

Design and Implementation of an Interface Circuit for DC Motor Speed Control Using the PWM Technique

Firas Mohammed Ali Al-Raie 

Electrical Engineering Department, University of Technology/ Baghdad.
Email: firas@ieee.org

Received on: 7/6/2012 & Accepted on: 8/11/2012

ABSTRACT

In this paper an electronic interface circuit for controlling the speed of a small permanent magnet DC motor with the aid of a personal computer (PC) has been designed and implemented. The basic operation of the proposed circuit is based on the pulse width modulation (PWM) using a pulse generator with variable pulse width to drive the motor circuit. The ON and OFF time periods of the PWM signal are changed by means of mechanical relays connected to the pulse generator circuit. These relays are connected at the same time to the parallel port of the PC via a buffer and a driver. A computer program written in C++ programming language was used to generate the proper digital sequence needed to drive the control circuit in three different speed levels. This circuit was thereafter implemented and tested successfully.

Keywords: PC Interfacing, DC Motor Speed Control, C++.

تصميم وتنفيذ دائرة ترابط بيني للسيطرة على سرعة محرك تيار مستمر باستخدام تقنية تعديل عرض النبضة

الخلاصة

في هذا البحث تم تصميم وتنفيذ دائرة ترابط بيني الكترونية للتحكم بسرعة محرك تيار مستمر صغير بمغناطيس دائمى بواسطة الحاسوب. يعتمد مبدأ العمل الأساسي للدائرة المقترحة على تقنية تعديل عرض النبضة وذلك باستخدام مولد نبضات مع تغيير عرض النبضات المجهزة لتشغيل المحرك. وقد تم التحكم بزمان التوصيل والفصل لنبضات المذبذب عن طريق مرحلات مبربوطة إلى دائرة مولد النبضات. وتم ربط المرحلات إلى المنفذ المتوازي للحاسوب بواسطة دائرتي عازل ومشغل تيار. كما تمت كتابة برنامج بلغة C++ لتوليد التتابع الرقمي الملائم لتشغيل دائرة التحكم وبثلاث سرعات. وتم تنفيذ الدائرة العملية واختبارها بنجاح مختبرياً.

INTRODUCTION

It is widely needed to control the speed of a DC motor automatically by the PC. In order to implement the interfacing process between a DC motor and the PC, it is required to design an electronic circuit to connect the motor into one of the PC's ports. This circuit is utilized to direct the flow of signals outgoing from or incoming to the PC's port.

Besides, a computer program should be written to control the motor speed automatically. The most widely used PC ports for interfacing is the parallel port (LPT port), communication ports (COM Ports), and the USB ports.

There are, in general, three techniques used for DC motor speed control. The choice of the specific method depends on the required application. These methods are the field current control method, input voltage control method, and the pulse width modulation method.

FIELD CURRENT CONTROL METHOD

It is well-known that the rotational speed of DC motors is inversely proportional to the magnetic field around their poles. By adding a variable resistance in series with the field winding, the field current and thereby the magnetic flux of poles can be changed. Accordingly, the motor speed can also be varied. However, this method is used in separately excited DC motors and can not be used with permanent magnet motors.

INPUT VOLTAGE CONTROL METHOD

The speed of DC motors is dependent on the supply input voltage. When changing the applied voltage, the motor speed can thereby be varied. This method was used to control the speed of a permanent magnet DC motor in a previous work [1].

PULSE WIDTH MODULATION METHOD

This method is the most efficient one for varying DC motor speed [2]. According to this method, the average value of the applied voltage is changed by applying a pulsed waveform to the motor with variable pulse width as shown in Figure 1. This technique is known as *Pulse Width Modulation* (PWM). As shown from Figure 1, the average value of the applied voltage increases with the increase in pulse width. Based on this fact, the motor speed can be changed by varying the ratio between the pulse width t_{on} and period time T_{period} of the applied signal. The average value of the applied voltage is given by:

$$V_{av} = \frac{t_{on}}{T_{period}} V_{max} \quad \dots (1)$$

where V_{max} is the amplitude of the applied pulse signal.

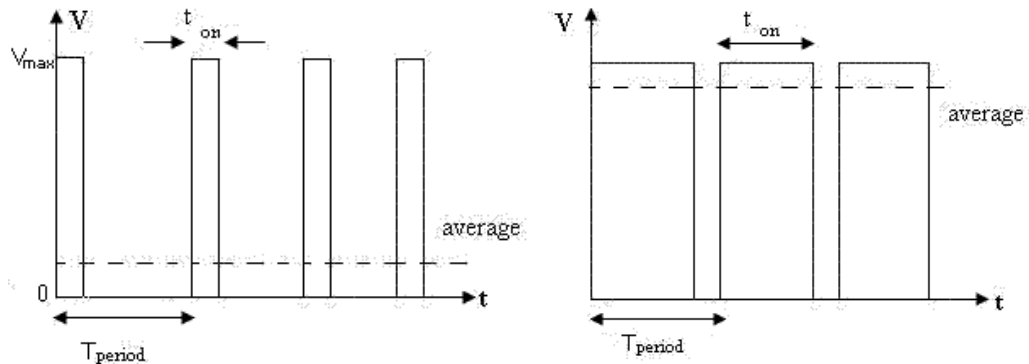


Figure (1) The Pulse Width Modulation Technique.

The pulse width modulation technique has been adopted throughout the design of the motor interface circuit in this paper.

PULSE WIDTH MODULATION CIRCUIT

In order to generate a pulse signal with variable pulse width, it is required to use a pulse generator circuit. The most widely used integrated circuit for such purpose is the 555 timer. This timer can be connected to operate as an oscillator (astable multivibrator) as shown in Figure 2. This circuit is very flexible, simple, cost effective, and can provide signals with frequency up to 1 MHz [3].

The ON time of the generated pulse waveform, t_{ON} , is dependent on resistor R_1 , and is given by [3]:

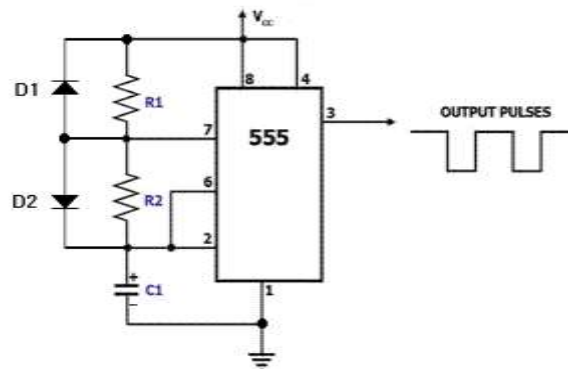
$$t_{ON} = 0.69C_1R_1 \quad \dots (2)$$

During the ON period, diode D_2 is forward biased and therefore resistor R_2 is shorted.

On the other hand, the OFF time, t_{OFF} , is dependent solely on resistor R_2 , and is given by [3]:

$$t_{OFF} = 0.69C_1R_2 \quad \dots (3)$$

During the OFF period, diode D_1 is forward biased and therefore resistor R_1 is shorted.



Figure(2) A Pulse Generator Circuit Using the 555 Timer.

The period of the generated pulse waveform is thus:

$$T = t_{ON} + t_{OFF} \quad \dots (4)$$

Substituting (2) and (3) in equation (4) yields:

$$T = 0.69C_1(R_1 + R_2) \quad \dots (5)$$

In order to change the pulse width of the output signal of the timer, resistor R_1 can be changed by connecting it as a potentiometer.

DC MOTOR DRIVING CIRCUIT

DC motors can either be driven by relays or transistors [4]. Figure 3 depicts a typical DC motor driving circuit using power transistor. This circuit can be connected to the output of the pulse generator to provide the motor with the proper required DC current.

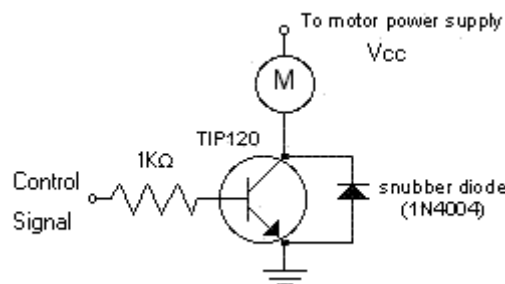


Figure (3) Typical DC Motor Driver Using Power Transistor.

DESIGN OF THE CONTROL CIRCUIT

The basic idea behind the control circuit involves using the 555 timer to work as an astable multivibrator so that it provides a pulse signal with variable pulse width. The pulse width of the generated signal can be controlled by changing the resistor

values on which the charging time and the discharge time depend. The generated signal is then provided to a driving transistor connected to the motor to increase the signal's current level. The values of charging and discharge resistors of the timer are controlled by mechanical relays connected to the data lines of the PC's parallel port via a buffer and a driver IC. Figure 4 presents the block diagram of the proposed control circuit.

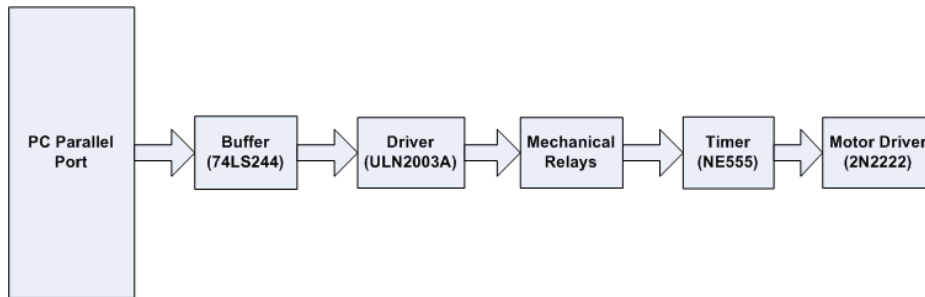


Figure (4)A Block Diagram of the Control Circuit.

Figure (5) shows the schematic diagram of the circuit which was developed from the block diagram of Figure (4).

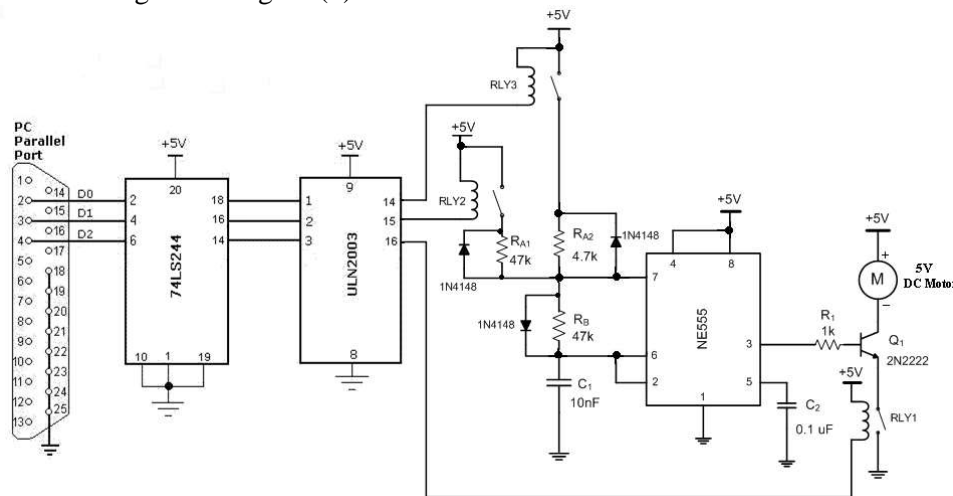


Figure (5) The Schematic Diagram of the Control Circuit.

This circuit is used to control the speed of a 5V permanent magnet DC motor in three levels via the PC's parallel port. The circuit operation is based on the pulse width modulation (PWM) technique by changing the average value of the voltage applied to the DC motor. As shown from the schematic diagram of the circuit, the DC motor is connected to the PC parallel port (LPT1) via an interface circuit consisting of an IC buffer (74LS244), driver IC (ULN2003A), three control relays (RLY1, RLY2, RLY3), 555 astable multivibrator circuit, and a driving transistor (2N2222). The 555 timer operates as a pulse generator providing the driving

transistor with a pulse signal of variable pulse width. The ON time (pulse width) of the generated waveform can be varied by changing the charging resistor of the timer circuit. This has been done through inserting or splitting the required resistor by means of mechanical relays. The operation of the relays, in turn, is controlled by the computer program which is written in the programming language C++. First, when RLY1 is ON and both RLY2 and RLY3 are OFF, the output of the timer is set to 'logic 1', thereby driving the motor with its maximum speed. When RLY2 becomes ON together with RLY1, the 555 timer generates a pulse signal with 50% duty-cycle which will reduce the average applied voltage of the motor and its speed. In this case, the charging resistor is R_{A1} and the discharge resistor is R_B which equals R_{A1} in this case. In the third case, when RLY3 is turned ON together with RLY1, the charging resistor becomes R_{A2} (where $R_{A2}=0.1 R_B$) reducing the ON time of the pulse signal and consequently the speed of the motor to the lower limit. The switching diodes (1N4148) are used to ensure that R_B is shorted during the charging period, and both R_{A1} and R_{A2} are shorted throughout the discharging period of the timer circuit. This is necessary to establish the ON and OFF times properly. Table-1 summarizes the ON/OFF operation conditions of the relays and the corresponding speeds of the DC motor.

Table (1) Relay Operating States and the Generated PC Sequences.

RLY3	RLY2	RLY1	Equivalent Digital Sequence	Motor Speed
OFF	OFF	OFF	000	Stop
OFF	OFF	ON	001	Maximum
OFF	ON	ON	011	Medium
ON	OFF	ON	101	Minimum

THE CONTROL PROGRAM

The computer program was written in C++ language. The program prompts the user to select a certain speed amongst three levels. The user's selection is then stored at an integer variable choice and the proper digital sequence is generated and stored at another integer variable data based on Table (1) above. The value of the integer variable data is placed at the PC's parallel port (LPT1) using the function outportb. LPT1 has been identified logically from its address (0378H) [5]. As soon as the motor starts running, it can be stopped by pressing any key from the keyboard by means of the function kbhit. Figure (6) presents the flow chart of the control program, while the program code is provided in the appendix.

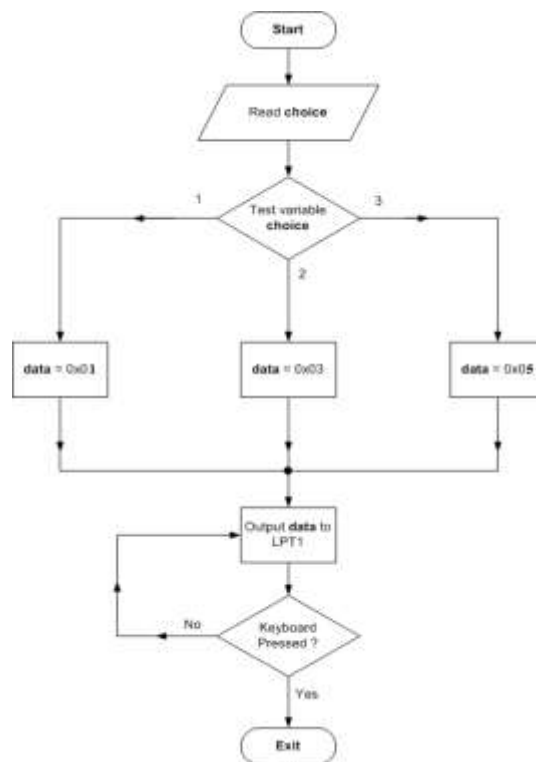


Figure (6) A Simplified Flow Chart of the Control Program.

DESIGN OF THE POWER SUPPLY

The power supply circuit should be capable of providing a sufficient current to the control circuit. The current drawn from each relay can be calculated as:

$$I_{RLY} = \frac{V_{DC}}{R_{RLY}} \quad \dots (6)$$

Where V_{DC} is the operating DC voltage of the relay, and R_{RLY} is the resistance of the relay coil. R_{RLY} was measured by an ohmmeter to be 56Ω , while $V_{DC} = 5V$. In this case $I_{RLY} = 90 \text{ mA}$.

It is noticed from Table-1 that in motor states 3 and 4, there are two relays operating at the same time. So, the maximum current consumed by the relays will be $2 I_{RLY}$. The current drawn by the motor, I_M , was measured to be 35 mA . Based on these facts, the total current consumed by the entire control circuit is:

$$\begin{aligned}
 I_T &= 2I_{RLY} + I_M + I_{timer} + I_{Buffer} + I_{Driver} \quad \dots (7) \\
 &= 2 \times 90 + 35 + 10 + 10 + 15 \\
 &= 250 \text{ mA}
 \end{aligned}$$

The power supply should thus provide a 5V regulated DC voltage with an output current more than 0.25 A. The designed power supply circuit was constructed from a 0.5A, 9V step-down transformer, bridge rectifier, capacitive filter, and an LM7805 voltage regulator. This regulator module is capable to provide a maximum load current of 1A approximately. Figure 7 shows the power supply circuit.

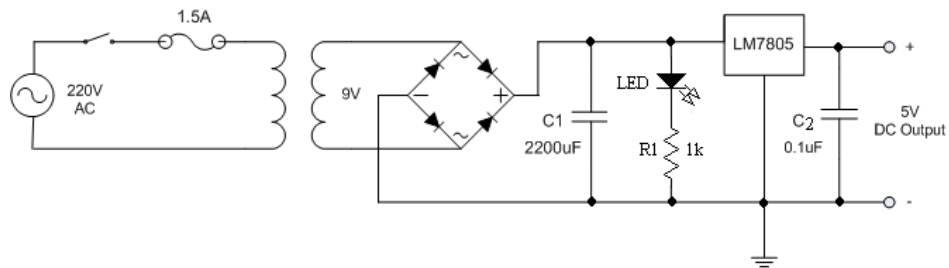


Figure (7) The Power Supply Used in the Designed Control Circuit.

PRACTICAL CIRCUIT CONSTRUCTION AND TESTING

Figure (8) depicts the constructed control circuit after assembly. As shown from this picture, both the power supply circuit and the electronic control circuit were assembled on the same Vero-board. The LEDs appearing at the data lines inputs of the circuit and in the power supply are used for indicating purposes. The circuit was connected to the PC's parallel port using a suitable parallel cable. The program was thereafter run successfully to verify the three speeds of the DC motor.

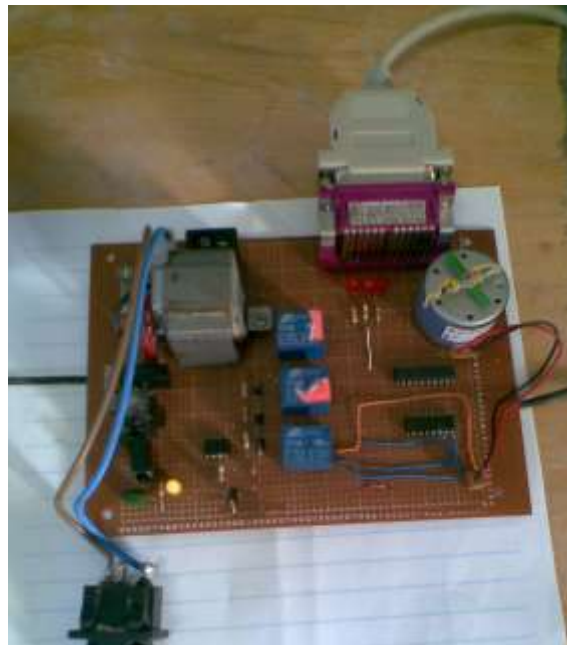


Figure (8) A Picture Showing the Constructed DC Motor Control Circuit.

CONCLUSIONS AND RECOMMENDATIONS

A control circuit has been designed and tested to change the speed of a DC motor in three states automatically using a computer program written in C++. The proposed circuit technique uses the pulse width modulation method which is very efficient for DC motor speed control. The circuit features are simplicity, low cost, easy to implement, flexibility for future development, and efficient operation. The mechanical relays in this circuit can be replaced with solid state relays or opto-couplers to reduce the noise generated by the circuit as well as to decrease the consumed current. The control program can be modified by re-writing it using the Visual C++ language which is compatible with the operating system Windows XP. The circuit can also be developed to drive a higher current DC motor by replacing the transistor with a power transistor and increasing the output current of the power supply using large capacity transformer and voltage regulator.

REFERENCES

- [1] Firas M. Ali Al-Raie, "Interface Circuit Allows Users to Control DC Motor's Speed", *Electronic Design*, Vol. 57, Issue 20, October 2009, p. 58.
- [2] J. Karthic, "PC-Based DC Motor Speed Controller", *Electronics for You Magazine*, Vol. 36, No.6, June 2004, pp. 67-68.
- [3] Bogart, T. F. *Electronic Devices and Circuits*, 4th Edition, Prentice-Hall, 1997.
- [4] Pei An, *PC Interfacing Using Centronic, RS232 and Game Ports*, Newnes Publishing Company, 1998.
- [5] Peter H. Anderson, "Use of a PC Printer Port for Control and Data Acquisition", *The Technology Interface Journal*, Vol.1, No.1, 1996, Online Journal, <http://technologyinterface.nmsu.edu>.

APPENDIX (The C++ Control Program Code)

```
#include<iostream.h>
#include<conio.h>
void main( ){
    int address = 0x0378,choice,data;
    cout<<"\n DC MOTOR SPEED CONTROL USING THE
    IBM PC"<<endl;
    cout<<"\nPress and key to continue"<<endl;
    getch();
    start:clrscr();
    cout<<"\nSelect one of the following
    choices:"<<endl;
    cout<<"1) Maximum Speed"<<endl;
    cout<<"2) Medium Speed"<<endl;
    cout<<"3) Minimum Speed"<<endl;
    cout<<"4) Exit"<<endl;
    cin>>choice;
    switch(choice)
```

```
{
  case 1:data=0x01;
  break;
  case 2:data=0x03;
  break;
  case 3:data=0x05;
  break;
  case 4:goto termination;
  default:cout<<"Invalid choice"<<endl;
           cout<<"Press any key to continue"<<endl;
           getch();
           goto start;    }
cout<<"\nPress any key to stop the motor";
while(!kbhit())
  outportb(address,data);
  outportb(address,0x00);
  goto start;
  termination:clrscr();    }
```