

УДК 004.42

**МИКРОПРОЦЕССОРНАЯ СИСТЕМА УПРАВЛЕНИЯ ШАГОВЫМ
ДВИГАТЕЛЕМ С ПОМОЩЬЮ ПРОТОКОЛА UART****Бакалин А. Р., Миронец В. В., Голубев Л. П.**

Киевский национальный университет технологий и дизайна

В статье рассматриваются вопросы микропроцессорного управления шаговым двигателем по протоколу UART. Описана разработанная система управления шаговым двигателем по командам, выдаваемым с персонального компьютера.

Ключевые слова: шаговый двигатель, микропроцессор, автоматизация, система управления, UART

При автоматизации различных процессов в качестве активного элемента системы управления очень часто используются шаговые двигатели. Обычно управление ими осуществляется с помощью микропроцессоров или микропроцессорных систем (аналогов Ардуино). Однако в этом случае управление шаговым двигателем осуществляется по загруженной в микропроцессор программе. Часто возникает необходимость динамически изменять параметры работы шагового двигателя – скорость вращения, направление и величину угла поворота и др.

Постановка задачи

Поэтому необходимо обеспечить возможность динамического изменения параметров микропроцессорной системы, построенной с использованием шагового двигателя. Для решения этой задачи необходимо разработать программно-технический комплекс, позволяющий динамически изменять параметры системы с помощью персонального компьютера, передавая соответствующие команды в систему через выбранный COM порт по протоколу UART.

Результаты исследования

В результате наших исследований разработана автоматизированная система управления шаговыми двигателями. Она осуществляет оперативное управление параметрами шагового двигателя – изменением угла поворота, скорости вращения, вплоть до полной (экстренной) остановки двигателя.

Электронные компоненты системы:

- микропроцессорная система Arduino Uno;
- драйвер двигателя ULN2003;

- шаговий двигатель 28BYJ-48;
- персональний комп'ютер (ноутбук).

Arduino Uno – флагманська платформа для розробки на базі мікроконтролера ATmega328P. На Arduino Uno передбачено все необхідне для зручної роботи з мікроконтролером: 14 цифрових входів/виходів (6 з них можуть використовуватися в якості ШИМ-виходів), 6 аналогових входів, кварцевий резонатор на 16 МГц, роз'єм USB, роз'єм живлення, роз'єм для внутрішньоплатного програмування (ICSP) і кнопка скидання [1, 3].

Мікроконтролер ATmega328P

Серцем платформи Arduino Uno є 8-бітний мікроконтролер родини AVR-ATmega328P.

Мікроконтролер ATmega16U2

Мікроконтролер ATmega16U2 забезпечує зв'язок мікроконтролера ATmega328P з USB-портом комп'ютера. При підключенні до ПК Arduino Uno визначається як віртуальний COM-порт. Прошивка мікросхеми 16U2 використовує стандартні драйвери USB-COM, тому установка зовнішніх драйверів не потрібна.

Піни живлення

- **VIN:** Напруга від зовнішнього джерела живлення (не пов'язана з 5 В від USB або іншим стабілізованим напругою). Через цей вивід можна як подавати зовнішнє живлення, так і споживати струм, якщо до пристрою підключено зовнішній адаптер.
- **5V:** На вивід поступає напруга 5 В від стабілізатора плати. Цей стабілізатор забезпечує живлення мікроконтролера ATmega328P. Запитувати пристрій через вивід 5V не рекомендується – в цьому випадку не використовується стабілізатор напруги, що може призвести до перегріву плати.
- **3.3V:** 3,3 В від стабілізатора плати. Максимальний струм вивода – 50 мА.
- **GND:** Виводи землі.
- **IOREF:** Вивід надає платам розширення інформацію про робочу напругу мікроконтролера. В залежності від напруги, плата розширення може переключитися на відповідний джерело живлення або впливати на перетворювачі напруги, що дозволить їй працювати як з 5 В, так і з 3,3 В пристроями.

- Порты ввода/вывода.
- **Цифровые входы/выходы:** пины 0–13.
- Логический уровень единицы – 5 В, нуля – 0 В. Максимальный ток выхода – 40 мА. К контактам подключены подтягивающие резисторы, которые по умолчанию выключены, но могут быть включены программно.
- **ШИМ:** пины 3, 5, 6, 9, 10 и 11.
- Позволяют выводить 8-битные аналоговые значения в виде ШИМ-сигнала.
- **АЦП:** пины А0–А5.
- 6 аналоговых входов, каждый из которых может представить аналоговое напряжение в виде 10-битного числа (1024 значений). Разрядность АЦП – 10 бит.
- **ТWI/I²C:** пины SDA и SCL.
- Для общения с периферией по синхронному протоколу, через 2 провода. Для работы – надо использовать библиотеку Wire.
- **SPI:** пины 10(SS), 11(MOSI), 12(MISO), 13(SCK).
- Через эти пины осуществляется связь по интерфейсу SPI. Для работы надо использовать библиотеку SPI.
- **UART:** пины 0(RX) и 1(TX)
- Эти выводы соединены с соответствующими выводами микроконтроллера ATmega16U2, выполняющей роль преобразователя USB-UART. Используется для коммуникации платы Arduino с компьютером или другими устройствами через класс Serial.

Светодиодная индикация

Имя светодиода	Назначение
RX и TX	Мигают при обмене данными между Arduino Uno и ПК.
L	Светодиод вывода 13. При отправке значения HIGH светодиод включается, при отправке LOW – выключается.
ON	Индикатор питания на плате.

Раз'єм USB Type-B.

Раз'єм USB Type-B призначений для прошивки платформи Arduino Uno з допомогою комп'ютера.

Раз'єм для зовнішнього живлення.

Раз'єм для підключення зовнішнього живлення від 7 В до 12 В.

ICSP-раз'єм для ATmega328P.

ICSP-раз'єм призначений для внутрисхемного програмування мікроконтролера ATmega328P. З використанням бібліотеки SPI дані виводи можуть здійснювати зв'язок з платами розширення за інтерфейсом SPI. Лінії SPI виведені на 6-контактний раз'єм, а також продубльовані на цифрових пинах 10(SS), 11(MOSI), 12(MISO) і 13(SCK).

ICSP-раз'єм для ATmega16U2.

ICSP-раз'єм призначений для внутрисхемного програмування мікроконтролера ATmel.

Шаговий двигачель може точно перемещатися на мінімально можливий кут, називаний кроком. Для практичних завдань можна вважати, що кроковий мотор трохи схожий на сервопривід. Можна задати йому повернутися в певне положення і можна розраховувати отримати достатньо стабільні результати в декількох повторних експериментах. Звичайно, сервоприводи обмежені кутом повороту в діапазоні від 0 до 180°, кроковий ж двигачель може обертатися неперервно, подібно двигачелю постійного струму. Перевагою крокових двигачелів є те, що можна досягти значно більшої ступені контролю над рухом. До недоліків крокових двигачелів можна віднести декілька більш складного управління, ніж в випадку з сервами або моторами постійного струму [2,4].

28BYJ-48 – це маленький, дешевий, 5 вольтовий кроковий моторчик з редуктором. Передаточне число редуктора у нього приблизно 64:1, що дозволяє отримати повністю достатній крутячий момент для моторчика такого розміру і швидкість порядку 15 об/хв. З певними програмними хитрощами для поступового прискорення можна досягти більш ніж 25 об/хв.

Дешевизна і мініатюрність робить 28BYJ ідеальним варіантом для невеликих робототехнічних застосувань, а також для знайомства з управлінням кроковими двигачелями, використовуючи Arduino. Нижче представлено детальна специфікація крокового двигачеля 28BYJ-48.

Тип мотора	Униполярний шаговий двигатель
Число фаз	4
Подключеніе	5-выводов (к контроллеру двигателя)
Рабочее напряжение	5-12 вольт
Частота	100 Гц
Сопротивление по постоянному току	50 Ом \pm 7%(25°C)
Частота под нагрузкой	> 600 Гц
Частота на холостом ходу	> 1000 Гц
Крутящий момент	> 34.3 мН*м (120 Гц)
Момент самопозиционирования	> 34.3 мН*м
Стопорящий момент	600-1200 г*см
Тяга	300 г*см
Сопротивление изоляции	> 10 МОм (500 В)
Класс изоляции	A
Шум	< 35 дБ (120 Гц, без нагрузки, 10 см)
Режим шага	Рекомендован полушаговый режим (8-шаговая управляющая сигнальная последовательность)
Угол шага	Полушаговый режим: 8-шаговая управляющая сигнальная последовательность (рекомендовано). 5.625 градусов на шаг, 64 шага на оборот внутреннего вала мотора. Режим полного шага: 4-шаговая управляющая сигнальная последовательность. 11.25°/шаг, 32 шага на оборот внутреннего вала двигателя.
Передаточное отношение редуктора	Производителем заявлено 64:1. Однако, некоторые пытливые товарищи с форума Arduino разобрали редуктор и определили, что, в действительности передаточное число равно 63.68395:1. Это означает, что в рекомендованном полушаговом режиме мы имеем: 64 шага на оборот мотора

	помноженное на передаточное число 63.684 \approx 4076 шагов на полный оборот.
Подключение к контроллеру ULN2003	A (синий), B (розовый), C (желтый), D (Оранжевый), E (красный, средний вывод обмоток)
Вес	30г

Двигатель имеет четыре обмотки, которые запитываются последовательно, чтобы повернуть вал с магнитом.

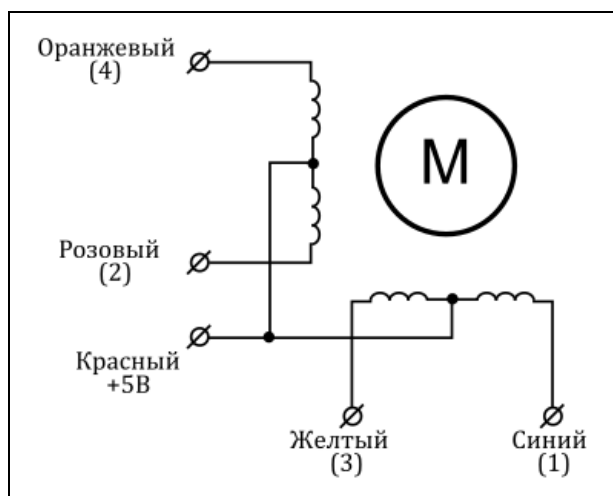


Рис.1. Подключение выводов шагового двигателя 28BYJ-48

Когда используется полношаговый метод управления, две из четырех обмоток запитываются на каждом шаге. Идущая вместе с Arduino IDE библиотека Stepper использует такой способ. В техническом руководстве к 28BYJ-48 сказано, что предпочтительным является использование метода полушага, при котором сначала запитывается только 1 обмотка, затем вместе первая и вторая обмотки, затем только вторая обмотка и так далее. С 4 обмотками это дает 8 различных сигналов, как показано в таблице ниже [6].

Провод	→ Направление вращения по часовой стрелке (1-2 фазы)							
	1	2	3	4	5	6	7	8
4 Оранжевый	■	■						■
3 Желтый		■	■	■				
2 Розовый				■	■	■		
1 Синий						■	■	■

Подключение драйвера шагового двигателя ULN2003 к Arduino Uno.

Плата драйвера шагового двигателя на базе микросхемы ULN2003, представляющей собой массив транзисторов, включенных по схеме Дарлингтона, позволяет достаточно просто управлять мотором 28BYJ-48, используя микроконтроллер. В нашем случае, в качестве управляющего микроконтроллера мы выберем плату Arduino Uno с микроконтроллером ATmega328. Помимо самой микросхемы ULN2003AN, на плате имеется пятиконтактный разъем для подключения к шаговику и четыре светодиода, показывающих, какая из обмоток запитана в текущий момент времени.



Также сбоку расположен джампер (два вывода под четырьмя резисторами), установка которого позволяет подавать питание на шаговый двигатель. Замечу, что питать мотор от 5 В Arduino не рекомендуется, так как мотор может потреблять ток, превышающий возможности Arduino. Лучше использовать внешний 5-12 В источник питания, выдающий ток не менее 1 А. Четыре управляющих входа помечены как *IN1-1N4* и должны быть подключены к четырем цифровым выводам Arduino.

Подключите выводы *IN1*, *IN2*, *IN3* и *IN4* к пинам 3, 4, 5 и 6 Arduino UNO. Положительный контакт источника питания необходимо подключить к выводу, помеченному как «+», а землю источника питания к выводу «-» на плате контроллера. Если для питания Arduino и мотора используются различные источники питания, то необходимо объединить выводы «земля» источников вместе.

Бібліотека *AccelStepper*

Стандартная библиотека Stepper, идущая с Arduino IDE, поддерживает только полношаговый метод управления и имеет сильно ограниченные возможности. Использовать ее можно только в очень простых приложениях, в которых используется только один мотор. Применение библиотеки Stepper для управления 28BYJ-48 является не самым эффективным решением [5, 7].

Есть решение получше – это использовать библиотеку Accel Stepper. Эта библиотека очень хорошо работает совместно с шаговым мотором 28BYJ-48 (мотор почти не греется), а также поддерживает ускорение, что позволяет заставить мотор вращаться быстрее. Библиотека использует код, не блокирующий шаги и включает немало других приятных особенностей.

- Объектно-ориентированный интерфейс для 2, 3 или 4-выводных шаговых двигателей.
- Поддержка ускорения и замедления.
- Поддержка одновременно нескольких шаговых двигателей с независимой работой для каждого мотора.
- Функции API не используют функцию `delay` и не прерывают работу.
- Поддержка выбора функции для реализации шага позволяет работать совместно с библиотекой `AFMotor`.
- Поддержка контроллеров шаговых двигателей таких как `Sparkfun EasyDriver`(основанных на микросхеме драйвера 3967).
- Поддержка низких скоростей.
- Расширяемый API.
- Поддержка подклассов.

Программа для подключения шагового двигателя 28BYJ-48 к Arduino Uno с помощью драйвера ULN2003 приведена ниже;

```
#include <Stepper.h>
const int stepsPerRevolution = 256;
Stepper myStepper(stepsPerRevolution, 2, 4, 3, 5);
void setup() {
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
```



```
myStepper.setSpeed(70);  
Serial.begin(9600);  
}  
void loop() {  
  Serial.println("clockwise");  
  myStepper.step(stepsPerRevolution * 4);  
  digitalWrite(2, LOW); //Для економії енергії снимаем напругу з  
  контактів 2 и 3  
  digitalWrite(3, LOW); //Контакти по умовчанию були активні для  
  удержання тяги на валу  
  delay(2000);  
  Serial.println("counterclockwise");  
  myStepper.step(-stepsPerRevolution * 4);  
  digitalWrite(2, LOW);  
  digitalWrite(3, LOW);  
  delay(2000);  
}
```

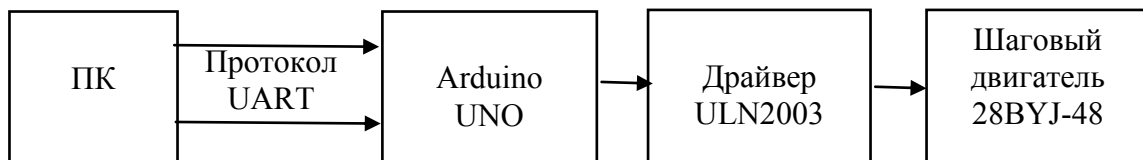


Рис. 3. Схема управління шаговим двигателем з допомогою ПК, використовуючи протокол UART

Алгоритм роботи програмно-технічного комплексу

1. Необхідно зібрати установку в відповідності со схемою (рис. 3).
2. Пользователь на ПК запускає програму ContrStep і заповнює поля форми, відповідаючі за швидкість і напрямлення обертання шагового двигателя.
3. Після натискання кнопки Start інформація з ПК по протоколу UART передається в скетч на Arduino UNO.
4. Далі виконується програма з параметрами, прийнятими з ПК, яка через драйвер ULN2003 керує шаговим двигателем.

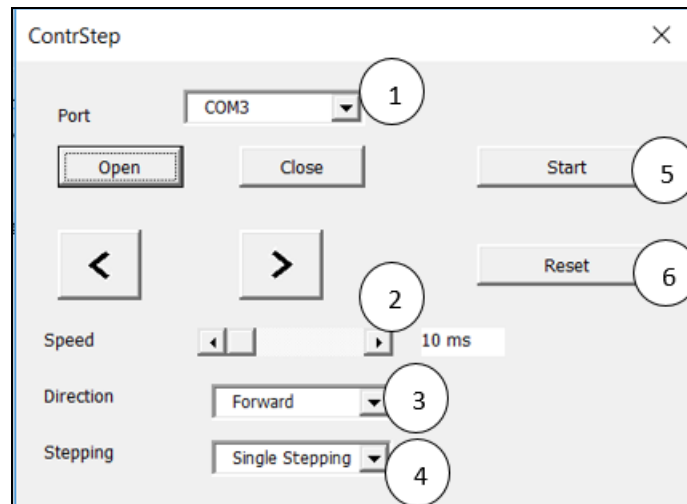


Рис. 4. Окно программы ContrStep:

- 1 – ComboBox1 – комбинированный список номеров портов;
- 2 – ScrollBar1 – Полоса прокрутки скорости вращения шагового двигателя;
- 3 – ComboBox2 – комбинированный список направлений движения;
- 4 – ComboBox3 – комбинированный список величины шага двигателя;
- 5 – CommandButton1 – командная кнопка вызова процедуры передачи параметров шаговому двигателю;
- 6 – CommandButton2 – командная кнопка вызова процедуры сброса параметров шагового двигателя;

Ниже приведен фрагмент программы ContrStep подключения устройства по COM порту:

```
MSComm1.CommPort = 3
MSComm1.Settings = "9600,N,8,1"
MSComm1.InputLen = 0
On Error Resume Next
MSComm1.SThreshold = 2  'Число байт передачи для срабатывания
события
MSComm1.RThreshold = 3  'Число байт приема для срабатывания
события
MSComm1.PortOpen = True
If Err Then
MsgBox "Com" & MSComm1.CommPort & ": not available. Change
CommPort property to another port."
Exit Sub
End If
```

Для ввода информации через последовательный порт, используя элемент управления MSCComm1 можно воспользоваться таким программным кодом:

```
Dim InString as String
' Retrieve all available data.
MSCComm1.InputLen = 0

' Check for data.
If MSCComm1.InBufferCount Then
' Read data.
InString = MSCComm1.Input
End If
```

Для вывода информации с помощью элемента управления MSCComm1 используется метод MSCComm1.Output:

```
Dim Buffer as Variant
' Set and open port
MSCComm1.CommPort = 1
MSCComm1.PortOpen = True
Buffer = Chr$(KeyAscii)
MSCComm1.Output = Buffer
```

В скетче Arduino UNO для обеспечения обмена данными между ПК и микропроцессором используются следующие команды:

Функция `Serial.begin()` иницирует последовательное соединение и задает скорость передачи данных в бит/с (бод). При соединении через порты входа/выхода 0 и 1 могут быть использованы разные значения скорости, требуемые устройством, с которым будет осуществляться обмен данными.

Функция `Serial.available()` получает количество байтов (символов), доступных для чтения из последовательного интерфейса связи. Это те байты, которые уже поступили и записаны в буфер последовательного порта. Буфер может хранить до 128 байтов.

Функция `Serial.read()` считывает очередной доступный байт из буфера последовательного соединения.

Функция Serial.println() передает данные через последовательный порт как текст ASCII вместе со следующим за ним символом переноса строки (символ ASCII 13 или '\r') и символом новой строки (ASCII 10 или '\n').

Пример использования вышеперечисленных функций:

```
int in=0; // переменная для хранения полученного байта
void setup() {
    Serial.begin(9600); // устанавливаем последовательное соединение
}
void loop()
{
    if (Serial.available() > 0) // если есть доступные данные
    {
        // считываем байт
        in = Serial.read();
        // отсылаем то, что получили
        Serial.println(in,DEC);
    }
}
```

Выводы

В результате проведенных исследований разработан метод доступа управления параметрами шагового двигателя с ПК по протоколу UART. Благодаря этому любой пользователь (даже незнакомый с программированием микропроцессоров и шаговых двигателей) может управлять основными параметрами работы шагового двигателя: скоростью вращения, направлением и величиной шага двигателя. Разработанный программно-технический комплекс продемонстрировал на практике простоту и надежность управления шаговым двигателем с ПК по протоколу UART с помощью ПК.

ЛИТЕРАТУРА

1. Массимо Банци Arduino для начинающих волшебников / Массимо Банци – М. : Рид Групп, 2012. – 128 с.
2. Стюарт Болл Р. Аналоговые интерфейсы микроконтроллеров / Р. Стюарт Болл – М.: Издательский дом «Додэка XXI», 2007. – 360 с.
3. Петин В. Arduino и Raspberry Pi в проектах Internet of Things. / В. Петин – СПб. : БХВ-Петербург, 2016. – 320 с.

4. Карвінен Т. Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi. / Т. Карвінен, К. Карвінен, В. Валтокари – М. : Вильямс, 2015. – 448 с.
5. Блум Дж. Изучаем Arduino. Инструменты и методы технического волшебства / Дж. Блум – СПб. : БХВ-Петербург, 2015. – 336 с.
6. Таранушенко С. Arduino, датчики и сети для связи устройств / С. Таранушенко – СПб. : БХВ-Петербург, 2015. – 544 с.
7. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino / У. Соммер – СПб. : БХВ-Петербург, 2015. – 256 с.
8. Офіційна документація проекту Arduino [Електронний ресурс]. Режим доступу: <http://www.arduino.cc>
9. Авторські матеріали з сайту «Паяльник» [Електронний ресурс]. Режим доступу: <http://www.cxem.net>

Мікропроцесорна система управління кроковим двигуном за допомогою протоколу UART

Бакалін А. Р., Миронець В. В., Голубєв Л. П.

Київський національний університет технологій та дизайну

У статті розглядаються питання мікропроцесорного управління кроковим двигуном за протоколом UART. Описана розроблена система управління кроковим двигуном по командам, що видаються з персонального комп'ютера.

Ключові слова: кроковий двигун, мікропроцесор, автоматизація, система управління, UART

Microprocessor systems control the stepper motor using UART Protocol

Bakalin A. R., Myronec V. V., Golubev L. P.

Kiev National University of Technology and Design

The article deals with the microprocessor control stepper motor for UART protocol. The developed control system for stepper motor commands issued from the personal computer.

Keywords: stepping motor, a microprocessor, automation, system management, UART