# ROBOTIC CAR USING 8051 MICROCONTROLLER

MPMC PROJECT REPORT

*SUBMITTED BY*

*BATCH-D8*


Y.Nagaraju (20481A04P2)

Y.Chandra Sekhar (20481A04O4)

P.Vasavi Satya (20481A04N5)

S.Gnana Pavan (20481A04K6)


*Under the Guidance of*

*Dr .B.Rajasekhar*



Department Of Electronics and Communication Engineering

**SR GUDLAVALLERU ENGINEERING COLLEGE**

(An Autonomous Institute With Permanent Affiliation to JNTUK, Kakinada)

SESHADRI RAO KNOWLEDGE VILLAGE

GUDLAVALLERU-521356

ANDHRA PRADESH


2022

**SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE**

(An Autonomous Institute with Permanent Affiliation to JNTUK)

SESHADRI RAO KNOWLEDGE VILLAGE GUDLAVALLERU-521356

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINERING**

# CERTIFICATE

This is certify that the project report entitled " **ROBOTIC CAR** " is a bonafied record of work carried by

**Y. Nagaraju** (20481A04P2)

**P.Vasavi Satya** (20481A04N5)

**Y.Chandra Sekhar** (20481A04O4)

S.Gnana Pavan (20481A04K6)

Under my guidance and supervision in partial fulfillment of the requirements for award of the degree of Bachelor Of Technology in Electronics and Communication Engineering of Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru**.**

**(Dr . B. Rajasekhar)**
PROJECT GUIDE

**(Dr. Y. RAMAKRISHNA)**
HEAD OF THE DEPARTMENT

# ACKNOWLEDGEMENT

We are glad to express our deep sense of gratitude to **Dr. B. Rajasekhar** Professor, Electronics and Communication Engineering for his guidance and cooperation in completing this project. Through this we want to convey our sincere thanks to him for inspiring assistance during our project.

We express our heart felt gratitude and deep in debtedness to our beloved Head of the Department **Dr. Y.RAMAKRISHNA**, for his great help and encouragement in doing our project successfully.

We also express our gratitude to our principal **Dr.G.V.S.N.R.V PRASAD** for his encouragement and facilities provide during the course of the project.

We express our heart felt gratitude to our **Faculty members** and **Lab Technicians** for providing great support for us in completing our project.

We think one and all who have rendered help us to directly or indirectly, in the completion of this work.

# CONTENTS

# ABSTRACT

The objective of this project is “How to control the speed and direction of the DC motor using 8051 micro controller “Interfacing Dc motor of 8051 forms and essential part in designing embedded robotic projects. A well designed 8051 -DC motor system has essentially two parts. Firstly an 8051 with the required software to control the motor and secondly a suitable driver circuit. In this project we are using IC L293D for interfacing 8051 with DC motor.

The two input buttons can be used to control the output generated by the microcontroller. The signal is the fed to the DC motor through motor driver. Thus the speed of the DC motor can controlled and to control the direction of the DC motor we can use a circuit called an H-Bridge**.** This H-Bridge is an electronic circuit that can driver the motor in both direction

## Software Requirements

- ✓ Keil Software
- ✓ Proteus Software
- ✓ 8051 microcontroller
- ✓ L239D motor driver IC
- ✓ Push buttons
    - Move forward
    - Move backward
    - Move right
    - Move left
- ✓ DC motors
- ✓ 9v battery(to energise motor)
- ✓ 5v battery(to energise L239D IC)
- ✓ Vss-5v
- ✓ ground

# CHAPTER 1

# INTRODUCTION

Button controlled robotic cars are a type of robot that can be controlled using push buttons. These robots typically consist of a chassis, wheels, motors, a microcontroller, and push buttons. The microcontroller receives input from the push buttons and sends commands to the motors to control the movement of the car.

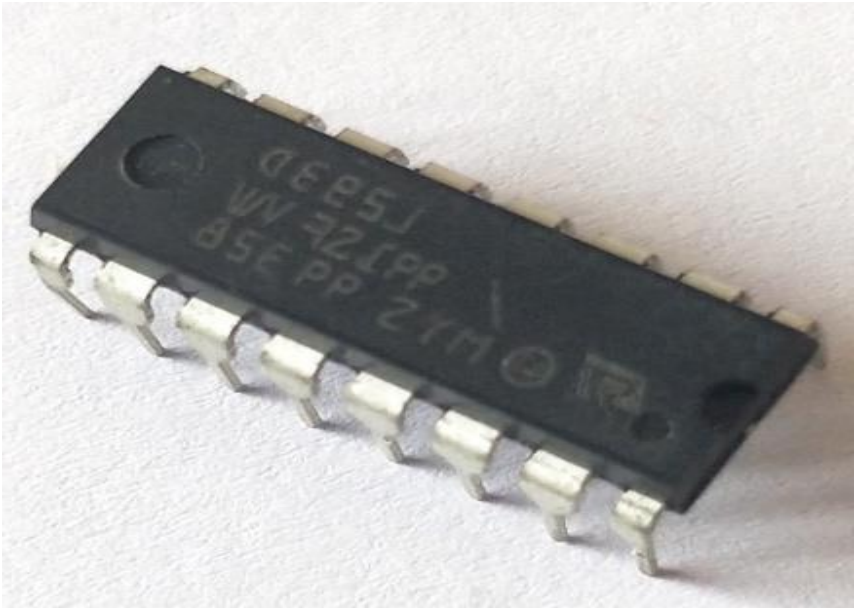
Button controlled robotic cars are popular among hobbyists and students who want to learn about robotics and microcontroller programming. They can be built using relatively simple components and programming techniques, making them accessible to beginners.

These robots can be customized and programmed to perform a variety of tasks, such as following a line, avoiding obstacles, or carrying small objects. They can also be designed to have different types of control inputs, such as remote control or voice control.

Button controlled robotic cars have a wide range of applications, including education, entertainment, and research. They can be used in classrooms to teach programming and robotics, in entertainment settings as interactive displays, or in research to explore new control systems and algorithms. Overall, button controlled robotic cars are an exciting and accessible way to learn about robotics and explore the possibilities of microcontroller programming.

## 1.1 BACKGROUND

The 8051 microcontroller is a widely used microcontroller in embedded systems and is known for its simplicity, low power consumption, and ease of use. It was developed by Intel in the 1980s and has since become a popular choice for many applications, including robotics.



Button controlled robotic cars using 8051 microcontroller are popular among hobbyists, students, and engineers who want to learn about microcontroller programming and robotics. They are relatively simple to build, requiring only a few basic components such as motors, wheels, push buttons, and a microcontroller. The microcontroller receives input from the push buttons and sends commands to the motors to control the movement of the car.

Button controlled robotic cars using 8051 microcontroller are a great way to learn about robotics and programming. They provide a hands-on way to explore the principles of robotics, including motor control, sensor input, and decision-making. They can also be customized and programmed to perform a variety of tasks, making them a useful tool for education, entertainment, and research.

Overall, button controlled robotic cars using 8051 microcontroller are a popular and accessible way to learn about microcontroller programming and robotics. They provide a platform for experimentation and creativity, and are a great way to introduce people to the exciting world of robotics.

## 1.2 Aim of the project

The aim of building button controlled robotic cars using 8051 microcontroller is to provide an accessible platform for learning about microcontroller programming and robotics. These robots can be built using relatively simple components and programming techniques, making them an excellent starting point for beginners.

## 1.3 Objective

The objective of building button controlled robotic cars using 8051 microcontroller is to develop a robot that can be controlled using push buttons and can perform a variety of tasks such as line following, obstacle avoidance, and object retrieval

## 1.4 Software requirements

### 1.4.1 Keil uVision5

The µVision IDE combines project management, runtime environment, build facilities, source code editing, and program debugging in a single powerful environment. µVision is easy-to-use and accelerates your embedded software development.

µVision supports multiple screens and allows you to create

individual window layouts anywhere on the visual

surface.


1.4.2 Proteus Software:---

The Proteus deign suite combines each of use

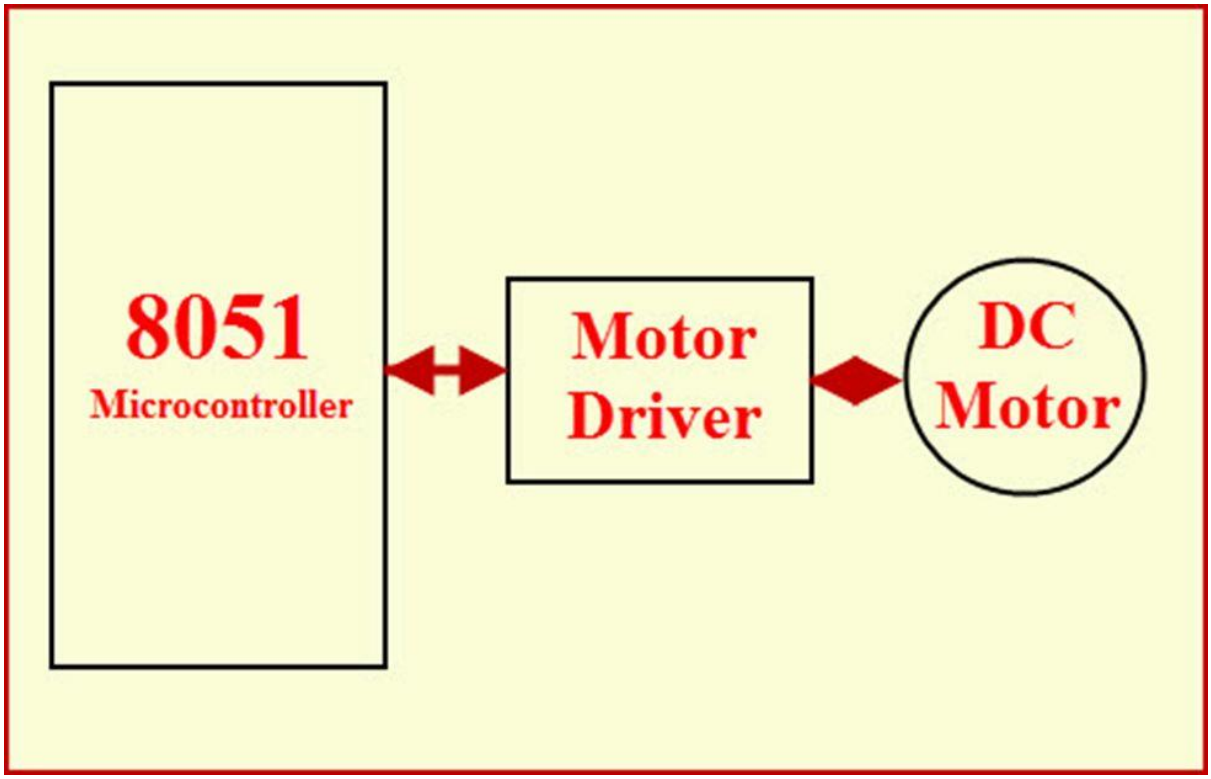
with a powerful feature set to enable the rapid

design, test and layout of professional printed circuit boards.

# CHAPTER 2

# BLOCK DIAGRAM AND EXPLANATION

## 2.1 Block diagram




## 2.2 Block diagram explanation:

The block diagram of button controlled robotic cars using 8051 microcontroller consists of several functional blocks that are connected together to form the overall system. Here is a brief explanation of each block:

Microcontroller: The microcontroller is the brain of the system. It receives input from the push buttons and sends commands to the motors to control the movement of the car. The 8051 microcontroller is a popular choice for this type of application due to its simplicity and ease of use.

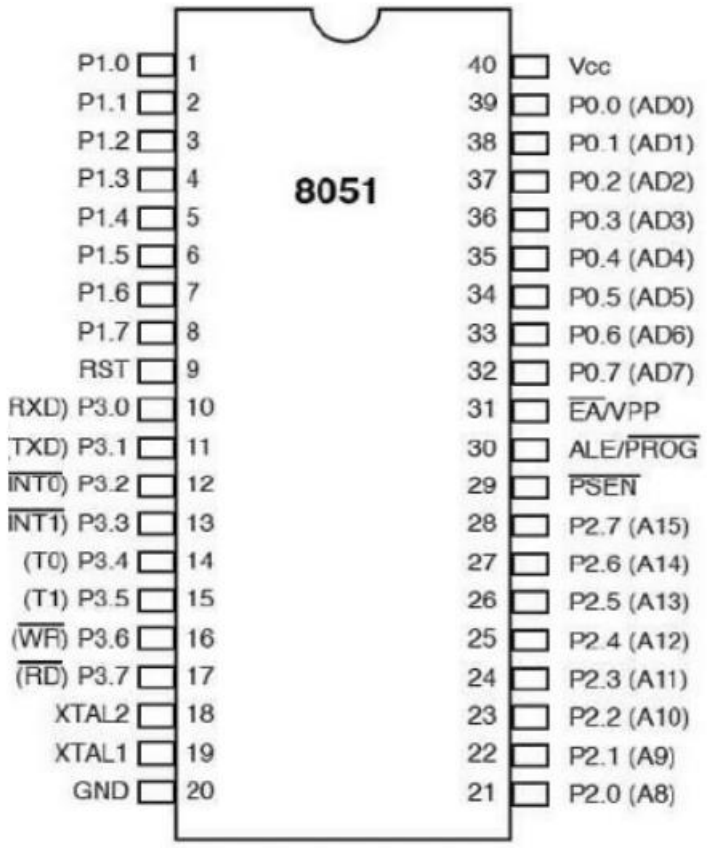
Motor driver: The motors are the mechanical components that drive the movement of the car. They are connected to the motor driver circuit and receive commands from the microcontroller to move the car forward, backward, left, or right.

# CHAPTER 3

# WORK TITLE EXPLANATION

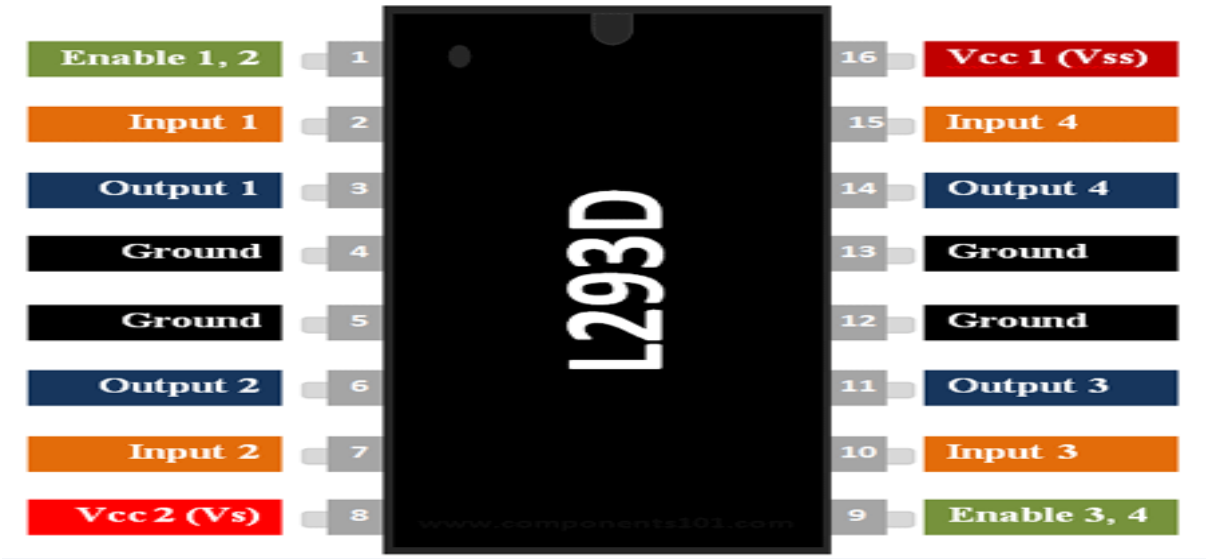
## 8051 Microcontroller

The 8051 microcontroller is a popular 8-bit microcontroller architecture that was first introduced by Intel in 1980. Since then, it has become one of the most widely used microcontrollers in the world due to its simplicity, reliability, and low cost. The 8051 microcontroller is widely used in a variety of applications such as robotics, industrial control, automotive systems, and consumer electronics.The 8051 microcontroller is programmed using low-level languages such as Assembly or high-level languages such as C. It is also supported by a variety of development tools such as integrated development environments (IDEs) and debuggers.Overall, the 8051 microcontroller is a versatile and popular microcontroller architecture that has been used in a wide range of applications due to its simplicity, low cost, and reliability.




## L293D

The L293D is a popular motor driver IC that is commonly used to drive DC motors in robotics and other applications. It is a versatile IC that can control up to two motors simultaneously and provides bidirectional control of each motor. The L293D can handle a maximum current of 600mA per channel and can operate at a voltage range of 4.5V to 36V.




## DC motor

DC motors are electric motors that convert electrical energy into mechanical energy through the interaction of magnetic fields. They are widely used in a variety of applications, including robotics, automation, automotive, and consumer electronics.The basic design of a DC motor consists of a rotor and a stator. The rotor is the moving part of the motor, while the stator is the stationary part. The rotor is typically made of a coil of wire that is wrapped around a central iron core, while the stator is made of a series of magnets that are arranged in a circular pattern around the rotor.

## 3.1 SOFTWARE CODE:

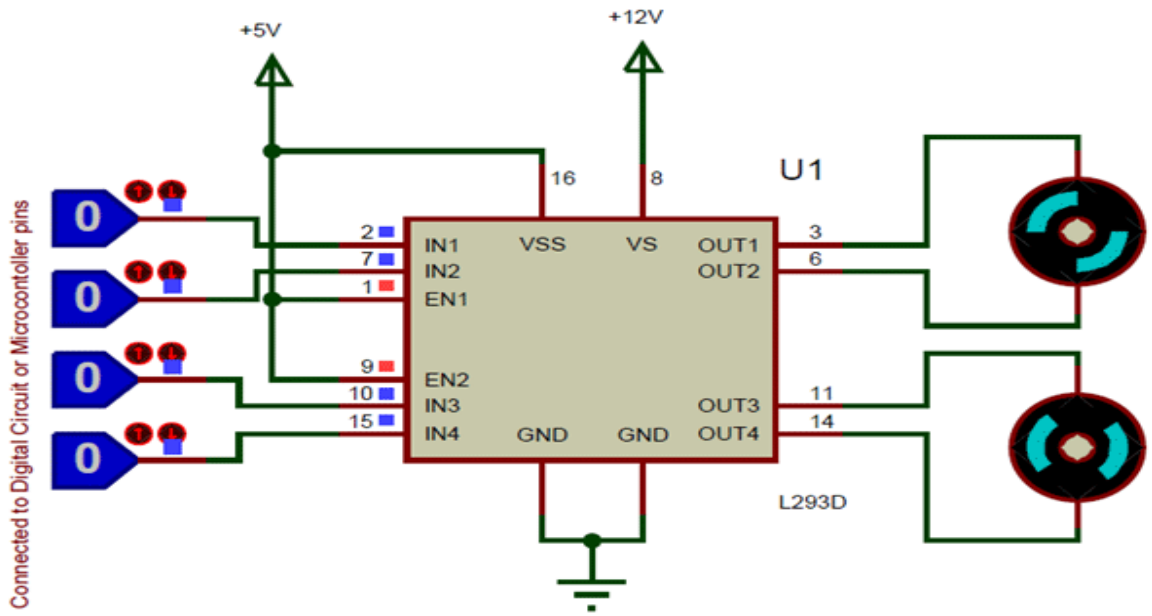
```
void main()
{
        while(1)
        {
                if(F==0)
                {
                        //ALL Motors Clock Wise Rotate
                        IN1 = 1;
                        IN2 = 0;

                        IN3 = 1;
                        IN4 = 0;
                }
                else if(REV==0)
                {
                        //ALL Motors Anti Clockwise Rotate
                        IN1 = 0;
                        IN2 = 1;

                        IN3 = 0;
                        IN4 = 1;
                }
```

```
        else if(R==0)
        {
            //Upper Two motors ClocK wise Rotate
            IN1 = 1;
            IN2 = 0;

            //Lower Two Motors STOP
            IN3 = 0;
            IN4 = 0;
        }
        else if(L==0)
        {
            //Upper Two Motors Anti Clockwise Rotate
            IN1 = 0;
           IN2 = 1;

            //Lower Two Motors Stop
            IN3 = 0;
            IN4 = 0;
        }
        else
        {
            //All Motors Stop
            IN1 = 0;
            IN2 = 0;

            IN3 = 0;
            IN4 = 0;
        }
}
```

# CHAPTER 4

# CIRCUIT DIAGRAM AND EXPLANATION

## 4.1 Circuit diagram




## 4.2 Construction:

Step 1: Connect the motor driver IC to the microcontroller using jumper wires. Use pins 2.0, 2.1, 2.2 and 2.3 of the microcontroller to control the direction of the motors through the motor driver IC. Connect the motor driver IC to the motors using jumper wires.

Step 2: Connect the wheels to the motors using screws.

Step 3: Attach the motors to the chassis using screws.

Step 4: Connect the push buttons to the microcontroller using jumper wires. Connect one push button to pin 3.3 and the other to pin 3.4 of the microcontroller.

Step 5: Connect the battery to the microcontroller using jumper wires. Use a 100uF capacitor to filter out the noise from the battery.

Step 6: Connect a 10uF capacitor between the Vcc and GND pins of the microcontroller to filter out any noise.

Step 7: Write the code to control the motors and read the push button inputs. When the first button is pressed, the car should move forward. When the second button is pressed, the car should move backward. When both buttons are pressed, the car should stop.

Step 8: Upload the code to the microcontroller and test the robotic car.

Note: This is a basic outline of the steps involved. Depending on the specific components you are using, there may be additional steps or modifications required. It is also important to consider safety precautions when building and testing the wearing protective eyewear and keeping the car away from edges or other hazards.

## 4.3 Working Principle:

The working principle of a button-controlled robotic car using 8051 microcontroller can be explained in the following steps:

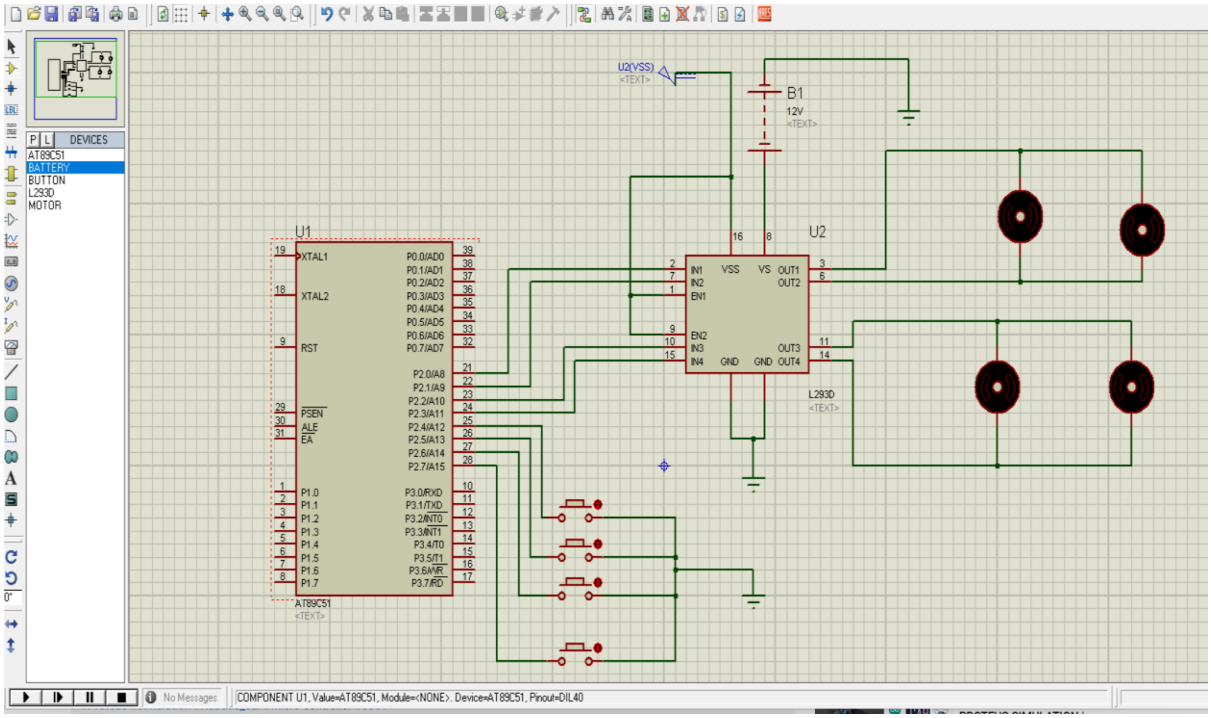
- The microcontroller is programmed to receive input signals from push buttons connected to its input pins. Each push button is connected to a different input pin of the microcontroller.
- The microcontroller reads the input signals from the push buttons and interprets them as commands to control the motors of the robotic car. For example, if the "forward" button is pressed, the microcontroller sends a signal to the motor driver to drive the motors forward.
- The motor driver receives the control signals from the microcontroller and amplifies them to provide sufficient power to drive the DC motors. The L293D motor driver IC is commonly used in robotic car projects as it can control two DC motors independently.
- The amplified signals are then sent to the DC motors, which rotate the wheels and propel the robotic car in the desired direction.
- The microcontroller continuously monitors the input signals from the push buttons and sends control signals to the motor driver to change the direction and speed of the motors as per the commands received from the buttons.

Overall, the working principle of a button-controlled robotic car using 8051 microcontroller involves a simple control system that allows the user to send commands to the robotic car using push buttons, which are interpreted by the microcontroller and translated into control signals for the motors.

# CHAPTER 5

# RESULTS AND ANALYSIS

## 5.1 Result




## 5.2 Advantages

- Simple and user-friendly: The button-controlled interface is simple and easy to use, making it ideal for beginners who are learning about robotics and programming.
- Low cost: Compared to other sophisticated control systems, a button-controlled robotic car using 8051 microcontroller is relatively low cost, making it affordable for hobbyists and students.
- Easy to customize: As the interface is controlled using push buttons, it is easy to customize the control system by adding or removing buttons or changing their functions.
- Robust and reliable: The 8051 microcontroller is a well-established and widely used microcontroller, known for its robustness and reliability. The use of DC motors also adds to the reliability of the system, as they are simple and durable components that are widely used in many applications.
- Can be used for educational purposes: The simplicity of the control system makes it a suitable platform for educational purposes, allowing students to learn about programming, electronics, and robotics in a hands-on and interactive way.

## 5.3 Applications

- Education: Robotic cars using 8051 microcontrollers are a great way to teach students about robotics, microcontrollers, and programming. They can be used in

schools and universities to provide hands-on experience with robotics and programming.

- Research: Robotic cars using 8051 microcontrollers can be used in research projects to study the behavior of robots and develop new algorithms for control and navigation.
- Prototyping: Robotic cars using 8051 microcontrollers can be used as a platform for prototyping new robotic systems. They provide a low-cost and flexible platform for developing and testing new ideas.

# CHAPTER 6

## CONCLUSION

Autonomous self-driving robotic car is not only a development but a revolution. Self-driving robotic cars are set to revolutionize the way we live. This is transformational technology, on the cutting-edge of robotics, machine learning, software engineering, and mechanical engineering. In 2014, in terms of automation, the standards of autonomous self-driving robotic cars were
standardized. With the spread of autonomous self-driving robotic cars, the role of individual property is put in the background and the car sharing use of more and more space and the transportation will be a service-oriented (Transport-as-a-Service).

## REFERENCES


- https://www.youtube.com/watch?v=tyC669X1EnY
- https://www.idconline.com/technical_references/pdfs/electronic_engineering/8051_Micro_controller_based_Robotic_Car.pdf