Joystick Based Stepper Motor Angle Controller **Using AVR MCU**

ASHUTOSH M. BHATT

A stepper motor is associated with parameters like speed, direction, number of revolutions and angle of rotation. There are different types of stepper motor controllers designed to control required parameters as per specific applications. Here is a project for controlling a stepper motor's angle of rotation up to 360°.

Controlling the angle of rotations of stepper motors finds many applications including the following:

Satellite dish antenna positioning. A satellite dish antenna should be in alignment with the satellite in space to receive signal at maximum strength. Therefore it is important to position the dish antenna at required angle.

Aileron, elevator and rudder positioning in aircraft. In aircraft, it is required to position the aileron, elevator and rudder at desired angles to increase or decrease lift, change direction and increase or decrease speed.

Rudder positioning on ship. To move the ship in a specific direction, it is important to set the angle of the rudder located in front of the propeller.

Tank guns or howitzers positioning. For hitting the bulls-eye (target), the tank gun (or howitzer) must be positioned at a specific angle.

One way to control the stepper motor's angle of rotation is by using a joystick. You can turn the joystick shaft at a specific angle to rotate the motor at the desired angle and, at the same time, setting of the angle of rotation can be displayed on the LCD screen.

As the joystick shaft is turned right or left, the motor immediately follows the joystick and rotates in the same direction by same angle. That is, if the shaft is turned counter-clock-wise, the motor also rotates in that direction through a similar angle, and vice versa.

Circuit and working

Circuit of the joystick based stepper motor angle controller using an AVR microcontroller (MCU) is shown in Fig. 1. It is built around AVR MCU ATMega16 (IC4), current driver chip ULN2003 (IC3), a 16x2 LCD module (LCD1) and a few other components.

In this project we use a 10-kilo-ohm joystick potmeter (JP1), which varies the voltage given as input to the built-in analogue-to-digital (ADC) converter of the MCU. The MCU, through the program, rotates the stepper motor at a desired angle as per the position of the joystick shaft.

JP1 is connected as input to the first channel of the built-in ADC at pin PA0. It is connected in such a way that when its shaft is rotated fully, voltage at PA0 increases from 0V to 5V.

PORTB pins PB0 through PB3 are connected to input pins of IC3. These pins drive stepper motor M1. Output of IC3 drives four coils of the unipolar stepper motor. The common terminal (point E) of motor coils and common pin 9 of IC3 are connected to + 5V.

PORTD of IC4 drives eight data pins

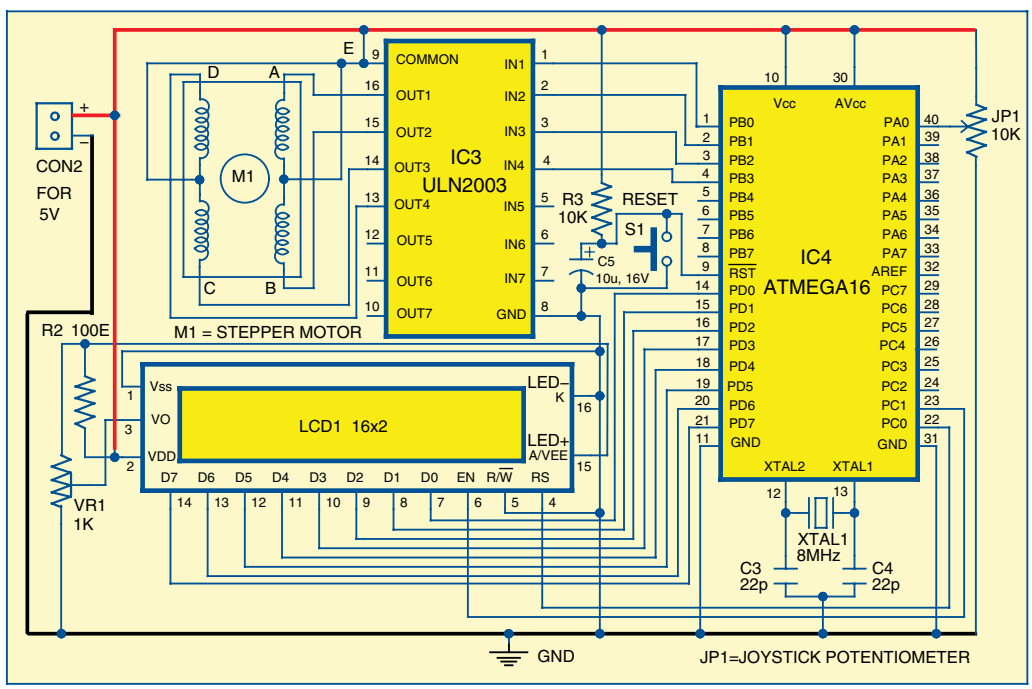


Fig. 1: Circuit diagram of the joystick based stepper motor angle controller using an AVR MCU

(D0 through D7) of LCD1. PORTC pins PC0 and PC1 of IC4 drive two control pins RS and EN of LCD1, respectively. Control pin R/$\overline{W}$ of LCD1 is connected to ground. A 1-kilo-ohm preset is connected to pin 3 of LCD1 to control the contrast of the display.

An 8MHz crystal is connected to crystal input pins of ATMega16 along with two ceramic capacitors forming the clock circuitry. Complete circuit operation is due to the software program embedded into the AVR MCU.

Power supply. The circuit requires 5V DC regulated voltage. The power supply circuit is shown in Fig. 2. 5V DC is derived from 230V AC mains through a transformer (X1), bridge rectifier DB107 (IC1), 7805 regulator (IC2) and a few other components. Glowing of LED1 indicates the presence of power in the circuit.

Software

The software program is written in C language. It is compiled using AVR Studio software.

Software logic. Software logic is to find two things: setting the angle of the motor rotation as per position of joystick shaft and positioning the stepper motor in predefined start angle at power on.

As the potmeter is rotated, analogue voltage input to the ADC pin is varied and so the corresponding digital input also varies. As analogue input varies from 0V to 5V, digital input varies from 0 to 1023 (because ATMega16 has 10-bit ADC resolution). This complete range of 0 to 1023 is divided into 12 different ranges such as 0-85, 86-171, 172-257 and so on. The motor angle is set as per the digital value listed in the table.

Motor rotation directly depends on its step resolution. Motor step resolution is 3.75°/pulse. That means, for each pulse applied, motor rotates by 3.75° only. But it is required to give four pulses in a sequence to all four motor coils. So the motor will rotate by 3.75° × 4 = 15°. Thus, we can rotate the motor in multiples of 15° such as 15°, 30°, 45°, 60°, 75° and so on.

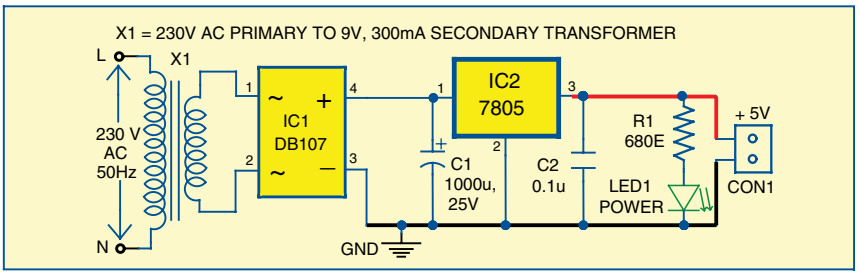


Fig. 2: Circuit of the power supply

PARTS LIST

Semiconductors:
IC1 - DB107 bridge rectifier
IC2 - 7805, 5V regulator
IC3 - ULN2003 driver
IC4 - ATmega16 MCU
LED1 - 5mm LED
LCD1 - 16×2 alphanumeric display module

Resistors (all 1/4-watt, ±5% carbon):
R1 - 680-ohm
R2 - 100-ohm
R3 - 10-kilo-ohm
VR1 - 1-kilo-ohm preset
JP1 - 10-kilo-ohm joystick potmeter

Capacitors:
C1 - 1000□F, 25V electrolytic
C2 - 0.1□F ceramic
C3, C4 - 22pF ceramic
C5 - 10□F, 16V electrolytic

Miscellaneous:
X1 - 230V AC primary to 9V, 300mA secondary transformer
XTAL1 - 8MHz crystal oscillator
M1 - 3.75°/step, 5V stepper motor (unipolar)
CON1 - 2-pin berg strip male connector
CON2 - 2-pin terminal connector
CON3 - 5-pin berg strip male connector
- 16-pin berg strip female connector for LCD1
S1 - Tactile switch

The logic is to find out the number of pulses that should be applied to the motor so it can rotate till the desired angle value is achieved. To do this, every time you select a rotation angle, the number of pulses to be applied to the motor is found using the relationship given below:

Number of pulses = 4 × (new set angle value – previous set angle value)/15

For example, let us say, the previously set angle value was 45° and the new set angle value is 75°. That means, the motor has to rotate for 30°. Putting values in the above equation,

Different Angles of Stepper Motor Rotation

Serial number	Range of digital values	Corresponding angles (in degree)
1	0-85	0
2	86-171	30
3	172-257	60
4	258-343	90
5	344-429	120
6	430-515	150
7	516-601	180
8	602-687	210
9	688-773	240
10	774-859	270
11	860-945	300
12	946-1023	330

Number of pulses = 4 × (75-45)/15 = 8

Therefore the motor rotates for 8 × 3.75° = 30°

If the new set angle value is less than the previous set angle value, the result will be negative and the motor will rotate in reverse direction.

Program functions. The program is made up of different functions as explained below.

Senddata( ) sends one byte data to the LCD on its data bus.

sendcommand( ) sends one byte command to the LCD on its data bus.

printstr( ) displays the message or string on the LCD.

inc_angle( ) increments the angle by 15° and displays on the LCD.

dec_angle( ) decrements the angle by 15° and displays on the LCD.

display_value( ) takes the integer angle value as input, converts all digits into ASCII and displays it as angle value on the LCD.

lcd_init( ) configures and initialises the LCD.

adc_init( ) initialises the built-in ADC of AT-Mega16.

rotate_motor ( ) rotates the stepper motor in clockwise or counter-clockwise direction as per the entered angle value till the motor reaches the required position.

Construction and testing

Actual-size, single-side PCB layout of the joystick based stepper motor angle rotation controller using AVR MCU is shown in Fig. 3 and its component layout in Fig. 4. Actual-size, single-side PCB layout of the power supply section is shown in Fig. 5 and its component layout in Fig. 6.

Use any 2-wire electrical cable between CON1 and CON2 to connect 5V power supply to the controller circuit. When the circuit is first switched on, the LCD displays the angle as 0°. The motor also rests at 0°. To set the desired angle, rotate the shaft of the joystick potmeter. As potmeter shaft is rotated counter-clockwise (increase) or clockwise (decrease), the angle is increased or decreased in steps of 15° and is displayed on the LCD. As soon as the angle is changed, the stepper motor starts rotating in order to reach the new angle. For example, if the angle is set to 30° using the potmeter, the motor immediately starts rotating and stops at 30°. If the angle is increased by rotating the potmeter in counter-clockwise direction, the motor also rotates in the same direction. Similarly, if the angle is decreased by rotating the potmeter in clockwise direction, the motor rotates clockwise.

EFY note. The circuit has been checked in EFY Lab with a regular potentiometer instead of a joystick. ●

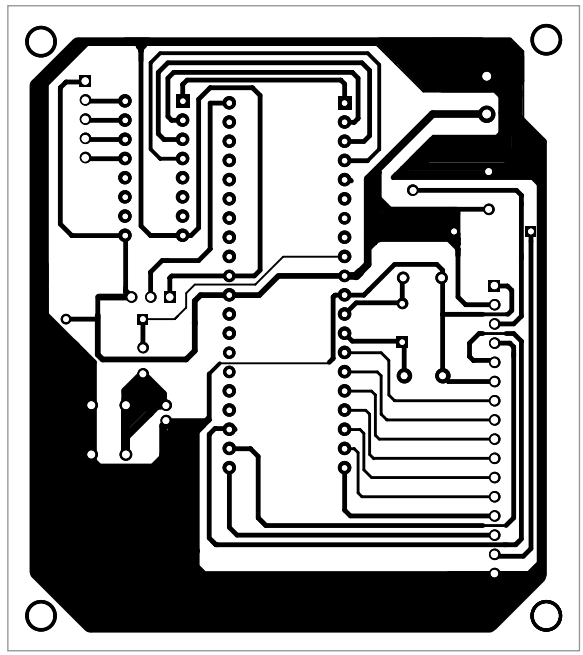


Fig. 3: Actual-size PCB layout of the joystick based stepper motor angle controller using AVR MCU

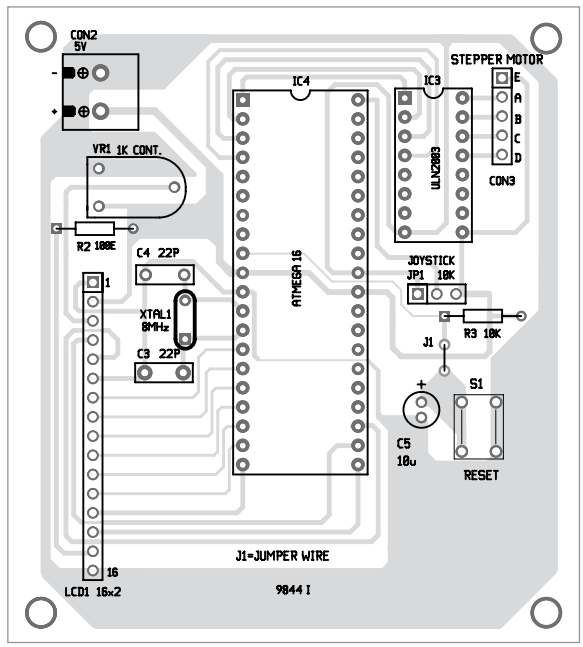


Fig. 4: Component layout of the PCB shown in Fig. 3

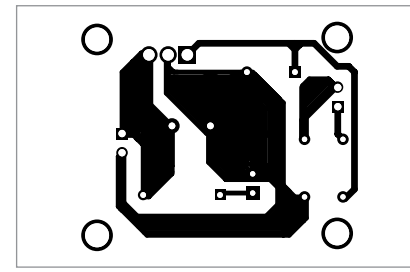


Fig. 5: Actual-size PCB layout of the power supply

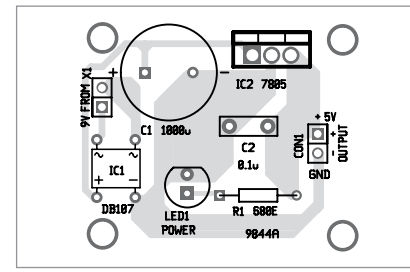
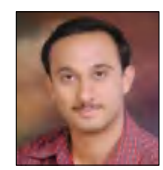


Fig. 6: Component layout of the power supply shown in Fig. 5

EFY Note

The source code of this project is included in this month's EFY DVD and is also available for free download at *source.efymag.com*



Ashutosh M. Bhatt is M.Tech in embedded systems. Currently, he is lecturer of electronics and radio engineering at government polytechnic, Jamnagar, Gujarat