

---

**RESEARCH ARTICLE**

## Development of a DC Motor Speed Control System Based on PID

**Qianyao Xu**

*School of Mechanical and Electrical Engineering, China Jiliang University, Hangzhou, China*

**Corresponding Author:** Qianyao Xu, **E-mail:** [xuqianyao0828@163.com](mailto:xuqianyao0828@163.com)

---

**ABSTRACT**

In the field of motion control systems, motor speed control plays a pivotal role. Various control algorithms and methods have been developed for this purpose, and one of the earliest strategies to emerge was analog control. Over time, it has evolved into a typical control structure with convenient parameter tuning capable of meeting general control requirements. This design is based on digital control algorithms, with the ST89C52 microcontroller as the control core. It generates pulse-width modulation (PWM) pulses controlled by a digital algorithm to regulate the speed of direct circuit (DC) motors. Additionally, it utilizes a photoelectric sensor to convert motor speed into pulse frequency, which is fed back to the microcontroller, thereby achieving closed-loop speed control and eliminating static speed errors.

**KEYWORDS**

PWM pulses, duty cycle, zero steady-state error regulation

**ARTICLE INFORMATION**

**ACCEPTED:** 20 October 2023

**PUBLISHED:** 03 November 2023

**DOI:** 10.32996/jmcie.2023.4.4.4

---

### 1. Introduction

#### 1.1 Overall Design Concept and Solution Selection

Our research is focused on the development of a DC motor speed control system. We employ control techniques to modulate the duty cycle of the voltage, thereby achieving control over the DC motor's speed. A PWM signal generation system is used to drive the motor. The motor speed control system primarily comprises the power supply module, main control module, motor drive module, speed measurement module, Bluetooth control module, and LCD display, as illustrated in Figure 1 (Overall Design Concept) and Figure 2 (Functional Description).

### 2. Selection of Design Options for Each Module

#### 2.1 Selection of Power Supply Unit Design

The power supply unit is responsible for converting other forms of energy into electrical energy and supplying it to the circuit. It serves as the foundation of the hardware system. Through USB conversion, it provides 5V voltage to the main control module, Bluetooth module, and motor speed sensing module, as depicted in Figure 3.

#### 2.2 Choice of Main Control Unit Design

The minimum system for a microcontroller, also known as the minimal application system, refers to a system that can operate with the fewest components. For the 51 series microcontroller, the minimum system typically includes the microcontroller itself, a crystal oscillator circuit, and a reset circuit (Docin, 1999). Below is a schematic diagram of the minimum system for a 51 microcontroller, as shown in Figure 4.

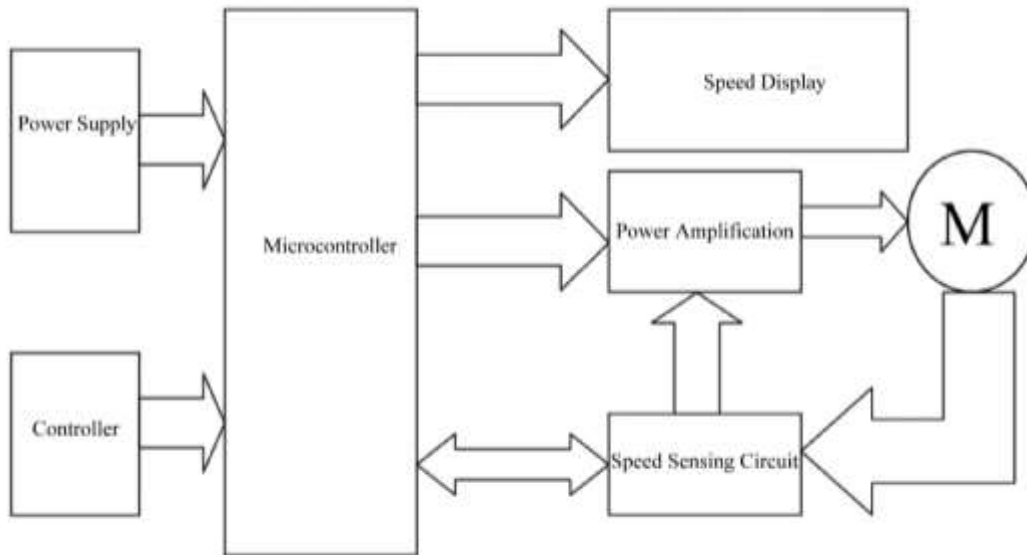


Figure 1. Overview of Our Method



Figure 2. Functional Description

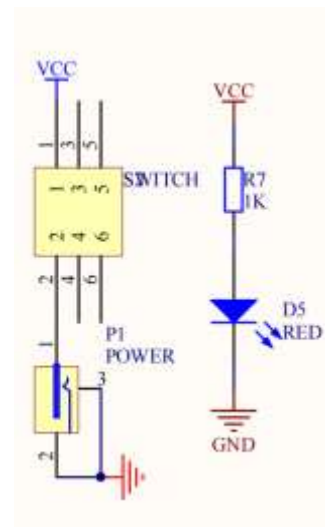


Figure 3. Power Supply Module

Detailed explanation is as follows:

(1) Reset Circuit:

This circuit is composed of a resistor and a capacitor connected in series. Based on the property that "the power-on voltage cannot change abruptly," it can be understood that when the system is powered on, the RST pin will go high, and the duration of this high level is determined by the RC values of the circuit (Docin, 1999). In a typical 51 microcontroller, if the high level on the RST pin lasts for more than two machine cycles, a reset will occur. Therefore, selecting appropriate values for RC ensures a reliable reset. The principle is to set the RC combination to produce a high level on the RST pin for at least two machine cycles.

(2) Crystal Oscillator Circuit:

A typical crystal oscillator frequency is 11.0592 MHz (as it provides accurate 9600 and 19200 baud rates for serial communication) or 12 MHz (for precise microsecond-level timing operations). In this circuit, a 12 MHz crystal oscillator is chosen.

(3) Microcontroller:

An STC89C52 microcontroller or another 52-series compatible microcontroller is needed. For pin 31 (EA/Vpp), when connected to a high level, the microcontroller starts executing from the internal ROM address 0000H after a reset. When connected to a low level, it starts executing from the external ROM address 0000H after a reset.

The STC89C52 microcontroller has a total of 40 pins. Two pins are for power (Vcc and GND), two for the crystal oscillator, one for reset, and one for EA/Vpp, leaving 34 pins. Pins 29 (PSEN) and 30 (ALE) have specific purposes when external data/program memory is used.

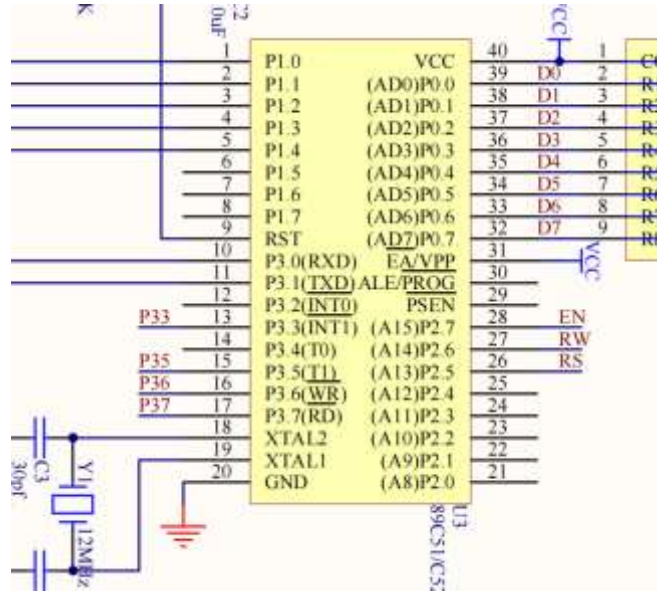


Figure 4. Main Control Unit Schematic

**2.3 Selection of Motor Speed Control Method**

After careful consideration, we have opted to employ PID (Proportional, Integral, and Differential) control to regulate the PWM duty cycle for motor speed control. Both practical experience and theoretical analysis have demonstrated that PID control is capable of meeting the control requirements of a wide range of industrial applications. It remains one of the most widely applied control algorithms to date. Additionally, the utilization of the incremental PID algorithm effectively reduces error accumulation, thereby achieving the goal of rapid and precise control.

$$u(n) = K_p e(n) + K_i \sum_{i=0}^n e(i) + K_d [e(n) - e(n - 1)] \quad \#(1)$$

**2.4 Selection of Remote Control Unit Design**

After comparing Wi-Fi and Bluetooth modules for communication, we found that Wi-Fi development posed higher challenges and costs. Consequently, we have opted for a Bluetooth module for data transmission. A Bluetooth module is a PCB board integrated with Bluetooth functionality, designed for short-range wireless communication. It connects to the microcontroller's serial port, enabling the microcontroller to communicate with other devices through a Bluetooth connection, as depicted in Figure 5.

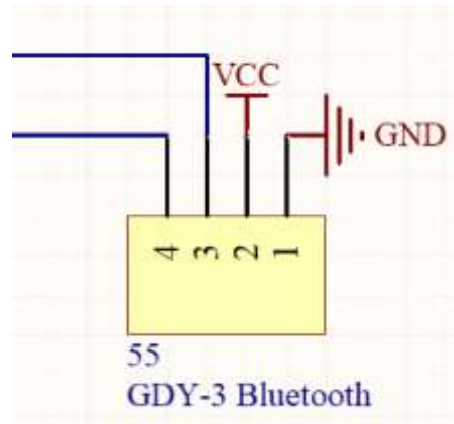


Figure 5. Bluetooth Module Schematic

### 3. Selection of Key Hardware Components

#### 3.1 Main Control Chip

For the main control chip, we have chosen the STC89C52. The 89C52 is a low-voltage, high-performance CMOS 8-bit microprocessor commonly referred to as a microcontroller (Yan, 2018). It features 4K bytes of flash programmable and erasable read-only memory (FPEROM), which can be erased and reprogrammed up to 100 times. This device is manufactured using ATMEL's high-density, non-volatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout (Docin, 1999; Yan, 2018). A physical representation of the STC89C52 is depicted in Figure 6.



Figure 6. Image of STC89C52

#### 3.2 The L298N Motor Driver

We chose the L298N motor driver. The L298N is a full-bridge driver chip with high voltage and high current capabilities. It operates at a high switching frequency. Each L298N module can independently control two DC motors and includes an enable control pin. Using this chip as the motor driver offers ease of operation, good stability, and excellent performance. When combined with a microcontroller, the L298N allows for precise control of the vehicle's speed. This speed control method provides excellent characteristics: smooth adjustment, a wide speed range, high overload capacity, and the ability to withstand frequent load shocks. It also enables frequent, smooth startups, braking, and reversal. A physical representation of the motor driver can be seen in Figure 7.



Figure 7. Image of Motor Drive Circuit

The L298N is a dual H-bridge motor driver chip, with each H-bridge capable of providing a current of 2A. The power supply voltage range for the power section is 2.5-48V, while the logic section operates on a 5V supply and accepts 5V TTL-level inputs. In general, the power supply voltage for the power section should be greater than 6V; otherwise, the chip may not function correctly.

The L298N can drive two motors, with OUT1 and OUT2 connected to one motor and OUT3 and OUT4 connected to the other. Pins 5, 7, 10, and 12 are used to input control levels for motor direction. ENA and ENB are used to control motor stopping. The STC89C52 microcontroller outputs two sets of PWM signals. Each set of PWM signals is used to control the speed of one motor, and two I/O pins can control the motor's direction. The control method and circuit are relatively simple: P10 and P11 control the direction of the first motor, and the input PWM1 controls the speed of the first motor. P12 and P13 control the direction of the second motor, and the input PWM2 controls the speed of the second motor. The motor driver circuit's schematic diagram is shown in Figure 8.

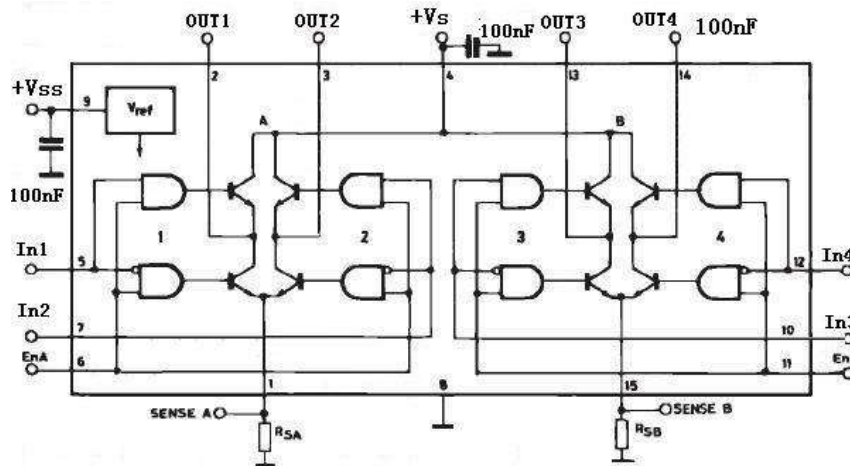


Figure 8. L298N Circuit Schematic

### 3.3 DC Motor

The structure of a DC motor consists of two main parts: the stator and the rotor. The stationary part of the DC motor, responsible for generating a magnetic field during operation, is called the stator. The stator typically consists of components like the motor housing, main magnetic poles, commutator segments, end caps, bearings, and the brush assembly. The rotating part of the motor is known as the rotor, and its primary role is to generate electromagnetic torque and induce electromotive force. It is considered the heart of the DC motor and is also referred to as the armature. The rotor comprises the rotor shaft, armature core, armature windings, commutator, and a cooling fan, among other components. The fundamental operating principle of a DC motor involves converting the alternating electromotive force induced in the armature coil into direct current electromotive force. This process is achieved through the action of a commutator in conjunction with the commutator brushes, which change the direction of current flow as the armature rotates. The direction of the force acting on the conductor is determined by the left-hand rule. This electromagnetic force forms a torque on the armature, referred to as electromagnetic torque, attempting to rotate the armature counterclockwise. If this electromagnetic torque can overcome the resistive torque on the armature, such as frictional resistance and other load torques, the armature will begin to rotate counterclockwise.

In our application, we use a 5-volt brushed DC motor. The primary structure of a brushed DC motor consists of the stator, rotor, and brushes. It generates rotational torque through a rotating magnetic field, which, in turn, produces kinetic energy. The brushes maintain contact with the commutator and play a role in electrical conduction and commutation as the motor rotates.

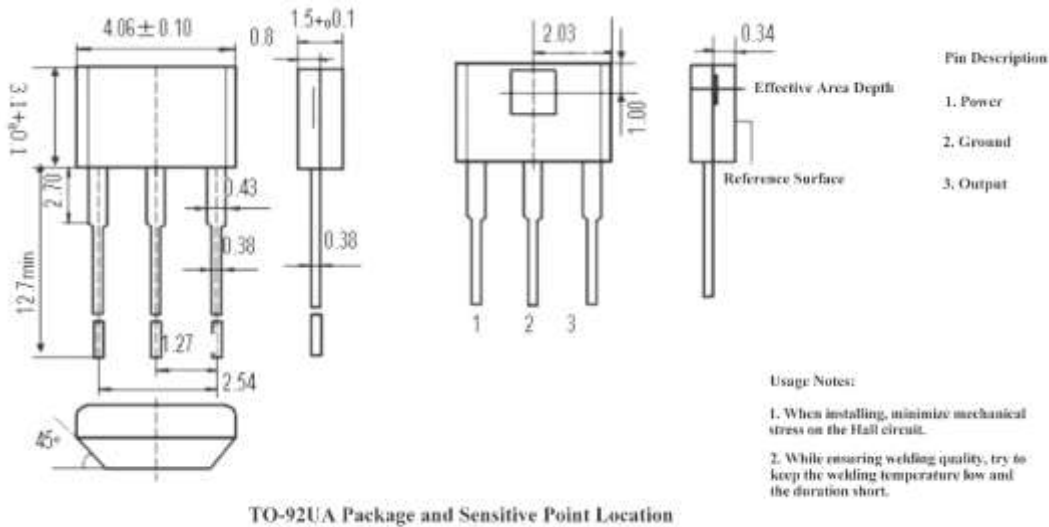
As the motor rotates, different coils or different poles of the same coil are energized at different times, ensuring that the N-S poles of the coil's magnetic field have the appropriate angular difference from the N-S poles of the nearest permanent magnet in the stator, with opposite magnetic polarities attracting and like polarities repelling, forces are generated, propelling the motor's rotation. The physical representation of the DC motor is shown in Figure 9.



Figure 9. Brushed DC Motor

### 3.4 Hall Sensor

The A3144E Hall element, also known as 44E or OH44E, is a Hall sensor or Hall switch integrated circuit that operates based on the Hall effect principle. It is constructed using semiconductor integration technology, creating a magnetically sensitive circuit. This sensor comprises a voltage regulator, Hall voltage generator, differential amplifier, Schmitt trigger, temperature compensation circuit, and open-collector output stage. Its input is the magnetic induction intensity, and its output is a digital voltage signal. The OH3172X Hall switch circuit is most suitable for responding to changes in a steep-slope magnetic field and for use in situations with weaker magnetic flux density. It consists of a reverse voltage protector, voltage regulator, Hall voltage generator, signal amplifier, Schmitt trigger, and collector/open-circuit output stage, as shown in Figure 10.



TO-92UA Package and Sensitive Point Location

Figure 10. Hall Sensor

### 3.5 JDY-31 Bluetooth Serial Module

For this project, we are using the JDY-31 Bluetooth serial module. The JDY-31 Bluetooth module is designed based on Bluetooth 3.0 SPP (Serial Port Profile) and supports data passthrough for Windows, Linux, and Android devices. It operates in the 2.4GHz frequency band, uses GFSK modulation, has a maximum transmission power of 8dB, and a maximum transmission distance of 30 meters. Users can easily configure device settings, such as the device name and baud rate, using AT commands, making it a flexible and user-friendly option.

The JDY-31 module is a slave module, and one of its major advantages is that it does not require AT command setup when programming on the MCU. This simplifies the operation, allowing us to send information directly over the serial interface for communication. Typically, we just need to connect to the module via the serial pins, and after successfully establishing communication with the Bluetooth module, it will send a "CONNECTED" message, as shown in Figure 11.

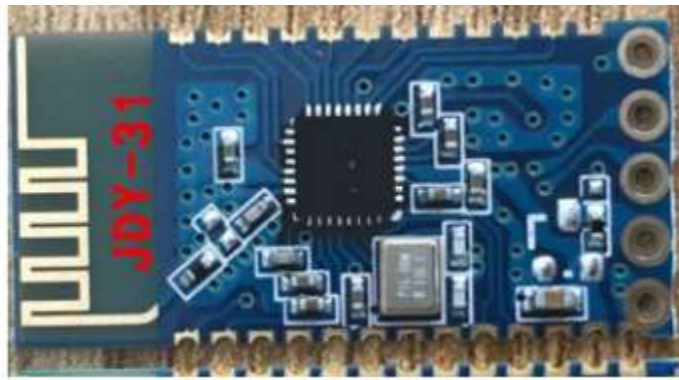


Figure 11. Image of JDY-31

### 3.6 LCD1602 Display Module

LCD (Liquid Crystal Display) that can simultaneously display 16×2 characters, totaling 32 characters. It is a dot-matrix LCD module designed specifically for displaying letters, numbers, symbols, and more.

The LCD1602 LCD display is a widely used character-based LCD display module. It consists of a character-based LCD screen, a control and drive main circuit (HD44780), its extended drive circuit (HD44100), and a few resistors, capacitors, and structural components assembled on a PCB board. The advantages of this display are its low power consumption, compact size, and low radiation.

The LCD1602 is primarily used for displaying numbers, letters, graphics, and a small number of custom characters. It can display two lines of 16 characters, and it has 16 pins, including an 8-bit data bus (D0-D7) and three control ports: RS, R/W, and EN. It operates at a voltage of 5V and includes a character contrast adjustment (V0) and a backlight source (AK), as shown in Figure 12.

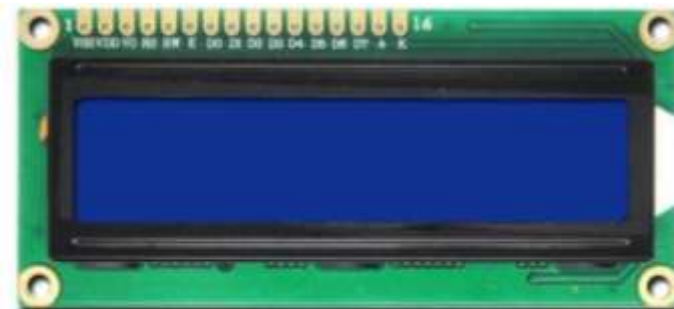


Figure 12. Physical Image of the LCD1602 Display Module

## 4. Software

### 4.1 Introduction to Keil

The main steps for software simulation development using Keil are as follows: writing the source code and saving it, creating a project, adding source files, configuring the project, compiling/assembling, linking to generate target files, and program debugging. Keil uses the concept of a "Project" to perform compilation, assembly, linking, and other operations on projects, not just individual source files. Creating a project, configuring it, and compiling/assembling it while generating target files is straightforward. Here are the steps: Write your source program in the source code editor and save it with the appropriate file extension (.asm for assembly or .c for C). Create a new project by selecting "File" -> "New Project" in Keil and save it (without a file extension, or you can use .uv2). After saving the project, a device selection dialog will appear. Choose the CPU and click "OK" to return to the main interface. In the Project Management window, you'll see "Target1" under the "Files" tab. Expand it by clicking the "+" sign. Then, select "Source Group1," right-click, and choose "Add File to Group 'Source Group1'." This will open a dialog to locate and add source files. After adding the files, click "Close" to return to the main interface. You'll see the files you added when you expand "Source Group1." Double-click a file name to open it. Configure the project by selecting "Target1" in the Project Management window. Then, choose "Project" -> "Option for Target 'Target1'" (or right-click and choose this option). This opens the project property settings dialog with eight tabs. You can set parameters such as the crystal frequency under the "Target" tab, configure your experimental simulation board under the "Debug" tab, and, if necessary, choose "Create Hex File" under the

"Output" tab. The settings in other tabs can generally be left at their default values. After configuring the project, press F7 (or click the corresponding icon on the compile toolbar) to compile, assemble, and link to generate target files.

Once you have successfully compiled, assembled, and linked the program, select "Debug" -> "Start/Stop Debug Session" (or press Ctrl+F5) to enter the program debugging state. Keil provides a simulation debugging feature that includes a powerful simulated CPU for program execution. You can debug your program in various ways, such as single-stepping (F11 or Debug -> Step), single-stepping through procedures (F10 or Debug -> Step Over), or full-speed execution. If you find errors in your program, you can use the online assembly feature to make changes without exiting the debugging environment. This eliminates the need to exit, modify the source program, recompile, and relink the project before entering the debugging state again.

#### 4.2 Motion Control Algorithm

The controller is one of the most widely used automatic controllers, known for its simple principles, broad applicability, and independent control parameters. In this project, we are using an incremental closed-loop speed control (as shown in Figure 13) to ensure rapid and stable adjustment of the motor.

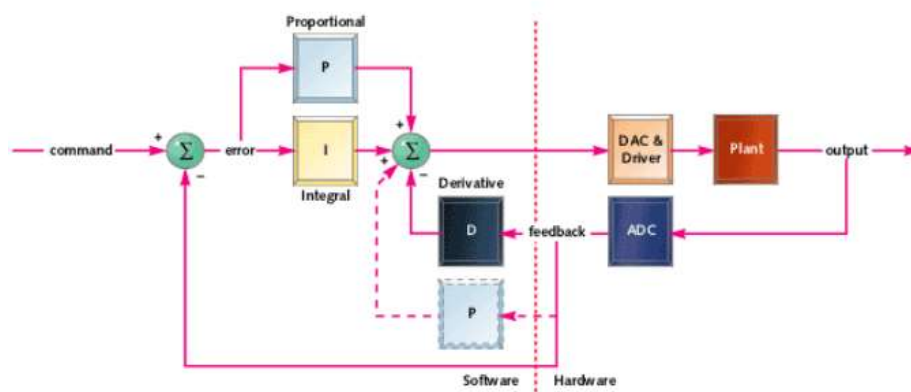


Figure 13. Incremental Closed-loop Speed Control

Here is core code:

```
unsigned char UKEY = 0;
#define kp 1 // Proportional coefficient
#define ki 0.4 // Integral coefficient
#define kd 0 // Derivative coefficient
float ee, eeh, p; // Error and accumulated error
float control(float uset, float feedback) // Equation
{
    float duk, D;
    eeh = ee;
    ee = uset - feedback;
    p += ee;
    D = ee - eeh;
    duk = kp * ee + ki * p - kd * D;
    return duk;
}
```

#### 4.3 Mobile Bluetooth App Interface

When we open the mobile app, it will appear as shown in Figures 5 and 6. Through the Bluetooth module, the app sends data to the motor. After a successful connection, it is accessible to enter the remote motor control mode. We can click on "accelerate" and "decelerate" to remotely control the motor's speed. The app also displays the actual motor speed on the LCD screen. We can use "forward" and "reverse" buttons to control the motor's direction.



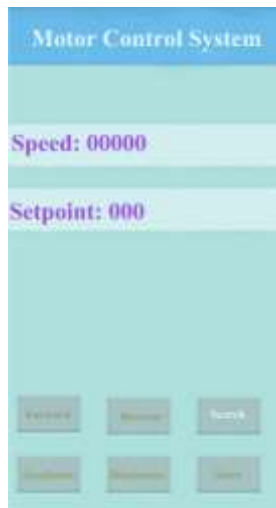


Figure 14. Bluetooth Serial Assistant App Interface

Here is core code:

```
void UART0_Send_Byte(unsigned char a)
{
    // ES = 0;          // Disable UART interrupt
    SBUF = a;
    UART0_TXD_BUSY = 1;
    while (UART0_TXD_BUSY); // Wait for successful transmission

    // ES = 0;          // Disable UART interrupt
    // TI = 0;          // Clear the transmit interrupt request flag
    // SBUF = a;        // Send the data
    // while (TI == 0); // Wait for the transmission to complete
    // TI = 0;          // Clear the transmit interrupt request flag
    // ES = 1;          // Enable UART interrupt
    // TH0 = 0;
    // TL0 = 0;
}

void UART0_Send_String(char *s)
{
    while (*s) // Check for the end of the string
    {
        UART0_Send_Byte(*s++); // Send the current character
    }
}

void Send_Data(void)
{
    static unsigned char Cnt = 0;
    if (++Cnt < 5)
    {
        return;
    }
}
```

## 5. Conclusion

After multiple rounds of debugging, we ultimately chose a PI (Proportional-Integral) control. Through experimentation, it was determined that the control worked best when KP (Proportional Coefficient) was set to 1, and KI (Integral Coefficient) was set to 0.4. The simulation results are shown in Figure 15, and the physical diagram is shown in Figure 16.

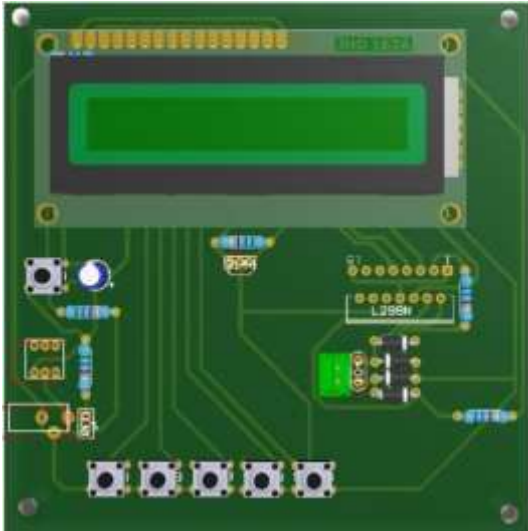


Figure 15. Simulation and Analog Diagram



Figure 16. Physical Image

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

## References

- [1] Docin. (1999). *Minimum system design of single chip microcomputer*. <https://www.docin.com/p-1999098623.html>
- [2] Yan, Y. N. (2018). *Design and implementation of music enjoyment digital music box based on single chip microcomputer control* [Paper presentation]. 2018 International Conference on Computational Science and Engineering (ICCSE 2018), Qingdao.