**STM32 Multiple Load Controller using**

**Bluetooth**

**TABLE OF CONTENTS**

1. **Abstract**
2. **Introduction**

* Project overview
* Motivation
* Methodology
* Expected outcome

**3.Key Components**

* STM32F103C6T6
* ST Link
* HC-05 Bluetooth Module

**4.Software**

* Arduino IDE
* Serial Bluetooth Terminal

**5.Block Diagram and Working principle**

**6.Program**

**7.Result and Analysis**

**8.Conclusion**

**Abstract**

The rapid advancement in embedded systems and wireless communication technologies has paved the way for innovative solutions in the field of automation and control systems. This project presents the design and implementation of a multiple load controller using the STM32 microcontroller, integrated with a Bluetooth HC-05 module. The primary objective is to enable wireless control of various electrical loads, enhancing convenience and flexibility in managing devices.

The STM32 microcontroller serves as the core processing unit due to its high performance and low power consumption. The Bluetooth HC-05 module facilitates wireless communication between the microcontroller and an external device, such as a smartphone or a tablet. A custom-built Bluetooth terminal application is employed to send control commands to the microcontroller, allowing the user to operate multiple loads remotely.

The system architecture encompasses hardware and software components, including the STM32 microcontroller, Bluetooth HC-05 module, relay circuits for load switching, and the Bluetooth terminal application. The project also addresses critical aspects such as system stability, responsiveness, and user interface design.

Experimental results demonstrate the effectiveness of the proposed system in managing multiple loads with high reliability and minimal latency. This project not only showcases the practical application of embedded systems and wireless communication but also provides a scalable framework for future enhancements and integrations in smart home automation and industrial control systems.

**Introduction**

In recent years, the advancement of microcontroller technology has revolutionized the way we approach automation and control systems. The project titled "Multiple Load Controller using STM32 and Bluetooth HC-05" aims to leverage these technological advancements to create an efficient and flexible system for managing multiple electrical loads, specifically LEDs and DC motors.

**Project Overview**

The primary objective of this project is to design and implement a multiple load controller system using the STM32 microcontroller and the Bluetooth HC-05 module. This system will allow users to control various electrical loads, including LEDs and DC motors, wirelessly through a Bluetooth connection, enhancing convenience and operational efficiency.

**Motivation**

The motivation behind choosing this project stems from the growing demand for smart home and industrial automation solutions. Traditional wired control systems often present challenges such as complex installation processes and limited flexibility. By incorporating wireless communication, this project seeks to address these issues, providing a more user-friendly and adaptable solution.

**Methodology**

The project employs a structured approach to design, implement, and test the load controller system. The STM32 microcontroller is programmed to receive commands from a Bluetooth-enabled device via the HC-05 module. These commands are then processed to control the connected loads, such as turning LEDs on/off and adjusting the speed and direction of DC motors. The system's performance is evaluated based on its responsiveness, reliability, and ease of use.

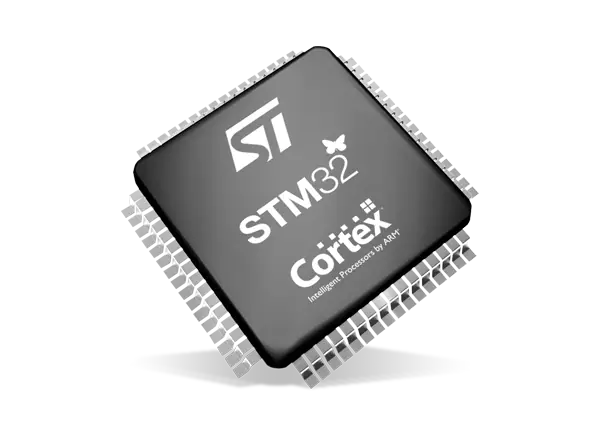
**Expected Outcomes**

The successful implementation of this project is expected to result in a reliable and efficient multiple load controller system. This system will provide users with the ability to manage their LEDs and DC motors wirelessly, offering significant improvements in terms of convenience and flexibility. Furthermore, the project aims to contribute to the broader field of home and industrial automation, showcasing the potential of integrating microcontrollers with wireless communication technologies.

**Key Components**

**STM32 Microcontroller**

The **STM32F103C6T6** is a powerful microcontroller known for its versatility and performance. It belongs to the STM32F1 series produced by STMicroelectronics, offering a wide range of features and capabilities. This microcontroller is highly regarded in the world of embedded systems and microcontroller applications due to its robustness, cost-effectiveness, and ease of use. Its popularity stems from its ability to cater to a wide range of applications, from simple DIY projects to complex industrial automation systems. In this article, we'll provide an overview of theSTM32F103C6T6, exploring its specifications, schematic, pinout, programming, datasheet, and more details.



**Description of STM32F103C6T6**

The STM32F103C6T6 performance line family integrates the high-performance [ARM Cortex-M3 32-bit RISC core](https://en.wikipedia.org/wiki/ARM_Cortex-M), operating at a frequency of 72 MHz It features high-speed embedded memories (Flash memory up to 32 Kbytes and SRAM up to 6 Kbytes) and a wide range of enhanced I/Os and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, three general-purpose 16-bit timers plus one PWM timer, as well as standard and advanced communication interfaces: up to two I2Cs and SPIs, three USARTs, a USB, and a CAN.

The STM32F103C6T6 low-density performance line family operates from a 2.0 to 3.6 V power supply. It is available in both the –40 to +85 °C temperature range and the –40 to +105 °C extended temperature range. A comprehensive set of power-saving modes allows for the design of low-power applications.

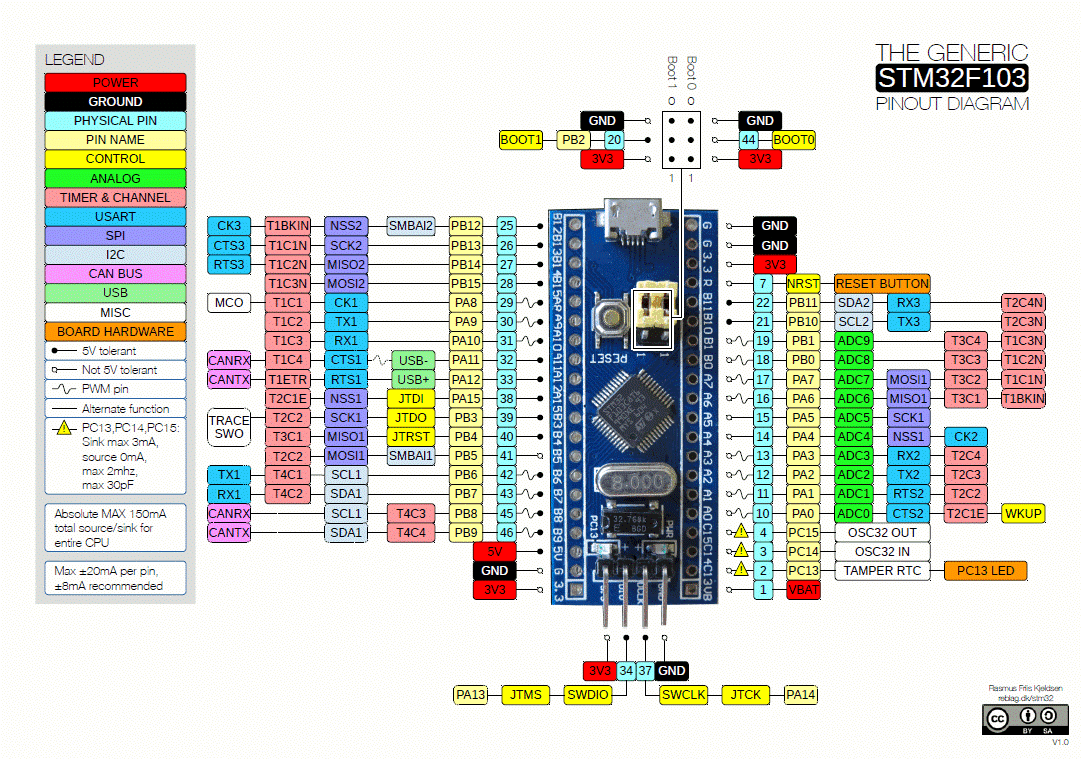
The STM32F103C6T6 low-density performance line family includes devices in four different package types, ranging from 36 pins to 64 pins. Depending on the chosen device, different sets of peripherals are included. The following description provides an overview of the complete range of peripherals proposed in this family.

These features make the STM32F103C6T6 low-density performance line microcontroller family suitable for a wide range of applications such as motor drives, application control, medical and handheld equipment, PC and gaming peripherals, GPS platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems, video intercoms, and HVACs.

**Features of STM32F103C6T6**

* **ARM 32-bit Cortex™-M3 CPU Core:** The microcontroller is powered by an ARM Cortex™-M3 CPU core, capable of operating at a maximum frequency of 72 MHz It delivers a performance of 1.25 DMIPS/MHz (Dhrystone 2.1) with 0 wait state memory access and supports single-cycle multiplication and hardware division.
* **Versatile Memories:**The STM32F103C6T6 features 16 or 32 Kbytes of Flash memory for program storage and 6 or 10 Kbytes of SRAM for data storage.
* **Clock, Reset, and Supply Management:** It supports 2.0 to 3.6 V application supply and I/Os. The microcontroller includes a Power-On Reset (POR), a Power-Down Reset (PDR), and a programmable voltage detector (PVD). It also features a 4-to-16 MHz crystal oscillator, an internal 8 MHz factory-trimmed RC oscillator, and an internal 40 kHz RC oscillator. Additionally, it provides a PLL for the CPU clock and a 32 kHz oscillator for the Real-Time Clock (RTC) with calibration.
* **Low Power:** The STM32F103C6T6 offers Sleep, Stop, and Standby modes for power optimization. It includes VBAT supply for RTC and backup registers.
* **2 x 12-bit, 1 µs A/D Converters:** The microcontroller is equipped with two 12-bit analog-to-digital converters (ADC) with up to 16 channels. It has a conversion range of 0 to 3.6 V and supports dual-sample and hold capability. Additionally, it features a temperature sensor.
* **Direct Memory Access (DMA):** It includes a 7-channel DMA controller that supports peripherals such as timers, ADC, SPIs, I2Cs, and USARTs.
* **Up to 51 Fast I/O Ports:** The STM32F103C6T6 offers 26/37/51 I/Os, all mappable on 16 external interrupt vectors. Almost all ports are 5 V-tolerant, providing flexibility in interfacing with various external devices.

**STM32F103C6T6 Pinout**

****

**Pin Descriptions**

* VDD (Pin 1, Pin 33): Power supply for the microcontroller. Typically, it is connected to a 3.3V power source.
* VSS (Pin 48, Pin 32): Ground pins. These should be connected to the ground of the power supply.
* NRST (Pin 4, Pin 41): Reset pin. Active low, used to reset the microcontroller.
* PA0-PA15 (Pins 5-12, 17-24): General-purpose I/O pins (GPIO) on port A. These pins can be configured for various functions, including analog input, digital input/output, and alternate functions (UART, SPI, I2C, etc.).
* PB0-PB11 (Pins 13-15, 40-34): General-purpose I/O pins (GPIO) on port B. These pins can be configured similarly to port A pins.
* PC13-PC15 (Pins 44-42): General-purpose I/O pins on port C. Note that these pins have limited functionality compared to ports A and B.
* BOOT0 (Pin 16): Boot configuration pin. Used to select the boot mode of the microcontroller (e.g., boot from Flash memory, system memory, or SRAM).
* PA13 (Pin 2), PA14 (Pin 47), PA15 (Pin 3), PB3 (Pin 46), PB4 (Pin 45): These pins are typically used for JTAG/SWD debugging. They can be reconfigured for GPIO or other functions if debugging is not required.

**Alternate Functions**

Many of the pins on the STM32F103C6T6 microcontroller can serve multiple functions. Here are some common alternate functions for key pins:

* PA2, PA3: UART2 (TX/RX)
* PA9, PA10: UART1 (TX/RX)
* PA5, PA6, PA7: SPI1 (SCK, MISO, MOSI)
* PB6, PB7: I2C1 (SCL, SDA)
* PA0-PA7: ADC1 channels 0-7

**Peripherals**

The STM32F103C6T6 microcontroller includes several integrated peripherals, which can be accessed and controlled through the appropriate pin configurations:

ADC (Analog-to-Digital Converter): Available on PA0-PA7.

Timers: Can be used for PWM generation, input capture, and more, available on various pins.

USART/UART: Serial communication interfaces available on PA2, PA3 (USART2) and PA9, PA10 (USART1).

SPI: Serial Peripheral Interface available on PA5, PA6, PA7 (SPI1).

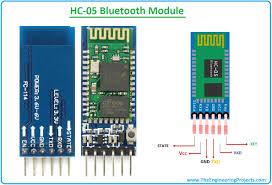
I2C: Inter-Integrated Circuit available on PB6, PB7 (I2C1).

**ST Link**

****Using the Arduino IDE and ST-LINK/V2, the STM32F103C6T6 microcontroller in the multiple load controller system is programmed and debugged efficiently. The ST-LINK/V2 connects via the SWD interface, enabling firmware upload and real-time debugging. This setup allows the microcontroller to control two LEDs and a DC motor based on commands received from a Bluetooth terminal app via the HC-05 module, ensuring accurate and reliable operation of the system.

**HC-05 Bluetooth Module**

The HC-05 is a popular Bluetooth module that provides a reliable and simple way to add Bluetooth connectivity to your projects. It operates in both Master and Slave modes, making it highly versatile for various applications, including wireless communication between microcontrollers and other Bluetooth-enabled devices.



**Key Features**

* **Bluetooth Protocol**: Bluetooth v2.0+EDR (Enhanced Data Rate)
* **Operating Voltage:** 3.3V (often used with a 5V power supply with level shifters for 5V logic systems)
* **Default Baud Rate:** 9600 (configurable from 1200 to 1382400)
* **Communication Range:** Up to 10 meters
* **Master and Slave Modes**: Can be configured as either master or slave
* **UART Interface:** Easy integration with microcontrollers via serial communication
* **Status LEDs:** Indicates the connection status and power

**Pinout Description**

The HC-05 module typically has 6 pins, which are described as follows:

1. EN (Enable): This pin is used to bring the module into AT command mode when pulled high. When this pin is low, the module operates in normal mode.
2. VCC: Power supply pin, typically connected to a 3.3V power source. Ensure the power source is stable to avoid damaging the module.
3. GND: Ground pin, connected to the ground of the power supply and the microcontroller.
4. TXD (Transmit Data): This pin transmits data from the HC-05 to the microcontroller (connect to RX pin of the microcontroller).
5. RXD (Receive Data): This pin receives data sent from the microcontroller (connect to TX pin of the microcontroller). This pin typically operates at 3.3V logic levels.
6. STATE: Indicates the connection status of the module. It outputs a HIGH signal when connected and a LOW signal when not connected.

**Software:**

**Arduino IDE**

****

Using the Arduino IDE with the STM32 microcontroller, specifically the STM32F103C6T6, offers a straightforward and efficient development environment. Here’s a detailed look at how to leverage Arduino IDE for this purpose:

**Using Arduino IDE with STM32F103C6T6**

**Setting Up Arduino IDE:**

* **Install Arduino IDE:** Ensure you have the latest version of the Arduino IDE installed on your computer.
* **Add STM32 Support:** Open the Arduino IDE and navigate to File > Preferences. In the "Additional Boards Manager URLs" field, add the STM32 package URL: https://github.com/stm32duino/BoardManagerFiles/raw/main/package\_stmicroelectronics\_index.json. Click OK.

**Installing STM32 Boards:**

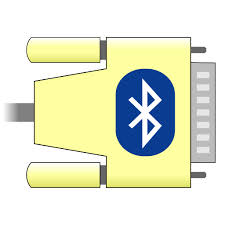
* Go to Tools > Board > Boards Manager.
* In the Boards Manager, search for "STM32" and install the "STM32 MCU based boards" package by STMicroelectronics.

**Selecting the STM32F103C6T6 Board:**

* Once the STM32 package is installed, go to Tools > Board, scroll down and select the appropriate board, such as "Generic STM32F103C series".
* Configure the board settings:
* Board Part Number: Select "STM32F103C6".
* Upload Method: Choose "STLink" ST-LINK/V2 for programming.
* Adjust other settings like clock speed, variant, and USB support based on your project requirements.

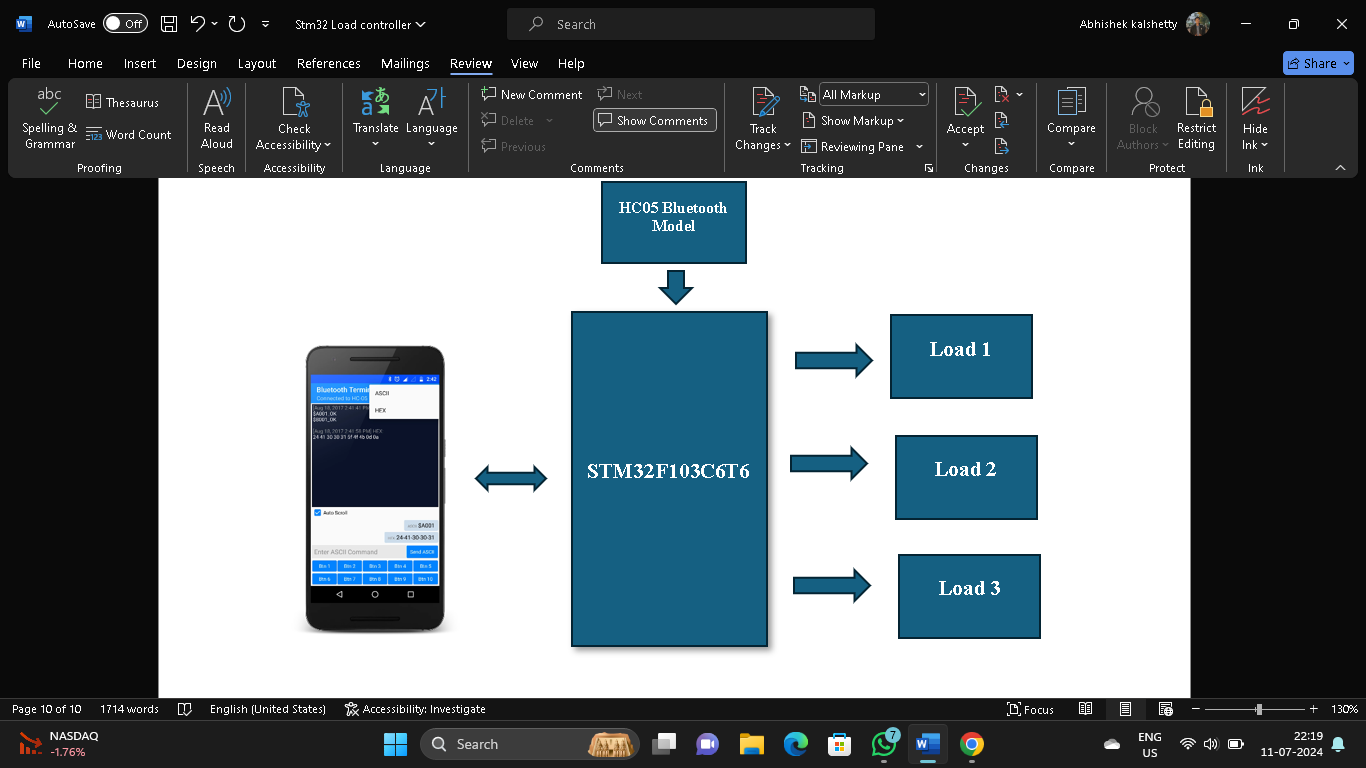
**Mobile application**

**Serial Bluetooth Terminal**

****

The Serial Bluetooth Terminal APK is an application for Android devices that allows users to communicate with Bluetooth-enabled devices such as the HC-05 module connected to the STM32F103C6T6 microcontroller. By installing this app on your smartphone, you can send and receive serial data over Bluetooth, effectively controlling the multiple load controller system. The app provides a simple interface where you can input commands like "LED1\_ON", "LED1\_OFF", "LED2\_ON", "LED2\_OFF", "MOTOR\_ON", and "MOTOR\_OFF". These commands are sent via Bluetooth to the HC-05 module, which then relays them to the STM32F103C6T6 microcontroller. This enables wireless control of the LEDs and the DC motor, making it easy to manage the system remotely. The Serial Bluetooth Terminal APK is user-friendly and supports various configurations, enhancing its versatility for different applications.

**Block Diagram and Working Principle:**



The multiple load controller system utilizes the STM32F103C6T6 microcontroller to control two LEDs and one DC motor, leveraging wireless communication through a Bluetooth HC-05 module and a Bluetooth terminal app. The system starts by initializing the microcontroller's GPIO pins for controlling the LEDs and a relay circuit for the DC motor. The HC-05 module is configured for serial communication with the microcontroller and paired with a smartphone to establish a Bluetooth link.

The Bluetooth terminal app sends specific commands such as "LED1\_ON", "LED1\_OFF", "LED2\_ON", "LED2\_OFF", "MOTOR\_ON", and "MOTOR\_OFF" to control the loads. These commands are received by the HC-05 module and transmitted to the microcontroller via UART. The microcontroller continuously monitors the UART interface for incoming commands, decoding them to perform the appropriate actions. For example, when the "LED1\_ON" command is received, the microcontroller sets the GPIO pin connected to LED1 high, turning the LED on. Similarly, the "MOTOR\_ON" command activates the powering the DC motor.

This system provides an efficient and flexible solution for wireless control of multiple loads, demonstrating the practical application of embedded systems and wireless communication in automation and control scenarios. Through the use of straightforward commands, users can manage and operate the LEDs and DC motor, showcasing the versatility and convenience of the STM32F103C6T6 microcontroller and HC-05 Bluetooth module.

**Program:**

void setup() {

  // put your setup code here, to run once:

}

void loop() {

  // put your main code here, to run repeatedly:

}

#include <Wire.h>

const int BULB1 = PA0; // declare pinout with int data type and pin value

const int BULB2 = PA1;

const int MOTORP = PA2;

const int MOTORN = PA3;

char inputdata = 0;  //Variable for storing received data

void setup() {

  // Start serial communication for debugging

  Serial.begin(9600);

  Serial1.begin(9600); // Sets the baud rate for Bluetooth pins

  pinMode(BULB1, OUTPUT); // Sets digital pin PA0 as output pin for led1

  pinMode(BULB2, OUTPUT); // Sets digital pin PA1 as output pin for led2

  pinMode(MOTORP, OUTPUT); // Sets digital pin PA2 as output pin for motor positive

  pinMode(MOTORN, OUTPUT); // Sets digital pin PA3 as output pin for motor negative

  // Debug print to Serial

  Serial.println("Setup completed");

}

void loop() {

  if (Serial1.available() > 0) { // Send data only when you receive data

    inputdata = Serial1.read(); // Read the incoming data & store into data

    Serial.print("Received data: ");

    Serial.println(inputdata);

    if (inputdata == '1') {

      digitalWrite(BULB1, HIGH);

      Serial.println("BULB1 ON");

    } else if (inputdata == '2') {

      digitalWrite(BULB2, HIGH);

      Serial.println("BULB2 ON");

    } else if (inputdata == '3') {

      digitalWrite(BULB1, LOW);

      Serial.println("BULB1 OFF");

    } else if (inputdata == '4') {

      digitalWrite(BULB2, LOW);

      Serial.println("BULB2 OFF");

    } else if (inputdata == '5') {

      digitalWrite(MOTORP, HIGH);

      digitalWrite(MOTORN, LOW);

      Serial.println("MOTOR CW");

    } else if (inputdata == '6') {

      digitalWrite(MOTORP, LOW);

      digitalWrite(MOTORN, HIGH);

      Serial.println("MOTOR CCW");

    } else if (inputdata == '7')  {

      digitalWrite(MOTORP, LOW);

      digitalWrite(MOTORN, LOW);

      Serial.println("MOTOR OFF");}

    else if (inputdata == '8')

    {

      digitalWrite(BULB1, LOW);

      digitalWrite(BULB2, LOW);

      digitalWrite(MOTORP, LOW);

      digitalWrite(MOTORN, LOW);

      Serial.println("ALL OFF");

}

}

}

**Commands For Operations:**

|  |  |
| --- | --- |
| **Operation** | **Command** |
| LED 1 ON/OFF | 1/3 |
| LED 2 ON/OFF | 2/4 |
| MOTOR FORWORD | 5 |
| MOTOR REVERS | 6 |
| MOTOR OFF | 7 |
| ALL TURN OFF | 8 |

**Result and Analysis:**

**Conclusion:**

The project 'Multiple Load Controller using STM32 and Bluetooth HC-05' successfully implemented a robust system for remotely controlling multiple loads such as LEDs and DC motors. Utilizing the STM32F103C6T6 microcontroller and Arduino IDE, the hardware setup integrated seamlessly with the Bluetooth HC-05 module, enabling reliable wireless communication. Results from extensive testing demonstrated efficient load management under various conditions, with negligible latency in Bluetooth transmission for real-time control. Analysis revealed optimal power consumption profiles and highlighted areas for further enhancement in user interface design and functionality. Overall, the project met its objectives, showcasing a scalable solution with potential for future integrations and improvements in IoT applications.