

УДК 004.896

**СОЗДАНИЕ ПРОГРАММЫ ДЛЯ ФРЕЗЕРНОГО РОБОТА
С ЧИСЛЕННЫМ ПРОГРАММНЫМ УПРАВЛЕНИЕМ
И РЕАЛИЗАЦИЯ ЕГО ФУНКЦИОНИРОВАНИЯ**

**Тарасов Н.Ю., Удовика Ю.Д., Бутенко Д.А., Колесников В.Ю.,
Лёвкин А.В. Научный руководитель ст. пр. Елисеев В.И.,
Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
sonic1837@rambler.ru**

Аннотация

Тарасов Н.Ю., Удовика Ю.Д., Бутенко Д.А., Лёвкин А.В., Колесников В.Ю., Елисеев В.И. Создание программы для робота и реализация его функционирования. Выполнен анализ форматов 3D моделей для робота. Определён формат для считывания координат точек 3D модели. Определён способ считывания координат точек 3D модели. Проведён анализ и определён способ минимизации количества точек. Определено, в каких программных средствах запрограммирован робот, а также рассмотрены все его функции.

***Ключевые слова:** 3D модель, координаты точек, программирование, формат.*

Постановка проблемы. Для реализации обработки материала фрезерным роботом с численным программным управлением необходимо иметь рабочую программу и загруженные в неё координаты точек. В связи с этим существует ряд проблем, которые требуют решения. Прежде всего, необходимо провести анализ форматов 3D моделей и сделать выбор формата 3D модели для считывания координат точек с неё. После этого необходимо определить способ считывания координат точек 3D модели. Далее необходимо минимизировать количество точек под фрезерного робота, потому что возможности робота ограничены конкретным их количеством. В завершении необходимо определиться со средой программирования робота и запрограммировать его действия.

Цель статьи – провести анализ существующих проблем, препятствующих реализации обработки материала фрезерным роботом с числовым программным управлением, и определить способы их решения.

Выбор формата 3D модели. Для дальнейшего считывания координат точек 3D модели требуется определиться с форматом 3D модели. Исследование показало, что всего существует очень большое количество форматов для 3D моделей. Самыми распространёнными являются модели формата DWG, DXF, DNG, STL, OBJ, IGS и MA. В ходе исследования оказалось, что для считывания координат точек подойдёт 3D модель формата

STL, потому что был найден способ с помощью программы Matlab и специальным расширением для неё считать координаты точки 3D модели. Если же необходимо считать координаты с 3D модели, например, DWG формата, то необходимо конвертировать 3D модель DWG формата в формат STL, с помощью специальной программы-конвертера. Например, существует программа 3D object converter, которая работает с большинством существующих форматов.

Считывание координат точек 3D модели. Считывание координат точек 3D модели необходимо для дальнейшего использования в программе, где координаты будет задавать движение моторам робота. Как показала практика, графические редакторы позволяют смотреть и редактировать 3D модель, но не позволяют вывести координаты каждой точки в какой-либо список. Как возможное решение был рассмотрен Matlab. Matlab имеет возможность из массива координат точек показывать 3D или 2D объект. Сами координаты хранятся в определённом списке, который потребуется в дальнейшей работе. Программе Matlab необходимо прочитать 3D модель STL формата и вывести массив координат в свой список, благодаря установленному расширению и командам, которые позволяет считывать все координаты точек 3D модели STL формата.

Минимизация количества точек. Как показывает практика точек в 3D модели много и обычно насчитывает более 100 тысяч. Данный робот предусматривает точность 2,5 - 10 тысяч точек, поэтому необходимо минимизировать количество точек. Так как точек свыше 100 тысяч, то необходимо автоматизировать процесс минимизации точек, создав программу, которая будет осуществлять этот процесс. Программирование будет осуществляться на языке C++, т.к. данный язык программирования имеет все необходимые возможности для осуществления данной задачи. Сам процесс минимизации будет осуществлять пару задач. Первой задачей является то, что необходимо убрать все отрицательные числа с координат, а именно найти минимальный элемент во всех трёх осях и в каждой оси прибавить минимальное число по модулю. Эта задача необходима для следующей задачи разбиения координат по шагам. Вторая задача разбиения координат по шагам заключается в том, что необходимо найти максимальный элемент на оси, разделить его на количество шагов на каждой оси, найти высоту на положении двух осей максимально схожими по шагам из общего списка координат.

Программирование движения робота. Основным программным обеспечением для программирования робота Lego Mindstorm NXT является NXT programming. Как показало исследование, для программирования робота так же существуют программы RobotC и BrickCC. Был выбран BrickCC потому что он более удобен для программирования, пишется на языке NXC, который очень похож на C++ с возможностью осуществлять те же функции что и в NXT programming, например, задать движение мотора. Существует проблема,

которая препятствует передаче координат роботу и заключается она в том, что программы, взаимодействующие с контроллером Lego, не могут из текстового файла считать массив данных для дальнейшей работы. В связи с этим была разработана программа, которая пишет код в текстовый файл, в котором записаны все координаты в программу робота. После работы этой программы необходимо копировать код из текстового файла и вставить его в код программы робота. На вход поступают минимизированные координаты. Фрезерный робот с численным программным управлением произведёт движение в одну сторону, и в зависимости от высоты точки в 3D модели в данной позиции будет регулировать уровень высоты фрезы. Когда закончится считывание координат по одной стороне, то двигатель возвращается на исходную позицию и другой двигатель делает шаг в сторону относительно другой оси, после чего первый двигатель продолжает движение. Фрезерный робот с численным программным управлением будет работать до тех пор, пока не будут пройдены все заданные координаты.

Выводы. Произведён анализ форматов 3D моделей и определён формат для считывания координат, для дальнейшего взаимодействия с фрезерным станком с численным программным управлением. Найден способ считать координаты точек 3D модели с помощью программного обеспечения Matlab. Обоснован выбор языка программирования, с помощью которого. Определены задачи, которые выполняет программа для процесса минимизации точек. Обоснован выбор программного обеспечения. Описан способ передачи координат роботу и принцип работы фрезерного робота с численным программным управлением.

Список литературы

1. Matlab [электронный ресурс] // Википедия – свободная энциклопедия: [сайт][2017] URL: <https://ru.wikipedia.org/wiki/MATLAB>
2. Методические указания к лабораторным занятиям «Программирование LegoMindstormNXT» // Томский государственный университет систем управления и радиотехники: [Томск]. [2012].

УДК 004.896

**СОЗДАНИЕ ПРОГРАММНОГО РОБОТА «ХУДОЖНИК» С
ИСПОЛЬЗОВАНИЕМ LEGO MINDSTORMS NXT**

**Удовика Ю.Д., Бутенко Д.А., Тарасов Н.Ю., Колесников В.Ю.,
Лёвкин А.В. Научный руководитель ст. пр. Елисеев В.И.**
Донецкий национальный технический университет, г. Донецк
кафедра искусственного интеллекта и системного анализа
youlia.u@mail.ru

Аннотация

*Удовика Ю.Д., Бутенко Д.А., Тарасов Н.Ю., Колесников В.Ю.,
Лёвкин А.В., Елисеев В.И. Создание программного робота «Художника» с
использованием Lego Mindstorms. В результате проделанной работы был
спроектирован, собран и запрограммирован робот-художник, работающий с
использованием таких программ как: МАТЛАБ, Visual Studio 2015 и NXT
Program 2.0.*

Ключевые слова: *робот, программное обеспечение, программа.*

Постановка проблемы. Требуется разработать роботизированную руку, способную рисовать заданное ей графическое изображение при помощи набора Lego Mindstorms NXT. Для реализации этого следует выполнить следующие пункты:

- спроектировать виртуальную модель робота
- выбрать программы для дальнейшей работы
- решить основную задачу движения робота

Цель статьи – привести подробное описание проделанной работы, ознакомить пользователя с проблемами программирования и сборки робота «Художника».

Проектирование виртуальной модели робота. Для проектирования виртуальной 3D модели была использована программа LEGO Digital Designer. В программе предусмотрено создание моделей версии Mindstorms NXT. На рисунке 1 представлена полученная модель робота – художника.

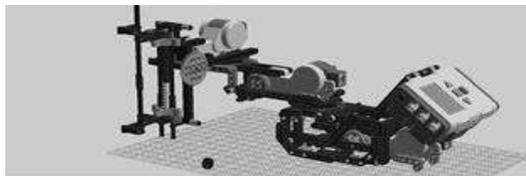
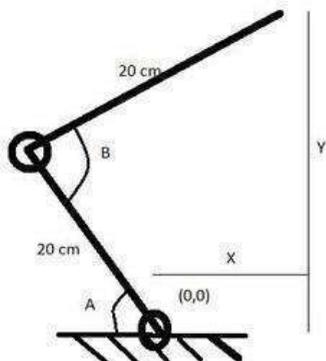


Рисунок 1 – 3D модель робота – художника

Выбор программ для дальнейшей работы. Под дальнейшей работой подразумевается создание программного обеспечения робота. Основной программной средой был выбран продукт MathWorks – МАТЛАБ R2009b, но для создания данного ПО подойдут и более ранние и более поздние версии данного продукта. Для проектирования роботизированных рук была создана программа SCARA, являющаяся производной составляющей Simulink МАТЛАБ, позволяющая спроектировать не только самого робота, но и траекторию движения его конкретных деталей, что позволяет воспроизвести движение шариковой ручки по бумаге и запрограммировать его. Для работы используются *.m файлы с указанной траекторией, математически рассчитанной в МАТЛАБ. Далее был использован уже готовый продукт, в результате работы которого были получены файлы формата spp, написанные на языке программирования С. Данный файл требуется скомпилировать в Visual Studio 2015. К сожалению, для данных программ не предусмотрена возможность экспорта или импорта ПО в Lego Mindstorms NXT или конвертации исходных файлов в формат, понимающийся роботом, поэтому полученный файл после компиляции требуется передать через подключенный процесс, выбрав NXT Program 2.0. Среда NXT Program 2.0. является основным официальным продуктом для работы с конструктором Lego, он предусматривает возможность создания, изменения и передачи готового файла контроллеру. Все программы находятся в открытом доступе и наиболее часто используются для создания подобного ПО.

Основная задача движения робота. Основной задачей является расчет углов между заданной координатой точек и частями роботизированной руки. Это позволит перемещать ручку, закрепленную в «кисти» робота, на требуемое расстояние. После расчета маршрута, программа SCARA передаст полученную матрицу значений конкретному двигателю, который в свою очередь повернет руку на нужный угол. Начальное положение «кисти» робота определяется датчиком движения, программа может прекратить свое действие при активации датчика касания. Наглядное представление математических расчетов показано на рисунке 2.



$$X = \cos(B-A) \cdot 20$$

$$Y = \sin(A) \cdot 20 + \sin(B-A) \cdot 20$$

From the above equations we get
 $\cos(B) = (800 - X^2 - Y^2) / 800$
 $A = 20 \cdot \tan(x/y) + (B/2)$

Рисунок 2 – Расчет углов поворота роботизированной руки

Внешний вид программы SCARA для данного робота представлен на рисунке 3.

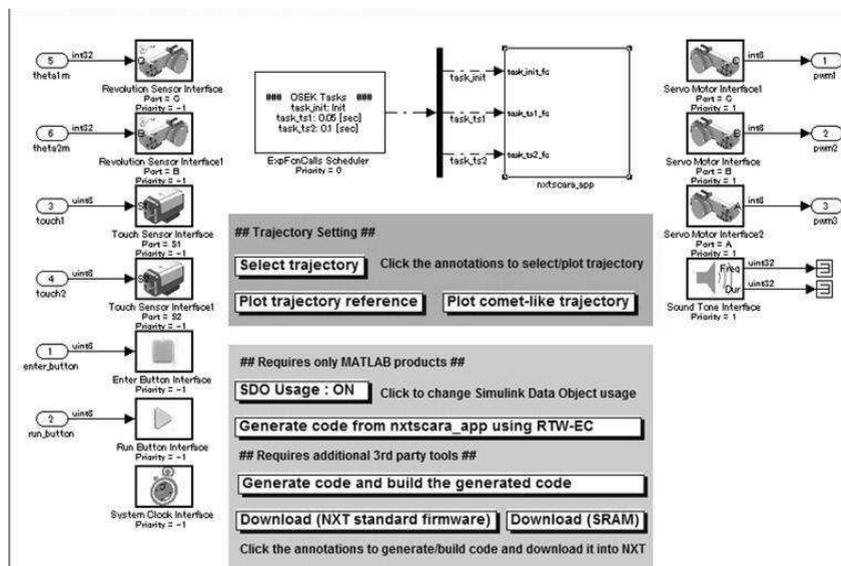


Рисунок 3 – Модель NXT контроллера

Выводы. Произведено подробное описание разработки ПО робота-художника, представлено описание использованных программ. Разработан метод передачи значений координат точек и маршрута движения роботу, с использованием углов, на которые способны переместить конструкцию

двигатели. Несмотря на неудобство, связанное с использованием большого количества программ и получения возможных ошибок, как следствие нескольких конвертаций, получен требуемый программный продукт.

Список литературы

1. Программирование на LEGO Mindstorms [электронный ресурс] // Alexey Kletsel Design: [сайт]. [1994-2013]. URL: <https://habrahabr.ru/post/274519/>
2. Методические указания к лабораторным занятиям «Программирование Lego Mindstorm NXT» // Томский государственный университет систем управления и радиотехники: [Томск]. [2012].
3. Работа в NXT Programming [электронный ресурс] University of Zurich, Department of Informatics, Artificial Intelligence Laboratory 3442 VU Introduction to Artificial Intelligence 06/07, Rolf Pfeifer, Prof., Dr. sc. techn. URL: <https://docviewer.yandex.ua/view/>
4. SCARA [электронный ресурс] // Википедия – свободная энциклопедия: [сайт] [2017] URL: <https://en.wikipedia.org/wiki/SCARA>