БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **69**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **69**

София • 2018 • Sofia

Low-cost Control of Stepper Motors with Multi-purpose for Robotics

Ognyan Manolov, Svetoslav Noykov

European Polytechnic University, Pernik,
Institute of Robotics - BAS, Sofia
Email: ognyan.manolov@epu.bg

Abstract: *In this paper a low-cost and reliable control system for bipolar stepper motor is presented. It is capable to control up to six stepper motors. The system includes a low-cost and robust microcontroller ARDUINO Uno and up to six STP-030 modules for stepper motor driven (one module per motor driving), which are reliable and simple to control. The proposed control system is very promising for variety of applications in robotics.*

Keywords: *microcontroller ARDUINO, stepper motor driver, stepper motor control*

1. Introduction

The stepper motors manifest some important features that make them irreplaceable for some applications like in robotics, industrial automations, medical machinery, security, consumer electronics, satellite, etc. Such applications include driving of different mechanisms and transmissions as well as driving and precise positioning of variety of actuators, manipulators, reading sensor systems, instruments, etc. The main advantages of stepper motors are as follows [1]:

- the rotation angle of the motor is proportional to the input pulse;
- the motor has full torque at standstill; precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non cumulative from one step to the next;
- excellent response to starting, stopping and reversing; very reliable since there are no contact brushes in the motor, therefore the life of the motor is simply dependant on the life of the bearing;
- the motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control; it is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft;
- a wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

According to the type of winding arrangements for the electromagnetic coils, stepper motors can be divided into following main classes: ***unipolar*** and ***bipolar***. Among them, bipolar ones are especially common. Because windings are better utilized, bipolar motors are more powerful than a unipolar motor of the same weight. This is due to the physical space occupied by the windings. A unipolar motor has twice the amount of wire in the same space, but only half used at any point in time, hence is 50% efficient (or approximately 70% of the torque output available) [2].

Though a bipolar stepper motor is more complicated to drive, the using of a proper driver board means this is much less difficult to achieve. Also, another class stepper motors exist: multi-phase stepper motors, but their high price limits their application.

To run a stepper motor, a proper control system is normally required, which includes two different kinds of modules: 1) a controller to create step and direction signals (at ±5 V normally) and 2) a driver module which can generate the necessary current to drive the motor.

The stepper controller drives minimum 3 lines - traditionally labelled "step", "dir", "enable" - which carry motion information to the stepper driver. (Often these 3 lines are opto-isolated). The stepper controller is typically a pure digital logic device, and requires relatively little power.

The stepper driver connects to the 4 thick wires of the stepper motor. It contains the big power transistors, and requires a thick power cable to a DC power supply, because all the power to drive the motors runs through it [3].

In this paper, we present a low-cost and reliable control system for bipolar stepper motors. It is capable to control up to six stepper motors. The system includes a low-cost and robust microcontroller ARDUINO Uno [4] and up to six stepper motor driver boards STP-030 [5] (one driver board per a motor), which are reliable and simple to control.

2. Description of the basic modules of developed system for bipolar stepper motors control

2.1. The microcontroller

The microcontroller board ARDUINO Uno is based on the *ATmega328P* [4] and it is shown on Fig. 1. The board is completed with 14 digital input/output pins (of which 6 can be used as PWM outputs) and 6 analog inputs, with 16 MHz quartz crystal, a USB port, a power jack, an ICSP header and a reset button. It contains all recourses needed to support the microcontroller. The connection to a computer may be established through the USB port by means of a USB cable.

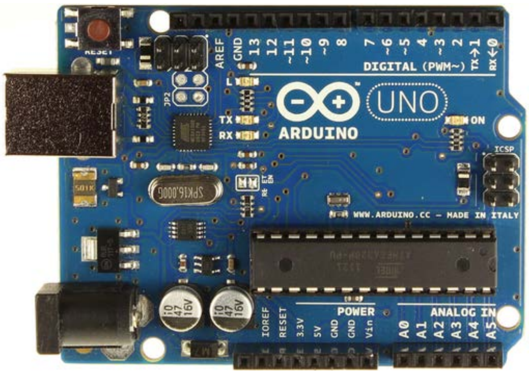

Fig. 1. The microcontroller ARDUINO Uno

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. The Arduino language (based on Wiring) is implemented in C/C++. The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and to communicate with them.

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ***xxx.ino***. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information.

The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow programmer to verify and upload programs, to create, open, and save sketches, and to open the serial monitor [6].

There are two special functions that are a part of every Arduino sketch: ***setup()*** and ***loop()***. The ***setup()*** is called once, when the sketch starts. It is intended to do setup tasks like setting pin modes or initializing libraries. The ***loop()*** function is called repeatedly and is heart of most sketches. It is necessary to include both functions in every sketch [7]. Further information is available on the Arduino© web page https://www.arduino.cc/

2.2. The stepper motor driver board

The driver module, which is used in the control system for bipolar stepper motors, are based on the stepper motor driver board STP-030 [5]. A photos of the STP-030 is given in Fig. 2.

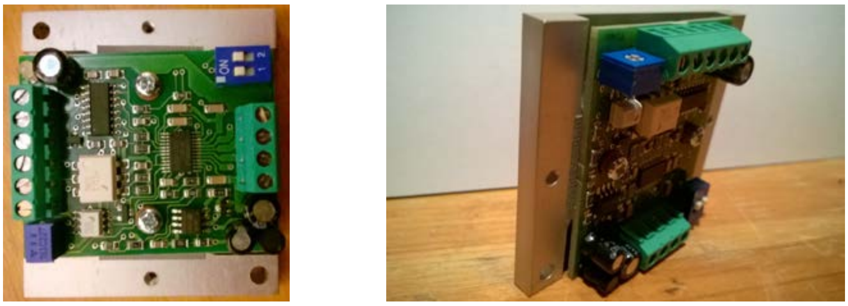

Fig. 2. The stepper motor driver module STP-030.

This module is designed to operate a bipolar stepper motor in full-, half-, quarter-, and eighth- step modes at up to 1.5A current per coil. The STP-030 interface is an ideal fit for applications where a complex microprocessor is unavailable. The microcontroller ARDUINO Uno is excellent choice to control such driver board. All of the control inputs of STP-030 are opto-isolated.

The driving control of the stepper motor is being carried out by means of the following signals:

- ***Step Input (STEP)*** . A low-to-high (or high -to- low) transition (depend on the sign of Viso) on the STEP input advances the motor one increment. The size of the increment is determined by the state of two micro- switches, dedicated for this purpose.
- ***Direction Input (DIR)***. This determines the direction of rotation of the motor.
- ***Enable Input (ENABLE)***. This input enables/disables the output drivers and this way the stepper motor is turned on or off.
- ***Input for supply voltage (Viso)*** for the opto-isolated part. The value of this voltage must match the value of the above described control signals. The sign of this voltage determines the active transitions or active levels of the above described control signals.

Additional details about functioning and characteristics of STP-030, are in [5].

3. The system for bipolar stepper motor control - experimental solutions

3.1. Hardware and software for a bipolar stepper motor control

On the basis of a microcontroller ARDUINO Uno and a stepper motor driver board STP-030, it was created a simple system for control of a bipolar stepper motor which block diagram is given in Fig. 3.

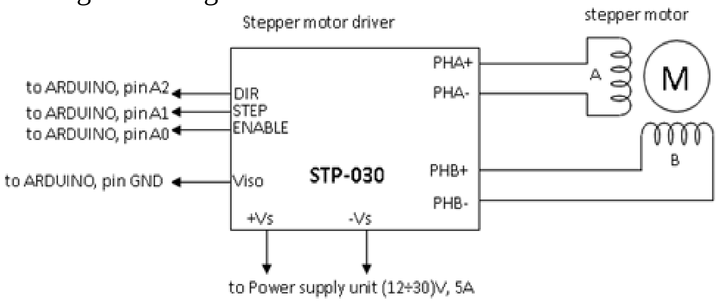

Fig. 3. Block diagram for a stepper motor control system, based on STP-030.

The ENABLE Input of STP-030 is connected to ARDUINO, pin A0; the STEP Input of STP-030 is connected to ARDUINO, pin A1; DIR Input of STP-030 is connected to ARDUINO, pin A2; Viso Input of STP-030 is connected to ARDUINO, pin GND. The operator or higher-level control unit should prepare the necessary commands for control of the stepper motors in proper format - command line, containing six consecutive symbols.

This command line is sent to the microcontroller ARDUINO Uno through the USB connection. The command line should contain six symbols in the following format:

$$\mathbf{X\ \ Y_1\,Y_2\ \ Z_1\,Z_2\,Z_3}$$

where:

X=1: Sets one from both possible directions of movement of the stepper motor

X=2: Sets the opposite direction of movement of the stepper motos

X=3: Switch off the stepper motor

Y_1Y_2: These digits set the pulse length ***t*** that determines the speed of the movement of the stepper motor: $\boldsymbol{t = 10 * Y_1 + Y_2}$

$Z_1Z_2Z_3$: These digits set the number of steps ***n*** which should complete the stepper motor: $\boldsymbol{n = 100 * Z_1 + 10 * Z_2 + Z_3}$

In the beginning of the corresponding developed sketch for Arduino, the following variables are included:

```
…………………………………….
int enablePin = A0;           // ENABLE line of STP-030 connected to
                              //ARDUINO, digital pin A0
int stepPin = A1;             // STEP line of STP-030 connected to
                              //ARDUINO, pin A1
int dirPin = A2;              // DIR line of STP-030 connected to
                              //ARDUINO, digital pin A2
int incomingByte = 0;         // to save the received symbol X
int incomingByte2 = 0;        // to save the received symbol Y1
int incomingByte3 = 0;        // to save the received symbol Y2
int incomingByte4 = 0;          // to save the received symbol Z1
int incomingByte5 = 0;          // to save the received symbol Z2
int incomingByte6 = 0;          // to save the received symbol Z3
int numSteps = 0;               // steps to be executed from the motor,
                                // calculated based on incomingByte4,
                                // incomingByte5, incomingByte6
int pulse = 5;                  // pulse width of the signal for motor velocity
                                //control, calculated, based on
                                //incomingByte2, incomingByte3
int i;                          // motor steps counter
……………………………………………………………………………
```

The function ***setup()*** sets the pin modes of the Arduino as well as the serial connection:

```
……………………………………………………………………………..
void setup()
{
pinMode(enablePin, OUTPUT);     // sets the pin as output
pinMode(dirPin, OUTPUT);        // sets the pin as output
pinMode(stepPin, OUTPUT);       // sets the pin as output
Serial.begin(9600);             // sets the serial connection
digitalWrite(enablePin, HIGH);  // enable MOTOR
}
…………………………………………………………………………
```

The next function- ***loop()***, called repeatedly generates the corresponding motor control signals, based on the current received command line and passes to the driver board STP-30. Here the code of this function is not presented, because the text is very long and the volume of the article will be significantly expanded.

3.2. The experimental set-up for a bipolar stepper motor control

The experimental set-up which includes a microcontroller ARDUINO Uno and two stepper motor driver boards STP-030 is presented in Fig. 4.

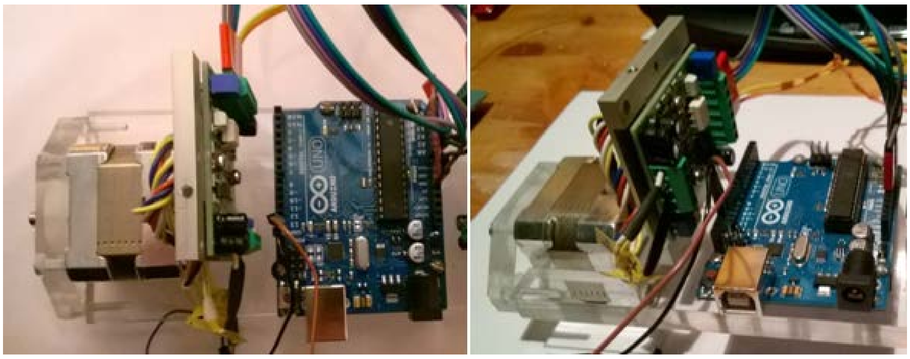

Fig. 4. An instance for experimental control system for bipolar stepper motor.

Conclusion

The stepper motor are very common in robotics and automation: they are widely used for driving in many modern robotized industrial machines, unmanned vehicles and aircrafts, etc., where controlled movement is required, including precise control of rotation angle, speed, position and synchronism. In particular, the bipolar stepper motors are preferable for such kind of applications, because they are compact and low-cost. The proposed control system, designed for control of bipolar stepper motors, is very promising for variety of applications in robotics.

References

1. Agarwal, T., Stepper Motor – Types, Advantages & Applications, https://www.elprocus.com/stepper-motor-types-advantages-applications/ , *from* EL-PRO-CUS, Copyright 2015 © Elprocus
2. Stepper motor, https://en.wikipedia.org/wiki/Stepper_motor , *from* Wikipedia, the free encyclopedia, Wikimedia Foundation, Inc.
3. Stepper motor driver, http://reprap.org/wiki/Stepper_motor_driver , *from* RepRap.org
4. Arduino Uno Rev. 3, https://store.arduino.cc/arduino-uno-rev3 , *from* ©Arduino
5. Stepper motor driver board STP-030, http://tinel.com/uploads/STP-030.pdf, Tinel Electronics Ltd., Sofia, (in Bulgarian)
6. Arduino Software (IDE), https://www.arduino.cc/en/Guide/Environment , *from* ©Arduino
7. Sketch, https://www.arduino.cc/en/Tutorial/Sketch , *from* ©Arduino
8. Jacob Apkarian, Aerial vehicle, Patent Application Publication, Pub. №: US 2010/0224723 A1, United States, Pub. date: Sep. 9, 2010.

Низкозатратное управление шаговыми двигателями с многоцелевым предназначением для робототехники

Огнян Манолов, Светослав Нойков

European Polytechnic University, Pernik,
Institute of Robotics - BAS, Sofia
Email: ognyan.manolov@epu.bg

Резюме: *Представлена недорогая и надежная система управления биполярными шаговыми двигателями, которая способна управлять до шести двигателей. Система состоит из недорогого и надежного микроконтроллера ARDUINO Uno и до шести плат драйверов шаговых двигателей STP-030 (одна плата на двигатель), которые надежны и просты в управлении. Предлагаемая система управления очень перспективна для разных применений в робототехнике.*