A fractional step solution for step motors

V.A. Dimaki, A.A. Sosin, and D.Yu. Shestakov

Institute of Atmospheric Optics,
Siberian Branch of the Russian Academy of Sciences, Tomsk

Received July 28, 2006

A simple method allowing operation of a step motor in the fractional step mode is considered. A scheme is suggested of a power supply constructed on the basis of the AT90S2313 microcontroller which supports the performance of the step motor in the considered mode. A control program for the microcontroller is described. The presented results are supposed to be successfully used to construct the devices with a high positioning accuracy of separate units and elements.

Introduction

Today step motors (SM) are widely used in constructing various measuring tools (including spectral ones) and other devices, which require high positioning accuracy of separate units and microprocessor control.[1] The latter is achieved due to the use of a special logic and high-precision chip-realizable drivers. In addition to driver microchips there are more composite[2,3] microchips with an integrated controller.

Step motors have a wide range of angular resolutions. Special precision SMs with a resolution of 1.8° or 0.72° per step are used to construct high-precision instruments. It is necessary to note, that the increase of positioning accuracy usually results in either the scheme complication of a digital-type driver or rise in prices of equipment due to the use of dedicated-chip drivers.

Another method to increase the positioning accuracy is based on the use of standard SMs operating in the fractional step mode controllable by an external program. In this work, a device providing the microstep operating mode of a SM is suggested together with a control program for its maintenance.

The device is designed for the use in MDP-12 and MDP-23 monochromators.

1. The SM drive circuitry

The SM drive circuitry includes power-circuit transistor switches and an AT90S2313 microcontroller (Fig. 1).

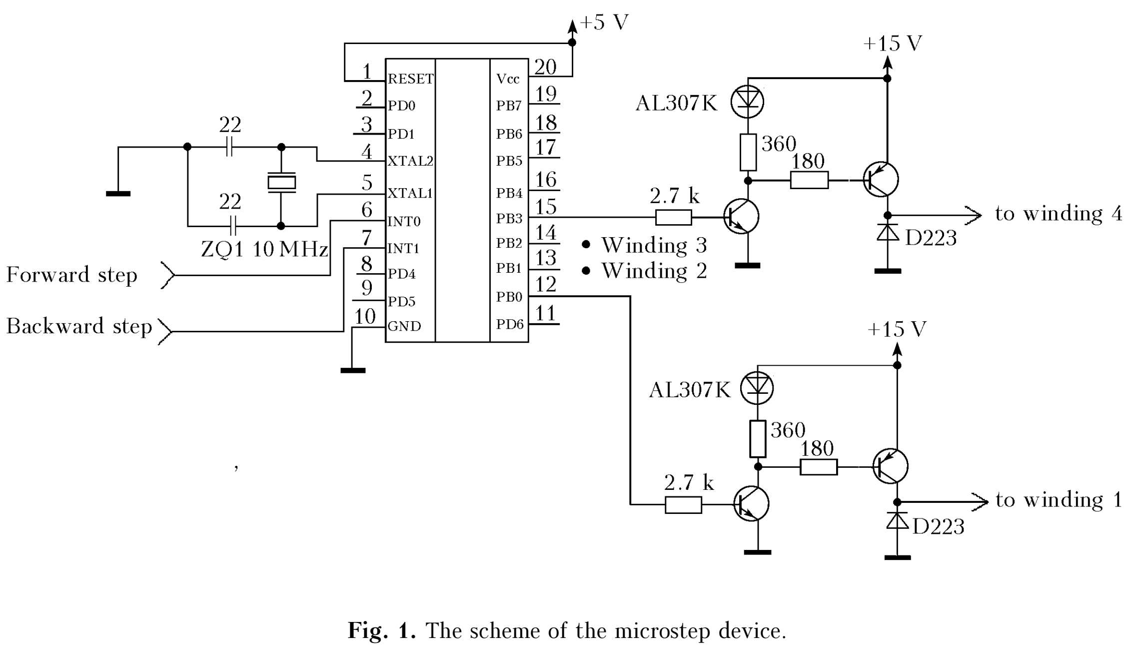

Fig. 1. The scheme of the microstep device.

0235-6880/07/02 163-03 $02.00

The principal idea is the following. If the current is flows in two windings (e.g., 1 and 2) in turn, then the motor rotor takes an intermediate position between poles corresponding to these windings. If in this case the flowtime is the same in neighboring windings, then the rotor is in the position of 1/2 step from the neighboring poles due to magnetic field redistribution. Varying the ratio of flowtimes in windings, i.e., pulse duration modulating (PDM), it is possible to force the rotor to take any position between poles by means of rotating with any fractional step.

In our case, 1/16-step operating mode has been realized. The software PDM has been realized with the period 160 μs. Thus, if the flowtime in one winding is t_1, then in the neighboring one it is $t_2 = 160\ \mu s - t_1$. The flowtime discreteness was chosen equal to 10 μs, i.e., the rotor could rotate between the poles by 1/16 of the step. When arriving a controlling pulse, an external controller interrupt is generated. Controlling pulse arriving to controller's pin 6 generates the external interrupt INT0; in this case, the flowtime in neighboring windings changes by 10 μs and the rotor clockwise rotates by 1/16 step. Arriving the controlling pulse to pin 7 generate the external interrupt INT1 which results in the counter-clockwise rotation of the rotor by 1/16 of the step.

2. Microcontroller program

The peculiarities of the operation algorithm of the SM drive circuitry can be explained more clear when considering the below microcontroller program realized on the C language. This program is assumed to be "stitched in" the controller with a standard programmer (see, e.g., Ref. 4).

```
// SM control program
// Microstep
#include <90s2313.h>
#include <delay.h>
int a, n, m;
unsigned char b, c;

interrupt[2] void takt_sd_pered(void) // External interrupt INT0
{ // Forward microstep
#asm
    cli
#endasm
n=n+1;
if(n==m)
{b=c; c=c<<1; n=0;
if(c==16){c=1;}
}
#asm
    sei
#endasm
} // End of the interrupt 2 function

interrupt[3] void takt_sd_nasad(void) // External interrupt INT1
{ // Backward microstep
#asm
    cli
#endasm
n=n–1;
if(n<0)
{n=m–1; c=b; b=b>>1;
if(b<1){b=8;}
}
#asm
    sei
#endasm
} // End of the interrupt 3 function

void main()
{
int ii;
// B port initialization (setting)
DDRB=0xff; // Pins are configured like exits
PORTB=0x00; // Set zero voltage at all pins of the port
DDRD=0x00;
PORTD=0xff;

MCUCR=15; // Interrupts are rising-edge generated
// at entrances INT0 and INT1, pins 6 and 7
#define xtal 10000000L

#asm
    sei // Interrupt enable
#endasm
SREG= SREG | 128;
GIMSK=GIMSK | 192;
a=1; // Initial state of the port pins
PORTB=a;
m=16; // Microstep is 1/16 of the step
b=1; c=2; n=0;
while (1) {
PORTB=b; // PDM
for(ii=1; ii<=(m–n); ii++) { delay_us(10); }
PORTB=c; //
for(ii=1; ii<=n; ii++) { delay_us(10); }
}
} // End of the program
```

To control SM windings, four low-order digits of the microcontroller's pin B are used. The variables b and c determine, to which winding the voltage is applied: to the first winding if b = 1, to the second one if b = 2, to the third one if b = 4, and to the fourth winding if b = 8. The winding reconnection is realized in the endless while-loop. The flowtime is divided between the windings depending on the variables m and n. The variable n increases by one and the flowtime changes by 10 μs when external interrupt INT0 is generated (clock pulsing to pin 6). The rotor in this case rotates by 1/16 of the step. When n = m, the next couple of windings is connected by means of the statements b = c; c = c ≪1; n = 0. When c = 16, assign 1 to c; in this case, b = 8 and the fourth and first windings are coupled. Next couple is the first and second windings. Thus, windings are reconnected circularly and the SM rotor rotates.

The interrupt INT1 is generated when clock pulse arrives to microcontroller's pin 7. The variable n decreases by one and the rotor rotates by 1/16 of the step in the opposite direction. If n is negative the reconnection to other couple of windings is realized by means of the operators $n = m - 1$; $c = b$; $b = b \gg 1$. If b turns out to be less than unity, assign 8 to b; in this case $c = 1$, the first and fourth windings are connected. Windings reconnection is also realized circularly but in the opposite direction.

Conclusion

As is seen from the program text, the suggested algorithm allows any fraction step to be obtained (the variable m here is the step fractional number). Electric circuit is quite simple, does not includes expensive elements, and accessible even to unskilled users. The type of transistors in power switches depends on the used SM power. In this work, transistors KT837G with small heat sinks have been used for the DSHI-200-2 step motor. The AT90S2313 microcontroller, control program, and programming methods were described in Ref. 4.

This device has been designed for the control of monochromators with SMs to increase the rotation accuracy of diffraction gratings; also, the device is used to control a SM of a coordinate table (plotting table) which is a part of a laser cutting setup. Figure 2 shows the board with two control devices for two SMs of the plotting table.

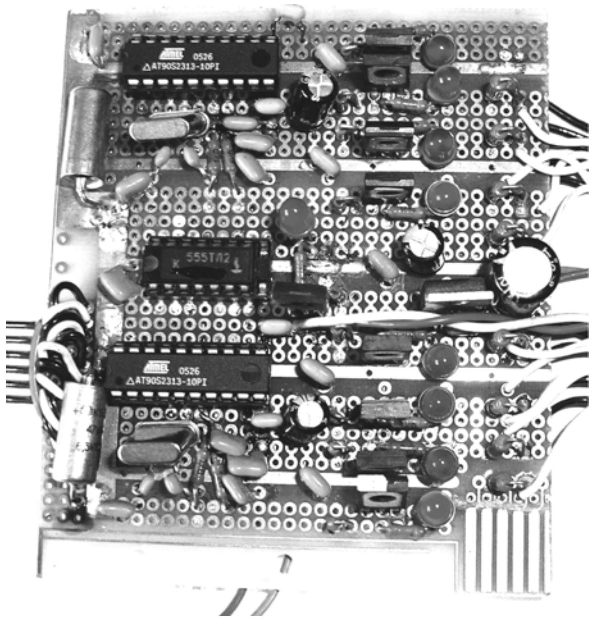

Fig. 2. The view of the designed board.

Fractional step mode of a SM allowed laser cutting refinement. Step fragmentation less than 1/16 does not result in increase of positioning accuracy due to mechanical end plays in the drives.

References

1. http://www.cs.uiowa.edu/~jones/step/
2. http://www.motionex.com/cmotor/engref.htm
3. http://www.eio.com/stprprod.htm
4. M.S. Golubtsov and A.V. Kirichenkova, *AVR Microcontrollers: from Simple to Complicated* (Solon-Press, Moscow, 2004), 237 pp.