University of Sri Jayewardenepura

Faculty of Applied Sciences

I.C.T 305 2.0

Embedded System

SMART VACUUM CLEANER

Project Report

Hashini Wijenayake

AS2020906

Acknowledgments

I would really like to place on record my deep sense of gratitude to Dr. M.D.R. Perera, Senior lecturer in Computer Science, Department of Computer Science, Faculty of Applied Sciences, University of Sri Jayewardenepura, Gangodawila, Nugegoda, Sri Lanka for his generous guidance, assist and beneficial suggestions. I specific my honest assistance to the embedded system project.

Contents

1.Introduction

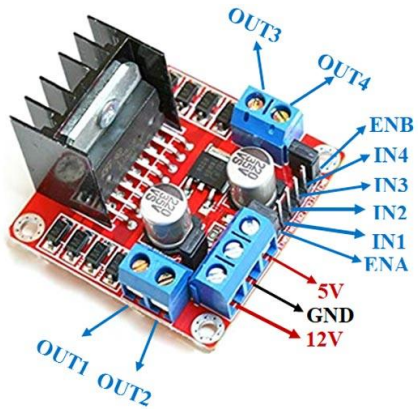
Time-saving and efficient solutions are essential in today's fast-paced world. One of the best examples of this trend is the Smart Vacuum Cleaner Robot, which runs on Arduino. This project report examines the cutting-edge capabilities and benefits of this clever robot, which is made to improve upkeep and clean your living areas on its own.

2. Components & Design Overview

2.1 Arduino Uno board

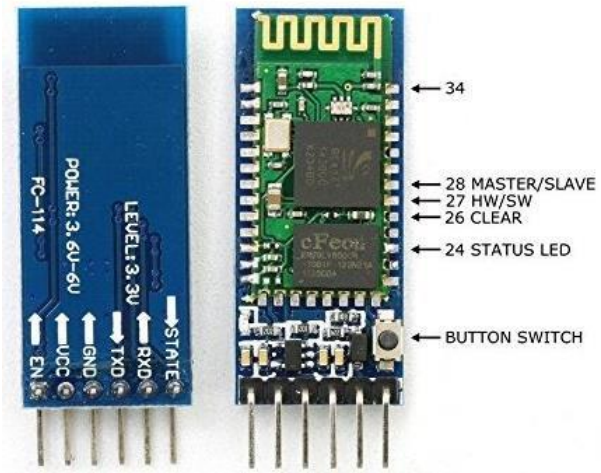
The Arduino Uno is a popular open-source microcontroller board designed for beginners and hobbyists interested in electronics and programming. It is based on the ATmega328P microcontroller and features a user-friendly interface for connecting sensors, actuators, and other components. The board includes digital and analog input/output pins, a USB interface for programming, and a simple integrated development environment (IDE) that makes it easy to write and upload code. Arduino Uno is widely used for a wide range of projects, from simple blinking LED experiments to more complex robotics and automation applications. It's a versatile and accessible platform for learning about electronics and creating interactive projects.

2.2 LM298 Motor drive



The LM298 motor driver is a popular integrated circuit (IC) used to control and drive DC motors and stepper motors in a variety of applications. It can handle higher current loads and provides both direction and speed control for motors. The L298N features dual H-bridge motor driver circuits, making it suitable for bi-directional control of two motors or a single stepper motor. It can be controlled easily by a microcontroller or other digital logic devices, making it a valuable component in robotics, automation, and other projects where motor control is required. The IC offers built-in protection features and is widely used for driving motors in hobbyist and industrial applications.

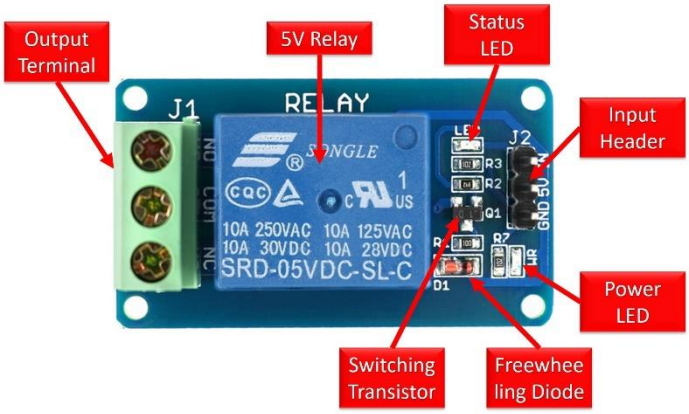
2.3 HC-06 Bluetooth module



The HC-06 Bluetooth module is a compact and cost-effective wireless communication module that enables Bluetooth connectivity for various electronic devices. It is based on the Bluetooth

2.0 specification and supports the Serial Port Profile (SPP), making it easy to establish a wireless serial connection between devices. The HC-06 module is often used for wireless data transmission, allowing microcontrollers, such as Arduino, to communicate with smartphones, tablets, or other Bluetooth-enabled devices. It is commonly used in projects like remote control systems, data logging, and wireless sensor networks, offering a simple and affordable way to add Bluetooth capabilities to your electronic projects.

2.4 Relay Module



A relay module is an electrical component used to control high-voltage devices with a low-voltage signal. It contains one or more relays that act as switches and is commonly used in applications like home automation and industrial control. Relay modules provide electrical isolation and enhance safety in electronics projects.

2.5 Gear Motors

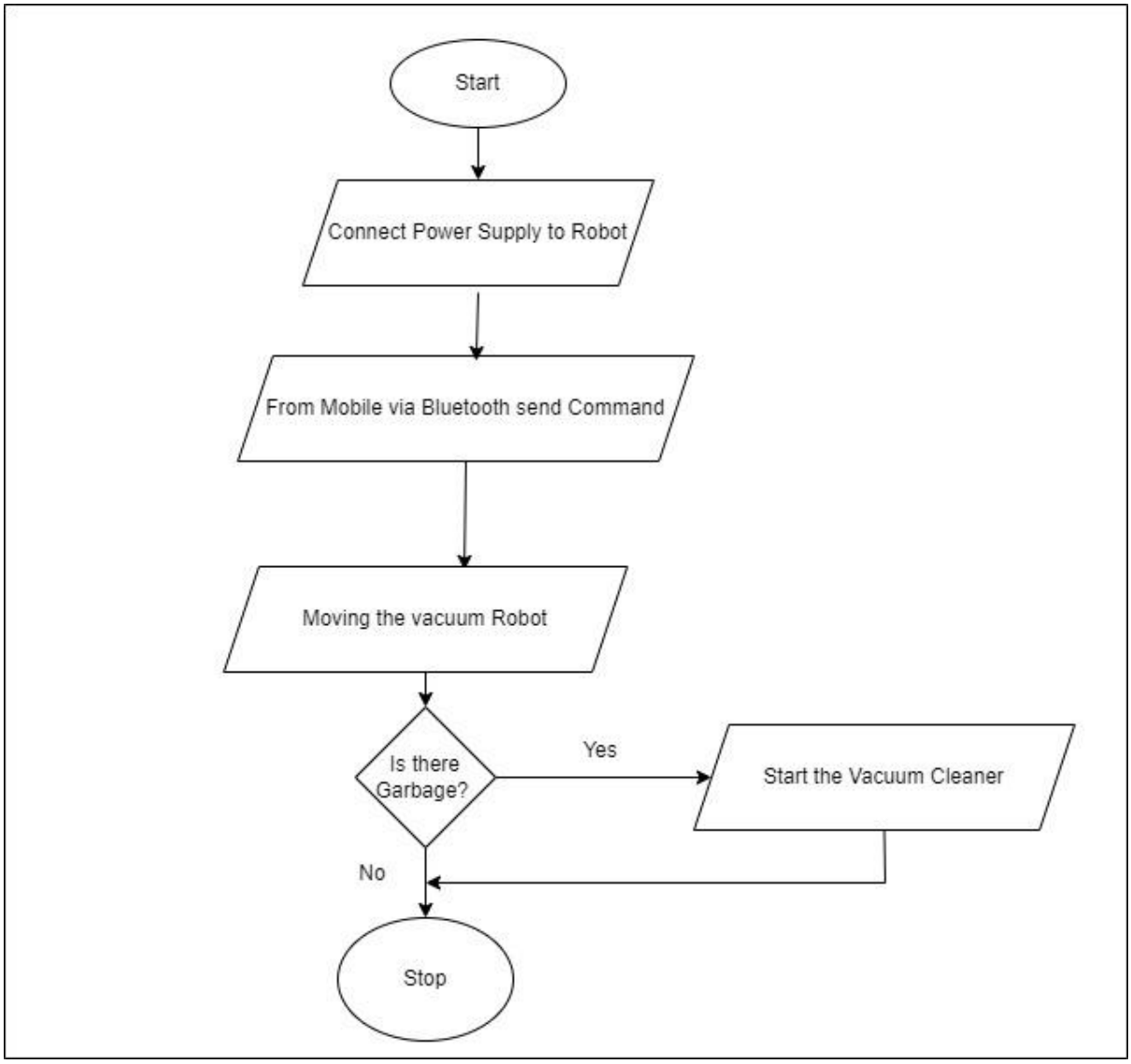
Gear motors are a type of electric motor equipped with an integrated gearbox. These motors combine the advantages of an electric motor's power with the gear reduction capabilities of a gearbox. The gearbox, which contains gears of various sizes, helps control the speed and torque of the motor's output shaft. Gear motors find extensive use in applications where precise control of speed and power is essential, such as in robotics, industrial machinery, and automotive

systems. They come in various sizes and gear ratios, making them versatile for a wide range of projects and devices.

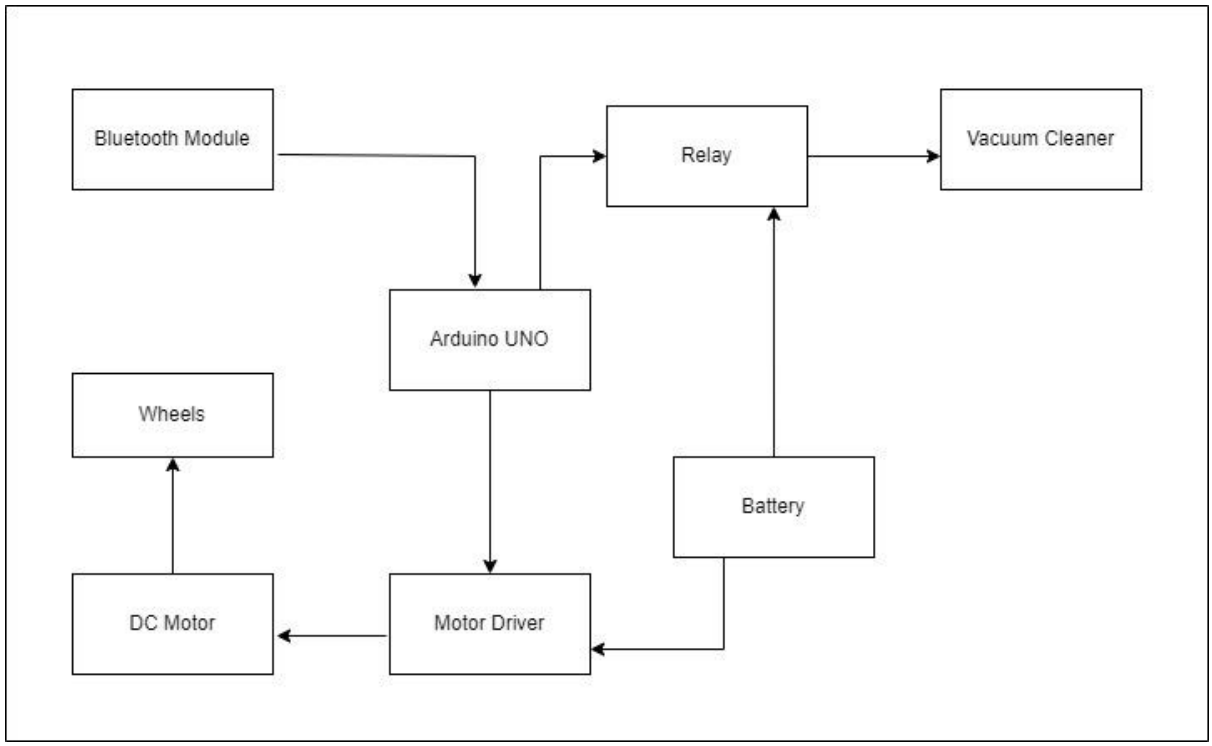
2.5 3.7 Battery

2.6 Wheels

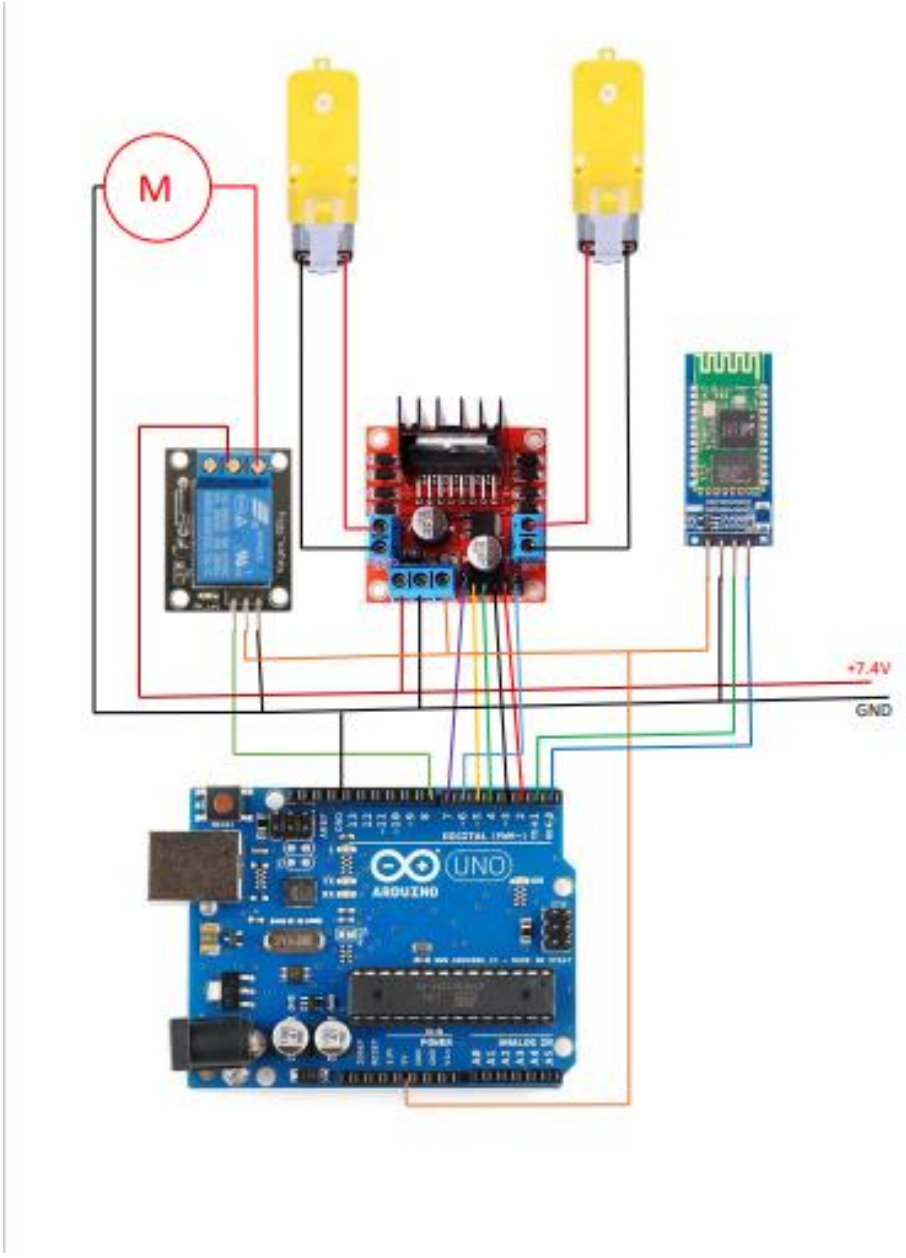
3.1 Flow Chart



3.2 Block Diagram



3.3 Circuit Diagram



3.4 Code

```
// Define motor control pins
#define in1 2
#define in2 3
#define in3 4
#define in4 5

void setup() {
  // Set motor control pins as OUTPUT
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // Set additional pins as OUTPUT
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);

  // Start serial communication
  Serial.begin(9600);

  // Set initial state for additional pins
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, HIGH);
}

// Variables for motor control
int x, y, x1, y1, m1, m2;

void loop() {
  // Check if there is data available on the serial port
  if (Serial.available()) {
    char c = Serial.read();  // Read the incoming character

    // Read an integer value from serial if the character is 'A'
    if (c == 'A') {
      x = Serial.parseInt();
    }

    // Switch statement to perform actions based on the received character
    switch (c) {
      case 'F':
        motor(x, x);
        break;
      case 'B':
```

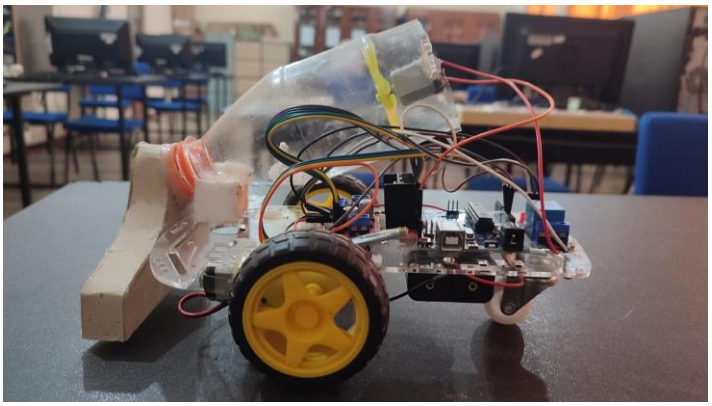
```
        motor(-x, -x);
        break;
      case 'L':
        motor(x, -x);
        break;
      case 'R':
        motor(-x, x);
        break;
      case 'G':
        motor(x, x/2);
        break;
      case 'I':
        motor(x/2, x);
        break;
      case 'H':
        motor(-x, -x/2);
        break;
      case 'J':
        motor(-x/2, -x);
        break;
      case 'S':
        motor(0, 0);
        break;
      case 'X':
        digitalWrite(8, LOW);
        break;
      case 'x':
        digitalWrite(8, HIGH);
        break;
    }
  }
}

// Function to control the motors based on input values
void motor(int i, int j) {
  // Ensure motor control values are within a valid range
  if (i > 255) {
    i = 255;
  }
  if (i < -255) {
    i = -255;
  }
  if (j > 255) {
    j = 255;
  }
  if (j < -255) {
    j = -255;
  }
```

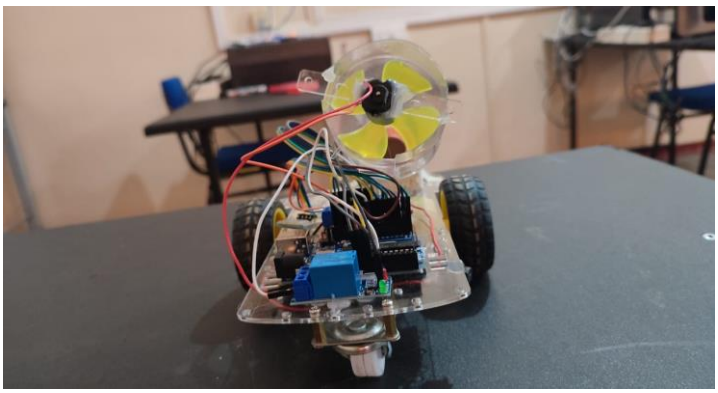
```
  // Control the motors based on input values
  if (i > 0) {
    analogWrite(in1, i);
    analogWrite(in2, 0);
  } else {
    analogWrite(in1, 0);
    analogWrite(in2, i * (-1));
  }

  if (j > 0) {
    analogWrite(in3, j);
    analogWrite(in4, 0);
  } else {
    analogWrite(in3, 0);
    analogWrite(in4, j * (-1));
  }
}
```

4. Budget

Component	Price (RS)
Arduino Uno board	1900
LM298 Motor drive	430
HC-06 Bluetooth module	1050
Relay Module	970
Gear Motors x 3	480
3.7 Battery x 2	1882
Wheels x 2	1100
Jumper Wires	100
SPST Rocket Switch	15
USB cable	200
Total	8,127

5. Structure of Prototype





6. Issues & Limitations

- **Limited Suction Power**
- **Small Dust Capacity**
- **Battery Life**
- **Lack of Advanced Sensors**

7. Conclusion

This smart vacuum cleaner robot facilitates effective floor cleaning with sweeping operations. It reduces the labor cost and saves time also and provides efficient cleaning. Bluetooth module and android application can be used to control the robot.

8. Future Implementations

In place of Bluetooth technology, GSM MODEM will be used.and obstacle detection will be added.

8. Reference

[1] Paulo Pinheiro, Eleri Cardozo, Jacques Wainer, and Eric Rohmer-"Cleaning Task Planning for an Autonomous Robot in Indoor Places with Multiples Rooms" published in2015

[2]T. Palleja, M. Tresanchez, M. Teixido, J. Palacin "Modeling floor-cleaning coverage performances of some domestic mobile robots in a reduced scenario" published in 2010

[3]Amit Sharma, Akash Choudhary, Akshay Gaur, and Amit Rajpurohit. "FULLY AUTOMATED HYBRIDHOME CLEANING ROBOT." Published in 2018

[4] Dr.J.Hameed Hussain,R.Sharavanan for "FLOOR CLEANING MACHINE BY REMOTE CONTROL" in 2017

[5]Akanksha Vyas, Satyam Chourasia, Shubham Antapurkar, and Raghvendra Prasad. "ARDUINO BASED DRY & WET AUTOMATIC FLOOR CLEANER." in2018