

УДК: 004.414.28:371.694

РАЗРАБОТКА УЧЕБНО-ДЕМОНСТРАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ МАКЕТОМ ГРАФОПОСТРОИТЕЛЯ

Статья поступила в редакцию 22.01.2016, в окончательном варианте 01.03.2016.

Боскебеев Калычбек Джэетмишбаевич, кандидат технических наук, доцент, Кыргызский государственный технический университет им. И. Раззакова, 720044, Кыргызская Республика, г. Бишкек, пр. Мира, 66, e-mail: kboskebeev@mail.ru

Акматалиева Жазгул Зарылбековна, аспирант, Кыргызский государственный технический университет им. И. Раззакова, 720044, Кыргызская Республика, г. Бишкек, пр. Мира, 66, e-mail: zakmatalieva@list.ru

Алимсеитова Жулдыз Кенесхановна, аспирант, Кыргызский государственный технический университет им. И. Раззакова, 720044, Кыргызская Республика, г. Бишкек, пр. Мира, 66, e-mail: zhuldyz_al@mail.ru

Обоснована целесообразность использования в учебном процессе студентов инженерно-технических специальностей вузов учебно-демонстрационной системы (УДС), имитирующей управление графопостроителями, катерами, головками станков с числовым программным управлением (ЧПУ) и аналогичными устройствами. Охарактеризованы некоторые теоретические вопросы управления исполнительными элементами указанных устройств. Описана структура G-кода, используемого в рамках такого управления. Для представленной в статье УДС подробно рассмотрены следующие вопросы: формирование G-кода для использования на макете графопостроителя или станке с ЧПУ; процесс обработки этого кода на ПЭВМ с наглядным отображением хода выполнения отдельных операций (действий); обеспечение параллельного воспроизведения этих операций на макете графопостроителя; технические решения в отношении механической и электрической частей макета графопостроителя. Сделан вывод о том, что использование предлагаемой УДС обеспечивает необходимую эффективность и наглядность учебного процесса, в том числе в рамках применения активных форм обучения.

Ключевые слова: математическая модель, алгоритм управления, фигура, рисование, перо, технологический процесс, программа, G-код, координаты точек, шаговый двигатель, макет графопостроителя, технические решения

DEVELOPMENT OF EDUCATIONAL-DEMONSTRATION SYSTEM MANagements LAYOUT OF PLOTTER

Boskebeev Kalychbek Dzh., Ph.D. (Engineering), Associate Professor, Kyrgyz Technical University named after I. Razzakov, 66 Mir Ave., Bishkek, 720044, Kyrgyz Republic, e-mail: kboskebeev@mail.ru

Akmatolieva Jazgul Z., post-graduate student, Kyrgyz Technical University named after I. Razzakov, 66 Mir Ave., Bishkek, 720044, Kyrgyz Republic, e-mail: zakmatalieva@list.ru

Alimseyitova Juldyz K., post-graduate student, Kyrgyz Technical University named after I. Razzakov, 66 Mir Ave., Bishkek, 720044, Kyrgyz Republic, e-mail: zhuldyz_al@mail.ru

Expediency is reasonable of the use in the educational process of students of engineer-technical specialties of universities learning of the educational-demonstration system (EDS) of imitating management plotters, cutters, heads of machine-tools with numerical control (MTwNC) and analogical devices. Some theoretical questions of management are described by the executive elements of the indicated devices. The structure of G-code used within the framework of such management is described. For presented in the article of EDS in detail next questions are considered: forming of G-code for the use on the layout of plotter or machine-tool with MTwNC; process of treatment of this code on PC with the evident reflection of course of performance of separate operations (actions); providing of the parallel reproducing of these operations on the layout of plotter; technical decisions in regard to mechanical and electric parts of layout of plotter. Drawn conclusion that the use of offered EDS provides necessary efficiency and evidentness of educational process, including with the use of active forms of educating.

Keywords: mathematical model, algorithm of managements, figure, drawing, feather, technological process, program, G-code, coordinates of points, foot-pace engine, layout of plotter, technical decisions

Введение. Обеспечение эффективности обучения студентов многих инженерно-технических специальностей (ИТС) вузов в настоящее время требует ознакомления их с принципами работы ряда периферийных устройств ЭВМ, станков с числовым программным управлением (ЧПУ) и др. Процедуры такого обучения затрудняются следующими обстоятельствами: высокой стоимостью таких устройств; отсутствием наглядности процессов выработки (формирования), обработки и фактической реализации команд управления (G-кодов) в таких устройствах. Как следствие, применение современных образцов такой техники в рамках учебного процесса оказывается нецелесообразным. Поэтому целью данной статьи является представление разработанной учебно-демонстрационной системы (УДС) по управлению указанными устройствами и особенностей (возможностей) использования этой УДС в учебном процессе студентов ИТС. При этом в данной статье графопостроитель взят в качестве управляемого устройства в силу наглядности его использования. Аналогичное по своему содержанию управление осуществляется и в режущих плоттерах (катерах), которые сейчас широко применяются в рекламном деле.

Опыт использования рассматриваемых классов устройств в Кыргызской Республике. С 1980 по 2000 г. на машиностроительном заводе им. Ленина (г. Фрунзе, ныне г. Бишкек) инженеры-технологи, работавшие в сфере машиностроения, разрабатывали программы для станков с ЧПУ «Fanuc» (обрабатывающие центры типов IP-500 и IP-800). Указанные ЧПУ использовались для выполнения следующих технологических операций: фрезерование; токарная обработка; шлифование; перфорирование; укладка; сортировка; позиционирование; распиловка; прессование. Управляющие программы создавали на G-коде и отлаживали их в САПР, работающей на базе ЭВМ СМ-1420. На ней использовалась операционная система реального времени (ОС РВ), объем оперативной памяти составлял 256 Кбайт. К этой ЭВМ был подключен графопостроитель, на котором отлаживался G-код, а после завершения отладки программа выводилась на перфоленгу. В сборочном цехе эта перфоленга считывалась в ЧПУ Fanuc, которая управляла исполнительными механизмами станка через шкаф устройства связи с оборудованием (реле, автоматы, блок питания на 3 фазы, амперметры, вольтметры и др.).

В настоящее время описанные технологии не используются – в силу их нерентабельности. Однако частные предприниматели в Кыргызской Республике используют станки с ЧПУ (деревообрабатывающие, лазерные и пр.). Поэтому рынок республики требует квалифицированных специалистов, способных обслуживать такие устройства, разрабатывать соответствующие программы на G-коде. Поэтому представляется целесообразной разработка и использование описываемой в статье УДС, которая может быть использована при обучении в наглядной форме бакалавров, магистров и аспирантов.

Характеристика базовых технических решений, использованных в УДС. В качестве привода «исполнительной части» УДС применен шаговый двигатель (ШД), который является наиболее простым по эксплуатационным характеристикам. В качестве датчиков обратных связей использованы концевые выключатели, установленные на позиции, соответствующей установке пера графопостроителя в начало координат. Управление пером (подъем, опускание) осуществляется электромагнитом с помощью программы на G-коде.

Шаговый двигатель поворачивается за каждый импульс управления на определенный фиксированный угол. Чем меньше шаг, тем большая точность вращения может быть достигнута. Современные ШД могут обеспечить маленький шаг (до 1000 шагов за один оборот). Частота вращения задается числом импульсов (шагов) в секунду. Поэтому вместо показателя «частота вращения» часто используется термин «шаговая частота вращения». Соотношение между частотой вращения и шаговой частотой вращения задается формулой (1):

$$\omega = (60f) / K, \quad (1)$$

где ω – частота вращения (об./мин.); f – шаговая частота вращения; K – число шагов ШД на один оборот.

Упрощенная блок-схема «исполнительной части» УДС представлена на рисунке 1. Электрические импульсы с LPT порта компьютера поступают в регулятор, сигнал из которого подается на вход электрической следящей системы, управляющей поворотом ротора ШД. Последний, в свою очередь, вращает винт. При этом гайка перемещается в направляющем элементе, изменяя положение пера (перемещаемого объекта) относительно основания по одной оси. Для второй (перпендикулярной) оси схема управления перемещениями пера аналогична.

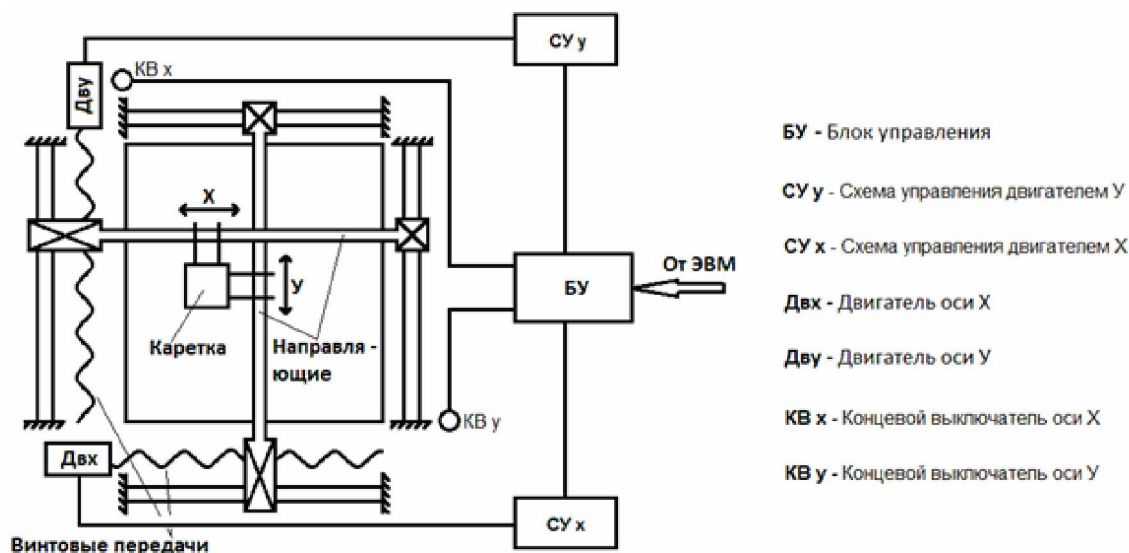


Рис. 1. Функциональная схема «исполнительной части» УДС

Блок управления (БУ) контролирует работу схем управления двигателями по каждой оси, а также принимает данные от концевых выключателей. Схемы управления (СУ) двигателями по осям (СУ_у, СУ_х) представляют собой драйверы ШД. Двигатели осей (Дв_х, Дв_у) приводят в движение соответствующие этим осям винтовые передачи. Концевые выключатели на схеме – это (КВ_х, КВ_у).

Математическая модель работы УДС. Построим такую модель с учетом характеристик электромеханического приводного механизма, включающего винтовую передачу и ШД. В работе [7] рассчитаны передаточные отношения по скоростным и силовым параметрам.

$$R_1 = 2\pi / (Sn); \quad R_2 = S / (2\pi\eta), \quad (2)$$

где R_1 – передаточное отношение (число) по скоростному параметру; R_2 – передаточное отношение (число) по силовому параметру; S – шаг резьбы, n – число заходов резьбы, η – коэффициент полезного действия ШД и винтовой пары.

В винтовой паре модель передачи скоростей и момента описывается такими выражениями [7]:

$$V = R_1 * \dot{z}; \quad M = R_2 * F; \quad \dot{x} = \dot{g} + \dot{z}, \quad (3)$$

где V – скорость вращения якоря ШД; M – вращающий момент на гайке; $F = m\ddot{x}$ – сила сопротивления; m – масса платформы; x – абсолютное перемещение объекта (координата); g – абсолютное перемещение основания; z – относительное перемещение объекта.

Момент, приложенный к валу ШД, и усилие, создаваемое поступательным движением объекта, определяются зависимостью по формуле (4):

$$M = (1/2)FDtg(\Psi + \rho), \quad (4)$$

где D – диаметр окружности, на которой располагаются центры ШД; Ψ – угол подъема винтовой линии по цилиндру диаметром D ; ρ – приведенный угол трения качения. Для управления перемещением исполнительных механизмов УДС целесообразно использовать алгоритм Брезенхэма [5, 6, 10], который обеспечивает необходимую точность перемещения пера. При проектировании УДС мы применяли теорию систем и разнообразные графические модели, методы и средства моделирования [1–4]. Предлагаемые методы управления исполнительными механизмами имеют единую закономерность или общие теоретические основания [9, 10]. Каждая из моделей может рассматриваться как абстрактный формализованный объект, причем они опираются на строгий математический базис. Общепринято, что любое устройство (система) подобного рода должно проектироваться с трех точек зрения: во-первых – его состояния; во-вторых – действия; в третьих – его поведения (управление и контроль) [4, 5].

Каждое представление может рассматриваться как некоторый тип или аспект рассмотрения понятия системы. В качестве инструмента изучения используется известная математическая модель, которая позволяет теоретически объяснить перемещения пера макета графопостроителя между двумя заданными точками (рис. 2) по известному алгоритму Брезенхэма [10].

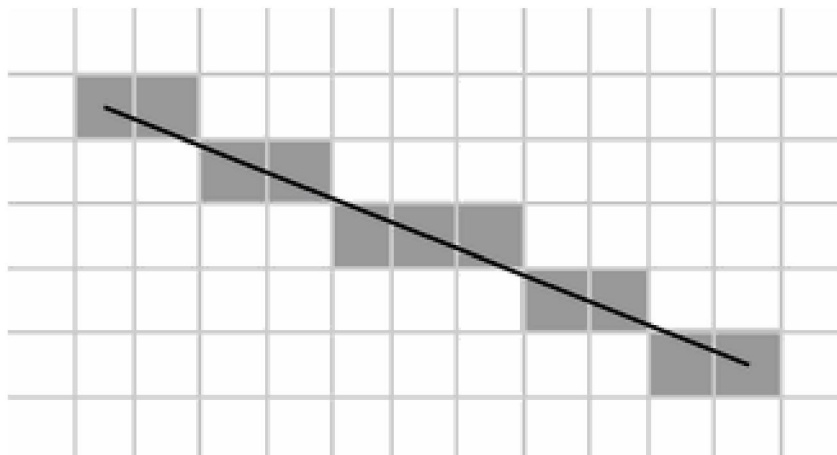


Рис. 2. Иллюстрация перемещения пера по алгоритму Брезенхэма

При смене направления вращения ШД возникает люфт, который описан (учтен) в блок-схеме (рис. 3).

Формула, описывающая линию, соединяющую две точки, взята из [10]:

$$y - y_0 = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0). \quad (5)$$

Из (5) получается рабочая формула:

$$y = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0) + y_0. \quad (6)$$

Линия проводится между двумя точками – соответственно матрице номеров (они увеличиваются вправо и вниз – по рисунку 2). Этот алгоритм подробно представлен на рисунке 3.

Формирование G-кода при проектировании. Для формирования G-кода используем программу Sprint-Layout 6.0. Например, в этой программе нарисуем многоугольник и экспортируем его в текстовый файл с G-кодом «многоугольник.BPR». Он содержит последовательные координаты точек по осям X и Y этого многоугольника (рис. 4).

Первая строка соответствует координатам точки по оси X = 104,06 мм, по оси Y = 74,60 мм. При DO2 перо поднято, а при DO1 перо опущено. Каждая запись содержит координаты точек по осям X и Y. Совокупность этих записей используется для перемещения пера макета между заданными точками. Этот G-код является входным файлом для программы на языке Pascal (в среде Delphi7), которую авторы разработали для управления ШД макета графопостроителя. Студенты могут формировать G-код в программе Sprint-Layout 6.0. и сохранять его в файлах на флеш-накопителе. Затем на лабораторных занятиях с помощью УДС они смогут проверить с помощью своих файлов как работает реальный макет графопостроителя.

Обработка G-кода в рамках использования УДС. Драйвер (программа) управления УДС написана на языке Паскаль, так как ее легко отлаживать в любой операционной системе (Windows 98, Windows XP, Windows 7, Windows 8). Мы используем ШД с четырьмя обмотками, следовательно, подключаем четыре транзистора в цепи управления. Для управления ШД требуется программная подача импульса из компьютера через LPT порт на базу соответствующего транзистора для перемещения исполнительного механизма на заданный отрезок. Транзистор открывается, и ток протекает от эмиттера в коллектор через обмотку ШД. При этом ротор ШД вращается до тех пор, пока не достигнет заданной координаты по нужной оси. Ниже приведена формула расчета длительности импульса. Она получена экспериментально для повышения скорости ШД (исполнительного механизма).

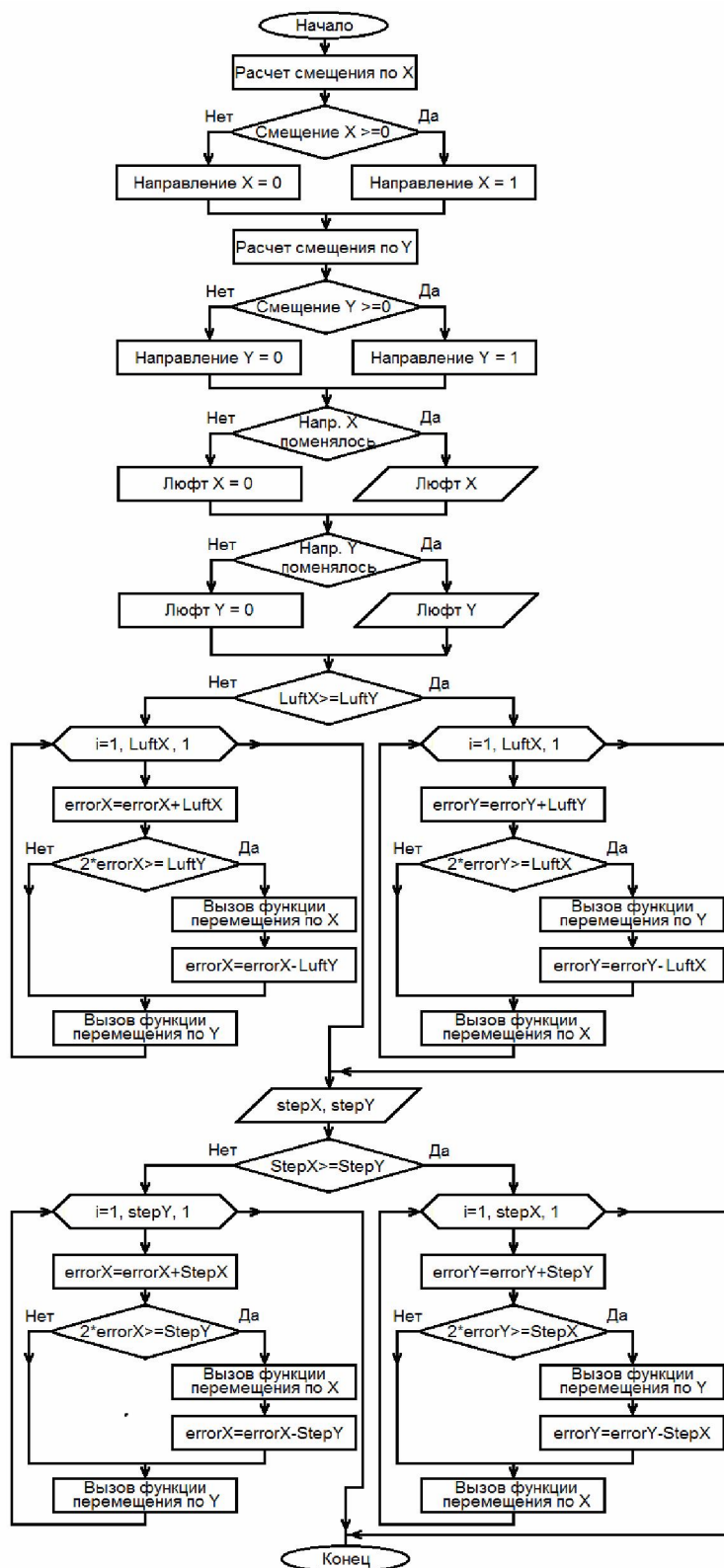


Рис. 3. Блок схема реализации алгоритма Брезенхема

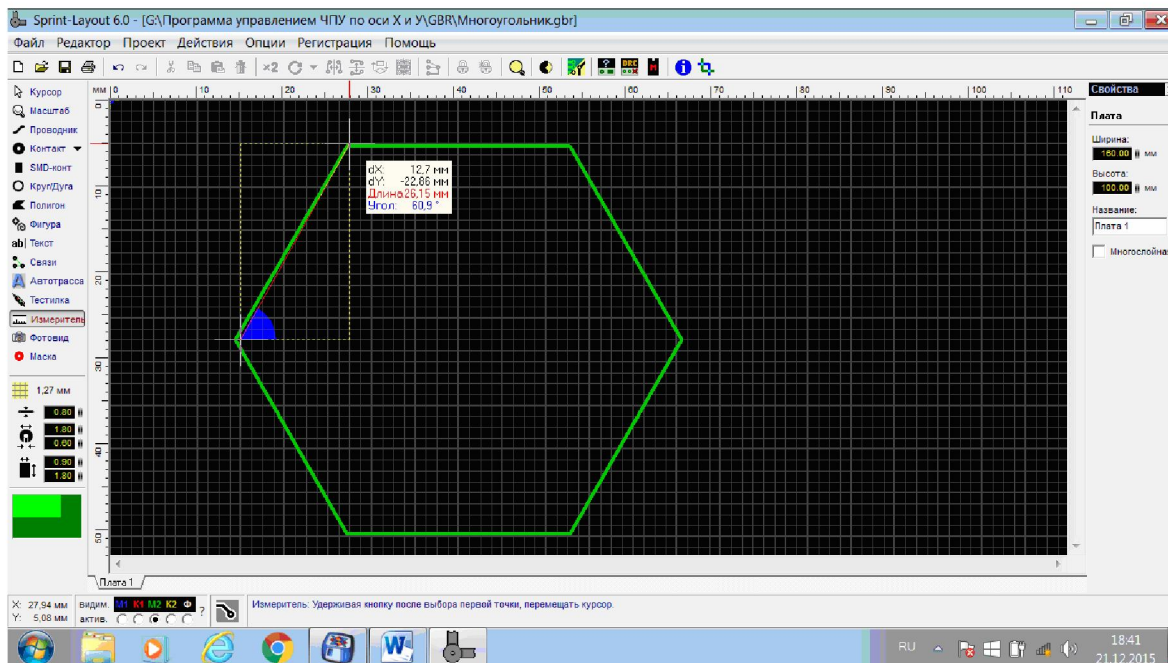


Рис. 4. Диалоговое окно построения шестиугольника в программе Sprint-Layout 6.0

Ниже приведен фрагмент текстового файла, отражающий содержание файла «многоугольник.BPR» (рис. 5).

```
X10406Y7460D02* <OK>
X11806Y9885D01* <OK>
X14606Y9885D01* <OK>
X16006Y7460D01* <OK>
X14606Y5035D01* <OK>
X11806Y5035D01* <OK>
X10406Y7460D01* <OK>M02*
```

Рис. 5. Листинг G-кода для вычерчивания многоугольника

Длительность импульса задержки T для исполнительного механизма определяем по формуле (7):

$$T = \frac{t}{(360/S)V}, \quad (7)$$

где t – время (минута, секунда, миллисекунда); S – шаг ШД (градусы); V – скорость подачи ШД (минута/ миллиметр).

Перемещение (смещение) по резьбе винта (C) в миллиметрах определяем по формуле (8):

$$C = \frac{HR}{(360/S)}, \quad (8)$$

где HR – шаг резьбы винта (в мм).

В описываемой программе сначала открывается входной текстовый файл «многоугольник.BPR». В левом верхнем углу окна программы (рис. 6) размещен текстовый файл «многоугольник.BPR». Программа определяет координаты двух точек и рассчитывает перемещение курсора на экране по модели Брезенхема (см. выше).

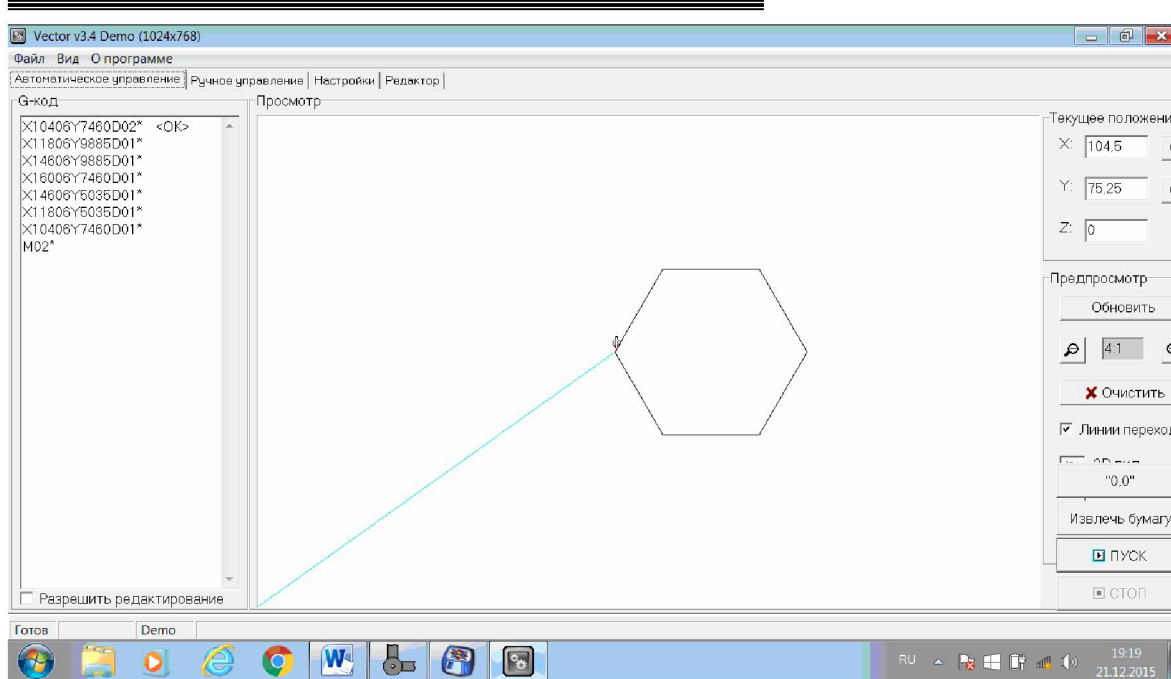


Рис. 6. Диалоговое окно, отражающее перемещение курсора в первую точку

Приведем фрагмент кода программы на языке Pascal для считывания входного текстового файла «многоугольник.BPR» (рис. 7).

```

image1.Canvas.MoveTo(0, (image1.Height-1)); for i:=0 to (memo1.Lines.Count-1) do begin s:=memo1.Lines[i];
if s="" then s:='0';
//Если пустая строка чтобы не было ошибки
if Pos('<', memo1.Lines[i])<>0 then goto konec;
//Проверка чтобы не рисовать уже отрисованные строки
case s[1] of 'X': begin
//Начали обработку (перевод) G-кода в координаты
x:=s; y:=s;
delete(x, Pos('Y', x), length(s)); delete(x, 1, 1); delete(y, 1, length(x)+2);
d:=y; delete(y, Pos('D', y), length(s));
delete(d, 1, length(y)+2); delete(d, Pos('*', d), 1);
//Закончили обработку G-кода
if SpeedButton3D.Down=false then
begin if d='1' then //Рисуем линию черным
begin image1.Canvas.Pen.Color:=clBlack;
image1.Canvas.LineTo(round(strToFloat(x)/100*masshtab),
round(strToFloat(y)/100*masshtab)+image1.Height-1);
end; if d='2' then //Линия перехода - рисуем голубым
begin if CheckBoxLine.Checked=true then
begin image1.Canvas.Pen.Color:=clAqua;
image1.Canvas.LineTo(round(strToFloat(x)/100*masshtab),
-round(strToFloat(y)/100*masshtab)+image1.Height-1);
else
End
begin
image1.Canvas.moveTo(round(strToFloat(x)/100*masshtab),
-round(strToFloat(y)/100*masshtab)+image1.Height-1);
end; end; end else
begin if d='1' then //Рисуем линию черным

```

```

begin      image1.Canvas.Pen.Color:=clBlack;      im-
age1.Canvas.LineTo(round(strToFloat(x)/100*masshtab)+round(strToFloat(y)/200*masshtab),
image1.Height-1-0-round(strToFloat(y)/200*masshtab));
end;      if d='2' then //Линия перехода - рисуем голубым
begin      if CheckBoxLine.Checked=true then
begin      image1.Canvas.Pen.Color:=clAqua;
mage1.Canvas.LineTo(round(strToFloat(x)/100*masshtab)+round(strToFloat(y)/200*masshtab),
image1.Height-1-0-round(strToFloat(y)/200*masshtab)); end      else
begin      im-
age1.Canvas.MoveTo(round(strToFloat(x)/100*masshtab)+round(strToFloat(y)/200*masshtab), image1.Height-1-0-
round(strToFloat(y)/200*masshtab)); end; end; end; end;
'D'; 'G'; end; konec;; end; end;

```

Рис. 7. Фрагмент кода программы для считывания исходных данных и отрисовки объекта на экране ПЭВМ

При выполнении второй записи программы G – кода, курсор перемещается в третью точку (рис. 8).

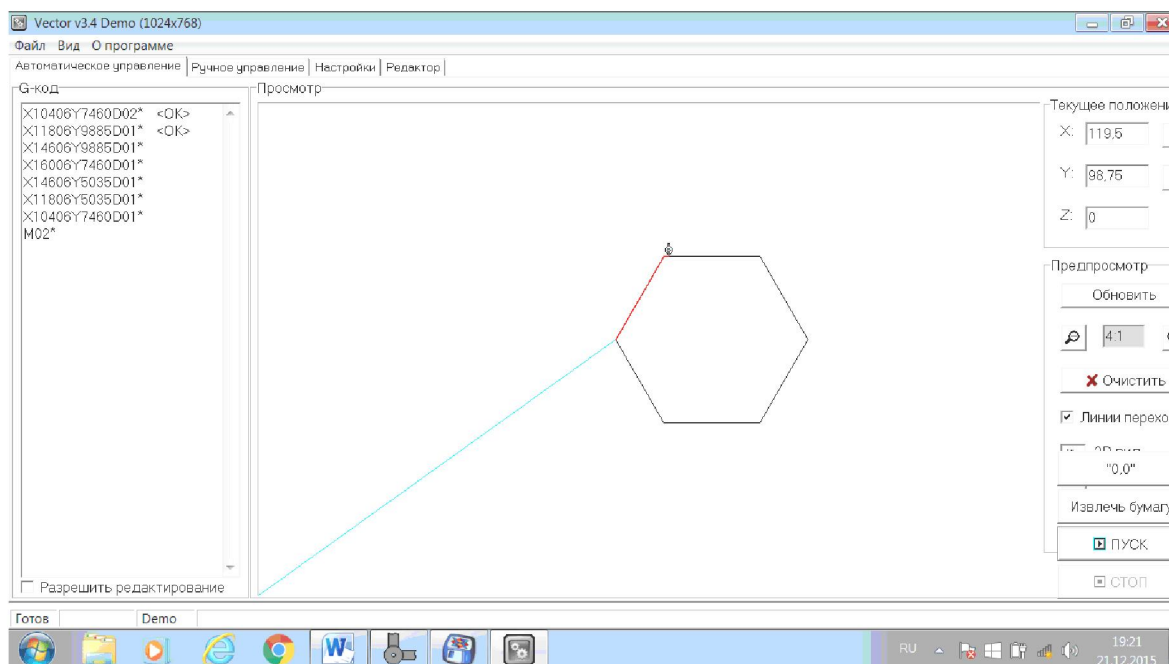


Рис. 8. Диалоговое окно, представляющее ход выполнения G-кода

Ниже приведен код программы на языке Pascal (рис. 9) для управления пером макета графопостроителя, включая краткие комментарии к выполняемым действиям.

```

image1.Canvas.Pen.Color:= clRed;
if SpeedButton3D.Down = false then
begin
//Если инструмент опущен - рисовать красным
if PosZ=0 then
begin
image1.Canvas.LineTo(round(PosX*masshtab),-round(PosY*masshtab)+image1.Height-1);
end else //Если инструмент поднят - не рисовать
begin
image1.Canvas.MoveTo(round(PosX*masshtab),-round(PosY*masshtab)+image1.Height-1); end; im-
age2.Left:=round(PosX*masshtab)-14;
image2.Top:=-round(PosY*masshtab)+image1.Height-1-14;

```



```
end else begin
//Если инструмент опущен - рисовать красным
if PosZ=0 then begin image1.Canvas.LineTo(round(PosX*masshtab)+round(PosY/2*masshtab),
0-round(PosY/2*masshtab)+image1.Height-1); End else
//Если инструмент поднят - не рисовать
Begin image1.Canvas.MoveTo(round(PosX*masshtab)+round(PosY/2*masshtab),
0-round(PosY/2*masshtab)+image1.Height-1); end;
image2.Left:=round(PosX*masshtab)+round(PosY/2*masshtab)-14;
image2.Top:=-round(PosY/2*masshtab)+image1.Height-1-14; end; end;end;
```

Рис. 9. Фрагмент кода программы для управления движением пера графопостроителя

Этот прототип программы, можно использовать для отладки G-кода на УДС, включающей макет графопостроителя – в т.ч. и в режиме пошагового выполнения, когда можно наблюдать за перемещениями «пера» в реальном масштабе времени. Отметим, что в процессе занятий студенты могут менять G-код в текстовом редакторе и смотреть, как это повлияет на работу макета.

Механическая часть конструкции УДС. В конструктивном плане УДС представляет собой двухкоординатную платформу с рабочим полем размерами 210 x 297 мм (соответствует стандартному листу формата А4). Ось X представляет собой портал, перемещающийся по двум параллельным линейным направляющим диаметром 12 мм. Длина направляющих – 445 мм. Ось Y – каретка, перемещающаяся по двум параллельным линейным направляющим диаметром 8 мм, установленным на портале. Направляющие оси X неподвижно закреплены между двух отрезков стального профиля с прямоугольным сечением и длиной 270 мм. Рабочий стол, выполненный из ламинированного ДСП, закреплен под каркасом оси X с помощью винтов. Размер стола – 470 x 210 мм. Перемещение портала и каретки осуществляется с помощью ходовых винтов М6 с шагом резьбы 1 мм. На портале и каретке в свою очередь имеются бронзовые гайки с резьбой соответствующего размера. Ходовые винты приводятся в движение шаговыми двигателями. На каретке оси Y установлен механизм подачи инструмента, представляющий собой электромагнит с креплением для рабочего инструмента. Внешний вид макета графопостроителя приведен на рисунках 10 и 11.



Рис. 10. Общий вид макета графопостроителя

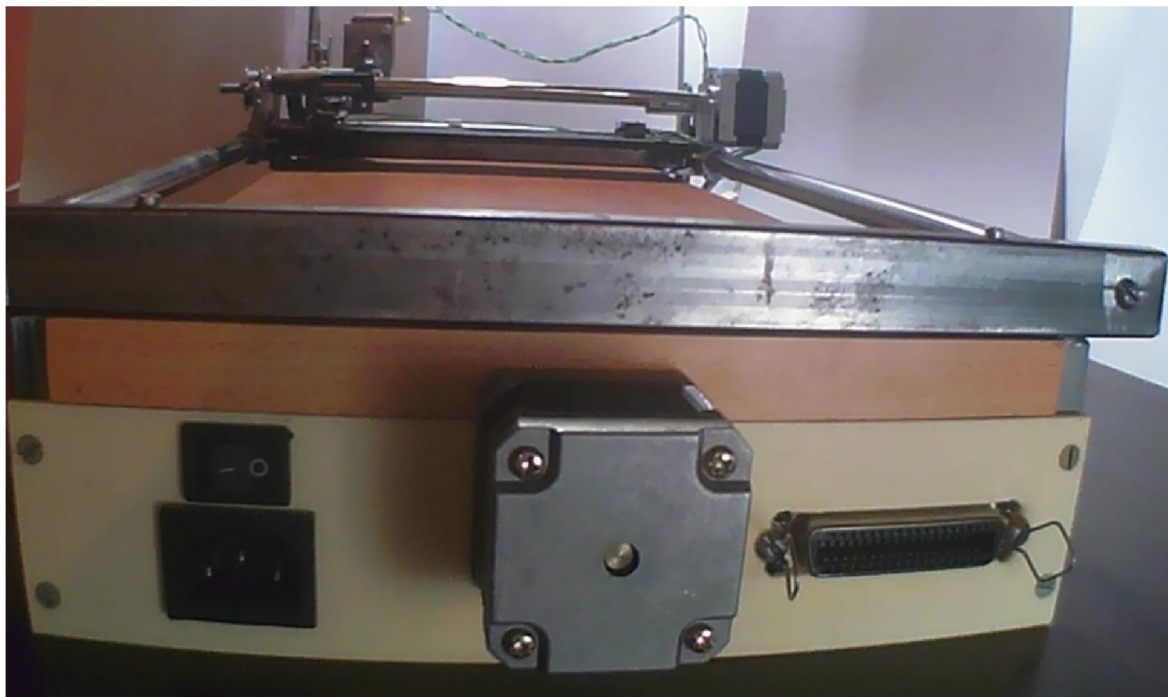


Рис. 11. Задняя панель макета графопостроителя

Для установки рабочего инструмента в координаты (0, 0) на платформе смонтированы концевые выключатели механического типа. При сборке механической части УДС были использованы детали от механического принтера EPSON снятого с эксплуатации. Это позволило избежать расходов на изготовление нужных частей макета по специальному заказу.

Принципиальная электрическая схема УДС. Электронная часть УДС и сетевой источник питания закреплены снизу рабочего стола. На заднюю панель платформы выведено следующее: разъём для подключения сетевого шнура; выключатель питания; 36-контактный разъём Centronics для подключения к LPT порту компьютера. Разработанный макет графопостроителя выполнен с использованием достаточной старой ПЭВМ, имеющей выходной LPT порт. Эту ПЭВМ в силу ее технических характеристик уже нельзя использовать для работы с современными программами САПР, поэтому такое продление ее «жизненного цикла» является вполне оправданным.

Выбранная для реализации принципиальная электрическая схема [8] приведена на рисунке 12. Основные функции в блоке управления выполняет микроконтроллер Atmega8A (U2). На выходы 14, 15 с выводов 2, 3 порта LPT поступают сигналы «Шаг» и «Направление» по оси X. С выводов 4, 5 на выходы 16, 17 – аналогичные сигналы по оси Y. Через выходы микроконтроллера 2, 3, 4, 5, 6, 11, 12, 13 осуществляется управление ключевой схемой, выполненной на транзисторах VT1–VT8.

Резисторы R1–R8 служат для ограничения тока базы транзисторов. В коллекторную цепь транзисторов включены обмотки шаговых двигателей M1 и M2: VT1–VT4 для первого двигателя; VT5–VT8 – для второго.

Средние точки обмоток двигателей подключены к плюсу источника питания. Поскольку нагрузка индуктивная, то в схеме применены защитные диоды VD1–VD8. Эти диоды увеличивают время спада тока в обмотках ШД. Через них, последовательно с ограничительным резистором R10, ЭДС самоиндукции «возвращается» в источник питания +5 В.

Такое решение снижает максимально возможную частоту вращения ШД. Если убрать диоды из схемы, то время спада тока окажется минимальным и двигатель сможет вращаться быстрее. Однако при этом возникают опасные для транзисторов броски ЭДС самоиндукции, которые могут превысить допустимое для них напряжение 60 В и, как следствие, привести к повреждению транзисторов.

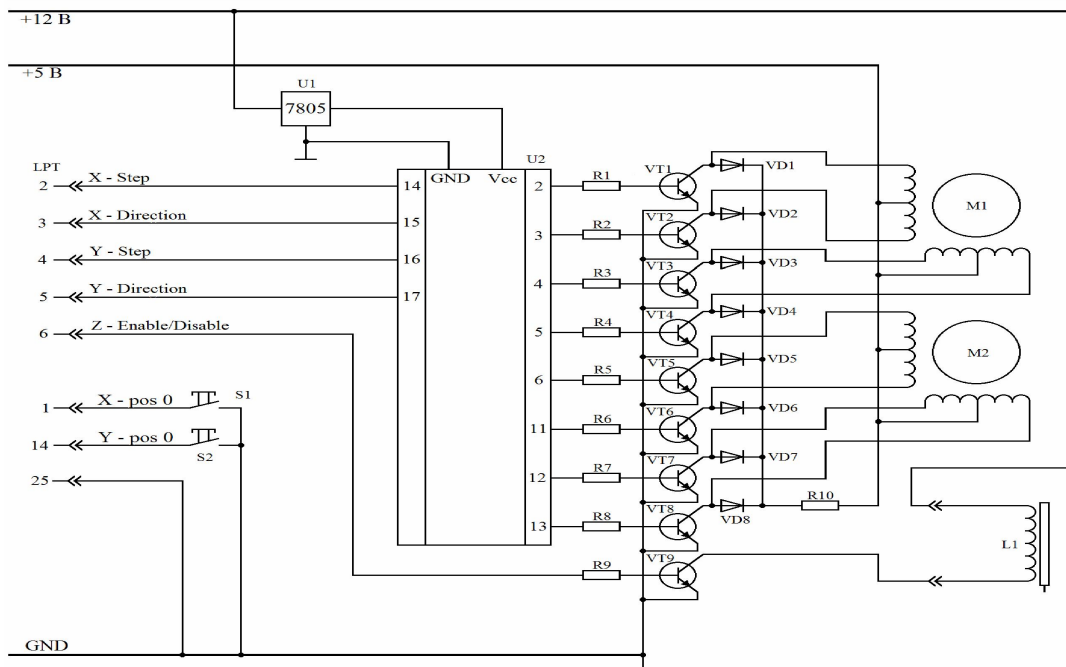


Рис. 12. Принципиальная электрическая схема макета УДС

Для обеспечения стабильной работы микроконтроллера в цепи питания установлен линейный стабилизатор напряжения LM 7805 (U1). Включение и выключение электромагнитного привода подачи инструмента (L1) осуществляется с помощью транзистора VT9, база которого через резистор R9 подключена к 6-ому выводу порта LPT. Этот узел питается от отдельного источника питания +12 В. Концевые выключатели S1 и S2, смонтированные на платформе, подключены к выводам 1 и 14 порта LPT. Вывод 25 LPT подключен к общей точке (GND), т.е. к «земле».

Приведенная электрическая схема достаточно проста для понимания студентами, допускает имитационное моделирование ее работы в программах создания принципиальных электрических схем – это важно с точки зрения ее исследования в учебных целях.

Выводы. 1. Предложен подход, обеспечивающий получение адекватной модели рисования фигуры на макете графопостроителя УДС. 2. Созданный макет позволяет наглядно объяснять студентам разных специальностей принципы аппаратно-программного управления рассматриваемым классом периферийных устройств. 3. При этом в рамках учебного процесса удастся наглядно показать функциональные взаимосвязи таких объектов: математической модели; алгоритма вычислений; методов программирования на языке Паскаль в Delphi7.

Список литературы

1. Батырканов Ж. И. Использование фреймово-продукционной модели представления знаний в системе управления предприятием / Ж. И. Батырканов, К. Д. Боскебеев // Прикаспийский журнал: управление и высокие технологии. – 2015. – № 1 (29). – С. 100–112.
2. Боскебеев К. Д. Модель интеллектуальной обучающей системы на основе теории систем / К. Д. Боскебеев // Материалы Международной научно-практической конференции, посвященной 70-летию образования Волгоградского государственного аграрного университета. – Волгоград, 28 января 2014. – С. 331–336.
3. Боскебеев К. Д. Систематизация базы знаний в информационных системах / К. Д. Боскебеев // Вестник Таджикского государственного университета политики и бизнеса. – 2014. – № 2 (58). – С. 255–263.
4. Брумштейн Ю. М. Сравнительный анализ функциональности программных средств управления проектами, расширяемых по модели SAAS / Ю. М. Брумштейн, И. А. Дюдиков // Прикаспийский журнал: управление и высокие технологии. – 2014. – № 4 (28). – С. 34–51.
5. Кормен Т. Алгоритмы. Построение и анализ / Т. Кормен. – Москва : Московский центр непрерывного математического образования, 2001. – 960 с.

6. Роджерс Д. Алгоритмические основы машинной графики / Д. Роджерс. – Москва : Мир, 1989. – С. 54–63.
7. Рыбак Л. А. Моделирование электромеханического привода с гибридным шаговым двигателем роботизированной платформы / Л. А. Рыбак, Н. Н. Черкашин, А. А. Гунькин, А. В. Чичварин // Современные проблемы науки и образования. – 2014. – № 6.
8. Тимченко Ю. Н. Доработка схемы управления шаговым двигателем / Ю. Н. Тимченко // Международный научно-исследовательский журнал. – 2013. – № 7–2(14).
9. Цой Ю. Р. О математических моделях эволюционных алгоритмов / Ю. Р. Цой // Перспективные информационные технологии и системы. – 2006. – № 2. – С. 42–47.
10. Bresenham J. E. Algorithm for computer control of a digital plotter / J. E. Bresenham // IBM Systems Journal 4. – 1965. – P. 25–30.

References

1. Batyrkanov J. I., Boskebeev K. D. Ispolzovanie freymovo-produktsionnoy modeli predstavleniya znaniy v sisteme upravleniya predpriyatiem [Use the frame and productional model of representation of knowledge in control system by an enterprise]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Management and High Technologies], 2015, no. 1 (29), pp. 100–112.
2. Boskebeev K. D. Model intellektualnoy obuchayushchey sistemy na osnove teorii sistem [Model of the intellectual teaching system on the basis of theory of the systems]. *Materialy Mezhdunarodnoy nauchno-prakticheskoy konferentsii, posvyashchennoy 70-letiyu obrazovaniya Volgogradskogo gosudarstvennoy agramogo universiteta* [Proceedings of the International Scientific and Practical Conference devoted 70 years of the Formation of the Volgograd State Agricultural University], Volgograd, January, 28, 2014, pp. 331–336.
3. Boskebeev K. D. Sistematzatsiya bazy znaniy v informatsionnykh sistemakh [Systematization of base of knowledge is in the informative systems]. *Vestnik Tadzhijskogo gosudarstvennogo universiteta politiki i biznesa* [Bulletin of the Tadjik State University of Politics and Business], 2014, no. 2 (58), pp. 255–263.
4. Brumshteyn U. M., Diudikov I. A. Cravnitelnyy analiz funktsionalnosti programmnykh sredstv upravleniya proektami, rasprostranyaemykh po modeli SAAS [Analysis of comparative of functionality of programmatic facilities of management by projects, expandable on the model of SAAS]. *Prikaspiyskiy zhurnal: upravlenie i vysokie tekhnologii* [Caspian Journal: Management and High Technologies], 2014, no. 4 (28), pp. 34–51.
5. Kormen T. *Algoritmy. Postroenie i analiz* [Algorithms. Construction and analysis], Moscow, Moscow Center for Continuous Mathematical Education Publ. House, 2001. 960 p.
6. Rodjers D. *Algoritmicheskie osnovy mashinnoy grafiki* [Algorithmic bases of machine graphic arts], Moscow, Mir Publ., 1989, pp. 54–63.
7. Rybak L. A., Vherkashin N. N., Gunkin A. A., Chichvarin A. V. Modelirovanie elektromekhanicheskogo privoda s gibridnym shagovym dvigatelem robotizirovannoy platformy [Design of electromechanics drive with the hybrid foot-pace engine of robotized platform]. *Sovremennye problemy nauki i obrazovaniya* [Modern Problems of Science and Education], 2014, no. 6.
8. Timchenko U. N. Dorabotka skhemy upravleniya shagovym dvigatelem [Revision of management chart by a foot-pace engine]. *Mezhdunarodnyy nauchno-issledovatel'skiy zhurnal* [International Scientific and Research Journal], 2013, no. 7–2 (14).
9. Soi U. R. O matematicheskikh modelyakh evolyutsionnykh algoritmov [About the mathematical models of evolutionary algorithms]. *Perspektivnye informatsionnye tekhnologii i sistemy* [Perspective Information Technologies and Systems], 2006, no. 2, pp. 42–47.
10. Bresenham J. E. Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4, 1965, pp. 25–30.