**A**

**Mini – Project Report**

**on**

“*App Based Control and Monitor Using LPC2138***”**

**Submitted By**

|  |  |
| --- | --- |
| HARSH BANDGAR | 21410045 |
| RAJWARDHAN VICHARE | 21410058 |
| VISHVAJEET JAGTAP | 21410059 |
| ADITYA PACHORE | 21410066 |

**Under the guidance of**

***Mr. S. R. Khedkar***



Department of Electronics Engineering

Walchand College of Engineering, Sangli

(Government-Aided Autonomous Institute)

ESD LAB mini–Project A.Y. 2023-2024

**Abstract**

The " *App Based Control and Monitor Using LPC2138*" project embodies a sophisticated blend of hardware components, including the LPC2138 microcontroller, L293D motor driver, and HC-05 Bluetooth module, to establish a robust and user-friendly interface for controlling a set of four motors through a mobile application. At the core of this system is the LPC2138 microcontroller, functioning as the central processing unit responsible for interpreting commands received via Bluetooth and translating them into precise motor movements.

The L293D motor driver assumes a pivotal role in this project by efficiently managing the distribution of power to the four motors. Its capability to control the direction and speed of the motors enhances the overall responsiveness and agility of the system. This component ensures optimal performance, contributing to the seamless execution of the designated motor commands.

Facilitating wireless communication, the HC-05 Bluetooth module serves as a critical link between the mobile application and the hardware. This module enables users to transmit commands remotely, fostering a dynamic and versatile motor control experience. Its integration allows for the establishment of a reliable and efficient communication channel, laying the foundation for the responsive execution of motor control instructions.

The user interface of the mobile application plays a crucial role in enhancing the overall project experience. Through four distinct commands – "Forward," "Backward," "Right," and "Left" – users can intuitively dictate the direction and movement of the motors. The real-time feedback, manifested through LCD messages such as "Moving Forward," "Moving Backward," "Turning Right," and "Turning Left," further enriches the user experience by providing instant visual confirmation of the executed commands.

The project's strength lies in its seamless integration of Bluetooth communication with precise motor control, offering a versatile solution for applications that demand not only responsiveness but also specific directional movements. The implementation ensures user-friendly operation, making it adaptable and practical across various contexts, from robotics to home automation. In essence, the "Bluetooth Control System" stands as an innovative and practical solution, showcasing the potential of integrating cutting-edge technologies for efficient motor control in diverse applications.

**Contents**

**Chapter 1**

* 1. **Introduction**
  2. **Project Idea**
  3. **Problem definition**
  4. **Objectives**
  5. **Expected outcomes**

**Chapter 2**

**Literature Survey**

**Chapter 3**

**Design Methodology**

**3.1 System block diagram**

**3.2 Block description**

**3.3 Circuit diagram**

**3.4 List of Components**

**Chapter 4**

**Implementation**

**4.1 Mathematical Modelling**

**4.1.1 Objective 1: Configure HC-05 and LPC2138.**

* + 1. **Objective 2: Configure L293D and LPC2138.**
    2. **Objective 3:** **Design and implement a user interface for vehicle control using a Bluetooth App.**

**4.2 Final Program**

**4.3 Final Proteus simulation**

**Chapter 5**

**Result and Testing**

**5.1 Objective 1**

**5.2 Objective 2**

**5.3 Objective 3**

**5.4 System output**

**Chapter 6**

**Conclusion & Future Scope**

**Chapter 1**

**1.1 Introduction**

In the ever-evolving landscape of automation and remote-control systems, the "*App Based Control and Monitor Using LPC2138*" represents a pioneering venture into the realm of precise and intuitive motor control. This project harnesses the power of advanced technologies, integrating an LPC2138 microcontroller, an L293D motor driver, and an HC-05 Bluetooth module to enable seamless communication and manipulation of four motors through a user-friendly mobile application.

Motivated by the demand for responsive and directional motor control, the project aims to bridge the gap between user input and physical motor movements, enhancing the adaptability of such systems in various applications. The integration of Bluetooth technology not only facilitates wireless communication but also opens avenues for versatile and dynamic control, catering to scenarios where precision and specific motor movements are paramount.

At its core, the LPC2138 microcontroller serves as the intelligence hub, interpreting commands received through Bluetooth signals and orchestrating the synchronized movement of the four motors. This microcontroller not only acts as the processing unit but also provides a programmable interface for defining and adjusting motor control parameters, contributing to the adaptability and flexibility of the system.

The L293D motor driver plays a crucial role in optimizing motor performance by efficiently managing power distribution. This component ensures that the motors operate at their full potential, allowing for precise control over both speed and direction. With the ability to interpret high and low pulses, the L293D motor driver becomes the key player for achieving nuanced and responsive motor movements.

Facilitating the wireless communication aspect, the HC-05 Bluetooth module acts as the intermediary between the mobile application and the hardware, enabling users to remotely command the motors. The integration of this module not only extends the range of control but also simplifies the user experience, making it accessible to a broad spectrum of applications.

The project's user interface, manifested through a mobile application, introduces four distinct commands – "Forward," "Backward," "Right," and "Left." These commands empower users to dictate the precise movements of the motors, providing an intuitive and responsive control experience. The LCD messages, such as "Moving Forward," "Moving Backward," "Turning Right," and "Turning Left," offer real-time feedback, enhancing the overall user interaction and ensuring the successful execution of commands.

In conclusion, the "*App Based Control and Monitor Using LPC2138*" emerges as a cutting-edge solution at the intersection of advanced microcontroller programming, motor control technology, and wireless communication. As we delve into the intricacies of this project, its potential applications range from robotics and automation to user-friendly control systems in smart homes. The project not only embodies technological innovation but also exemplifies a practical and adaptable approach to motor control in the modern era.

**1.2 Project Idea**

The core idea behind the "*App Based Control and Monitor Using LPC2138*" project is to create an innovative and versatile motor control system that seamlessly integrates Bluetooth technology with precise directional movements. The project envisions leveraging an LPC2138 microcontroller, an L293D motor driver, and an HC-05 Bluetooth module to enable users to control the motion of four motors through a mobile application. This concept responds to the demand for user-friendly and dynamic motor control, allowing individuals to intuitively dictate forward and backward movements, as well as precise turns to the left or right, all through a wireless interface.

The project's uniqueness lies in its ability to cater to diverse applications where responsive and specific motor movements are essential. By harnessing the power of Bluetooth communication, the system breaks free from physical constraints, offering a wide range of possibilities for implementation. Whether applied in robotics, automation, or smart home systems, the Bluetooth Control System provides a solution that is not only technologically advanced but also adaptable to various contexts. The overarching project idea revolves around empowering users with a flexible and intuitive means of controlling multiple motors remotely, thereby opening avenues for innovation in the realm of motorized systems.

**1.3 Problem definition**

The "*App Based Control and Monitor Using LPC2138*" project is defined by its objective to develop an advanced motor control system using an LPC2138 microcontroller, L293D motor driver, and HC-05 Bluetooth module. The primary goal is to enable precise control of four motors via a mobile application, offering intuitive commands such as "Forward," "Backward," "Right," and "Left." The project focuses on seamless Bluetooth communication, real-time LCD feedback, and efficient power distribution to ensure responsive and user-friendly motor control. The envisioned outcome is a versatile system applicable in diverse contexts, providing a practical solution for directional motor manipulation.

**1.4 Objectives**

1. Configure L293D and LPC2138
2. Configure HC-05 and LPC2138.
3. Design and implement a user interface for vehicle control using a Bluetooth application.

**1.5 Expected Outcomes**

The expected outcomes of the "Bluetooth Control System" project are as follows:

* **Functional Motor Control System:** The successful implementation of the project will yield a fully operational motor control system. Users will be able to command the four motors wirelessly through a mobile application, with precise and responsive movements.
* **Real-time Feedback through LCD Display:** The LCD display will provide real-time feedback, confirming the execution of user commands. Messages such as "Moving Forward," "Moving Backward," "Turning Right," and "Turning Left" will be displayed, enhancing the user's understanding of the system's actions.
* **Seamless Bluetooth Communication:** The Bluetooth module's integration will ensure seamless communication between the mobile application and the hardware. Users can expect a reliable and responsive link for transmitting motor control commands.
* **Versatility in Motor Control:** The system will showcase versatility by enabling users to execute distinct commands, including "Forward," "Backward," "Right," and "Left." This adaptability allows the project to find applications in a variety of contexts requiring directional motor control.
* **User-friendly Interface:** The mobile application's user interface will be designed to provide an intuitive and user-friendly experience, allowing individuals, regardless of technical expertise, to command and control the motors effortlessly.
* **Enhanced Learning Experience:** Beyond its practical applications, the project is expected to serve as a valuable learning experience, particularly for enthusiasts interested in microcontroller programming, motor control systems, and the integration of Bluetooth technology.
* **Customization Potential:** The system's design will allow for customization to meet specific needs. Users may explore modifications such as adjusting control algorithms, incorporating sensors for feedback, or integrating the system with other devices for expanded functionalities.

**Chapter 2**

**Literature Survey**

The literature survey for the "*App Based Control and Monitor Using LPC2138*" project reveals a comprehensive exploration of related works and existing knowledge in the fields of microcontroller-based motor control, Bluetooth communication, and applications of similar systems. Various studies highlight the significance of incorporating Bluetooth technology for wireless control, emphasizing its reliability and versatility in diverse contexts.

Research in microcontroller-based motor control systems delves into methodologies for precise and responsive motor movements. Studies discuss the role of microcontrollers, such as the LPC2138, in interpreting commands and orchestrating efficient motor control. The integration of L293D motor drivers is explored for optimal power distribution, enhancing motor performance and control accuracy.

Literature also addresses the challenges and innovations in Bluetooth communication for remote control applications. The HC-05 Bluetooth module, a crucial component in the project, is examined for its effectiveness in facilitating seamless communication between mobile applications and hardware.

Additionally, literature sources shed light on the importance of user-friendly interfaces in motor control systems. Studies explore how mobile applications can be designed to offer intuitive controls, ensuring accessibility for users with varying levels of technical expertise. Real-time feedback through LCD displays, as proposed in this project, has been studied for its role in enhancing user interaction and system understanding.

The literature survey collectively contributes to understanding the state-of-the-art in microcontroller-based motor control, Bluetooth technology, and user interface design. The insights gained from existing research will inform the design and implementation of the "*App Based Control and Monitor Using LPC2138*" ensuring it aligns with contemporary practices and addresses challenges identified in the literature.

**References:**

* UM10120

LPC2131/2/4/6/8 User manual

* Various Blogs by EmbeTronics on Bluetooth Interfacing and motor control.

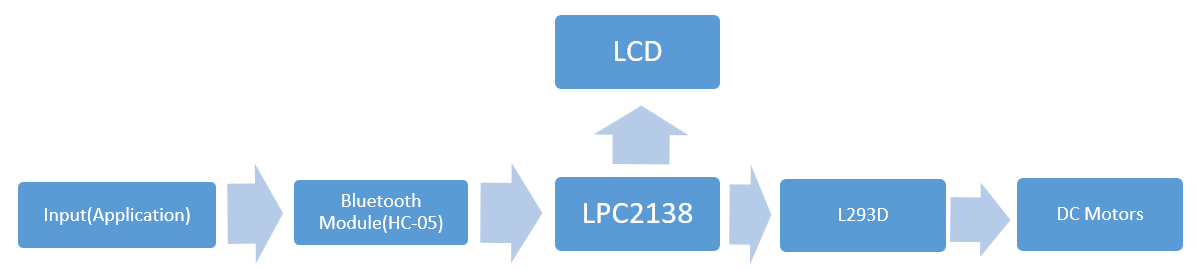
<https://embetronicx.com/>

* "Bluetooth Essentials for Programmers" by Albert S. Huang
* "Electric Motor Drives: Modelling, Analysis, and Control" by R. Krishnan

**Chapter 3**

**Design Methodology**

**3.1 System Block Diagram**



* 1. **Block Description**

1. **Input (Application):**

The project starts with user input through a mobile application, providing commands such as "Forward," "Backward," "Right," and "Left." This serves as the primary interface for users to control the motors wirelessly.

1. **Bluetooth Module (HC-05):**

The user commands from the application are transmitted wirelessly via Bluetooth to the HC-05 module. The HC-05 acts as a communication bridge, facilitating the transfer of control signals from the mobile application to the hardware components.

1. **LPC2138 Microcontroller:**

The HC-05 output is received by the LPC2138 microcontroller, which functions as the central processing unit. It interprets the Bluetooth signals, processes the user commands, and generates corresponding control signals for the motors based on the programmed logic.

1. **L293D Motor Driver:**

The control signals from the LPC2138 are then directed to the L293D motor driver. The L293D serves as an interface between the microcontroller and the DC motors, ensuring efficient power distribution and control over the motors' speed and direction.

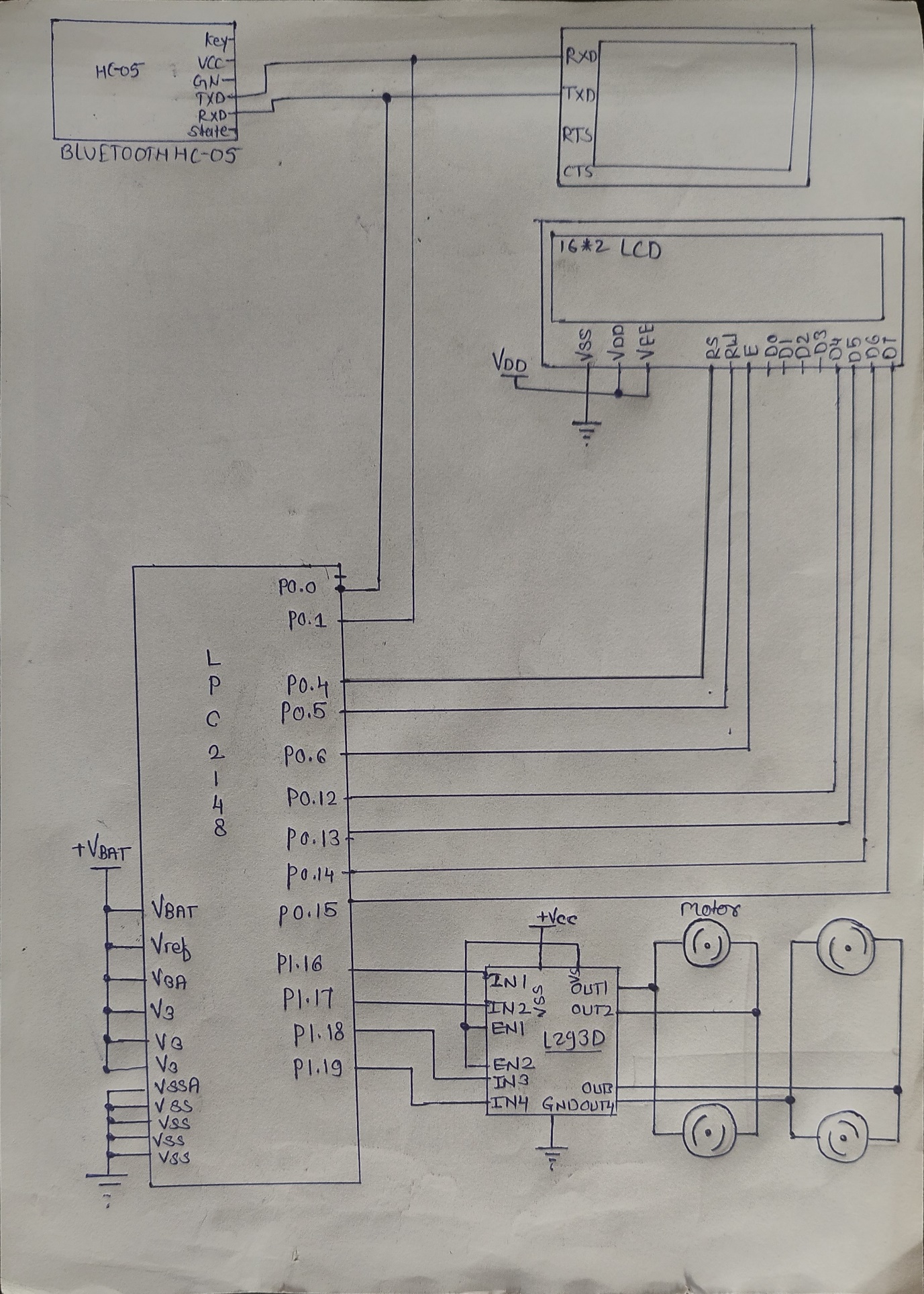
1. **DC Motors:**

The L293D outputs the processed signals to the connected DC motors. Depending on the specific command received, the DC motors will adjust their rotation, allowing the system to execute movements such as forward, backward, right turn, or left turn.

1. **LCD (Liquid Crystal Display):**

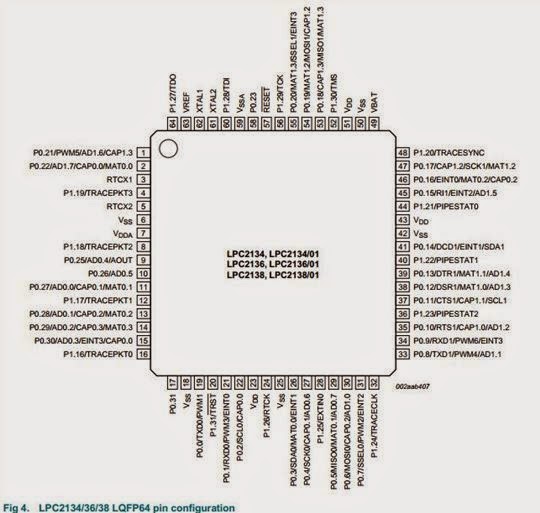
An LCD display is incorporated into the system to provide real-time feedback to the user. The LCD displays messages confirming the execution of commands, such as "Moving Forward," "Moving Backward," "Turning Right," or "Turning Left." This component enhances the user's understanding of the system's actions.

**3.3 Circuit Diagram**



**3.4 List of component**

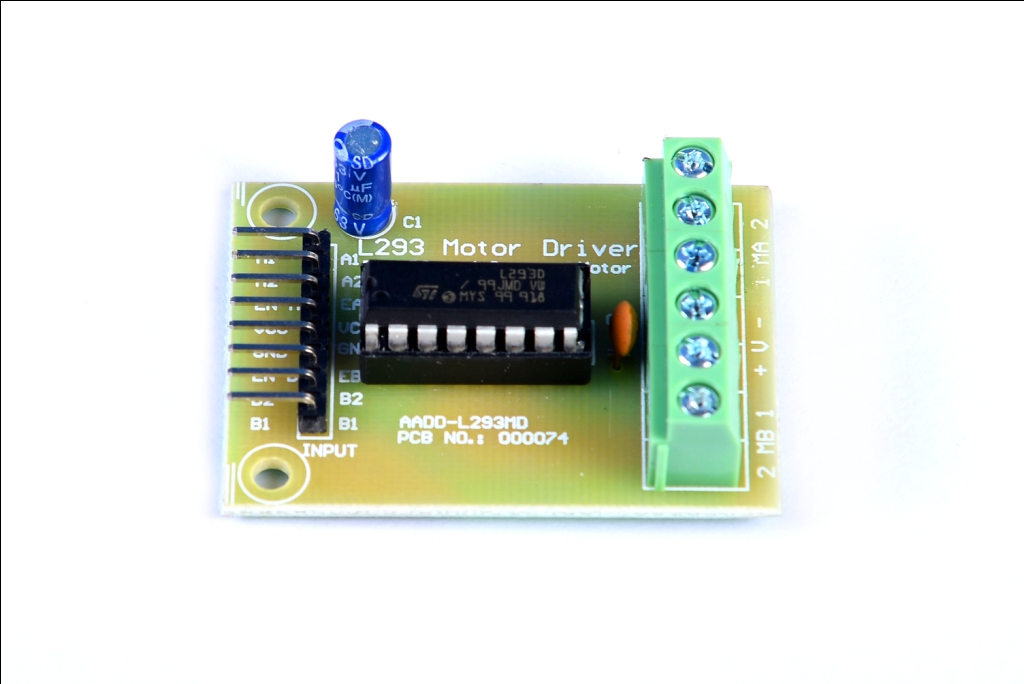
1)Microcontroller-LPC2138:

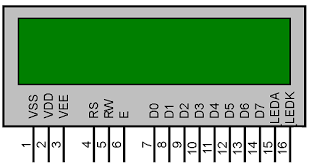
2) DC motor:



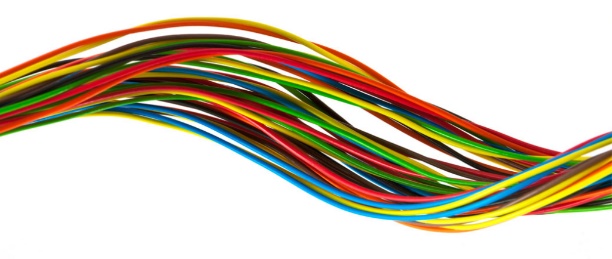
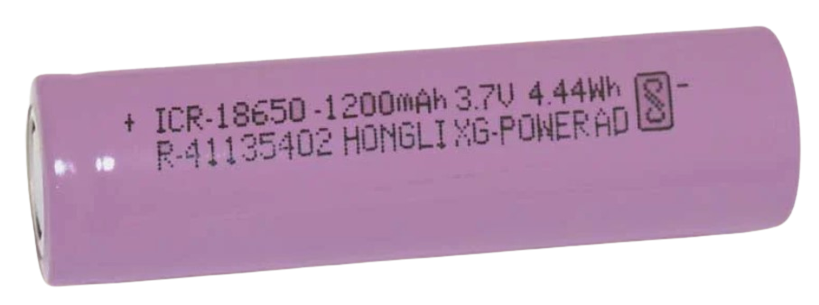
3) L293D Motor Driver :



4) 16 \* 2 LCD:



5) Wires and Power Supply

**Chapter 4**

**Implementation**

**4.1 Mathematical modelling**

Calculations of M and P:

M is a multiplying factor

Cclk = M x Fosc

Cclk is CPU clock

Fosc is Oscillator input to PLL

P is the division factor

It allows to keep the CCO frequency within range

Fcco = Cclk x2xP

Cclk = Fcco/(2 x P).

Cclk = M x Fosc = Fcco / (2 x P)

The PLL output frequency (when the PLL is both active and connected) is given by:

CCLK = MxFosc or CCLK = Fcco/(2 x P)

The CCO frequency can be computed as:

Fcco = CCLKx2xP or Fcco = FoscxMx2xP

**The PLL inputs and settings must meet the following:**

• Fosc is in the range of 10 MHz to 25 MHz.

• CCLK is in the range of 10 MHz to Fmax (the maximum allowed frequency for the microcontroller - determined by the system microcontroller is embedded in).

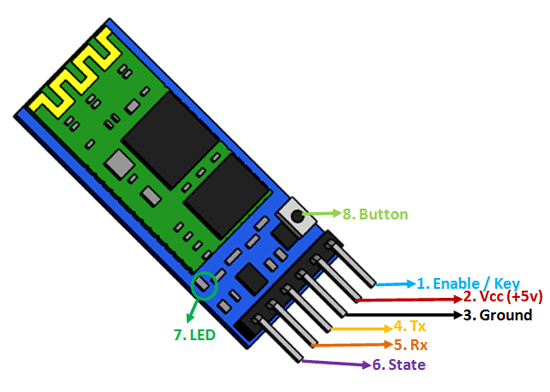
• Fcco is in the range of 156 MHz to 320 MHz.

**4.1.1 Objective 1: Configure HC-05 and LPC2138.**

**Explanation:**

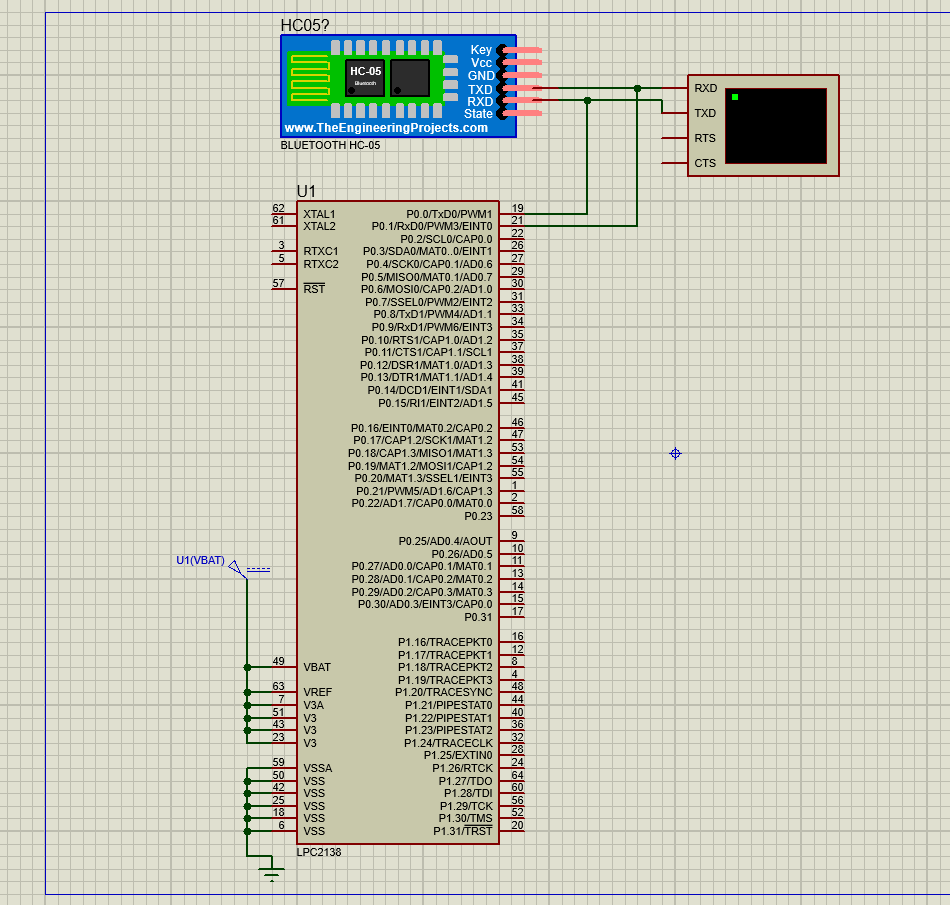
**HC-05 Bluetooth module:**

**Pin configuration:**



1. VCC (Pin 1):
   * Description: Power Supply
   * Function: Connect this pin to a power source to provide power to the module. The operating voltage is typically 3.3V, but some modules support a voltage range of 3.6V to 6V.
2. GND (Pin 2):
   * Description: Ground
   * Function: Connect this pin to the ground (0V) reference of your circuit.
3. TXD (Pin 3):
   * Description: Transmit Data
   * Function: This pin is used for serial communication. Connect it to the RX (receive) pin of the device (microcontroller or module) you are communicating with.
4. RXD (Pin 4):
   * Description: Receive Data
   * Function: This pin is used for serial communication. Connect it to the TX (transmit) pin of the device (microcontroller or module) you are communicating with.
5. STATE (Pin 5):
   * Description: State
   * Function: This pin provides information about the current state of the module. It can be used to check if the module is in a discoverable or non-discoverable state.
6. KEY (Pin 7):
   * Description: Key/Button
   * Function: This pin is often used to enter the module into AT (command) mode. It can be connected to a push-button or pulled HIGH or LOW based on the module's configuration.

**To configure L293D and LPC2138 Microcontroller, you need to follow these steps:**

****

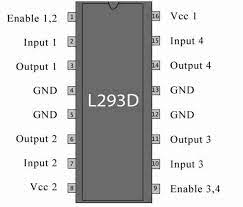
1. Connect the HC-05 Bluetooth module to the LPC2138 microcontroller:
2. Connect the pins of the HC-05 to the corresponding pins of the LPC2138 microcontroller. (Rx and Tx Pins)
3. Program the LPC2138 Microcontroller:
4. Write a program for signal receiving and transmitting
5. Use the Rx and Tx pins of the LPC2138 microcontroller to connect HC-05 Bluetooth module
6. Test the circuit:
7. Use Virtual Terminal to check the message receiving and transmitting is functioning properly

**4.1.2** **Objective 2: Configure L293D and LPC2138**.

**Explanation:**

**L293D Motor Driver:**

**Pin configuration:**

****

|  |  |  |
| --- | --- | --- |
| **Pin No.** | **Pin Name** | **Description** |
| **Power Supply Pins** | | |
| 1 | **VCC** | VCC pin is used to supply power to the motor. Its input voltage is between 4.5 to 36V. |
| 2 | **GND** | GND is a ground pin. It needs to be connected to the power supply ground(negative). |
| **Control Pins** | | |
| 1 | **IN1** | These pins are input pins of **Motor A**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop. |
| 2 | **IN2** |
| 3 | **IN3** | These pins are input pins of **Motor B**. These are used to control the rotating direction of Motor A. When one of them is HIGH and the other is LOW, Motor A will start rotating in a particular direction. If both the inputs are either HIGH or LOW the Motor A will stop. |
| 4 | **IN4** |
| **Speed Control Pins** | | |
| 1 | **ENA** | ENA pin is used to control the speed of **Motor A**. If a jumper is present on this pin, so the pin connected to +5 V and the motor will be enabled, then the Motor A rotates maximum speed. If we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor A. If we connect this pin to Ground the Motor A will be disabled. |
| 2 | **ENB** | ENB pin is used to control the speed of **Motor B**. If a jumper is present on this pin, so the pin connected to +5 V and the motor will be enabled, then the Motor B rotates maximum speed. If we remove the jumper, we need to connect this pin to a PWM input of the microcontroller. In that way, we can control the speed of Motor B. If we connect this pin to Ground the Motor B will be disabled. |
| **Output Pins** | | |
| 1 | **OUT1** & **OUT2** | This terminal block will provide the output for **Motor A**. |
| 2 | **OUT3** & **OUT4** | This terminal block will provide the output for **Motor B** |

**How Motor Driver Module Works:**

This module uses two techniques for the control speed and rotation direction of the DC motors. These are H-Bridge – For controlling rotation direction and PWM – For controlling the speed.

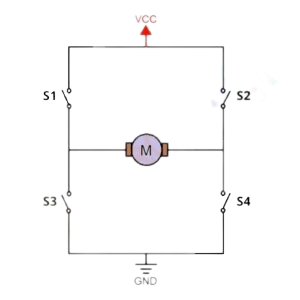
**H-Bridge Techniques:**

L293D motor driver module uses the H-Bridge technique to control the direction of rotation of a DC motor. In this technique, H-Bridge controlled DC motor rotating direction by changing the polarity of its input voltage.

An H-Bridge circuit contains four switching elements, like transistors (BJT or MOSFET), with the motor at the center forming an H-like configuration. Input**IN1, IN2, IN3, and IN4** pins actually control the **switches** of the H-Bridge circuit inside L293D IC.  
We can change the direction of the current flow by activating two particular switches at the same time, this way we can change the rotation direction of the motor.

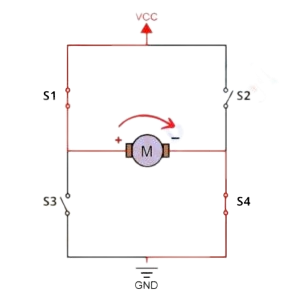
**Case 1**

When S1, S2, S3, and S4 all switches are open then no current goes to the Motor terminals. So, in this condition, the motor is stopped (not working).



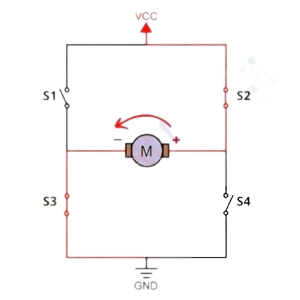
**Case 2**

When the switch S1 and S4 are closed, then the motor left terminal is getting a positive (+) voltage and the motor right terminal is getting a negative(-) voltage. So, in this condition motor start rotating in a particular direction (clockwise).



**Case 3**

When S2 and S3 switches are closed, then the right motor terminal is getting a positive (+) voltage and the left motor terminal is getting a negative (-) voltage. So, in this condition motor start rotating in a particular direction (anticlockwise).



**Module Specifications & Features:**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| Operating Voltage | 4.5V – 36V |
| Operating Current | 2A |
| Logic Voltage | 5V |
| Logical Current | 0-36mA |
| Maximum Power (W) | 25W |
| Driver Chip | L293D quadruple half-H driver IC |
| LED lights indicators | Power-On LED indicator |
| Drives motor | Drives up to 4 motors (2 for each motor output terminal block) or One Stepper Motor |
| Module Dimensions | 44 x 44 x 28 (L x W x H) mm |

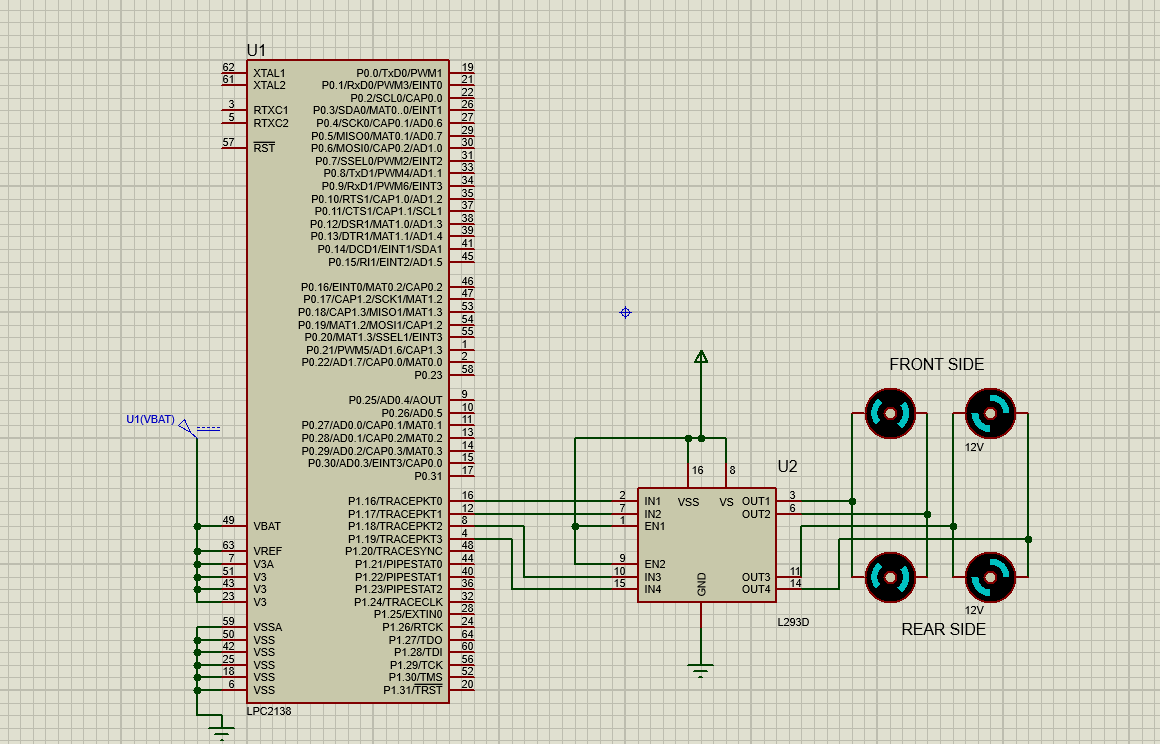
**Motor Driver Truth Tables**

| **Motor A truth table** | | | |
| --- | --- | --- | --- |
| **ENA** | **IN1** | **IN2** | **Description** |
| **0** | **N/A** | **N/A** | **Motor A is off** |
| **1** | **0** | **0** | **Motor A is stopped (brakes)** |
| **1** | **0** | **1** | **Motor A is on and turning backwards** |
| **1** | **1** | **0** | **Motor A is on and turning forwards** |
| **1** | **1** | **1** | **Motor A is stopped (brakes)** |

| **Motor B truth table** | | | |
| --- | --- | --- | --- |
| **ENB** | **IN3** | **IN4** | **Description** |
| **0** | **N/A** | **N/A** | **Motor B is off** |
| **1** | **0** | **0** | **Motor B is stopped (brakes)** |
| **1** | **0** | **1** | **Motor B is on and turning backwards** |
| **1** | **1** | **0** | **Motor B is on and turning forwards** |
| **1** | **1** | **1** | **Motor B is stopped (brakes)** |

**Applications**

* Control DC motors.
* Control stepping motors
* In Robotics

**To configure L293D and LPC2138 Microcontroller, you need to follow these steps:** ****

1. Connect the L293D motor driver to the LPC2138 microcontroller:
2. Connect the pins of the L293D to the corresponding pins of the LPC2138 microcontroller.
3. Connect the power supply (5V or 12V) to the L293D motor driver.
4. Configure the L293D motor driver:
5. Set the enable pins of the L293D to high, to enable the motor driver.
6. Set the direction pins (IN1 and IN2) of the L293D to control the direction of the motor.
7. Program the LPC2138 Microcontroller:
8. Write a program to control the direction of the motor.
9. Use the GPIO pins of the LPC2138 microcontroller to control the L293D motor driver.
10. Test the circuit:
11. Apply a voltage to the motor driver and check the direction and speed of the motor.

**4.1.3 Objective 3: Design and implement a user interface for vehicle control using a Bluetooth App.**

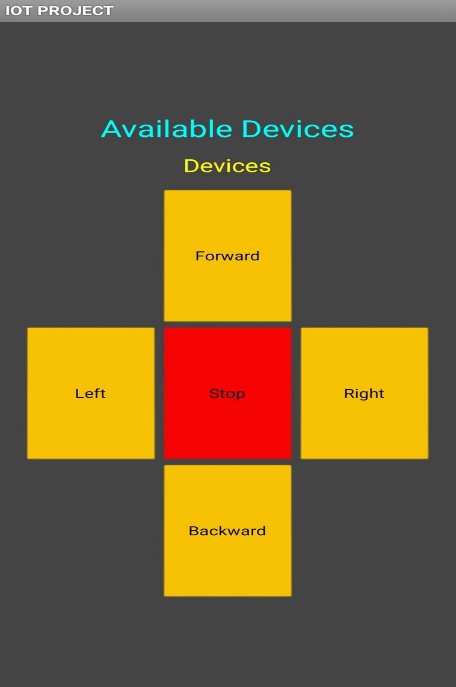
**Explanation:**

Basic structure to implement any User Interface (UI)

The program initializes the necessary hardware components including the LCD. The program enters a while loop that continuously reads the input from the app to adjust the motor direction.

Objective 4.1.3 outlines the task of designing and implementing a user interface for vehicle control through a Bluetooth application. This objective involves creating a visual and interactive interface that users can engage with to operate a vehicle. The design process encompasses crafting screens, controls, and layouts for the Bluetooth application. The user interface will incorporate controls for managing the direction of vehicle, such as Forward, Backward, Left and Right. The technology chosen for implementation is Bluetooth, requiring the application to establish a reliable connection with the vehicle, and the development phase involves frontend programming. Rigorous testing is crucial to ensure the functionality and security of the interface, considering the sensitive nature of vehicle control. Throughout the process, attention is given to user experience principles, aiming to deliver an intuitive and user-friendly interface for controlling the vehicle securely and effectively.

**Application Layout:**

****

**4.2 Final Program**

#include<lpc213x.h>

#include <stdint.h>

#include <stdlib.h>

#include <stdio.h>

#define bit(x) (1<<x)

#define delay for(i=0;i<=60000;i++)

void delay\_ms(unsigned int j) /\* Function for delay in milliseconds \*/

{

    unsigned int x,i;

    for(i=0;i<j;i++)

    {

    for(x=0; x<6000; x++);    /\* loop to generate 1 millisecond delay with Cclk = 60MHz \*/

    }

}

void LCD\_CMD(char command)

{

    IOPIN0 = ( (IOPIN0 & 0xFFFF00FF) | ((command & 0xF0)<<8) );

    IOSET0 = 0x00000040; /\* EN = 1 \*/

    IOCLR0 = 0x00000030; /\* RS = 0, RW = 0 \*/

    delay\_ms(5);

    IOCLR0 = 0x00000040; /\* EN = 0, RS and RW unchanged(i.e. RS = RW = 0)    \*/

    delay\_ms(5);

    IOPIN0 = ( (IOPIN0 & 0xFFFF00FF) | ((command & 0x0F)<<12) );

    IOSET0 = 0x00000040; /\* EN = 1 \*/

    IOCLR0 = 0x00000030; /\* RS = 0, RW = 0 \*/

    delay\_ms(5);

    IOCLR0 = 0x00000040; /\* EN = 0, RS and RW unchanged(i.e. RS = RW = 0)    \*/

    delay\_ms(5);

}

void LCD\_INIT(void)

{

    IODIR0 = 0x0000FFF0; /\* P0.12 to P0.15 LCD Data. P0.4,5,6 as RS RW and EN \*/

    delay\_ms(20);

    LCD\_CMD(0x02);  /\* Initialize lcd in 4-bit mode \*/

    LCD\_CMD(0x28);  /\* 2 lines \*/

    LCD\_CMD(0x0C);   /\* Display on cursor off \*/

    LCD\_CMD(0x06);  /\* Auto increment cursor \*/

    LCD\_CMD(0x01);   /\* Display clear \*/

    LCD\_CMD(0x80);  /\* First line first position \*/

}

void LCD\_CHAR (char msg)

{

        IOPIN0 = ( (IOPIN0 & 0xFFFF00FF) | ((msg & 0xF0)<<8) );

        IOSET0 = 0x00000050; /\* RS = 1, EN = 1 \*/

        IOCLR0 = 0x00000020; /\* RW = 0 \*/

        delay\_ms(2);

        IOCLR0 = 0x00000040; /\* EN = 0, RS and RW unchanged(i.e. RS = 1, RW = 0) \*/

        delay\_ms(5);

        IOPIN0 = ( (IOPIN0 & 0xFFFF00FF) | ((msg & 0x0F)<<12) );

        IOSET0 = 0x00000050; /\* RS = 1, EN = 1 \*/

        IOCLR0 = 0x00000020; /\* RW = 0 \*/

        delay\_ms(2);

        IOCLR0 = 0x00000040; /\* EN = 0, RS and RW unchanged(i.e. RS = 1, RW = 0) \*/

        delay\_ms(5);

}

void LCD\_STRING (char\* msg)

{

    unsigned int i=0;

    while(msg[i]!=0)

    {

LCD\_CHAR(msg[i]);

        i++;

    }

}

void LCD\_CLEAR\_LINE2(void)

{

    unsigned int k;

    LCD\_CMD(0xC0);   /\* Move cursor to the beginning of the second line \*/

    for(k = 0; k < 16; k++)

    {

        LCD\_CHAR(' ');  /\* Write a space \*/

    }

}

void LCD\_UPDATE\_LINE2(char\* msg)

{

    LCD\_CLEAR\_LINE2();  /\* Clear the second line \*/

    LCD\_CMD(0xC0);   /\* Move cursor to the beginning of the second line \*/

    LCD\_STRING(msg); /\* Write the new message \*/

}

void print()

{

    LCD\_INIT();

    LCD\_STRING("System Is ON !!!");

    LCD\_CMD(0x01);   /\* Clear display \*/

  delay\_ms(2);     /\* Wait for the command to complete \*/

    LCD\_STRING("Moving Condition");

}

unsigned int i;

void pll()

{

    PLLCON=0X01;

    PLLCFG=0X24;

    PLLFEED=0XAA;

    PLLFEED=0X55;

    while((PLLSTAT&(1<<10))==0);

    PLLCON=0X03;

    PLLFEED=0XAA;

    PLLFEED=0X55;

    VPBDIV=0x02;                         //pclk=30mhz

}

void ser\_int()

{

    PINSEL0|=0x05;

    U0LCR=0x83;

    U0DLL=195;

    U0DLM=0;

    U0LCR=0x03;

}

void tx(unsigned char c)

{

    while((U0LSR&(1<<5))==0);

    U0THR=c;

}

char rx()

{

    unsigned char a;

    while((U0LSR&(1<<0))==0);

    a=U0RBR;

    return a;

}

int main()

{

    unsigned char b;

    IODIR1=0xf0000;                     //Declaring as a output

    pll();

    ser\_int();

      print();

    while(1)

    {

        b=rx();

        tx(b);

        if(b == 'F')

        {

            //Forward/

            LCD\_CMD(0xCC);

            IOSET1=bit(16) | bit(18);           //IN1 = 1, IN3 = 1

            IOCLR1=bit(17) | bit(19);           //IN2 = 0, IN4 = 0

            LCD\_UPDATE\_LINE2("Moving Forward");

        }

        else if (b == 'B')

        {

            //Reverse/

            LCD\_CMD(0xCC);

            IOSET1=bit(17) | bit(19);           //IN2 = 1, IN4 = 1

            IOCLR1=bit(16) | bit(18);           //IN1 = 0, IN3 = 0

            LCD\_UPDATE\_LINE2("Moving Backward");

        }

        else if (b == 'R')

        {

            //Right/

            LCD\_CMD(0xCC);

            IOSET1=bit(16) ;                                            //IN1 = 1

            IOCLR1=bit(17) | bit(19) | bit(18);     //IN2 = 0, IN3 = 0, IN4 = 0

            LCD\_UPDATE\_LINE2("Turning Right");

        }

        else if (b == 'L')

        {

            //Left/

            LCD\_CMD(0xCC);

            IOSET1=bit(18) ;                                            //IN3 = 1

            IOCLR1=bit(16) | bit(17) | bit(19);     //IN2 = 0, IN1 = 0, IN4 = 0

            LCD\_UPDATE\_LINE2("Turning Left");

        }

        else if (b == 'S')

        {

            //Off/

            LCD\_CMD(0xCC);

            IOCLR1=bit(16) | bit(17) | bit(18) | bit(19); //IN1 = IN2 = IN3 = IN4 = 0

            LCD\_UPDATE\_LINE2("Stopping Vehicle");

            delay\_ms(20);

            LCD\_UPDATE\_LINE2("Vehicle Stopped");

        }

        else

        {

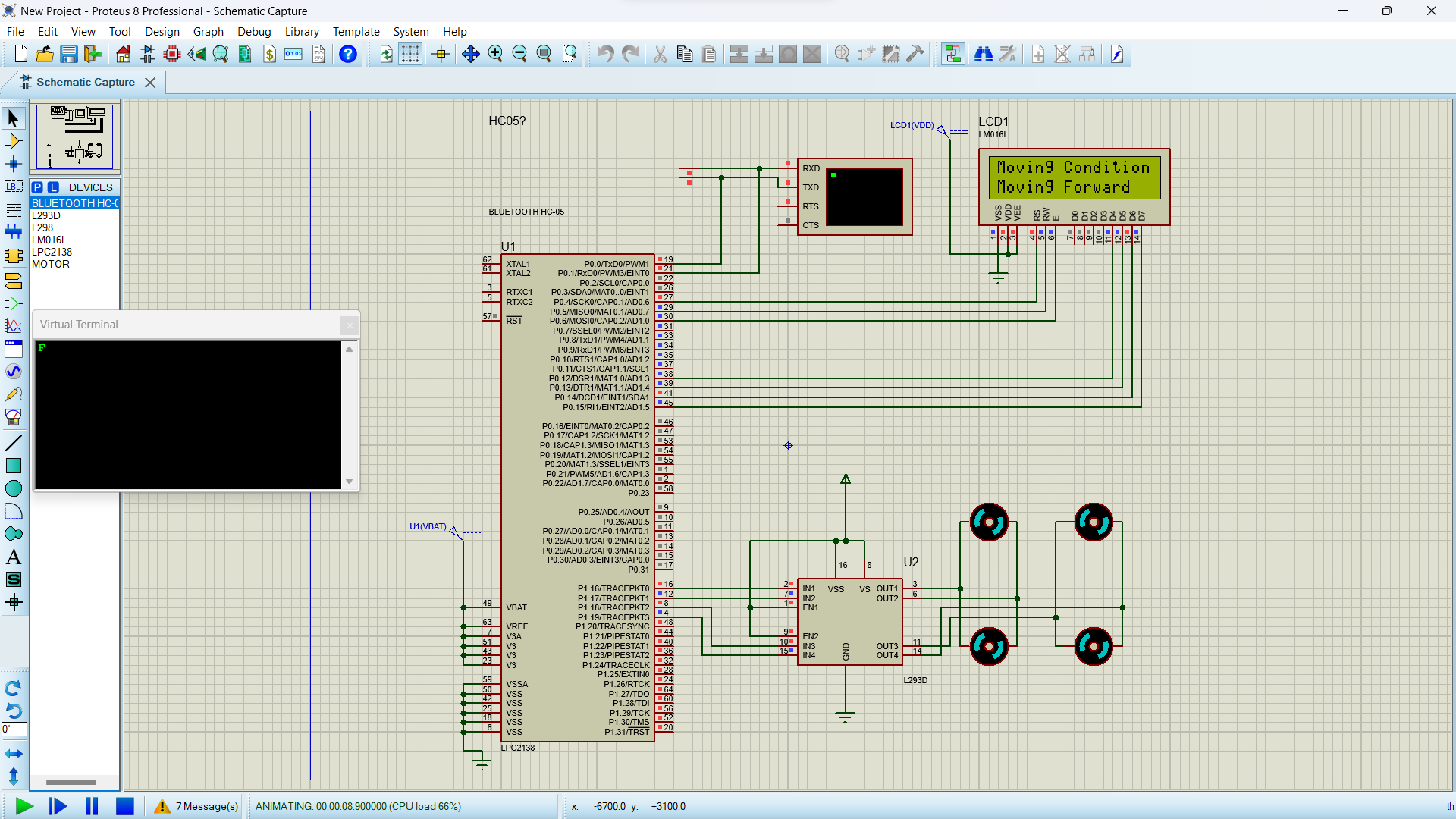
            LCD\_UPDATE\_LINE2("Vehicle Parked");

        }

    }

}

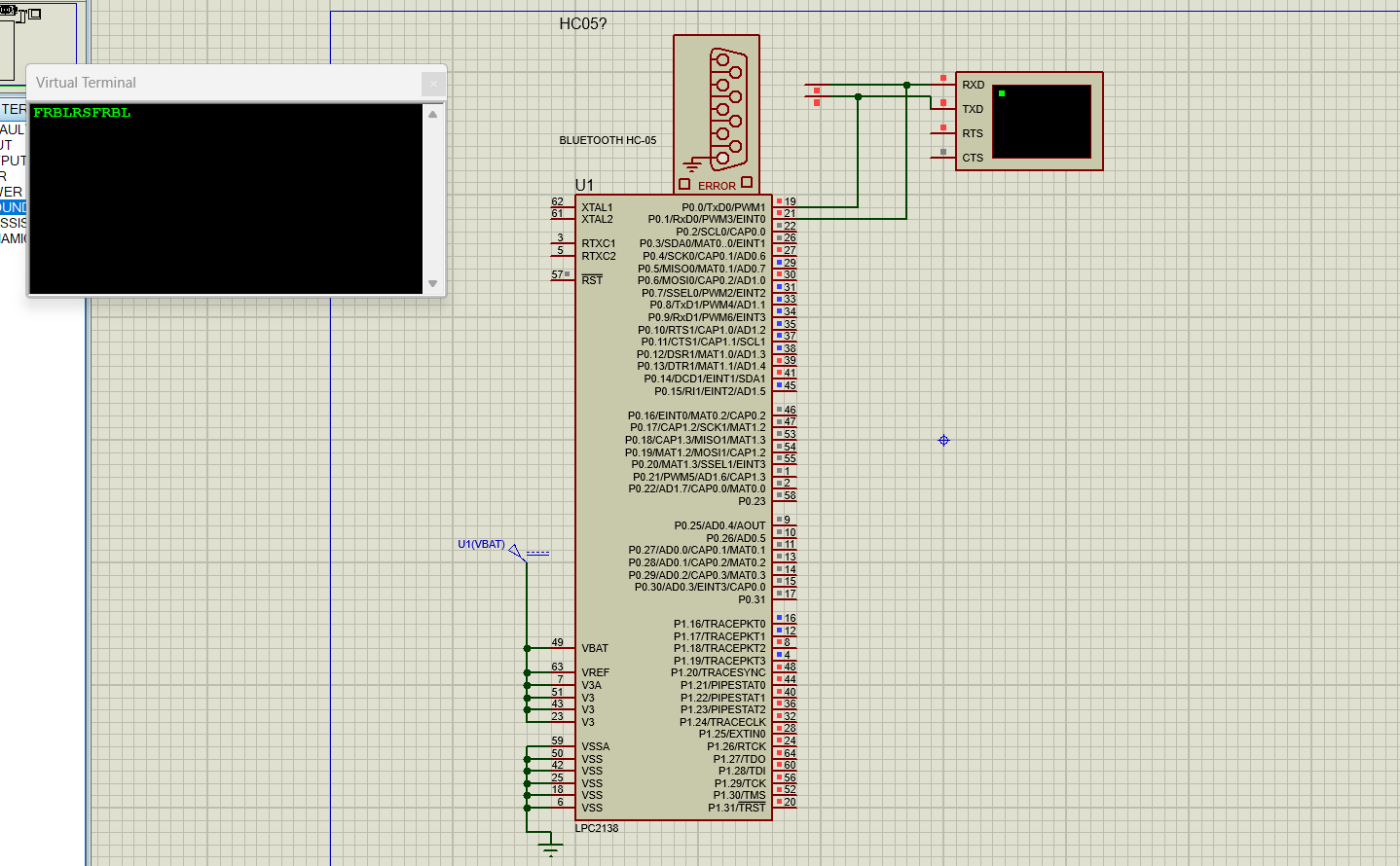
**4.3 Final Proteus simulation**



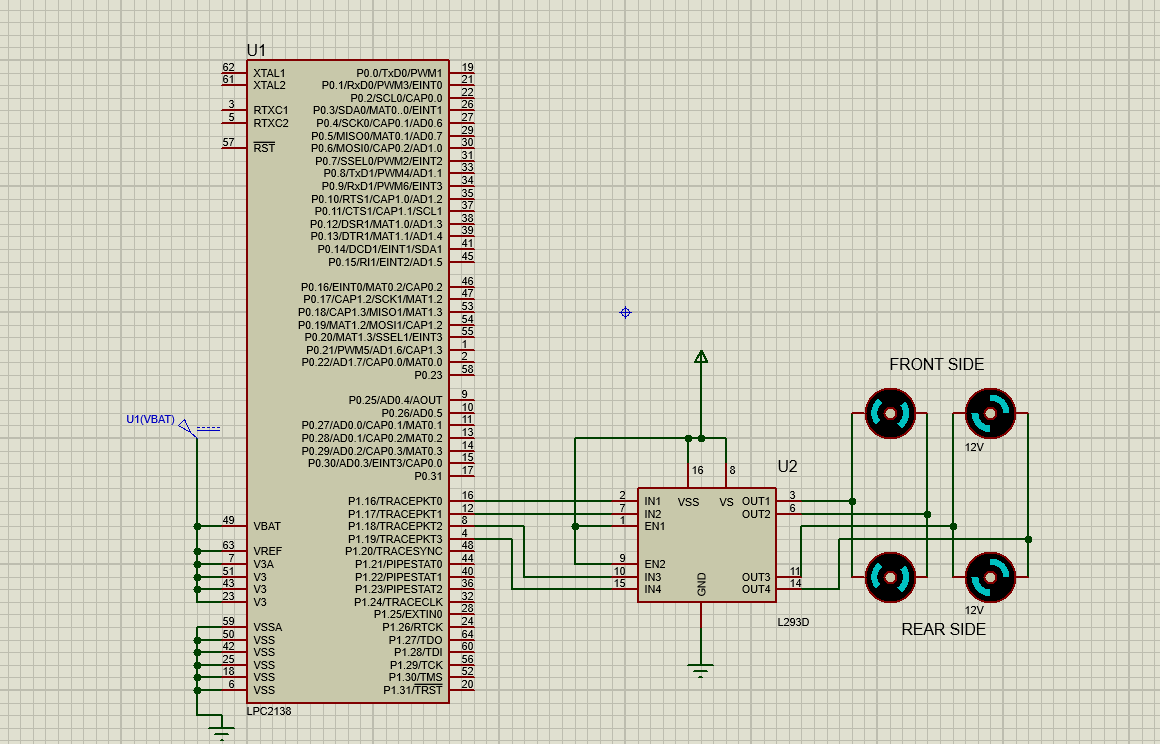
**Chapter 5**

**Result and Testing**

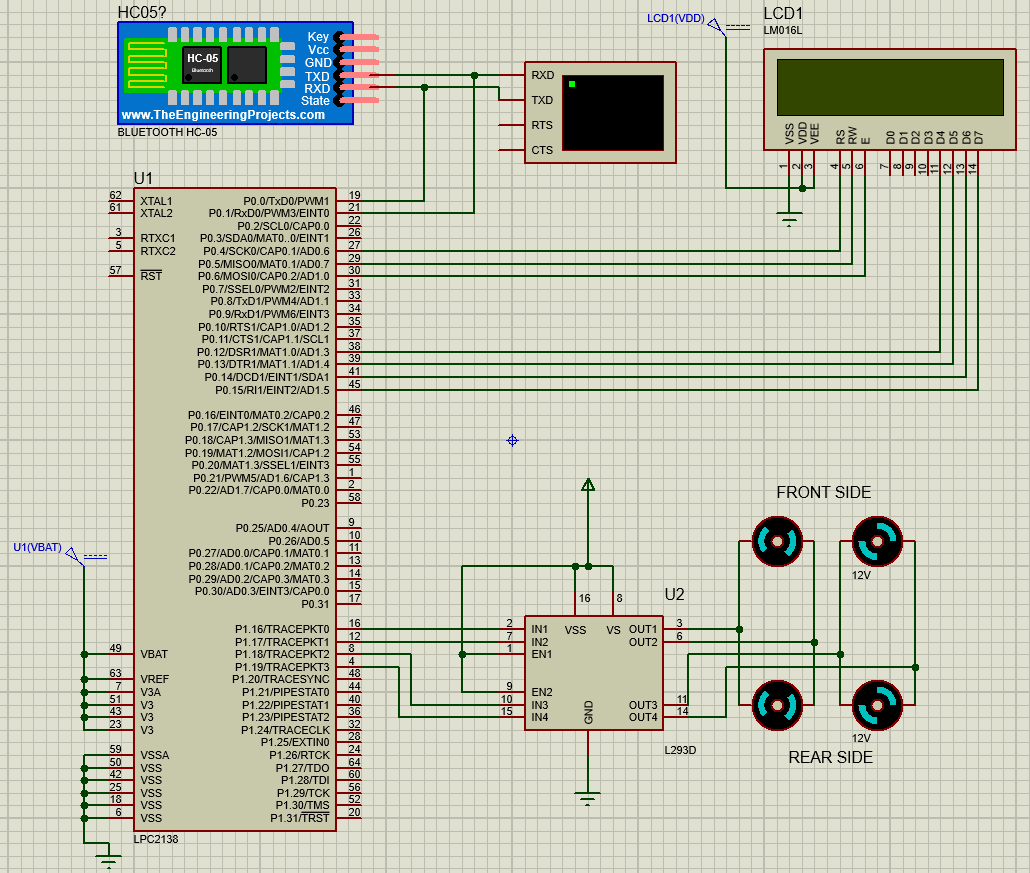
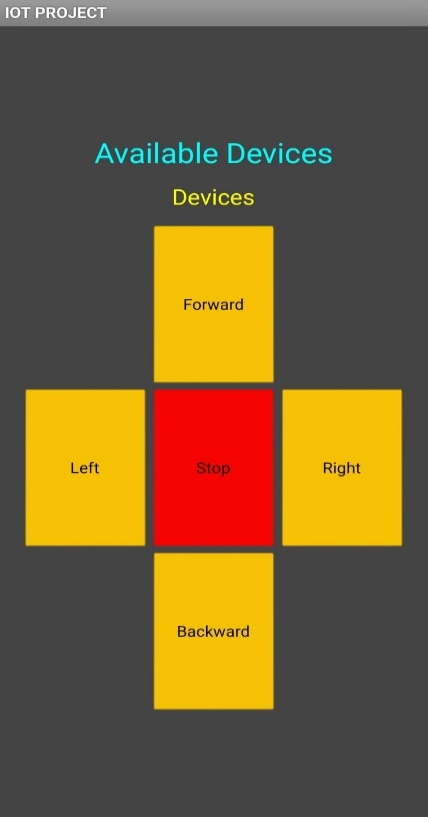
**5.1 Objective 1**

****

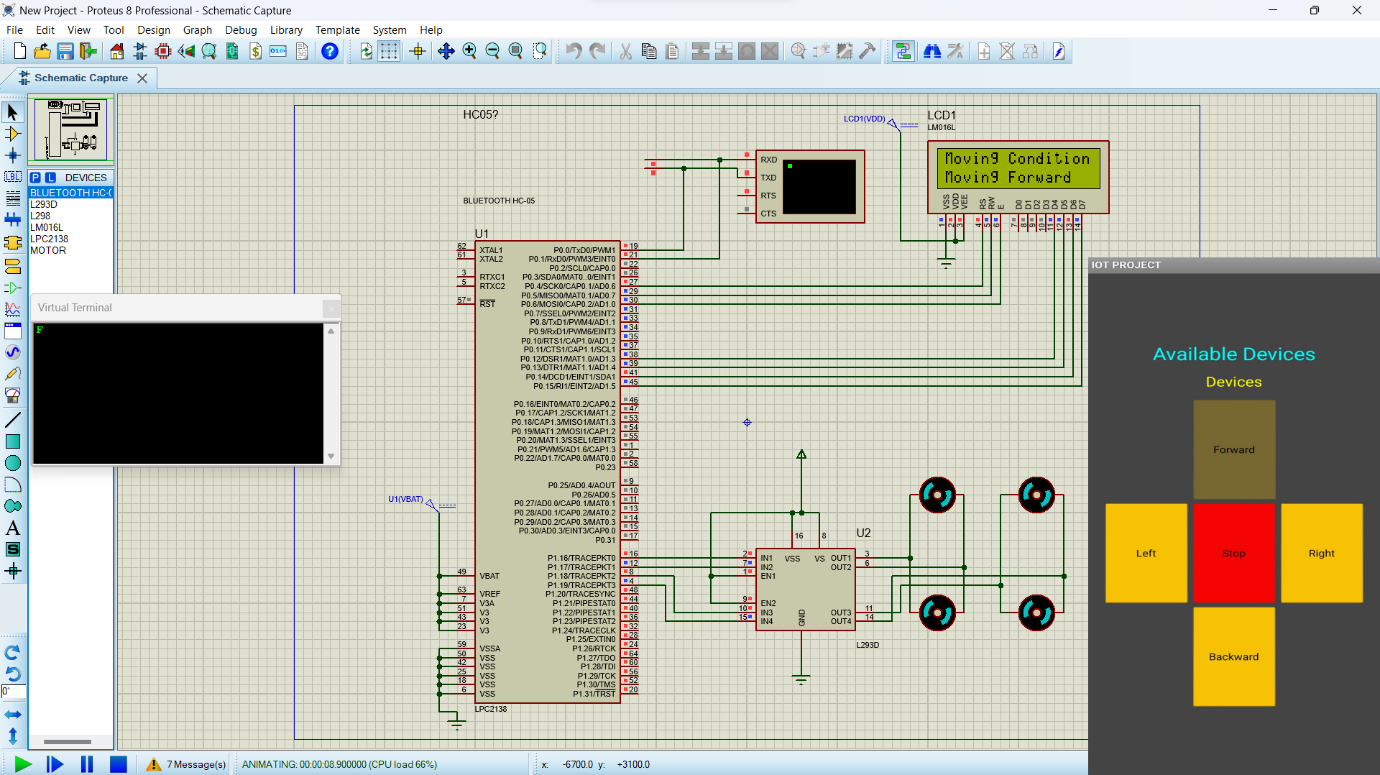
**5.2 Objective 2**

****

**5.3 Objective 3**



**5.4 System output**



**Chapter 6**

**Conclusion & Future Scope**

The " *App Based Control and Monitor Using LPC2138*" project has successfully demonstrated the seamless integration of advanced technologies to achieve precise and intuitive control of four motors. Through the harmonious collaboration of the LPC2138 microcontroller, L293D motor driver, and HC-05 Bluetooth module, the system provides users with a dynamic and versatile motor control experience.

The user-friendly mobile application interface, offering commands such as "Forward," "Backward," "Right," and "Left," has proven effective in translating user intentions into distinct motor movements. The accompanying LCD messages, including "Moving Forward," "Moving Backward," "Turning Right," and "Turning Left," offer real-time feedback, enhancing the overall user experience and providing a visual confirmation of executed commands.

The core microcontroller acts as the project's central processing unit, effectively interpreting Bluetooth signals and orchestrating precise motor movements. The L293D motor driver plays a pivotal role in optimizing motor performance, ensuring efficient power distribution and control over both speed and direction.

The HC-05 Bluetooth module serves as a reliable bridge between the mobile application and the hardware, enabling seamless and remote control. Its implementation has not only enhanced wireless communication but has also contributed to the project's adaptability and practicality in diverse applications.

In conclusion, the " *App Based Control and Monitor Using LPC2138*" project exemplifies the successful convergence of Bluetooth communication and precise motor control. Its innovative design, real-time feedback mechanisms, and user-friendly operation make it a valuable tool for applications requiring responsive and directional movement control. The project opens avenues for further exploration in robotics, home automation, and various industrial applications, showcasing the potential of integrating cutting-edge technologies to meet evolving control system needs. As a result, the project stands as a testament to the effectiveness and practicality of employing advanced technologies in motor control systems.

The mini project "*App Based Control and Monitor Using LPC2138*" can be further enhanced and expanded in several ways. Here are some future scope possibilities:

i) **Efficiency Monitoring:** Integrate sensors to measure motor parameters, enabling real-time data analysis for optimizing energy usage and identifying potential inefficiencies.

ii) **Closed-Loop Control:** Implement feedback mechanisms using sensors to dynamically adjust motor parameters, ensuring consistent and precise control under varying conditions.

iii) **Long Range Controls:** Extend control capabilities beyond Bluetooth by incorporating technologies like LoRa or GSM for remote and long-range motor control in diverse applications.

Overall, the mini project "*App Based Control and Monitor Using LPC2138*" has immense potential for further development and expansion in various domains, including robotics, automation, and industrial control.