World Scientific Research Journal (WSRJ) ISSN: 2472-3703
www.wsr-j.org

Design and Implementation of Stepper Motor Control System

Xianwen Hu, Jingwen Liu, Daobing Liu

Three Gorges University, College of Electrical and New energy, Yichang, 443000, China

Abstract: In this paper, the stepper motor is controlled by STC89C52 single-chip microcomputer. The chip ULN2003 drives the stepper motor and supplies the winding with power-on sequence according to the signal input by the controller. The four-in-one common anode digital tube is used as a display element and can display the number of rotation speeds and forward/reverse rotations, and the use of buttons to adjust the speed and adjust the direction makes the system more user-friendly.

Keywords: Tepping motor, single-chip microcomputer, positive inversion control.

1. DESIGN OF STEPPER MOTOR CONTROL SYSTEM FRAME

In the design of the general system, we hope to meet the requirements of the system as much as possible in a simple way. Usually through a master chip to control other integrated circuits and various functional modules. There are many factors to consider in terms of hardware, such as hardware performance, efficiency, whether the hardware can meet the functions that the system is to implement, hardware costs, and ease of operation. The software is determined based on the choice of master control.

The system design is based on a single-chip microcomputer and mainly includes keyboard control module, digital tube display module, motor drive module, and a power supply module. The overall system block diagram of this design is shown in Figure 1.

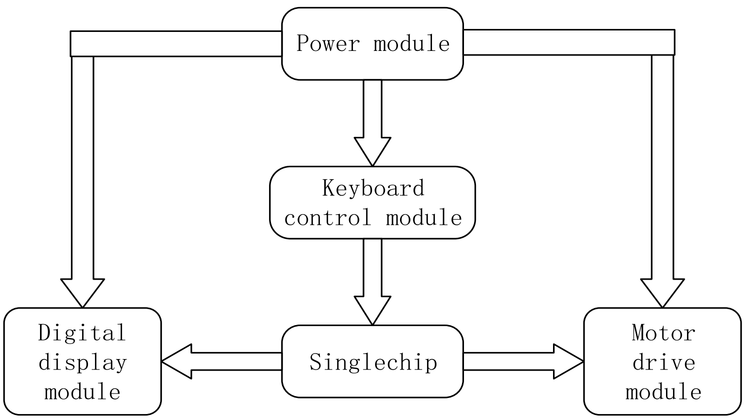

Figure 1. Design diagram of overall

2. ANALYSIS AND DESIGN OF SYSTEM SOFTWARE

2.1 Platform and language of software development

The Keil uvision series is a language software development system, it developed by German companies for 51 series single chip, and the language is the most common development of C language. Keil uvision4 is more flexible than the 2nd generation and 3rd generation software window management and has a high space utilization rate. The Keil Integrated Development Environment is a C language programming language specifically designed for microcontrollers. The Keil C integrated development environment is a development tool that integrates compilation and debugging of assembly language and C language editing. Keil C integrated development environment can be divided into two working interfaces, namely editing, compiling interface and debugging interface. At the same time, its integrated development environment can not only compile C language source programs and assembly language source programs unexpectedly, but also can simulate software debugging and hardware simulation debugging user programs to verify the correctness of user programs.

2.2 Main program

After the single chip is powered on, the timer is initialized and then the display subroutine is called. After power-on, character 0 is displayed in the first and third and fourth digits of the digital tube. Then determine if there is a key press. If not, return to initialization. If yes, make another judgment. After confirmation, call the function program.

```
void display()                    // Display function
{
wei3=1;wei2=1;wei1=1;wei0=0;led=tab[buf[3]];delay1m(1);led=0xff;
wei3=1;wei2=1;wei1=0;wei0=1;led = tab[buf[2]];delay1m(1);led=0xff;
wei3=1;wei2=0;wei1=1;wei0=1;led = tab[buf[1]];delay1m(1);led=0xff;
wei3=0;wei2=1;wei1=1;wei0=1;led = tab[buf[0]];delay1m(1);led=0xff;
}
```

Wei represents bit-choice, active-low, and drivers of the digital tube, respectively. Then the array is invoked and the time delay is displayed. Uchar buf [4]={0,10,0,0} is the caching of the digital tube display. Uchar code tab[11]= {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90,0xff}; Total digital signal is 0---9.

```
void delay1m(uint x)
{
uint i,j;
for (i=0;i<x;i++)               //Number x, about x
ms for(j=0;j<120;j++);            // Number 120, about 1 ms
}
```

This is a delay sub-function, used in the program to remove the impact of jitter, and also to display characters.

2.3 Simulation of System model

The stepper motor module implements a general model for two of the most popular stepper motor series: variable reluctance stepper motor and permanent magnet or hybrid stepper motor. The stepper motor model consists of electrical and mechanical parts. The electrical part is represented by an equivalent circuit and its configuration depends on the motor type. An equivalent circuit has been established, assuming that the magnetic circuit is linear (no saturation) and the mutual inductance between phases is negligible. The mechanical part is represented by a state space model based on the moment of inertia and viscous friction coefficient. As shown in FIG. 2, the operation model of the two-phase hybrid stepper motor is composed of a stepper driver, a pulse transmitter, and a stepper motor model.

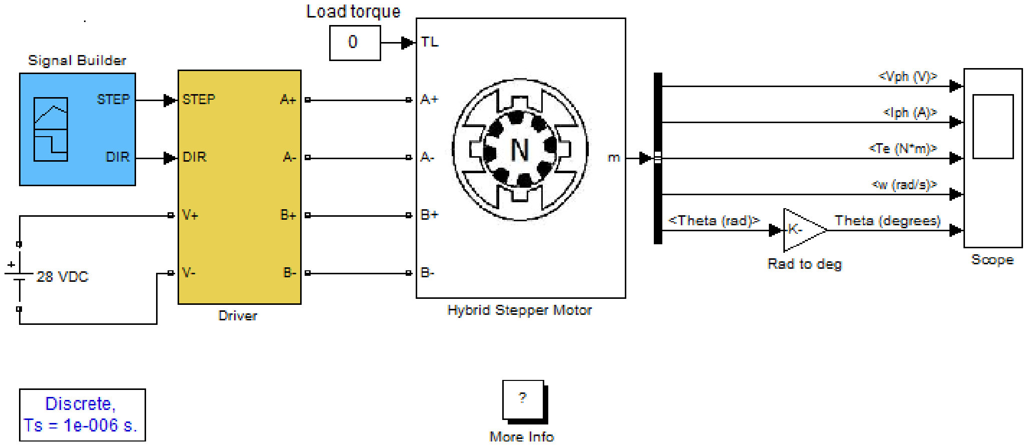

Figure 2. Simulation of motor model

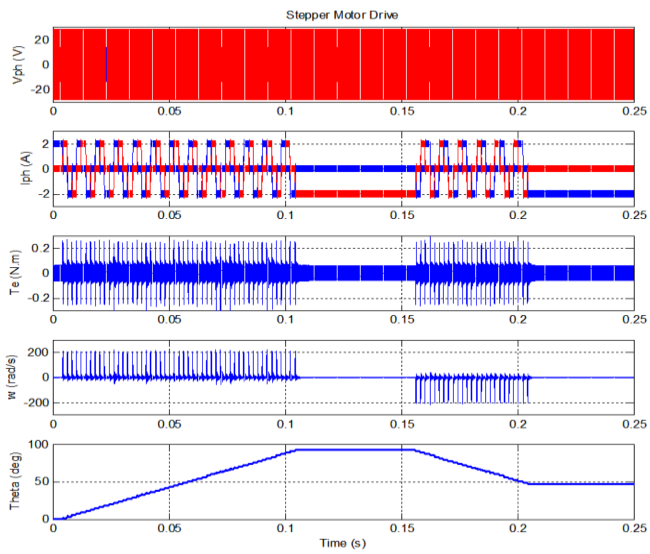

Figure 3. Simulation waveform of stepper motor system

The motor phase is fed by two H-bridge MOSFET PWM converters connected to a 28-V DC voltage source. The square wave current reference is generated using the current amplitude and stepping frequency parameters specified in the dialog window. Stepper drive movement is controlled by STEP and DIR signals received from external sources. The following waveform was obtained from a 0.25

second operation simulation driven by a stepper motor, in which the stepper was rotated within 0.1 second in the positive direction, stopped for 0.05 seconds, reversed for 0.05 seconds and stopped. The simulation results are shown in Figure 3.

3. ASSEMBLY AND DEBUGGING OF SYSTEM

3.1 Debugging and improvement of circuit assembly

After the component purchase is completed, perform the test before welding to ensure that the components are not damaged and can be used normally. When welding components, pay attention to the placement of components on the board. Reserve enough solder for each component. Note the positive and negative polarity of the components. Although it will not be burned, it must not work properly. When welding, it is necessary to avoid false soldering and short circuit soldering to avoid damage to components.

Circuit debugging is a check and adjustment of the hardware and software of the system. After the circuit is soldered, there is no guarantee that the soldering at each step is perfect. We need to debug the circuit to check the hardware to avoid short circuits or other soldering errors. After the hardware inspection, we also need to debug the software. After the program is burned, we can observe the power and whether it can complete the control of the stepper motor as we designed. If there are deviations, you need to modify the software. If you can perfectly realize the design of the system, the design will be successful.

There are two kinds of debugging methods, one is overall debugging, and the other is debugging each module. Integral debugging refers to the completion of hardware welding and the unified overall debugging after the software is written. This kind of debugging method will not have any missed welding, but since there are a lot of components used in the entire hardware part, once the overall debugging is out of order, it is difficult to quickly find out where the fault occurred. Debugging with the module as a unit can avoid this situation, but since every step requires debugging, there will naturally be some duplicate work.

There are several points to pay attention in this design and commissioning process. First, the working principle of the stepper motor must be determined. This is the basis of system design. The power supply must also be connected to a filter circuit to remove excess electromagnetic coupling and ensure stable operation of the circuit. Program compilation checks both syntax errors and logic errors. When there are no errors in the grammar, check the logic. It is through simulation that the stepper motor can work as expected.

3.2 Results of experimental

As shown in Figure 4, after the system is powered on, press the self-locking switch. First, the one or three or four digits of the digital tube display 0 characters, the power indicator light is on, and the motor speed is 0.

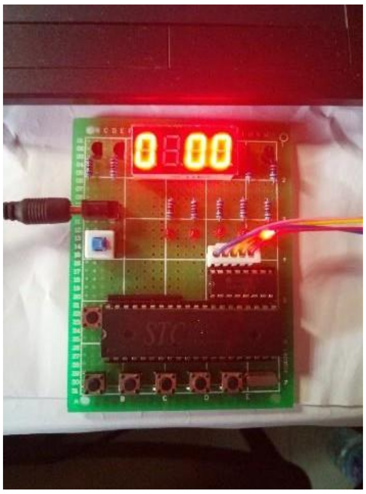

Figure 4. Initialization status after system power on

Then, when we press button 2, the first digit of the nixie tube will change to character 1 and the other two bits will be 0. Pressing button 1 will not change the system, and the motor rotation speed will still be 0 because the button parameter value has not changed.

Then press button 3, the fourth digit of the digital tube becomes 1 and the motor starts to rotate slowly. Press button 3 again, the fourth digit of the digital tube will change to 2, and the speed will increase accordingly. Each time you press button 3, the corresponding speed position will increase by 1 and the maximum position will be 10. If you accelerate again, you will go back to gear 1, then repeat the above steps. As shown in Figure 5 (a) and Figure 5(b).

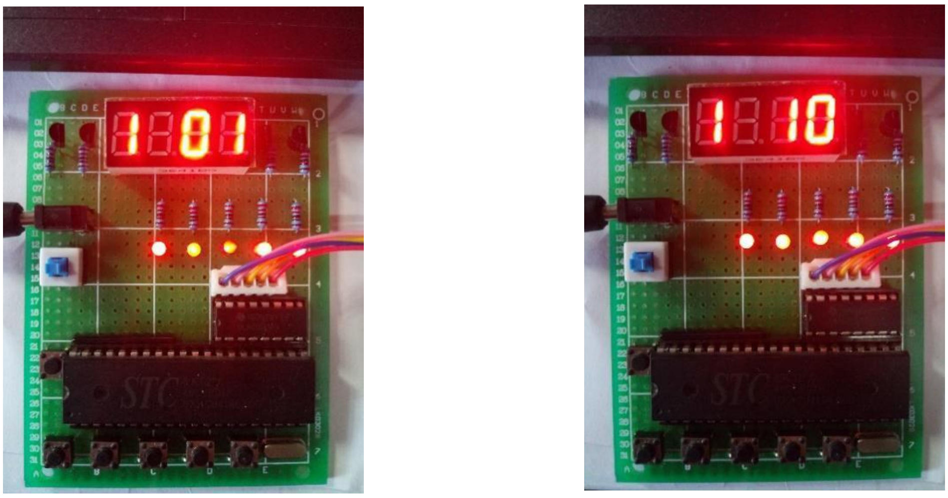

Figure 5. (a) The graph of the minimum speed (b) The graph of the maximum speed

The function of button 4 is the opposite of that of button 3, button 3 is acceleration, and button 4 is deceleration. If the current speed is 1st gear or the motor has no speed, pressing the deceleration key will directly jump to the maximum gear 10. If you press the deceleration button again, it will decrease by 1 step and change to 9. The key 5 is the pause/start key. If the current speed is 7th gear, press the key 5, and the digital tube 3 will display 0 and the motor will stop rotating. Press button 5 again, the motor will return to speed 7 and the motor will resume rotating. The fifth lamp is the power indicator, which lights up as long as the power is on. The other four lamps change the blinking state with the pulse frequency. When the motor speed is greater, the led lamp gradually goes to a stable level.

4. SUMMARY

This design takes the single-chip microcomputer as the core and controls the stepper motor through the pulse signal. By learning the function of each pin of the SCM and building the system structure, the ability of independent operation and thinking is improved. The research results prove that the research results obtained by using a single-chip microcomputer as a stepper motor controller are more accurate.

REFERENCES

[1] Yong Xu. Thermoelectric effects and topological insulators [J]. Chinese Physics B, 2016, 25 (11):58-66.

[2] Pedersen, S., Blanes-Vidal, V., Joergensen, H., Chwalibog, A., Haeussermann, A., Heetkamp,M. J. W., & Aarnink, A. J. A.. Carbon dioxide production in animal houses [J]. Agricultural Engineering International: CIGR Ejournal, 2008, 10 (3): 237-244.

[3] H.H. Mohammed, M.A. Grashorn, W. Bessei. The effects of lighting conditions on the behaviour of laying hens [J]. Arch.Geflügelk, 2014, 74 (3):197-202.