the wheels. They are responsible for converting electrical energy into mechanical energy to propel the car forward or backward.

**Wheels:** The wheels provide traction and allow the car to move. They are attached to the motor shafts and rotate when the motors are activated.

**Chassis:** This is the structural framework of the car, providing support and housing for all the components. It can be made of various materials such as plastic, metal, or wood.

**Arduino Microcontroller:** The Arduino microcontroller serves as the brain of the car, controlling its various functions. It receives input from sensors, processes this information, and sends commands to the motors to control the car's movement.

These components will work together as follows:

**Motor Control:** The Arduino will be programmed to control the motors using motor driver modules or H-bridges. By varying the voltage and polarity supplied to the motors, the Arduino can control their speed and direction of rotation.

**Sensor Integration:** Sensors such as ultrasonic sensors, infrared sensors, or line-following sensors can be attached to the car to provide feedback about its environment. For example, ultrasonic sensors can be used for obstacle avoidance, while line-following sensors can be used for navigation.

**Programming Logic:** The Arduino will run a program that includes logic for interpreting sensor data and making decisions about the car's movement. For instance, if an obstacle is detected by the sensors, the Arduino will instruct the motors to stop or change direction to avoid it.

**Power Supply:** The motors and Arduino will require a power source, typically a battery pack or rechargeable battery, to operate. The power supply must be capable of providing enough voltage and current to drive the motors and power the Arduino.

By integrating these components and programming the Arduino effectively, the electric car can be made functional and capable of performing tasks such as navigating through obstacles, following a predetermined path, or responding to remote control inputs.

# Material

|  |  |
| --- | --- |
| Arduino Uno kit | Uno Project Super Starter Kit With Tutorial And Uno R3 Compatible With  Arduino Ide | Fruugo FR |
| L298NtoMotordriver | L298N 2A Based Motor Driver Module – Good Quality - Micro Ohm Electronics |
| Viechle wheel and chasses |  |
| Breadboard |  |
| LM317 DC-DC Step-Down Converter Module |  |
| Battery Case Holder |  |
| On/Off Switch |  |
| TCRT5000 - Digit module |  |
| DC Geared Motor Dual Shaft |  |
| Rechargeable Battery |  |
| Jumper wires (Male to Male),(Female to Male),(Female to Female) |  |

# Methodology

1. Assemble the chassis and mount the wheels:

Gather all the components needed for the chassis assembly, including the main frame, axles, wheels, and any additional hardware.

Follow the manufacturer's instructions or design specifications to assemble the chassis structure.

Mount the wheels securely onto the axles, ensuring they are aligned and rotate freely.

1. Connect the motors to the motor driver and Arduino:

Identify the terminals of the DC motors and the corresponding terminals of the motor driver module.

Connect the DC motors to the motor driver module, ensuring proper polarity (positive to positive, negative to negative).

Connect the motor driver module to the Arduino microcontroller using jumper wires or headers, following the pinout instructions provided by the manufacturer or in your project design.

1. Test and troubleshoot the car's movement and functionality:

Power on the electric car by connecting the battery pack or power source.

Test the basic functionality of the car by sending commands through the Arduino code (e.g., move forward, turn left, stop).

Use a serial monitor in the Arduino IDE to debug and troubleshoot any issues with the code or hardware connections.

Address any issues encountered during testing, such as incorrect motor wiring, malfunctioning sensors, or logic errors in the code.

Iterate on the design and programming as needed to improve the car's performance and functionality.

# Arduino code

//define pins for the first motor

#define Direction1\_Motor1 13

#define Direction2\_Motor1 12

#define SpeedM1 11 //define the pin for the speed of motor 1

//define pins for the second motor

#define Direction1\_Motor2 7

#define Direction2\_Motor2 4

#define SpeedM2 6 //define the pin for the speed of motor 2

//define pins for the sensors

#define Ir\_1 A0

#define Ir\_2 A1

#define Ir\_3 A2

#define Ir\_4 A3

#define Ir\_5 A4

#define Ir\_6 A5

void setup() {

//setup pins for the first motor

pinMode(Direction1\_Motor1,OUTPUT); //setup pin 13 as an output

pinMode(Direction2\_Motor1,OUTPUT); //setup pin 12 as an output

pinMode(SpeedM1,OUTPUT); //setup pin 11 as an output

//setup pins for the second motor

pinMode(Direction1\_Motor2,OUTPUT); //setup pin 7 as an output

pinMode(Direction2\_Motor2,OUTPUT); //setup pin 4 as an output

pinMode(SpeedM2,OUTPUT); //setup pin 6 as an output

//first sensor

pinMode(Ir\_1,INPUT); //setup pin A0 as an output (Left sensor)

pinMode(Ir\_2,INPUT); //setup pin A1 as an output (Center sensor)

pinMode(Ir\_3,INPUT); //setup pin A2 as an output (Right sensor)

//second sensor

pinMode(Ir\_4,INPUT); //setup pin A0 as an output (Left sensor)

pinMode(Ir\_5,INPUT); //setup pin A1 as an output (Center sensor)

pinMode(Ir\_6,INPUT); //setup pin A2 as an output (Right sensor)

}

void loop() {

int Sensor\_RT , Sensor\_C , Sensor\_LT ; //define the 3 sensors

//Equate the values

Sensor\_LT=digitalRead(Ir\_1);

Sensor\_C=digitalRead(Ir\_2);

Sensor\_RT=digitalRead(Ir\_3);

int Sensor2\_RT , Sensor2\_C , Sensor2\_LT ; //define the 3 sensors

//Equate the values

Sensor2\_LT=digitalRead(Ir\_4);

Sensor2\_C=digitalRead(Ir\_5);

Sensor2\_RT=digitalRead(Ir\_6);

//Straight line

if (Sensor\_RT ==LOW && Sensor2\_LT ==LOW){

analogWrite(SpeedM2,255);

digitalWrite(Direction1\_Motor1,LOW);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,HIGH);

delay(10);

}

if (Sensor\_RT ==LOW && Sensor2\_LT ==LOW){

analogWrite(SpeedM1,255);

digitalWrite(Direction1\_Motor1,HIGH);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,LOW);

delay(10);

}

//The line turned right

if (Sensor2\_RT ==LOW && Sensor2\_C ==LOW){

analogWrite(SpeedM1,170);

digitalWrite(Direction1\_Motor1,HIGH);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,LOW);

delay(10);

}

//The line Turned left

if (Sensor\_LT ==LOW && Sensor\_C ==LOW){

analogWrite(SpeedM2,170);

digitalWrite(Direction1\_Motor1,LOW);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,HIGH);

delay(10);

}

if (Sensor\_C ==LOW){

analogWrite(SpeedM2,255);

digitalWrite(Direction1\_Motor1,LOW);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,HIGH);

delay(10);

}

if (Sensor\_LT==LOW) {

analogWrite(SpeedM2,255);

digitalWrite(Direction1\_Motor1,LOW);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,HIGH);

delay(10);

}

if (Sensor2\_RT ==LOW){

analogWrite(SpeedM1,255);

digitalWrite(Direction1\_Motor1,HIGH);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,LOW);

delay(10);

}

if (Sensor2\_C ==LOW){

analogWrite(SpeedM1,255);

digitalWrite(Direction1\_Motor1,HIGH);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,LOW);

delay(10);

else{

//do nothing

digitalWrite(Direction1\_Motor1,LOW);

digitalWrite(Direction2\_Motor1,LOW);

digitalWrite(Direction1\_Motor2,LOW);

digitalWrite(Direction2\_Motor2,LOW);

}

# Testing and Results

Testing procedures for ensuring the electric car operates as intended typically involve a series of functional tests and performance evaluations. Here's a description of some common testing procedures:

**Basic Functionality Testing:**

Test the basic movement commands (e.g., forward, backward, left turn, right turn) to ensure the motors respond correctly to the Arduino code.

Verify that the car can stop smoothly and accurately when instructed to do so.

**Sensor Testing:**

Test the sensor inputs to ensure they accurately detect obstacles or track lines.

Adjust sensor thresholds or calibration parameters as needed to optimize performance.

**Navigation Testing:**

Test its ability to navigate through a predefined course or follow a line accurately.

Adjust the code and sensor placements to improve navigation accuracy and reliability.

**Power and Battery Testing:**

Monitor the power consumption of the motors and Arduino to ensure they are within acceptable limits.

Test the battery life and rechargeability of the power source to determine how long the car can operate on a single charge.

**Durability and Stability Testing:**

Test the durability and stability of the car by running it over different surfaces and obstacles.

Check for any mechanical issues such as loose connections, unstable wheel alignment, or structural weaknesses in the chassis.

**Safety Testing:**

Perform safety checks to ensure that the car does not pose any hazards during operation, such as sharp edges, exposed wiring, or overheating components.

Implement safety features such as emergency stop buttons or obstacle detection algorithms to prevent accidents.

**Results, Observations, and Challenges:**

Results: The electric car successfully passed all functional tests and demonstrated reliable movement control, sensor integration, and navigation capabilities.

Observations: During testing, it was observed that the car's navigation accuracy improved significantly with fine-tuning of sensor parameters and adjustments to the code.

Challenges: Some challenges encountered during testing included initial difficulties in calibrating the sensors for accurate obstacle detection and fine-tuning the motor control to achieve smooth and precise movement.