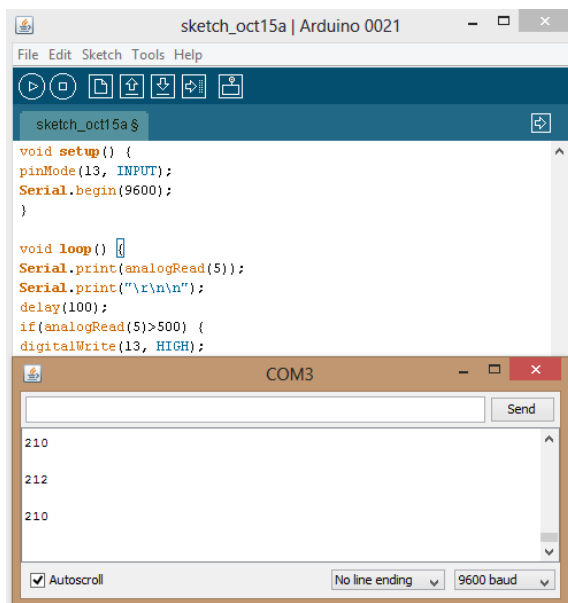


знакомыми близко с разработкой программного обеспечения. Язык программирования аналогичен используемому в проекте Wiring.

Программирование датчика и его компиляция представлены на рисунке 4.



```
sketch_oct15a | Arduino 0021
File Edit Sketch Tools Help
sketch_oct15a $
void setup() {
  pinMode(13, INPUT);
  Serial.begin(9600);
}
void loop() {
  Serial.print(analogRead(5));
  Serial.print("\r\n\n");
  delay(100);
  if (analogRead(5)>500) {
    digitalWrite(13, HIGH);
  }
}
```

COM3

210  
212  
210

Autoscroll No line ending 9600 baud

Рис. 4. Программирование датчика

В зависимости от уровня освещенности, датчик посылает нам цифровой сигнал, варьирующийся от 0 до 1000, где 0 – предел уровня яркости, а 1000 – темноты. В условиях данной работы использование цифрового сигнала нецелесообразно из-за низкой информативности и невозможности его использования в необходимых целях [1].

## ПРОГРАММИРОВАНИЕ РОБОТОВ LEGO MINDSTORMS НА ЯЗЫКЕ JAVA

Станкевич Ф.В.

Томский политехнический университет  
634050, Россия, г. Томск, пр-т Ленина,30  
E-mail: f.stankevich@mail.ru

В данной работе дано описание роботов LEGO Mindstorms, описан инструментарий для программирования таких роботов на языке Java, а также приведено описание простой задачи, иллюстрирующей возможности конструкторов LEGO Mindstorms.

На сегодняшний момент создание роботов является одним из актуальных вопросов. Однако конструирование и программирование робота не является тривиальной задачей. Процесс создания промышленного робота является достаточно сложным и дорогостоящим. В связи с этим для первого знакомства с роботами вполне подходят роботы, созданные компанией LEGO – LEGO Mindstorms.

LEGO Mindstorms – конструктор для создания программируемого робота. Первые LEGO Mindstorms были представлены в 1998 году. Спу-

Как видно на рисунке 4, светодиод включается при достижении датчиком значения 500 при помощи простого цикла, таким образом можно легко настраивать систему для управления отдельными элементами иллюминации по различным сценариям и уровням освещенности

### Заключение

Данная исследовательская работа отражает основы автоматического управления на базе многофункциональной платформы Arduino. В частности была разработана технология автономного освещения.

Применение системы Arduino гражданами среднего класса в бытовых целях финансово более выгодно по сравнению с другими микроконтроллерами подобного типа.

В дальнейшем предполагается продолжение изучения данного устройства контроля и проведение с ним целого комплекса работ.

### Литература

1. «Математические основы теории систем» за авторством Малышенко А.М. [Учебник]
2. Официальный сайт Arduino [Электронный ресурс]: <http://www.arduino.cc/>. Режим доступа: свободный.
3. Портал о платформе Arduino [Электронный ресурс]: <http://robocraft.ru/>. Режим доступа: свободный.
4. Российский сайт Arduino [Электронный ресурс]: <http://www.arduino.ru/>. Режим доступа: свободный.

стя 8 лет в 2006 году появилась первая модель LEGO Mindstorms NXT 1.0. В 2009 году появилась усовершенствованная модель LEGO-робота NXT 2.0. В начале 2013 года компания LEGO анонсировала новую модель LEGO Mindstorms EV3.



Рис. 1. LEGO Mindstroms NXT

На рисунке 1 приведено изображения робота LEGO Mindstorms NXT в сборе, с подключенным датчиком звука. В таблице 1 представлены технические характеристики LEGO Mindstorms NXT и EV3.

Таблица 1. Технические характеристики LEGO Mindstorms [1]

Характеристика/Модель	NXT	EV3
Процессор	ARM 7	ARM 9
Тактовая частота	48 MHz	300 MHz
FLASH память	256 Кб	16 Мб
Оперативная память	64 Кб	64 Мб
Слот расширения	–	SD
USB	–	v2.0
Bluetooth	v2.0	v2.1
Динамик	Basic	Premium
Размер экрана	100×60	178 ×128
Кол-во портов для двигателей	3	4

Наборы LEGO Mindstorms комплектуются набором стандартных деталей LEGO (палки, оси, колеса, шестерни) и набором, состоящим из сенсоров, двигателей и программируемого блока. Сенсоры и двигатели подключаются к программируемому блоку посредством кабелей и разъемов RJ-12. Ниже представлен список доступных сенсоров (датчиков):

Touch Sensor – датчик касания, реагирует на касание и последующее отпускание;

Light sensor – датчик света, выдает значение уровня освещенности;

Sound sensor – датчик звука, измеряет уровень звука в децибелах и его частоту в пределах 3-6 КHz;

Ultrasonic sensor – датчик расстояния, измеряет расстояние до объекта;

Compass sensor – компас, определяет ориентацию относительно сторон света;

Color sensor – цветовой датчик, определяет цвет;

Accelerometer sensor – акселерометр, измеряет ускорение по трем осям и угла наклона вдоль каждой оси;

RF ID Sensor – датчик идентификации, позволяет идентифицировать объекты (ключи), идущие в комплекте с датчиком в количестве 2 штук [2].

В комплект LEGO Mindstorms NXT 2.0 входит визуальный конструктор под названием NXT-G, позволяющий в графическом режиме составлять программы для LEGO-роботов, однако данный инструмент не позволяет создавать нетривиальные программы. Причина этого достаточно очевидна, графические языки программирования не позволяют эффективно управлять сложными приложениями и затрудняют понимание их логики.

Для создания более сложных программ лучше использовать один из высокоуровневых языков программирования. Так как задачи решаемые данными роботами носят преимущественно прикладной характер, то целесообразно использовать язык, оптимизированный для этих целей, например Java.

Для программирования на Java была выбрана свободно распространяемая интегрированная среда разработки (IDE) Eclipse. Данная среда разработки успешно интегрируется с библиотекой Lejos, которая была разработана специально для программирования LEGO Mindstorms [3].

Lejos содержит набор классов позволяющий легко управлять компонентами Mindstorms. Далее приведены некоторые классы, входящие в состав библиотеки:

LightSensor, UltrasonicSensor, SoundSensor – классы, предназначенные для работы с сенсорами;

NXTMotor, Movement, DifferentialPilot – классы, предназначенные для управления движением робота.

Работа с классами очень проста – достаточно передать необходимые параметры в конструктор класса, при создании экземпляра, например:

```
//S4 – порт подключения сенсора
new UltrasonicSensor(SensorPort.S4);
//C – порт подключения мотора
new NXTMotor(MotorPort.C);
//5.7f, 11.3f – диаметр колес и расстояние между ними
//B,C – порты подключения моторов
new DifferentialPilot(5.7f, 11.3f, Motor.B, Motor.C);
```

Для демонстрации работы с библиотекой Lejos ниже представлен фрагмент программы, который считывает значение освещенности и выводит его на дисплей робота.

```
//S3 – порт подключения сенсора
LightSensor sensor = new LightSensor(SensorPort.S3);
int lightValue = sensor.getLightValue();
//0,0 – координаты вывода на дисплей
LCD.drawInt(lightValue, 0, 0);
```

В рамках данной работы с помощью робота LEGO Mindstorms была разработана и отлажена программа, позволяющая роботу следовать ломаной линии, которая может иметь разрывы на своей протяженности. Пример ломаной линии, которой должен следовать робот, изображен на рисунке 2.

Для построения линии использовались листы ватмана и изоляционная лента черного цвета. К роботу был прикреплен датчик света (Light Sensor), направленный вниз, который позволил анализировать цвет поверхности, на которой находится робот. Линия имела разрывы не более 6-ти сантиметров и углы поворота не превышающие 135°.

По итогам испытания можно заключить, что робот успешно справился с поставленной задачей.

В представленной работе был сделан обзор роботов LEGO Mindstorms, а именно были рассмотрены компоненты конструктора и набор доступных сенсоров. Также были рассмотрены инструменты, позволяющие создавать программы для роботов – сочетание IDE Eclipse и библиотеки Legos. Для демонстрации возможностей описанных инструментов была решена задача следования роботом ломаной линии.

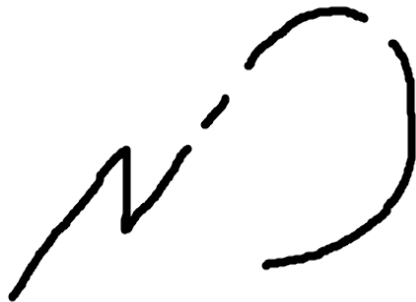


Рис. 2. Пример ломаной линии

### Литература

1. LEGO MINDSTORMS® EV3 Frequently Asked Questions [Электронный ресурс]. Дата доступа: 14.10.2013 Режим доступа: <http://mindstorms.lego.com/en-us/News/ReadMore/Default.aspx?id=476781>
2. Lego Mindstorms [Электронный ресурс]. Дата доступа: 14.10.2013 Режим доступа: <http://mindstorms.lego.com/en-us/products/default.aspx>
3. Java for LEGO Mindstorms [Электронный ресурс]. Дата доступа: 14.10.2013 Режим доступа: <http://www.lejos.org/>

## ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ КИНЕМАТИЧЕСКИХ ЦЕПЕЙ МЕХАНИЗМОВ – КОЛЕСНЫХ ЗВЕНЬЕВ

Светлякова Е.В.

Томский политехнический университет  
634050, Россия, г. Томск, пр-т Ленина,30  
E-mail: SvetlyakovaEkaterina@yandex.ru

### Введение

В современном автоматическом и автоматизированном производстве применяются сотни различных по структуре, управлению, компоновке и исполнению манипуляторов и роботов. Их число с каждым годом увеличивается. И в связи с этим возникает потребность в разработке формализованного способа описания кинематики подобных систем, который бы содержал всю информацию о структуре и параметрах манипуляторов. Такой способ описания должен быть удобен для решения задач, связанных с формированием баз данных о манипуляторах, их структурным и кинематическим анализом на ЭВМ. Кроме того, данный способ должен быть применим для описания не только манипуляторов и роботов с разомкнутыми кинематическими цепями, но и для сложных по структуре кинематических цепей роботов и других механизмов.

Разработанный профессором А. М. Малышenko и опубликованный в [1] способ описания структур и параметров кинематических цепей манипуляторов основан на формализованном описании всех входящих в цепь звеньев и сочленений. При этом каждый элемент описывается соответствующим его типу идентификатором и набором параметров, часть из которых отражает структурные связи в кинематической цепи, а остальные определяют собственные параметры этого звена или сочленения. Описание каждого элемента имеет структуру  $I(m,n,p)$ . Здесь  $I$  – идентификатор описываемого элемента;  $m$  – собственный номер

данного элемента, присвоенный в процессе описания цепи;  $n=[n_1,n_2,\dots,n_v]$  – вектор собственных номеров элементов цепи, с которыми непосредственно связан элемент  $m$ ;  $p=[p_1,p_2,\dots,p_\mu]$  – вектор конструктивных параметров элемента  $m$ , определяющих его геометрию и положение относительно других элементов. Параметры  $m$ ,  $n$ ,  $p$  являются позиционными, т. е. последовательность их записи строго определена.

### Формализованное описание различных типов сочленений и звеньев

#### 1. Сочленения.

Идентификаторы сочленений начинаются с символа  $S$  (сочленение). Их структура имеет вид  $StY$ , где  $t$  – число степеней относительной подвижности звеньев кинематической пары, образуемой данным сочленением;  $Y$  – символ, указывающий на тип сочленения. На позиции  $Y$  в [1] предлагается использовать:  $N$  – для жесткого соединения;  $W$  – для вращательной кинематической пары пятого класса;  $P$  – для поступательной кинематической пары пятого класса;  $G$  – для сочленения типа шарнира Гука;  $C$  – для цилиндрического сочленения четвертого класса;  $S$  – для сочленения сферического типа третьего класса.

Например, для неподвижного сочленения формализованное описание имеет вид  $SON(i, j, k, \gamma)$  и включает лишь один конструктивный параметр  $\gamma$  – угол поворота базисного