

Поскольку метод кодирования сигнала в штриховых кодах заключается в определенной последовательности чередования штрихов и пробелов различной ширины, профиль изображения содержит всю необходимую информацию для того, чтобы считать штриховой код.

Результат выполнения команды `ListPlot[list, Joined->True, PlotStyle->Black]` представлен на рисунке 4.



Рисунок 3 — Штрихкод, который подлежит обработке

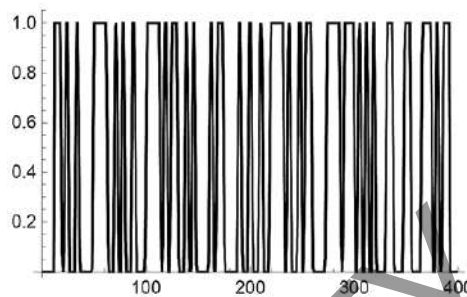


Рисунок 4 — Профиль изображения

Вычисляем среднее значение яркости по всему изображению с помощью функции `Mean`.

Находим начальную и конечную абсциссу первого максимума, затем длину по *Ox* второго максимума, тем самым определяем длину наименьшей штриховой линии.

```
For[i=1,i<=Length[list],i++,If[list[[i]]<m,list[[i]]=0,list[[i]]=1]
i=1;While[list[[i]]==0,i++;pos1=i];i=pos1;While[list[[i]]==1,i++;pos2=i-1];
i=pos2+1;pos3=0;While[list[[i]]==0,i++;pos3=i];i=pos3;pos4=0;While[list[[i]]==1,i++;pos4=i-1];
bisw=pos4-pos3+1.
```

Формируем бинарную строку, в которой содержатся 1 и 0 столько раз, какова длина соответствующей штриховой линии:

```
b="" ; i=pos1 ; k=1 ; s=0 ; While[i<=Length[list],If[list[[i]]==k,i++;s=s+1,b=b<>StringRepeat[ToString[k],
Round[s/bisw+0.25]];If[k==1,k=0,k=1];s=0].
```

Результат этой команды следующий: 1101001000011110010100100011110101100101000010110000100100
100111101001011000011110111010101000110001100011101011.

Первые 11 символов соответствуют началу штрихкода, последние 11 символов — его завершению.

Осуществляем декодирование подстрок длиной 11: `f[x_]:=ReplaceAll[x,{"10011101100"->"0","10011100
110"->"1",...,"11000111010"->"|"}]`.

Составляем массив подстрок длины 11: `list1=Table[StringTake[b,{11(i-1)+1,11i}],{i,1,IntegerPart[String
Length[b]/11]}];list1=Delete[list1,{{1},{-1},{-2}}]`.

Действуем на них правилом `f`: `StringJoin[f[list1]]`.

Результат, соответствующий коду (см. рисунок 3), это слово «wolfram».

Заключение. В работе описаны этапы построения гистограммы снимка, профиля изображения, а также его последующего распознавания.

Список цитируемых источников

1. Дьяконов, В. MATLAB. Обработка сигналов и изображений. Специальный справочник / В. Дьяконов. — СПб. : Питер, 2002. — 608 с.

УДК 621.91

А. А. Дыль, А. В. Шах

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ И РАЗРАБОТКА УПРАВЛЯЮЩЕЙ ПРОГРАММЫ ДЛЯ ИЗГОТОВЛЕНИЯ ПРОУШИНЫ ШТОКА ДОМКРАТА ПЕРЕДВИЖКИ НА СТАНКЕ С ЧИСЛОВЫМ ПРОГРАММНЫМ УПРАВЛЕНИЕМ

Введение. Современное машиностроительное производство трудно представить без станков с числовым программным управлением (далее — ЧПУ). Сегодня они широко применяются как на промышленных гигантах, так и на малых предприятиях. Успешное развитие машиностроительной промышленности невозможно без активного использования оборудования с ЧПУ и автоматизации производства.

В отличие от ручного управления автоматическое управление осуществляется по программе, в которой в том или ином виде воплощена информация чертежа детали. Человек при автоматическом управлении исключен из процесса управления станком, он участвует только в наладке станка или выполняет роль оператора. Автоматическое управление дает возможность существенно повысить производительность труда [1].

Цель проекта — компьютерное моделирование и разработка управляющей программы для изготовления проушины штока домкрата передвижки на станке с ЧПУ.

Основная часть. Прουшина служит для подвижного закрепления корпуса гидроцилиндра, она крепится на резьбу штока, соединяющегося с подвижным механизмом и поршнем гидроцилиндра.

Геометрическая модель проушины представлена на рисунке 1.

После создания нового проекта необходимо импортировать в САМ-систему твердотельную 3D-модель разрабатываемой детали, предварительно сохранив ее в формате, поддерживаемым используемой системой.

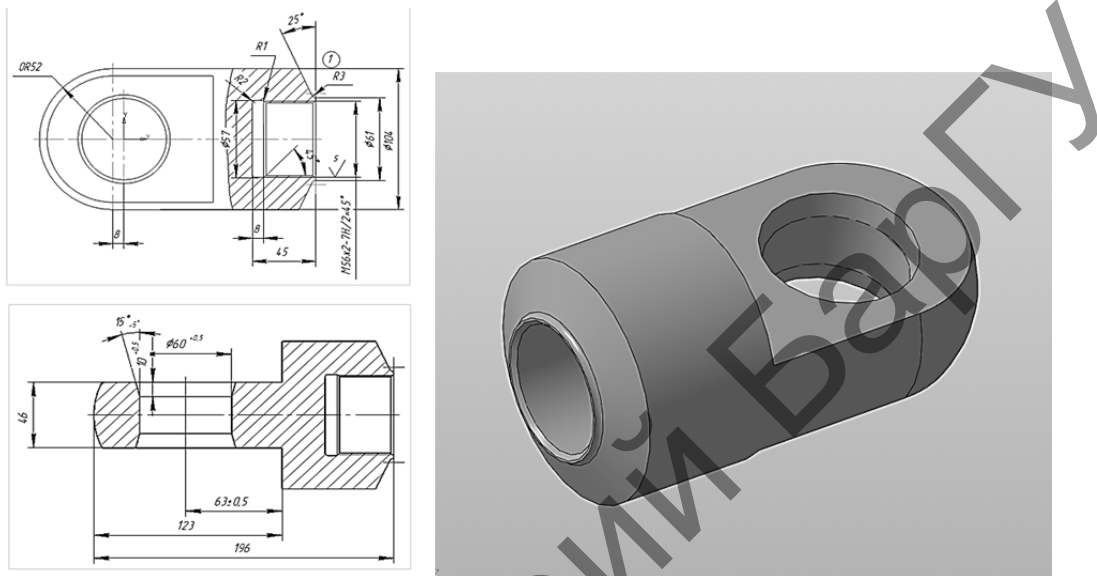


Рисунок 1 — Твердотельная модель проушины штока

После импортирования 3D-модели необходимо автоматически или вручную идентифицировать типовые элементы детали, которые требуется обработать, и запустить имитацию обработки детали. При этом FeatureCAM на основе встроенной в систему технологической базы знаний автоматически:

- выбирает режущий инструмент из заданного набора или встроенной базы данных инструмента;
- назначает параметры черновой и чистовой обработки;
- разбивает припуск на проходы и рассчитывает режимы резания в зависимости от характеристик материала;
- контролирует отсутствие зарезов и столкновений инструмента с деталью;
- рассчитывает управляющую программу в коде ISO.

После успешной имитации обработки детали появляется список производимых операций в установленном порядке, используемые инструменты и их типы, который так же можно редактировать, меняя порядок и скорость обработки поверхностей.

Список операций механической обработки представлен на рисунке 2.

Для управления многочисленными функциями станка с ЧПУ применяется довольно число различных кодов. Базовые коды программирования обработки представлены в таблице 1 [2].

Список операций					
<input checked="" type="radio"/> Автоматический порядок <input type="radio"/> Упорядочить вручную					
С.	Операция	Элемент	Инструмент	Подача	Скорость
	черн. проход 1	точение1	SW_Turn_80m...	* 0.030 мм/об	* 1200 м/мин
	чист.	точение1	SW_Turn_80m...	* 0.030 мм/об	* 1200 м/мин
	черн. проход 1	точение2	SW_Turn_80m...	* 0.030 мм/об	* 1200 м/мин
	чист.	точение2	SW_Turn_80m...	* 0.030 мм/об	* 1200 м/мин
	черн. проход 1	точение3	SW_Turn_80m...	* 0.030 мм/об	* 1200 м/мин
	чист.	точение3	SW_Turn_80m...	* 0.030 мм/об	* 1200 м/мин
	сверло	отверстие1	* конц.фрезаМ5...	0.653 мм/об	196 об/мин
Результаты					

Рисунок 2 — Список операций механической обработки

Т а б л и ц а 1 — Базовые коды программирования обработки

Код (функция)	Назначение и пример кадра с кодом
<i>Осевое перемещение</i>	
G00	Ускоренный ход — перемещение на очень высокой скорости в указанную точку G00 X10. Y20. Z25.
G01	Линейная интерполяция — перемещение по прямой линии на указанной скорости подачи G01 X10. Y20. F100
G02	Круговая интерполяция — перемещение по дуге по часовой стрелке на указанной скорости подачи G02 X10. Y20. R10. F100
G03	Круговая интерполяция — перемещение по дуге против часовой стрелки на указанной скорости подачи G03 X10. Y20. R10. F100
<i>Настройка</i>	
G20	Ввод дюймовых данных G20 G00 X10. Y20
G21	Ввод метрических данных G21 G00 X10. Y20
G90	Абсолютное позиционирование — все координаты отсчитываются от постоянной нулевой точки G90 G00 X10. Y20
G91	Относительное позиционирование — все координаты отсчитываются от предыдущей позиции G91 G00 X10. Y20
<i>Обработка отверстий</i>	
G81	Цикл сверления
G82	Цикл сверления с задержкой на дне отверстия
G83	Прерывистый цикл сверления
G85	Цикл растачивания отверстия
<i>Вспомогательные коды(функции)</i>	
M00	Запрограммированный останов — выполнение программы временно прекращается
M01	Запрограммированный останов по выбору — выполнение программы временно прекращается, если активирован режим останова по выбору
M03	Прямое вращение шпинделя
M04	Обратное вращение шпинделя — шпиндель вращается против часовой стрелки
M05	Останов шпинделя
M06	Автоматическая смена инструмента M06 T02
M08	Включение подачи охлаждающей жидкости
M09	Выключение подачи охлаждающей жидкости
M30	Конец программы, перевод курсора к началу программы

В результате работы программы мы получаем исполняемый код управляющей программы для станка с ЧПУ. Часть сгенерированного кода представлена на рисунке 3.

Язык G- и M-кодов, как и любой другой язык программирования, позволяет работать с подпрограммами и совершать переходы. Посредством функции подпрограммы основная (главная) управляющая программа может вызывать из памяти другую программу (подпрограмму) и выполнить ее определенное число раз [3].

```

Код УП
P1: Верхний резцедержатель
N25 G28 U0
N30 G28 W0
( OPERATION: ROUGH TURN ТОЧЕНИЕ1 )
N40 T101
N45 G50 S3000
N50 G96 S3937 M4
N55 G0 X9.8425 Z4.9213 M8
N60 X4.4882
N65 Z-0.1142
N70 G1 X-0.0787 F0.
N75 Z0.
N80 X-0.0509 Z0.0139
N85 G0 Z0.1714
( OPERATION: FINISH TURN ТОЧЕНИЕ1 )
N95 G0 X4.5365
N100 G50 S3000
N105 G96 S3937
N110 G0 X4.5365 Z0.1714 M8
N115 Z-0.08
N120 G1 X4.252 Z-0.1181 F0.
N125 X-0.0787
N130 X0.144 Z-0.0068
( OPERATION: ROUGH TURN ТОЧЕНИЕ2 )
N140 G0 X4.4882
N145 G50 S3000
N150 G96 S3937
N155 G0 X4.4882 Z-0.0068 M8
N160 X4.5669 Z-0.0394
N165 X4.1024

```

Рисунок 3 — Сгенерированный код

Заключение. Использование станков с ЧПУ имеет следующие преимущества: более высокий уровень автоматизации производства; производственная гибкость — для обработки разных деталей нужно всего лишь заменить программу; высокая точность и повторяемость обработки.

Увеличение парка станков с ЧПУ приводит к повышению требований к технологической подготовке производства, в том числе к качеству разработки управляющих программ.

Список цитируемых источников

1. Морщилов, М. В. Разработка управляющей программы для станков с ЧПУ : учебно-методическое пособие / М. В. Морщилов [и др.]. — М. : МАДИ, 2017. — 48 с.
2. Проектирование металлорежущих станков и станочных систем : справочник-учебник : в 3-х т. / под общ. ред. А. С. Проņикова. — М. : Изд-во МГТУ, 2000. — Т. 3 : Проектирование станочных систем — 584 с.
3. Ловыгин, А. А. Современный станок с ЧПУ и CAD/CAM-система / А. А. Ловыгин, Л. В. Теверовский. — М. : ДМК Пресс, 2012. — 279 с.

УДК 004.35

А. И. Калько, П. С. Трубенюк, Н. С. Миколайчук, К. Н. Избицкий, М. С. Толочко
Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

СИСТЕМА УПРАВЛЕНИЯ СВЕТОДИОДНОЙ МАТРИЦЕЙ НА МИКРОКОНТРОЛЛЕРЕ

Введение. В настоящий момент очень актуальна тематика робототехники в нашем мире. Программа, написанная с помощью IDE для Arduino, называется скетчем. IDE Arduino поддерживает языки C и C++ с использованием специальных правил структурирования кода. Он состоит всего из двух функций: настройки и цикла. Функция настройки используется для инициализации переменных, режимов входных и выходных контактов и других библиотек, необходимых в скетче. После вызова настройки цикл функций повторно выполняется в основной программе. Он управляет платой до тех пор, пока плата не будет отключена или перезагружена [1].

Основная часть. В данном исследовании основной задачей ставилась разработка системы управления светодиодной матрицы, в состав которой входят следующие компоненты:

- 1) светодиодная матрица 16X16 адресных светоизлучающих диодов;
- 2) микроконтроллер Arduino Mega 3.0;
- 3) приложение на телефоне, которое позволяет управлять данным устройством;
- 4) Bluetooth модуль связывающий микроконтроллер с приложением на телефоне для управления и манипуляции с матрицей.

Для разработки используется устройство фирмы Arduino, в основе которого лежит микроконтроллер Atmel, семейства AVR, ATmega328. Микроконтроллеры для Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через интерфейс USB. Поддержка загрузчика встроена в Arduino IDE и выполняется в один щелчок мыши.

Управляемая светодиодная матрица — это экран с разрешением 16x16 пикселей, такое разрешение экрана позволяет нам выводить текст с мобильного устройства по bluetooth. Bluetooth позволяет нам передавать информацию на матрицу, о светодиодах которые должны загореться и цвет которым они будут излучать.

Такая матрица может служить как:

- бегущая строка для отображения небольшого объема текстовой информации;
- выводить пиксель-арты любого светового диапазона;
- выводить анимацию огня или снегопада;
- так же, с мобильного устройства можно играть в игру «Тетрис», выводя изображение на матрицу, в то же время управляя фигурами на мобильном устройстве;
- так же матрицу можно использовать как источник света.

Выбор был сделан в пользу данного микроконтроллера, так как устройство должно производить некоторые вычисления и проверки, а также передавать и принимать данные с телефона и других устройств. Микроконтроллер имеет внутрисистемную само программируемую FLASH память, размер которой составляет 32 Кбайта при этом 2 Кб используются для загрузчика, что достаточно для размещения прошивки. Наличие двух 8-битных и одного 16-битного таймера, программируемого USART и трёх внешних прерыва-