

УДК 004.42

НАСТРОЙКА І МОДИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СТАНКА З ЧПУ

Голубєв Л. П., Ківа І. Л., Рафальський О. С., Шинкаренко Ю. В.

Київський національний університет технологій та дизайну

Мета. Дослідження і модифікація програмного забезпечення верстата з ЧПУ.

Методика. Теоретичні дослідження проводилися на базі спеціальних розробок в теорії управління методом моделювання і теорії прийняття рішень. Дослідження проводилися на системі числового програмного керування, побудованого на базі персонального комп'ютера (PC) і контролера управління рухом робочих вузлів верстата.

Результати. Досліджено технологію роботи системи числового програмного керування і виконано модифікацію програмного і технічного забезпечення для поліпшення роботи системи.

Наукова новизна. Удосконалено технологію роботи на верстаті з ЧПУ, розроблено новий підхід до вирішення завдання критичної поведінки системи в граничних точках монтажного столу (робочої області) і повернення різця в нульову точку.

Практична значимість. Розроблений метод вирішує проблему некоректної поведінки системи в граничних областях монтажного столу і може бути застосований в багатьох системах з числовим програмним забезпеченням.

Ключові слова: верстат з числовим програмним управлінням, G-код, Arduino, GRBL-прошивка

Системи числового програмного керування (СЧПУ) призначені для автоматизації роботи верстатного обладнання та здійснення обробки за заданою програмою. Сучасні обробні комплекси мають значну технологічну гнучкість і універсальність багато в чому завдяки наявності СЧПУ [1, 4].

Намічається тенденція використовувати універсальне обладнання (замість вузькоспеціалізованого верстатного оснащення і роботів-автоматів) навіть при великосерійному виробництві, тому що можливість легкого перенастроювання виявляється більш вагомим плюсом, ніж підвищення вартості обладнання. Крім того, виготовлення складних профільованих деталей без використання багатокоординатної програмної обробки інколи виявляється неможливо в принципі.

Найбільш яскраві переваги систем ЧПУ є:

- більш висока продуктивність обладнання;
- поєднання універсальності та точності обробки;
- спрощення виробничого процесу (пов'язаного, однак, з деяким ускладненням підготовчого етапу – розробки керуючих програм);

- малий розкид якості виробів в межах однієї партії випуску;
- швидкість переналагодження обладнання та переходу до випуску інших виробів;
- простота оснащення верстатів;
- легкість обслуговування і експлуатації і т. д.

Верстати з ЧПУ сьогодні є переважаючими в сучасному виробництві. Тому питання дослідження і удосконалення технології роботи систем з числовим програмним управлінням є на сьогоднішній день особливо актуальними.

Постановка завдання

Незважаючи на загальний підхід до побудови систем з числовим програмним управлінням, однотипною їх структурою необхідно розглядати кожну систему з числовим програмним забезпеченням окремо, тому що вони мають свої особливості.

Нами була досліджена технологія роботи на фрезерному трьохосьовому верстаті з числовим програмним управлінням. Створено працюючий макет верстата, проведені дослідження особливостей його роботи та виявлено наступні проблеми в його роботі:

- некоректна поведінка верстата при роботі на кордонах монтажної області (якщо в керуючій програмі вказані координати, що перевищують реально можливі координати робочої області верстата – то в цьому випадку можлива навіть поломка крокового двигуна).

- необхідно модифікувати програмне забезпечення для забезпечення можливості повернення ріжучого інструменту в нульову точку, не залежно від його положення на робочій поверхні.

Результати досліджень

Архітектура систем ЧПУ

Принцип роботи систем ЧПУ полягає у видачі мікроконтролером керуючого впливу (електричних імпульсів строго певної тривалості) на виконавчі механізми верстата, а також контролю їх переміщення («зворотний зв'язок») для реалізації руху ріжучого інструменту відповідно до заданої програми обробки [2, 5].

Виконавчими механізмами фрезерних верстатів є електродвигуни приводу інструментального порталу, а також електромотор шпинделя і ряд допоміжних систем.

При використанні крокових двигунів «зворотний зв'язок» з самого початку закладено в їх конструкцію – двигун однозначно «знає» на скільки кроків повернеться ротор при певній довжині імпульсу. Для більш потужних верстатів застосовуються серводвигуни, контроль переміщення яких здійснюється спеціальними датчиками

положення. Говорячи спрощено, для реалізації принципу числового програмного забезпечення система ЧПУ повинна «знати», куди переміщати ріжучий інструмент, і «відчувати» де він в кожен момент часу знаходиться. За перше відповідає програма обробки, а за друге – датчики положення інструменту [3, 6].

Отже, електронна система ЧПУ повинна включати наступні компоненти:

- мікропроцесор – для перетворення кодів програми в керуючі імпульси (а також контролю всіх основних і проміжних процесів верстата);
- оперативну пам'ять – для зберігання поточної інформації в процесі обробки;
- постійну пам'ять – для зберігання файлів керуючих програм, налаштувань обладнання та іншої допоміжної інформації;
- пристрій завантаження програм (наприклад, через USB-інтерфейс);
- пристрій управління (власний і/або зовнішній – плата підключення ПК).

Конструктивне виконання систем ЧПУ відрізняється широким розмаїттям. У процесі розвитку системи зазнавали значних змін – як за способом завантаження програм (перфокарти для ранніх систем і тривимірні твердотільні моделі для сучасних), так і за алгоритмом управління (замкнуті, розімкнуті і т. п.). Для сучасних верстатів характерна наявність ЧПУ, орієнтованого на максимальну інтеграцію з ПК [7].

Особливості сучасних систем ЧПУ

В даний час вдосконалення систем ЧПУ, як і будь-яких інших продуктів в ІТ-сфері, йде стрімкими темпами. При цьому основними тенденціями розвитку ЧПУ є:

- спрощення апаратної і програмної частини систем;
- повна сумісність з попередніми «еволюціями» (для запуску раніше напрацьованих програм);
- упор на розробку і вдосконалення програмного забезпечення (і таким чином розширення функціоналу існуючих систем ЧПУ);
- плавна «еволюція» технічних рішень (замість «революційних» змін) апаратної частини систем;
- відкритість систем – для виробників верстатного устаткування це означає широкі можливості для самостійного доопрацювання;
- багатоканальність – для реалізації одночасного запуску декількох керуючих програм на одній системі ЧПУ;
- підтримка алгоритмів високошвидкісної обробки.

На рис. 1 представлений зібраний авторами статті фрезерно-гравірувальний верстат з ЧПУ.

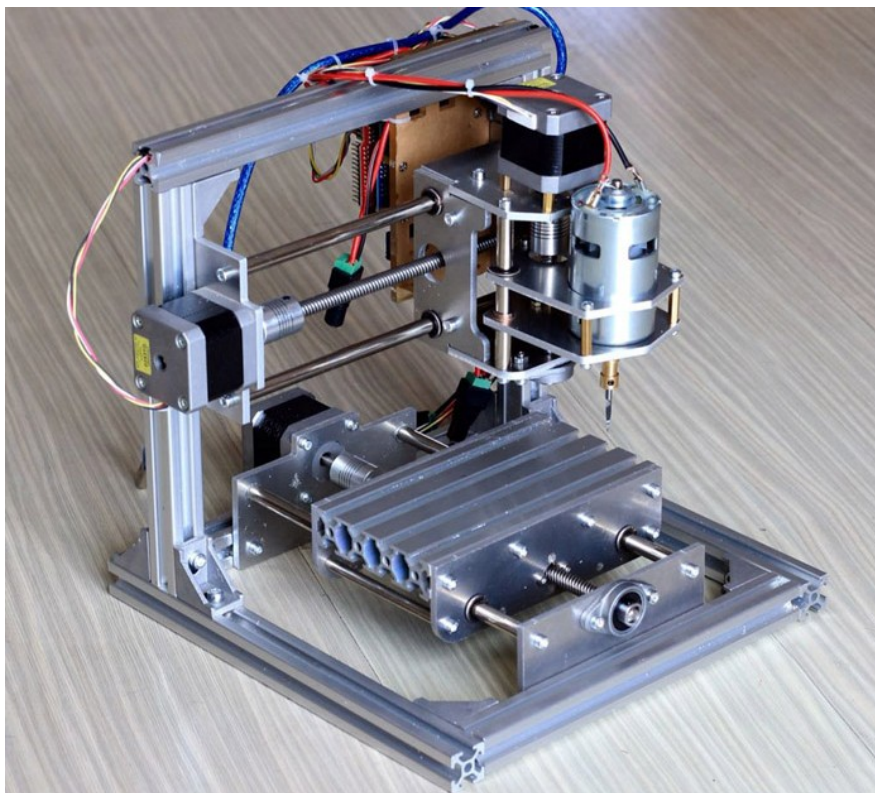


Рис. 1. Верстат з ЧПУ

Він складається з наступних частин: станини, напрямних, шпинделя, приводів подачі, системи ЧПУ, ріжучих інструментів.

Характеристики верстата:

- Хід по осях: 130 x 110 x 45мм;
- Точність обробки: 0.1 мм;
- Максимальна швидкість переміщення: 600мм / хв;
- Потужність двигуна шпинделя: 80Вт (24В);
- Патрон шпинделя: 3.125мм;
- Рама: Алюмінієвий профіль 2020;
- Робочий стіл: Алюмінієвий профіль 2080 шириною 150мм;
- Фланці: Сталь 4мм;
- Контролер: Arduino UNO з CNC Shield і трьома драйверами крокових двигунів, управління шпинделем через реле;
- Програмне забезпечення: програма gbrlcontrol і векторний редактор Inkscape.;

Центральною частиною всієї СЧПУ є підсистема управління. З одного боку, вона читає керуючу програму і віддає команди різних агрегатів верстата на виконання тих чи інших операцій. З іншого – взаємодіє з людиною, дозволяючи оператору верстата контролювати процес обробки.

В якості контролера СЧПУ була використана плата Arduino Uno. Керуюча програма – GRBL-прошивка повинна бути попередньо записана в пам'ять Arduino.

Основні настройки GRBL v.0.9 +.

Для виклику налаштувань GRBL у вікні монітора порту необхідно ввести команду \$\$.

GRBL відповідь списком всіх налаштувань:

\$ 0 = 10 (step pulse, usec);

\$ 1 = 25 (step idle delay, msec);

\$ 2 = 0 маска задає інвертування для сигналу кроку двигуна 00000000;

\$ 3 = 6 маска, що задає напрямки для двигунів 00000110;

\$ 4 = 0 (step enable invert, bool);

\$ 5 = 0 інвертувати значення кінцевиків;

\$ 6 = 0 (probe pin invert, bool);

\$ 10 = 17 режим виведення додаткової інформації, 17 = 16 + 1 показує позицію і значення кінцевиків;

\$ 11 = 0.010 (junction deviation, mm);

\$ 12 = 0.002 (arc tolerance, mm);

\$ 13 = 0 дюйми або міліметри? ставимо нуль, у нас міліметри;

\$ 20 = 0 програмний ліміт, вимикаємо – у нас апаратний;

\$ 21 = 1 апаратні датчики ліміту;

\$ 22 = 1 режим homing - відкат в нульову позицію до упору в кінцевики;

\$ 23 = 3 маска напрямку руху додому 00000011;

\$ 24 = 50.000 мінімальна швидкість кінцевиків;

\$ 25 = 3000.000 швидкість ходу до домашньої позиції mm / min;

\$ 26 = 250 захист від брязкоту контактів у кінцевиків, в мілісекундах;

\$ 27 = 1.000 відкат від нульової точки мм;

\$ 100 = 38.000 кількість кроків на 1мм переміщення по осі x;

\$ 101 = 38.000 кількість кроків на 1мм переміщення по осі y;

\$ 102 = 10.000 кількість кроків на 1мм переміщення по осі z;

- \$ 110 = 5000.000 максимальна швидкість мм / хв по осі x;
- \$ 111 = 5000.000 максимальна швидкість мм / хв по осі y;
- \$ 112 = 2000.000 максимальна швидкість мм / хв. по осі z;
- \$ 120 = 200.000 прискорення мм / с² по осі x;
- \$ 121 = 200.000 прискорення мм / с² по осі y;
- \$ 122 = 200.000 прискорення мм / с² по осі z;
- \$ 130 = 320.000 максимальна дистанція в мм по осі x;
- \$ 131 = 140.000 максимальна дистанція в мм по осі y;
- \$ 132 = 10.000 максимальна дистанція в мм по осі z;

Для створення програми виготовлення деталі використовувалася програма Inkscape. Додатково необхідно встановити в ній розширення – Інструменти GCode. З початку малюється макет виробу, далі він перетворюється в векторні фігури. Результат перетворення в векторне зображення наведено на рис. 2. Далі він редагується і формується програма виготовлення виробу (т. зв. G-код).

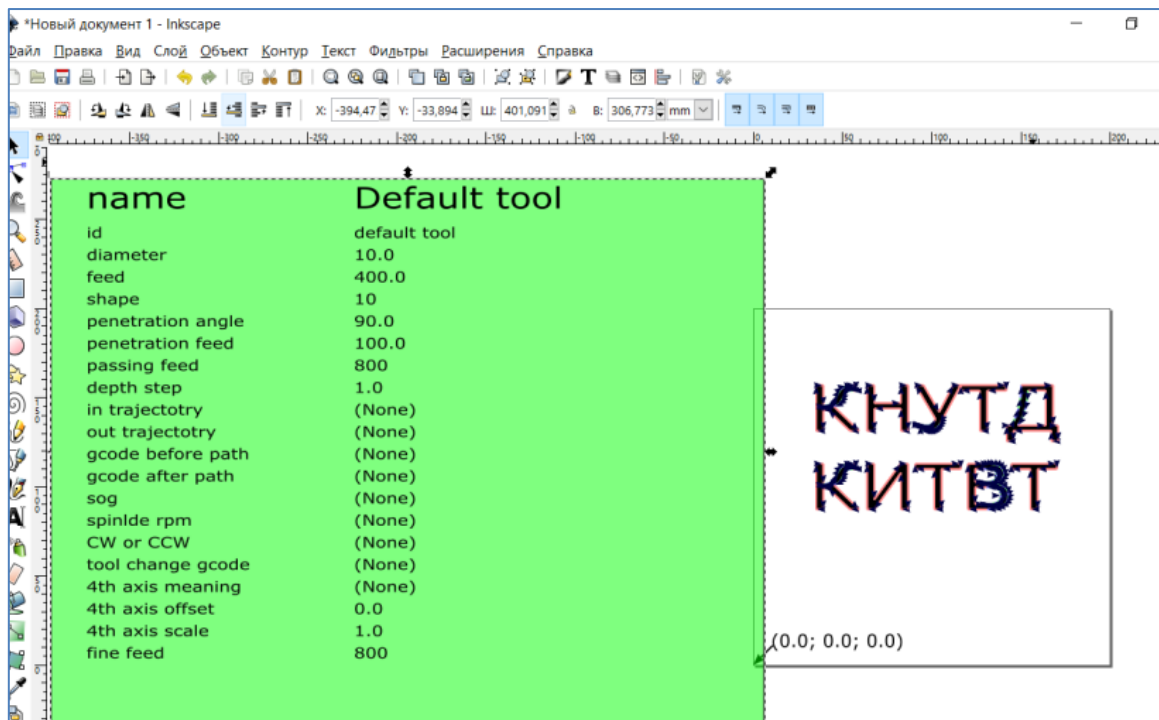


Рис. 2. Результат перетворення об'єкта в векторне зображення

Фрагмент отриманого G-коду наведено на рис. 3.

```
output_0019 — Блокнот
Файл  Правка  Формат  Вид  Справка

%
(Header)
(Generated by gcodetools from Inkscape.)
(Using default header. To add your own header create file "header" in the output dir.)
M3
(Header end.)
G21 (All units in mm)

(Start cutting path id: path4523)
(Change tool to Default tool)

G00 Z5.000000
G00 X48.057163 Y30.994534

G01 Z-0.125000 F100.0(Penstrate)
G01 X45.023050 Y30.994534 Z-0.125000 F400.000000
G01 X45.023050 Y24.627891 Z-0.125000
G01 X43.893812 Y24.627891 Z-0.125000
G01 X43.893812 Y30.994534 Z-0.125000
G01 X40.859698 Y30.994534 Z-0.125000
G01 X40.859698 Y31.847946 Z-0.125000
G01 X48.057163 Y31.847946 Z-0.125000
G01 X48.057163 Y30.994534 Z-0.125000
G00 Z5.000000

(End cutting path id: path4523)

(Start cutting path id: path4521)
(Change tool to Default tool)

G00 Z5.000000
G00 X40.397738 Y26.848699

G01 Z-0.125000 F100.0(Penstrate)
G02 X40.333116 Y26.327885 Z-0.125000 I-2.131032 J0.000000 F400.000000
G02 X40.158204 Y25.898311 Z-0.125000 I-1.614030 J0.406798
```

Рис. 3. Фрагмент G-коду програми обробки об'єкта

Як видно з малюнка, координати, отримані при проектуванні деталі можуть виходити за межі монтажної області (в цьому випадку можлива поломка крокового двигуна).

Цю проблему можна вирішити програмним шляхом, змінивши параметри прошивки GRBL:

\$ 130 - максимальна дистанція по осі x в мм;

\$ 131 - максимальна дистанція по осі y в мм;

\$ 132 - максимальна дистанція по осі z в мм.

Однак, таке рішення цієї проблеми не завжди є прийнятним - якщо неправильно визначена нульова точка верстата зміна параметрів GRBL не допоможе.

Тому вирішено було внести апаратні зміни в схему верстата, ввівши додатково 6 кінцевих вимикачів і кнопку повернення шпинделя в нульову точку.

Остаточна схема пристрою з урахуванням доопрацювання виглядає наступним чином (рис. 4):

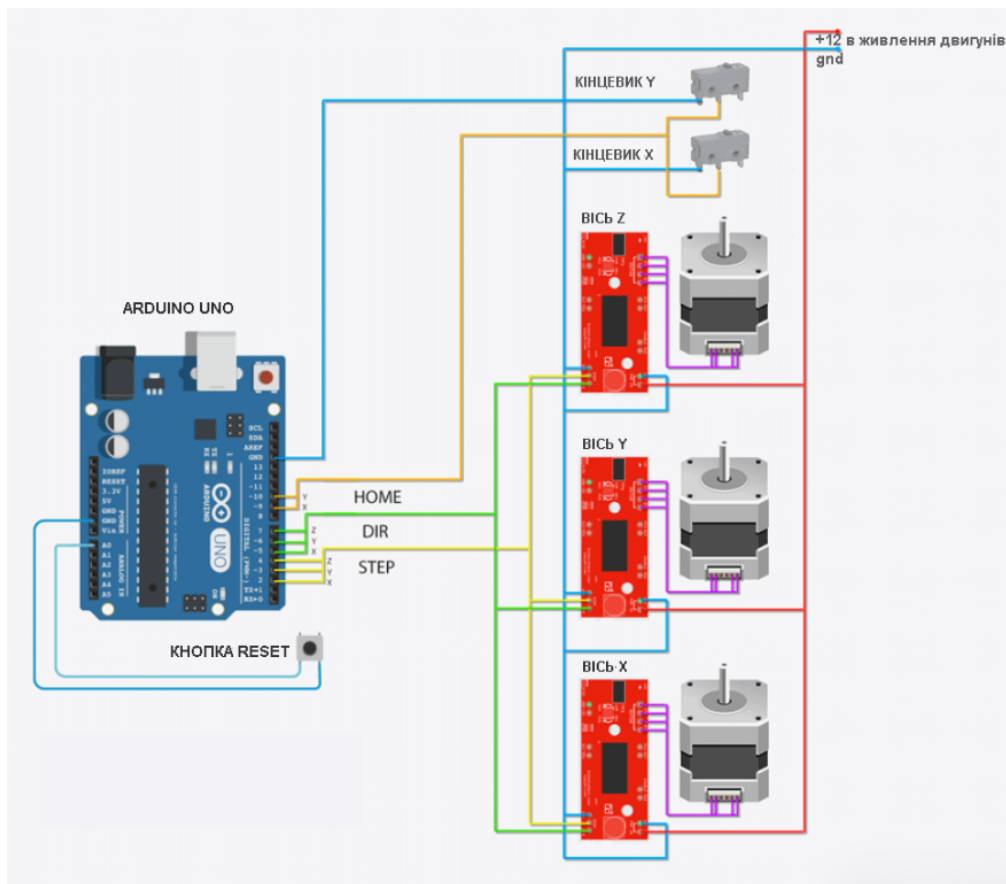


Рис. 4. Схема пристрою після модифікації

Додатково, необхідно виконати таку модифікацію прошивки, встановивши наступні параметри:

- \$ 21 = 1 -Активувати включення кінцевиків;
- \$ 22 = 1 - Активувати «Запустити цикл повернення в нульову точку»;
- \$ 23 = 3 -Встановити маску напрямку руху в нульову точку;
- \$ 24 = 25.000 -Встановити мінімальну швидкість руху до кінцевика;
- \$ 25 = 500.000 -Встановити мінімальну швидкість руху до кінцевика;
- \$ 26 = 25 - Встановити час спрацювання кінцевика (в мк сек.);
- \$ 27 = 3.000 - Встановити час відскоку кінцевика при спрацюванні (в мк сек.).

Кнопка RESET дозволяє виконати перезавантаження програми обробки деталі і встановити ріжучий інструмент в нульову точку.

Висновок

Виконана програмно-апаратна модифікація фрезерного верстата з ЧПУ істотно поліпшила технологію роботи верстата і усунула виявлені в процесі дослідної експлуатації системи проблеми.

Запропонований метод може бути використаний і для модифікації інших систем з числовим програмним управлінням, що базуються на використанні керуючої мікропроцесорної прошивки GRBL v.0.9 +.

Список використаних джерел

1. Сосонкин В. Л. Системы числового программного управления. Учебное пособие для вузов / В. Л. Сосонкин, Г. М. Мартинов. – М. : Логос, 2005. – 296 с.
2. Должиков В. П. Основы программирования и наладки станков с ЧПУ / В. П. Должиков. – Учебное пособие. – 2-е изд., перераб. и доп. – Томск, ТПУ, 2011. – 143 с.
3. Босинзон М. А. Современные системы ЧПУ и их эксплуатация / М. А. Босинзон. – М. : Академия, 2009. – 192 с.
4. Можчиль Б. В. Использование микропроцессоров при создании автоматизированных систем управления / Б. В. Можчиль, С. Ю. Фетисенко, Л. П. Голубев. // Технології та дизайн. – 2016. – № 3. – Режим доступу: http://nbuv.gov.ua/UJRN/td_2016_3_6.
5. Макацьора Д. А. Аналіз раціонального положення ножа та форми його поперечного перерізу в машинах типу «ДН» / Д. А. Макацьора, В. І. Князев // Вісник Київського національного університету технологій та дизайну. 2004. – № 1. – С. 159-163.
6. Столяров В. Г. Автоматизированное удаленное управление устройствами при помощи Ардуино / В. Г. Столяров, Л. П. Голубев. // Технології та дизайн. – 2016. – № 4. – Режим доступу: http://nbuv.gov.ua/UJRN/td_2016_4_12.
7. Ловыгин А. А., Васильев А. В. Современный станок с ЧПУ и CAD/CAM система. / А. А. Ловыгин, А. В. Васильев. – М. :Эльф ИПР,

References

1. Sosonkin, V. L. & Martinov, G. M. (2005). *Sistemy chislovogo programmno upravleniya. Uchebnoe posobie dlya vuzov* [Numerical control systems. Textbook for high schools]. Moscow: Logos [in Russian].
2. Dolzhikov, V.P. (2011). *Osnovy programmirovaniya i naladki stankov s CHPU* [Basics of programming and setting up CNC machines]. Tomsk, TPU [in Russian].
3. Bosinzon, M.A. (2009). *Sovremennye sistemy CHPU i ih ehkspluatatsiya* [Modern CNC systems and their operation]. Moscow: Akademiya [in Russian].
4. Mozhchil', B.V., Fetisenko, E.U., Golubev, L.P. (2016). *Ispol'zovanie mikroprocessorov pri sozdanii avtomatizirovannyh sistem upravleniya* [The use of microprocessors in the creation of automated control systems]. *Tekhnologii ta dizajn – Technology and design* 3(20). Retrieved from: http://nbuv.gov.ua/UJRN/td_2016_3_6. [in Ukraine].
5. Makat'ora, D.A. , Knyazev, V.I. (2004). *Analiz racional'nogo polozhennya nozha ta formi jogo poperechnogo pererizu v mashinah tipu «DN»* [Analysis of the rational position of the knife and its cross-sectional shape in the machines of the type "DN"]. *Visnik Kiivs'kogo nacional'nogo universitetu tekhnologij ta dizajnu- Bulletin of the Kiev National University of Technology and Design Vol.1, 159–163* [in Ukraine].
6. Stolyarov, V. G., Golubev, L.P. (2016). *Avtomatizirovannoe udalennoe upravlenie ustrojstvami pri pomoshchi Arduino* [Automated remote device management with Arduino]. *Tekhnologii ta dizajn –*

2016. – 286 с.

Technology and design 4(21). Retrieved from:

http://nbuv.gov.ua/UJRN/td_2016_4_12.
[in Ukraine].

7. Lovygin, A.A. & Vasil'ev, A.V. (2016). *Sovremennyy stanok s CHPU i CAD/CAM Sistema* [Modern CNC machine and CAD / CAM system]. Moscow:Elf IPR [in Russian].

Настройка и модификация программного обеспечения станка с ЧПУ

Голубев Л. П., Кива И. Л., Рафальский А. С., Шинкаренко Ю. В.

Киевский национальный университет технологий и дизайна

Цель. Исследование и модификация программного обеспечения станка с ЧПУ.

Методика. Теоретические исследования проводились на базе специальных разработок в теории управления методом моделирования и теории принятия решений. Исследования проводились на системе числового программного управления, построенное на базе персонального компьютера (PC) и контроллера управления движением рабочих узлов станка.

Результаты исследований. Исследована технология работы системы числового программного управления и выполнена модификация программного и технического обеспечения для улучшения работы системы.

Научная новизна. Усовершенствована технология работы на станке с ЧПУ, разработан новый подход к решению задачи критического поведения системы в граничных точках монтажного стола (рабочей области) и возврата резца в нулевую точку.

Практическая значимость. Разработанный метод решает проблему некорректно поведения системы в граничных областях монтажного стола и может быть применена во многих системах с числовым программным обеспечением.

Ключевые слова: станок с числовым программным управлением, G-код, Arduino, GRBL-прошивка

Configuring and modifying the software of CNC machine

Golubev L. P., Kiva I. L., Rafalsky A. S., Shynkarenko Yu. V.

Kiev National University of Technology and Design

Purpose. Research and modification of CNC machine software.

Methodology. Theoretical studies were carried out on the basis of special developments in the theory of control by modeling and decision-making theory. The research was carried out on a system of numerical software management, built on the basis of a personal computer (PC) and a controller for controlling the movement of machine working units.

Findings. the technology of the system of numerical control of the software and the modification of the software and technical support to improve the system.

Originality. Improving the technology of work on a CNC machine, developed a new approach to solving the problem of critical behavior of the system at the boundary points of the assembly table (working area) and the return of the cutter to the zero point.

Practical significance. The developed method solves the problem of incorrect behavior of the system in the boundary areas of the assembly table and can be applied in many systems with numerical software.

Keywords: machine numerical control, G-code, Arduino, GRBL-firmware