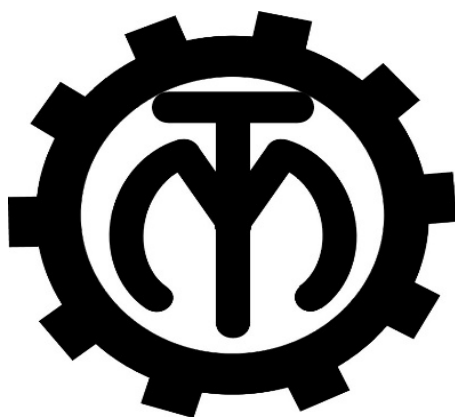


МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

# МОБИЛЬНАЯ ПРОМЫШЛЕННАЯ РОБОТОТЕХНИКА

*Методические рекомендации к лабораторным работам  
для студентов направления подготовки  
15.04.06 «Мехатроника и робототехника»  
очной и заочной форм обучения*



Могилев 2023

УДК 621.865.8  
ББК 32.816  
М74

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «22» ноября 2022 г.,  
протокол № 1

Составители: канд. техн. наук, доц. В. М. Шеменков;  
Е. Ю. Демиденко;  
О. Н. Кляус

Рецензент Д. М. Свирепа

Изложены методические рекомендации по выполнению лабораторных работ по дисциплине «Мобильная промышленная робототехника», а также теоретические положения в области программирования и наладки мобильных робототехнических систем.

Учебно-методическое издание

## МОБИЛЬНАЯ ПРОМЫШЛЕННАЯ РОБОТОТЕХНИКА

Ответственный за выпуск	В. М. Шеменков
Корректор	И. В. Голубцова
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2023

## Содержание

Инструкция по охране труда при проведении лабораторных работ.....	4
1 Лабораторная работа № 1. Компоновка мобильного робота Robotino.....	5
2 Лабораторная работа № 2. Принцип работы основных узлов мобильного робота Robotino.....	8
3 Лабораторная работа № 3. Система управления мобильного робота Robotino .....	11
4 Лабораторная работа № 4. Дистанционное управление мобильного робота Robotino.....	14
5 Лабораторная работа № 5. Программирование мобильных промышленных роботов в среде симулятора.....	17
6 Лабораторная работа № 6. Программирование мобильного робота Robotino в среде RobotinoView: функции распознавания цветов .....	20
7 Лабораторная работа № 7. Программирование мобильного робота Robotino в среде RobotinoView: функции анализа изображений .....	24
8 Лабораторная работа № 8. Программирование мобильного робота Robotino в среде RobotinoView: функции перемещений .....	28
Список литературы .....	32

## **Инструкция по охране труда при проведении лабораторных работ**

### ***Общие требования безопасности***

Допуск студентов к лабораторным занятиям производится только после инструктажа по технике безопасности, о чем делается соответствующая запись в специальном журнале (бланке).

### ***Требования безопасности перед началом работы***

1 Внимательно изучить содержание и порядок проведения лабораторной работы, а также безопасные приемы его выполнения.

2 Перед каждым включением оборудования предварительно убедиться, что его пуск безопасен.

### ***Требования безопасности во время работы***

1 Точно выполнять все указания преподавателя.

2 Не прикасаться к находящимся под напряжением элементам электрических цепей, к корпусам стационарного электрооборудования.

3 Запрещается во время работы оборудования снимать ограждения и предохранительные устройства, а также держать их открытыми.

4 Во время работы запрещается касаться руками нагретых, вращающихся и перемещающихся частей, вводить руки в зону движения.

### ***Требования безопасности по окончании работы***

1 Полностью выключить оборудование.

2 Привести в порядок рабочее место.

3 Предупредить преподавателя обо всех, даже малейших и незначительных, неисправностях оборудования.

### ***Требования безопасности в аварийных ситуациях***

1 В случае травмирования кого-либо немедленно доложить преподавателю.

2 При выходе оборудования из строя необходимо:

– отключить оборудование (обесточить);

– доложить преподавателю о случившемся, а в случае возгорания приступить к немедленной его ликвидации первичными средствами пожаротушения.

# 1 Лабораторная работа № 1. Компоновка мобильного робота Robotino

**Цель работы:** изучить компоновку мобильного робота Robotino.

## 1.1 Основные теоретические положения

**Устройство.**

Так как Robotino (рисунок 1.1) предназначается, главным образом, для обучения, то он выполнен модульно; все технические компоненты (электроприводы, датчики, камера) можно отключить от робота и изучить отдельно.

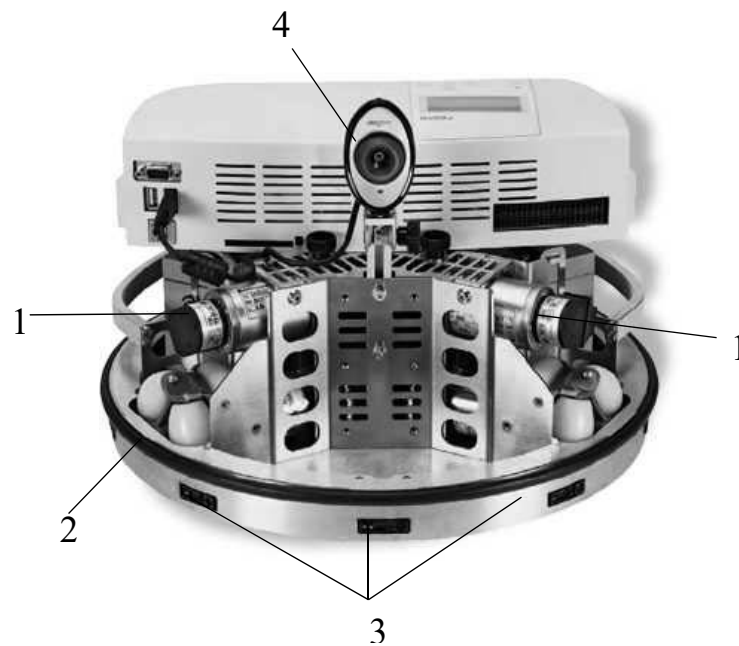


Рисунок 1.1 – Конструкция Robotino

Три двигателя привода обеспечивают перемещение системы во всех направлениях в горизонтальной плоскости, а также вращение вокруг вертикальной оси на месте. Система оснащена видеокamerой (4) и датчиками двух типов: цифровыми – для контроля фактической скорости (1) и аналоговыми – для измерения расстояния (3). Такое оснащение гарантирует выполнение широкого диапазона требований, предъявляемых к системам такого типа.

Каждый из датчиков может быть запрошен индивидуально через интерфейс ввода-вывода. Таким образом, можно избежать столкновения с препятствиями. Датчик антистолкновения (2) представляет собой узкую резиновую трубку, размещенную по периметру шасси Robotino. Внутри трубки расположены две электропроводящие полосы переключения, между которыми имеется малый зазор. При столкновении с препятствием эти поверхности замыкаются между собой, тем самым формируется сигнал для отключения привода. Такое

отключение возможно в случае столкновения с препятствием при перемещении в любом направлении.

Robotino состоит из следующих **подсистем**.

**Корпус робота** – стальной корпус с бампером, обеспечивающий лёгкий доступ к батареям, двигателям и портам робота.

**Подсистема питания** – две аккумуляторные батареи (12V/5Ah кислотной свинцовой (lead-acid) или 12V/9Ah NiMH), позволяющие роботу работать в автономной режиме несколько часов.

**Двигательная подсистема** – три двигателя постоянного тока, редуктор и ременная передача на роликовые колёса, позволяющие роботу двигаться в различных направлениях.

Для стабилизации скоростей вращения валов двигателей робота используются встроенные ПИД-регуляторы с предустановленными коэффициентами.

**Подсистема одометрии** – инкрементные энкодеры на валах двигателей.

**Подсистема ввода/вывода** – плата, выполняющая коммуникационную связь между компьютером робота и его датчиками, двигателями и интерфейсом ввода/вывода.

**Подсистема беспроводной связи с внешним управляющим компьютером** (Wi-Fi точка доступа).

**Бортовой вычислитель** – одноплатный промышленный компьютер формата PC/104+ (имеет шины ISA и PCI) с процессором на 300 МГц. В качестве памяти используется карточка Compact Flash (1024 MB), на которую установлена встроенная версия операционной системы Linux.

**Различные датчики робота** – 9 инфракрасных датчиков расстояния (на бампере), USB web-камера.

Robotino поддерживает интерфейсы: USB, Ethernet, VGA, 8 цифровых портов ввода-вывода, 8 аналоговых портов ввода.

*Система управления.*

Управлять роботом можно как с помощью программ, записанных в память робота, так и дистанционно – по каналу беспроводной связи Wireless LAN (WLAN).

Кроме того, возможно автономное программирование робота через подключенные к нему монитор и клавиатуру.

В комплекте с роботом идёт программа **Robotino View** – программное обеспечение для интерактивного графического программирования робота.

Robotino View позволяет посмотреть состояние управляющих сигналов робота и его датчиков.

Робот может быть запрограммирован в графическом режиме – простым выбором необходимых функциональных блоков из имеющегося списка и их последующим комбинированием, и настройкой (визуальное программирование).

Программное обеспечение робота можно разделить на две части: внутреннее программное обеспечение и внешнее.

Внутренне ПО робота – это его операционная система (Linux), которая осуществляет взаимодействие программного и аппаратного обеспечения робота, обработку внутренних команд и обмен данными со внешним управляющим компьютером. Также ко внутреннему ПО робота можно отнести программы, хранящиеся в собственной памяти робота для его автономного управления.

Внешнее программное обеспечение – это программы, реализующие связь и обмен данными с Robotino. Это могут быть:

- программы, написанные на языках программирования высокого уровня (C, C++, Java, C#);

- среды программирования и визуального проектирования: Matlab, Simulink, Labview, ROS, Microsoft Robotics Developer Studio (для них есть специальная библиотека для управления роботом);

- среда визуального проектирования Robotino View;

- виртуальный симулятор Robotino SIM, позволяющий управлять виртуальным Robotino.

## ***1.2 Порядок выполнения работы***

1 Изучить компоновку мобильного робота Robotino.

2 Изучить функциональное назначение внешних входов/выходов.

### ***Контрольные вопросы***

1 Какие основные элементы конструкции входят в компоновку мобильного робота Robotino?

2 Какие интерфейсы включены в компоновку мобильного робота Robotino?

3 Какие датчики включены в компоновку мобильного робота Robotino?

## 2 Лабораторная работа № 2. Принцип работы основных узлов мобильного робота Robotino

**Цель работы:** изучить принцип работы основных узлов мобильного робота Robotino.

### 2.1 Основные теоретические положения

Промышленные мобильные роботы состоят из разнообразных подсистем, выполняющих различные функции. В таблице 2.1 схематически представлены такие подсистемы и подфункции.

Таблица 2.1 – Подсистемы и подфункции промышленного робота

Подсистема	Подфункция
Кинематика	Установление пространственного отношения между деталью/инструментом и технологическим устройством
Привод	Преобразование и передача необходимой энергии на все оси движения
Кодовый датчик перемещения	Измерение положения и скорости для отдельных осей движения
Управление без обратной связи / с обратной связью	Сохранение, регулирование и контроль запрограммированной последовательности. Регулировка и управление числом оборотов и скоростями
Датчики	Измерение образца физических переменных и определение положения
Механизмы захвата	Захват заготовки, фиксация положения заготовки во время перемещения

*Общая информация по роботам со всенаправленным приводом.*

Основным элементом всенаправленного привода является так называемое разнонаправленное колесо или ролик, также известное как всенаправленное колесо. Эти колеса обычно имеют форму бочонка и крепятся на поворачивающейся поверхности основного колеса, их оси вращения расположены под прямым углом к оси вращения основного колеса (рисунок 2.1).

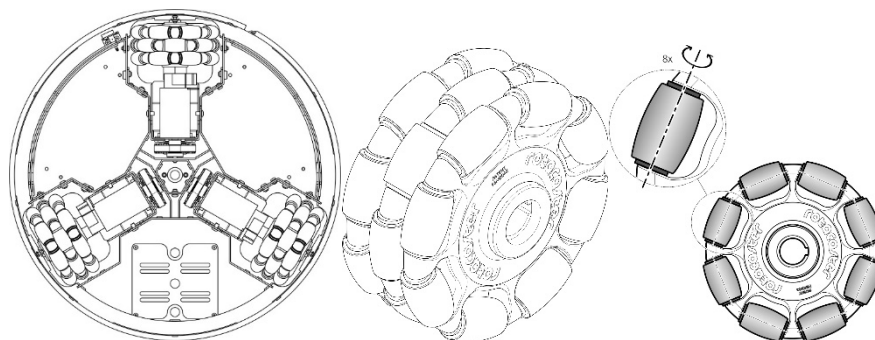


Рисунок 2.1 – Конструкция всенаправленного привода



Разнонаправленные колеса могут активно приводиться в движение двигателем, а также пассивно катиться в сторону на встроенных в колесо роликах.

Робота не надо отворачиваться от контролируемого объекта; он сохраняет свою линию визирования.

В случае использования симметричного привода колеса (разнонаправленные колеса или ролики) установлены под углом 120 град друг к другу.

На всенаправленном приводе колеса приводятся в движение так, чтобы все они вращались вместе.

#### *Движение тел.*

Robotino может двигаться в любом направлении: вперед, назад, вбок, по кругу с сохранением или без сохранения линии визирования.

И, следовательно, обладает тремя степенями свободы – двумя поступательными и одной вращательной.

#### *Система безопасности мобильного робота Robotino.*

Датчик защиты от столкновения Robotino состоит из так называемой чувствительной кромки (рисунок 2.2). Эта чувствительная кромка состоит из полимерного профиля разной формы со встроенной переключающей полостью. Внутри полости располагаются две отдельные проводящие зоны, которые замыкаются накоротко при надавливании на чувствительную кромку и генерируют сигнал для блока оценки. Чувствительная кромка Robotino работает по принципу тока холостого хода, поэтому возможно обнаружение потенциального разрыва кабеля и остановка Robotino.

Ток холостого хода – это электрический ток, который непрерывно протекает в цепи, даже если она неактивна. Если ток не протекает, то это происходит из-за разрыва кабеля или повреждения чувствительной кромки.

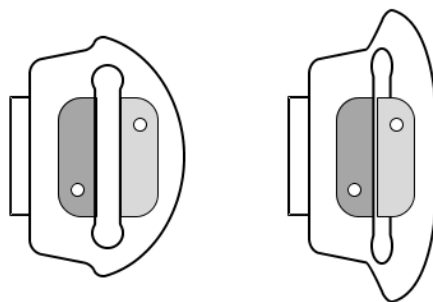


Рисунок 2.2 – Датчик защиты от столкновения

Чувствительные кромки преимущественно применяются в технологии обеспечения безопасности для предотвращения получения персоналом травм или повреждения оборудования или материалов в результате дробления или срезания. В медицинской технике они применяются на диагностическом

оборудовании, излучателях и подвижных защитных кожухах для защиты пациентов и персонала.

Также они применяются в дверях лифтов, автобусов и потолочных окнах с электроприводом для защиты пальцев.

Бампер доступен при контакте, подает сигнал 1 и в основном применяется для остановки Robotino в случае столкновения.

## 2.2 Порядок выполнения работы

Чтобы подключиться и управлять Robotino с использованием системы графического программирования Robotino View, необходимо выполнить последовательность действий, представленную в лабораторной работе № 1.

Построить при помощи библиотеки функциональных блоков в Robotino View схему, представленную на рисунке 2.3.

В соответствии с заданием преподавателя отрегулировать параметры скорости движения робота, скоростей его поворота и безопасного расстояния.

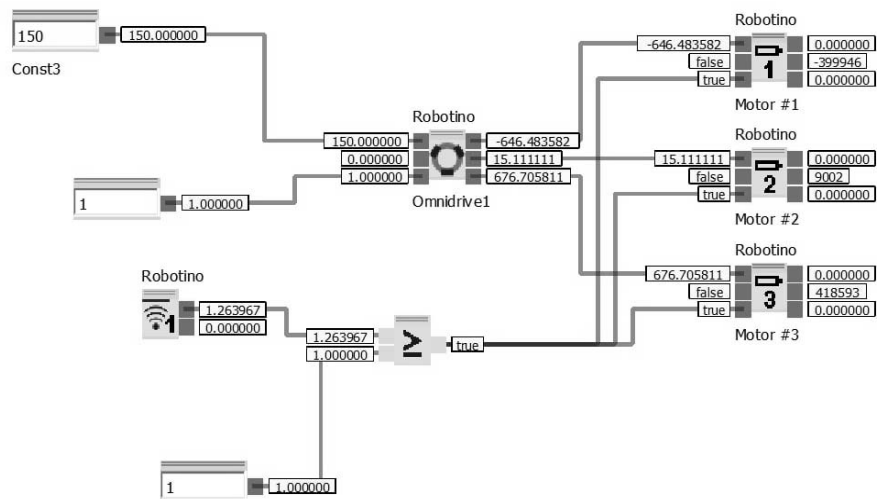


Рисунок 2.3 – Схема к выполнению лабораторной работы № 2

### Контрольные вопросы

- 1 Как работает система колес мобильного робота Robotino?
- 2 Как работает система безопасности мобильного робота Robotino?
- 3 Из чего состоит система колес мобильного робота Robotino?

### 3 Лабораторная работа № 3. Система управления мобильного робота Robotino

**Цель работы:** изучить систему управления мобильного робота Robotino.

#### 3.1 Основные теоретические положения

Robotino – это технический продукт Festo Didactic. Робот предназначен в основном для обучения, исследования и демонстрации мобильных роботизированных устройств. Robotino работает под управлением Linux (Ubuntu), но также может работать и на операционной системе Windows.

С Robotino можно взаимодействовать с помощью следующих трех стандартных программ: Robotino SIM, Robotino Factory, Robotino View. Существует возможность использования другого программного обеспечения для управления мобильной платформой, которое также официально поддерживается. Как правило, стандартные программы включают: программирование в LabVIEW; программирование в MATLAB; операционную систему программирования роботов (ROS) и др. Robotino также поддерживает средства прямого управления им: через Robotino Factory, интерфейс браузера и через вспомогательный контроллер или через программные продукты, упомянутые ранее.

Для того чтобы управлять Robotino через Robotino View, программа должна быть напрямую подключена к роботу. После подключения к Robotino через правильный IP-адрес можно загрузить созданную программу в систему. После загрузки Robotino View не нужно повторно подключать к системе, чтобы программа могла управлять роботом. Также после этого программа будет доступна через веб-интерфейс.

Robotino View содержит элементарные элементы управления для прямого управления Robotino без сложного программирования, как показано на рисунке 3.1. Этого можно добиться, подключив систему привода Robotino к панели управления в работающей программе. Также возможно подключить внешнее устройство управления и назначить его элементы для прямого управления.

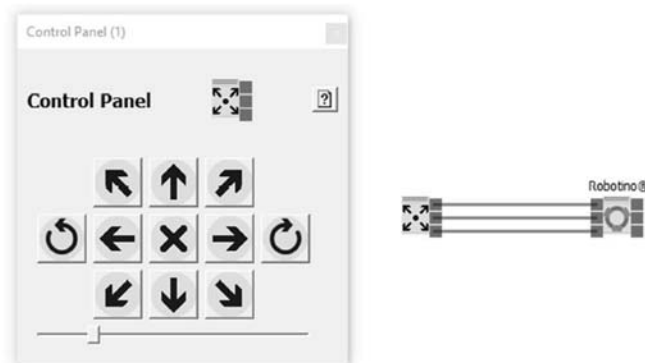


Рисунок 3.1 – Прямое управление Robotino через Robotino View

Запуск программы осуществляется нажатием зеленой кнопки воспроизведения, показанной на рисунке 3.2, вверху рядом с кнопкой сохранения. Текущую подпрограмму можно запустить, нажав вторую кнопку воспроизведения рядом с кнопкой паузы. Это полезно для проверки работы отдельных частей программы без прохождения полного цикла работы.

Чтобы остановить программу внешнего управления, необходимо нажать кнопку остановки. Кнопка остановки расположена в том же ряду, что и кнопки запуска на рисунке 3.2.

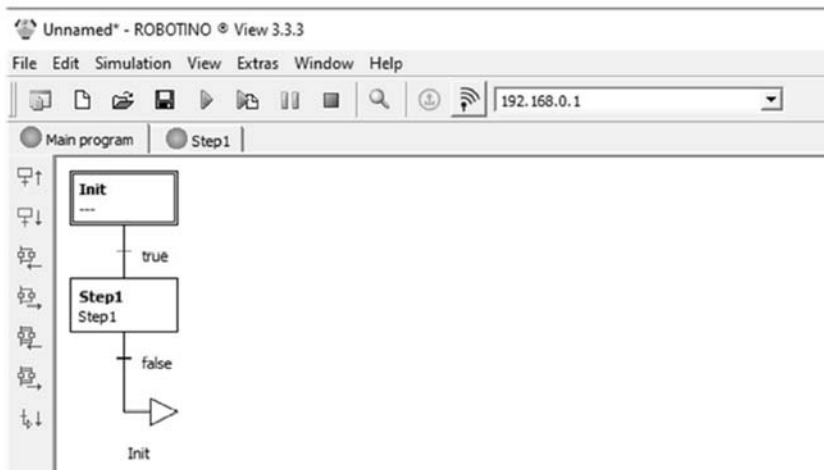


Рисунок 3.2 – Основная программа и панель инструментов Robotino View

### *Операционная система роботов (ROS).*

Robot Operating System (ROS) – это гибкая платформа (фреймворк) для разработки программного обеспечения роботов. Это набор разнообразных инструментов, библиотек и определенных правил, целью которых является упрощение задач разработки ПО роботов.

Целью создания ROS является создание «среды разработки, которая позволяет разработчикам ПО для роботов сотрудничать на глобальном уровне».

ROS сосредоточена на максимизации повторного использования кода при разработке. Основные характеристики, позволяющие это реализовать.

**Распределенные процессы.** Структура ROS создана в виде минимальных единиц исполняемых процессов (нод), и каждый процесс выполняется изолированно. Взаимодействие разных нод происходит только на уровне обмена сообщениями.

**Управление пакетами.** Несколько процессов, имеющих общую задачу, объединяются в пакеты. Управление пакетами подразумевает набор утилит, позволяющих автоматически скачивать, устанавливать и удалять пакеты. Пакетный менеджер гарантирует работоспособность и целостность установленных пакетов.

**Публичные репозитории и документация.** Каждый доступный пакет публикуется в публичном репозитории. Документация пакетов публикуется в единой системе, которая упрощает поиск необходимых пакетов.

Единое API. При разработке программы, использующей ROS, вы получаете простое и легко встраиваемое API. В примерах программ вы увидите, что использование API не сильно отличается от языка (C++ или Python). При этом нет разницы при использовании API, на каком языке была написана программа.

Поддержка различных языков программирования. Программа ROS предоставляет клиентские библиотеки для поддержки различных языков программирования. Наиболее популярны Python, C++, а также другие языки, такие как Lisp, JAVA, C#, Lua и Ruby.

Для использования ROS совместно с Robotino существуют ноды `robotino_node` и `robotino_rest_node`.

### ***3.2 Порядок выполнения работы***

Тест управления Robotino через ROS.

1 Установить необходимое ПО.

2 Запустить через командную строку скрипт `omnidriverosrun robotino_rest_node omnidrive.py`.

Если ROS работает не на компьютерной модели, необходимо указать IP-адрес: `Robotino.rosrunrobotino_rest_nodeomnidrive.pyROBOTINO_IP_ADDRESS`.

3 Через командную строку отдать команду роботу двигаться вперед со скоростью 0,1 м/с:

```
rostopic pub -r 10 /cmd_vel geometry_msgs/Twist '{linear: {x: 0.1, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0}}'
```

Здесь  $x$ ,  $y$  – скорости движения вперед и вбок, а  $z$  – скорость вращения робота.

### ***3.3 Задание***

На основе методических рекомендаций запустить управление робота Robotino по указанию преподавателя через Robotino View или через ROS.

### ***Контрольные вопросы***

1 На базе какой операционной системы функционирует система управления мобильного робота Robotino?

2 Какие типы системы управления поддерживает мобильный робот Robotino?

3 Что такое ROS (Robot Operating System)?

## **4 Лабораторная работа № 4. Дистанционное управление мобильного робота Robotino**

*Цель работы:* изучить дистанционное управление мобильного робота Robotino.

### **4.1 Основные теоретические положения**

Robotino использует высокопроизводительный встроенный процессор Embedded PC, в состав которого входят: процессор PC 104 с Realtime Linux-Kernel; SDRAM 64 Mb; Compact Flash Card 256 Mb; Ethernet, 2×USB, 2×RS232, 1×PS2, 1 параллельный порт и 1 VGA коннектор; протокол Wireless LAN (WLAN) в соответствии с 802.11 g и 802.11 b.

«Сердце» управления Robotino образует сервер, использующий приложение Linux. Дистанционное управление всенаправленным приводом с использованием протокола WLAN производится с помощью клавиатуры компьютера.

Мембранная клавиатура, встроенная в корпус блока управления, обеспечивает возможность управления следующими функциями Robotino без использования протокола WLAN: старт бортового процессора для запуска вычислительных операций; выбор языка меню; включение статуса информации; индикация состояния аккумуляторов; конфигурация сетевых соединений; выбор готовых автономных демонстрационных программ; TCP/IP – интерфейс, через который происходит связь с управляющим процессором с помощью протокола Wireless LAN.

С целью программирования и диагностики состояния монитор и клавиатура подключаются непосредственно к встроенному процессору PC 104. На базе Windows – C++ функциональной библиотеки можно подготовить приложения для управления Robotino через WLAN. С помощью WLAN можно напрямую соединиться с системой управления робота. Тогда станет возможной непосредственная передача сигналов управления приводами, контроль показания датчиков, масштабирование и анализ изображения видеокамеры в реальном времени.

Процессы и связи организуются с помощью Robotino View посредством соединения программных функциональных блоков. Программа управления соответствует представлению о движении робота и генерируется «интуитивно». Так достигается простое описание и программирование параллельных процессов (многозадачных режимов).

### **4.2 Порядок выполнения работы**

*Дистанционное управление робота с помощью веб-браузера.*

Чтобы управлять роботом Robotino с использованием веб-браузера, необходимо выполнить следующую последовательность действий:

– запустите Robotino, нажав главный выключатель питания на блоке управления примерно на 3 с;

– извлеките прилагаемый USB-накопитель WLAN из упаковки и вставьте его в одно из шести доступных USB-разъемов на передней панели блока управления Robotino. Как только операционная система распознает USB-накопитель WLAN, инициализируется точка доступа WLAN и запускается веб-сервер, который предоставляет веб-интерфейс для управления Robotino;

– подключите свой смартфон, планшет или записную книжку к указанной сети (например, Robotino.301.064). Сеть защищена паролем Robotino;

– введите пароль Robotino, когда вам будет предложено. Инициализация первого соединения может занять больше времени (до 60 с);

– откройте интернет-браузер и введите IP-адрес Robotino. Роботино веб-интерфейс отображается в вашем интернет-браузере. IP-адрес 172.26.1.1;

– чтобы протестировать Robotino, перейдите на страницу веб-интерфейса программы и запустите одну из демонстрационных программ. В качестве альтернативы вы можете переключиться на управление страницей веб-интерфейса, чтобы управлять перемещением Robotino индивидуально. Убедитесь, что для перемещения Robotino имеется достаточно большая свободная зона;

– нажмите главный выключатель Robotino примерно на 3 с, чтобы выключить систему.

#### *Дистанционное управление робота с помощью Robotino View.*

Чтобы управлять своим Robotino с использованием системы графического программирования Robotino View, необходимо выполнить следующую последовательность действий:

– запустите Robotino, нажав главный выключатель питания на блоке управления примерно на 3 с;

– установите на персональный компьютер Robotino View;


– подключите компьютер или ноутбук через WLAN к Robotino;


– запустите Robotino View на ПК или ноутбуке;

– введите IP-адрес вашего Robotino в поле редактирования на панели инструментов Robotino View (рисунок 4.1). По умолчанию IP-адрес равен 172.26.1.1;

– нажмите Enter или, в качестве альтернативы, нажмите на кнопку со значком антенны непосредственно слева от поля редактирования, чтобы установить соединение (рисунок 4.2). Как только соединение будет успешно инициализировано, кнопка и поле редактирования будут отключены, а качество соединения отобразится в поле рядом с полем ввода IP-адреса. Значение 100 соответствует наилучшему качеству связи;

– откройте один из установленных образцов Robotino View из каталога examples. Вы найдете этот каталог в каталоге установки Robotino View. В качестве альтернативы вы можете создать свою собственную простую программу управления. Для этого воспользуйтесь справочной системой Robotino View;

– нажмите кнопку  для запуска основной программы, чтобы использовать текущую программу в Robotino View для управления соответствующим Robotino;

– нажмите кнопку  для остановки программы управления. Управление немедленно прекращается;

– закройте открытую программу;

– закройте Robotino View;

– нажмите главный выключатель Robotino примерно на 3 с, чтобы выключить систему.

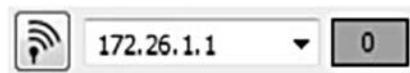


Рисунок 4.1 – Поле редактирования на панели инструментов Robotino View

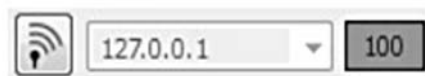


Рисунок 4.2 – Активное подключение Robotino View

### ***Контрольные вопросы***

1 Как организовать дистанционное управление мобильного робота Robotino с помощью веб-браузера?

2 Как организовать дистанционное управление мобильного робота Robotino с помощью среды RobotinoView?

3 Какие движения и функции мобильного робота Robotino подлежат дистанционному управлению?



## 5 Лабораторная работа № 5. Программирование мобильных промышленных роботов в среде симулятора

**Цель работы:** ознакомиться со средой симулятора Robotino SIM.

### 5.1 Основные теоретические положения

Robotino SIM – это программа для Windows, предназначенная для 3D-моделирования Robotino в заранее заданной виртуальной экспериментальной среде (рисунок 5.1). Имитационная модель Robotino включает в себя геометрическую модель с тремя всенаправленными ведущими колесами, камерой, девятью датчиками расстояния и цифровым датчиком в защитной планке шасси.

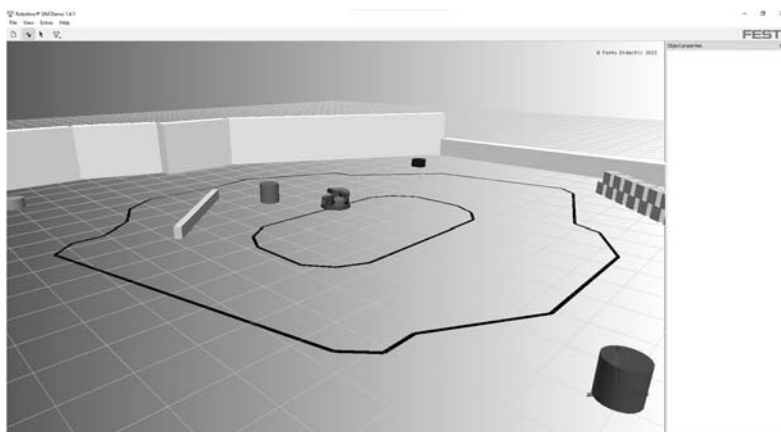


Рисунок 5.1 – Интерфейс программы-симулятора Robotino SIM

Программирование Robotino в среде Robotino SIM возможно при помощи Robotino View или одного из языков высокого уровня, таких как C++ или Java.

### 5.2 Порядок выполнения работы

Основную часть интерфейса занимает виртуальное пространство, в котором может перемещаться и взаимодействовать с объектами компьютерная модель мобильного робота Robotino. В правой части экрана располагается окно свойств объекта, в котором можно регулировать отдельные их параметры. Переключить отображение окна можно путем перехода во вкладку View и поставив или убрав галочку на пункте Show object properties (рисунок 5.2).



Рисунок 5.2 – Переключение отображения окна свойств объекта

Для перемещения камеры в виртуальном пространстве симулятора необходимо нажать клавишу С на клавиатуре или нажать на иконку Move camera на верхней панели интерфейса. Для переключения в режим выбора объектов необходимо нажать на клавишу Q или нажать на иконку Select object.

При выборе виртуальной модели Robotino в окне свойств объекта (рисунок 5.3) появятся следующие параметры:

Connection – параметры соединения с объектом;

Camera – параметры камеры  
Box transport – параметры перемещения объектов роботом;

Battery – параметры батареи.

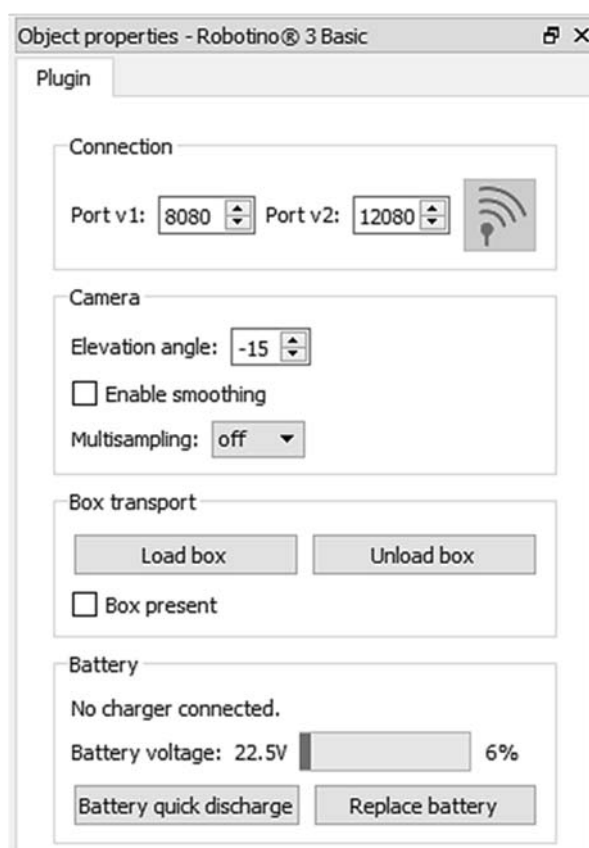



Рисунок 5.3 – Окно свойств виртуальной модели Robotino

Для вызова тестовой версии программы в Robotino View необходимо нажать на иконку  и выбрать одну из доступных в симуляторе программ (рисунок 5.4).

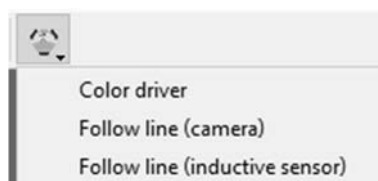



Рисунок 5.4 – Тестовые программы, доступные в Robotino Sim для Robotino view

Для создания нового окна программы и сброса всех изменений относительно начальных значений необходимо нажать сочетание клавиш `ctrl + N` или нажать иконку  .

### ***5.3 Задание***

1 Запустить указанную преподавателем тестовую программу в Robotino view.

2 Во время выполнения программы изменить свойства батареи компьютерной модели Robotino.

3 Обновить рабочую область программы.

### ***Контрольные вопросы***

1 В каком симуляторе возможно программирование мобильного робота Robotino?

2 Какие ограничения наложены на программирование мобильных промышленных роботов в среде симулятора?


3 Для каких задач рационально использовать среду симулятора?


## 6 Лабораторная работа № 6. Программирование мобильного робота Robotino в среде RobotinoView: функции распознавания цветов


**Цель работы:** изучить программирование функций распознавания цветов мобильным роботом Robotino в среде Robotino View.


### 6.1 Основные теоретические положения

Существуют следующие графические блоки для распознавания цветов Robotino:

 (Robotino/Image system/Camera) – блок, подключающий камеру робота;

 (Function block library/Image processing/Segmenter) – блок, присваивающий выбранным цветам порядковый номер;

 (Function block library/Image processing/Segment Extractor) – блок поиска текущего цвета сегмента и получения данных о координатах его центра;

 (Function block library/Image processing/Image Information) – блок выдачи ширины и высоты принимаемого изображения с камеры робота.

### 6.2 Порядок выполнения работы

На рисунке 6.1 представлена структура выполнения главной программы с условиями перехода на следующие шаги.

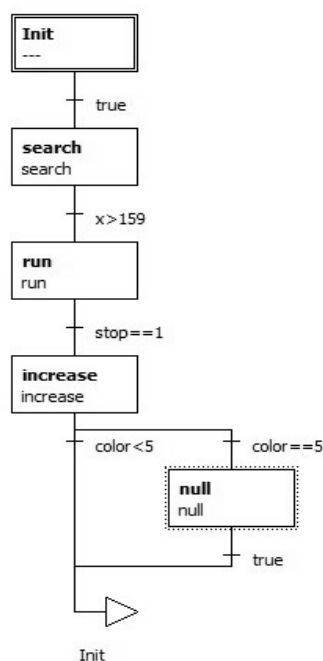


Рисунок 6.1 – Блок-схема главной программы

На шаге search (рисунок 6.2) робот производит поиск текущего цвета сегмента, совершая вращательное движение с заданной угловой скоростью.

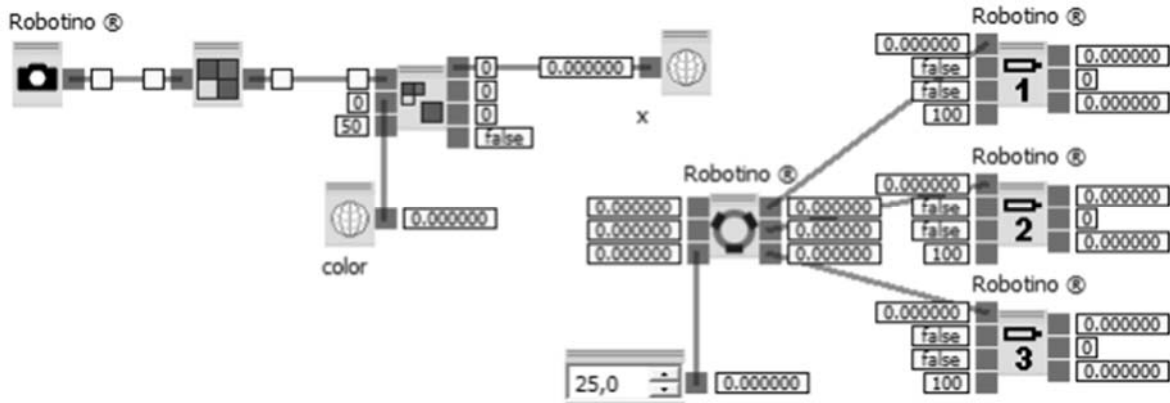


Рисунок 6.2 – Структура шага поиска цвета сегмента (search)

В блоке Segmenter (рисунок 6.3) при включенной симуляции присваиваются порядковые номера цветов всех сегментов. Для этого, во время вращения робота, при попадании в область видимости камеры нужного цвета ставиться флажок Freeze image, затем цвет выделяется мышкой и добавляется в директиву кнопкой '+'. Аналогично для других цветов.

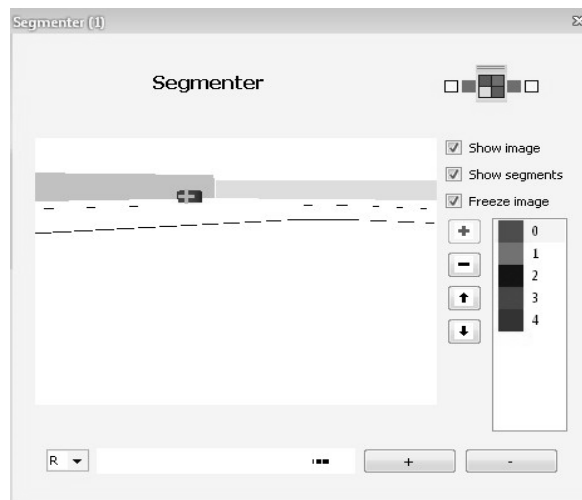


Рисунок 6.3 – Содержимое блока Segmenter

Изображение найденного сегмента с текущим порядковым номером цвета подводится к Segment Extractor (рисунок 6.4) к сигналу Input (Image).

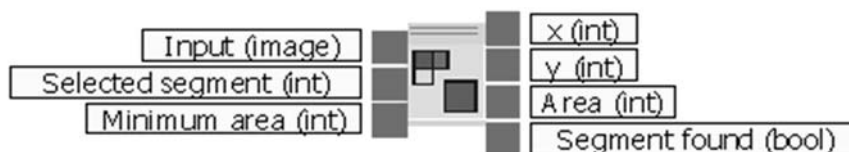


Рисунок 6.4 – Описание входных и выходных сигналов блока Segment Extractor

Input (image) – входящее изображение с Segmenter; Selected segment (int) – порядковый номер текущего цвета сегмента; Minimum area (int) – минимальная площадь сегмента для обнаружения; x (int) – текущая координата положения центра сегмента по оси X; y (int) – текущая координата положения центра сегмента по оси Y; Area (int) – текущая площадь сегмента; Segment found (bool) – логическая переменная, которая сообщает, что сегмент найден.

Канал Selected segment считывает текущий порядковый номер сегмента с переменной color. Текущая координата положения центра сегмента по оси X записывается в переменную x.

На следующий шаг gun программа переходит при условии, если  $x > 159$ , где 159 – это половина ширины входящего изображения с камеры (в данном случае 320 пикселей). Для камер с другим расширением эта величина будет определяться по формуле

$$\frac{\text{ширина изображения (в пикселях)}}{2} - 1. \quad (1)$$

В шаге gun (рисунок 6.5) константой задается скорость движения по оси X и сопоставляется центр изображения камеры с центром текущего сегмента, регулируя тем самым положение робота относительно оси Y.

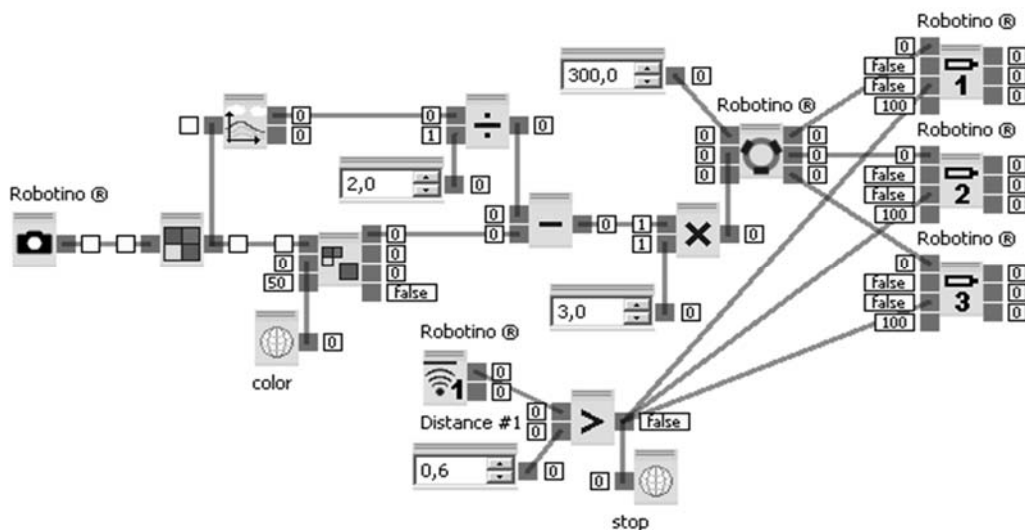


Рисунок 6.5 – Структура шага движения робота к текущему цвету сегмента

По мере приближения робота к объекту на датчик поступает сигнал. Когда сигнал с датчика превышает заданную константу, результат сравнения записывается в переменную stop. Условием перехода на следующий шаг является равенство этой переменной значению 1 (true).

В шаге increase (рисунок 6.6) к текущему значению color прибавляется единица, тем самым меняя искомый цвет.

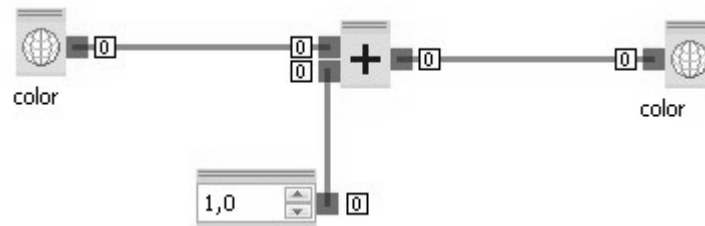


Рисунок 6.6 – Структура шага изменения искомого цвета сегмента

Затем в main program происходит разветвление по двум условиям. Если порядковый номер цвета меньше 5 (число заданных цветов), то программа возвращается в начало и выполняет вышеуказанные шаги для следующего цвета. Если порядковый номер цвета равен 5, то программа попадает на шаг null (рисунок 6.7), который обнуляет переменную color. Затем программа производит повторный поиск начального цвета.

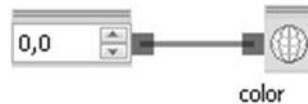


Рисунок 6.7 – Структура шага обнуления цвета сегмента

### 6.3 Задание

- 1 Составить программу по примеру, проверить работоспособность в Robotino SIM или на реальном роботе.
- 2 Составить свою программу, поменяв порядок поиска цветов роботом.

### Контрольные вопросы

- 1 В каком формате задаются цвета объектов в среде Robotino View?
- 2 Какой сенсор используется для определения цвета объектов?
- 3 Для чего необходимо определять цвета объектов?

## 7 Лабораторная работа № 7. Программирование мобильного робота Robotino в среде Robotino View: функции анализа изображений

**Цель работы:** изучить новые функциональные блоки и элементы среды Robotino View. Составить программу для автоматического перемещения мобильного робота по линии с помощью камеры.

### 7.1 Основные теоретические положения

Камера предоставляет прямое изображение происходящего, зафиксированное на данный момент времени. Камера подключена к надстройке управления через один из шести USB-разъемов. Функциональный модуль Camera (Камера) в Robotino View позволяет получать доступ к изображениям с веб-камеры, установленной на Robotino.

Настройки разрешения можно задавать в Robotino device dialogue (диалоговом окне устройства Robotino). В поле списка справа можно выбрать одно из возможных разрешений (320×240, 640×480, 800×600). В случае с распознаванием цветов обычно предпочтительнее более низкое разрешение, т. к. в этом случае изображение содержит меньше различных данных о цвете. Это позволяет упростить для программного обеспечения задачу распознавания связанных цветовых участков.

Robotino View предлагает большой выбор функциональных блоков для обработки изображения (рисунок 7.1).

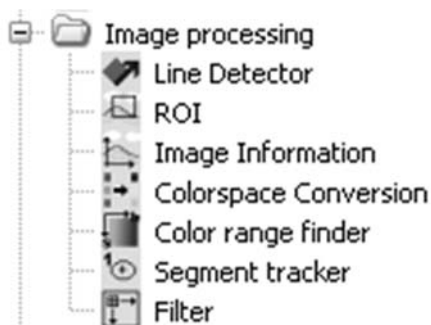



Рисунок 7.1 – Функциональные блоки для обработки изображения


### 7.2 Порядок выполнения работы

В данной лабораторной работе необходимо создать программу, которая позволит роботу поочередно искать и перемещаться по линии с помощью камеры. Для этого нам необходимы следующие графические блоки:

 (Robotino/Image system/Camera) – блок, подключающий камеру робота;



 (Function block library/Image processing/Segmenter) – блок, определяющий линию на входном изображении;

 (Function block library/Image processing/Image Information) – блок выдачи ширины и высоты принимаемого изображения с камеры робота.

В данном случае программа будет состоять лишь из одного шага (рисунок 7.2).

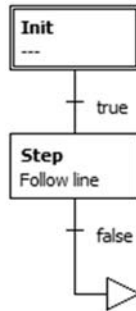


Рисунок 7.2 – Блок-схема главной программы

В шаге Follow line будет выполняться поиск линии и управление движением робота вдоль нее.

Блок обнаружения линии (line detector) определяет линии, как это показано на рисунках 7.3 и 7.4. Область поиска линии отмечена горизонтальными красными линиями. В нижней строке отмечается «Начало поиска» (search start). В верхней строке отмечаются «Начало поиска» + «Высота поиска (search height)».

Красный + отмечает край линии от темного к светлому, если смотреть слева направо.

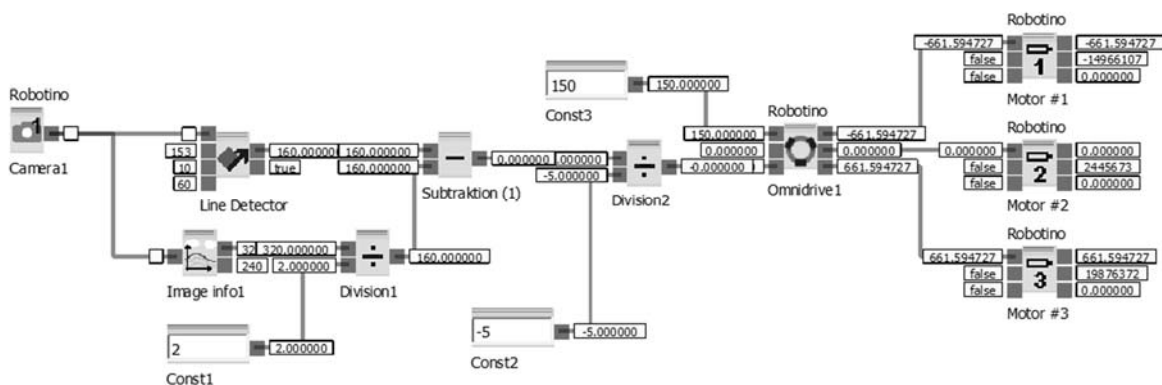


Рисунок 7.3 – Структура шага следования за линией (Follow line)

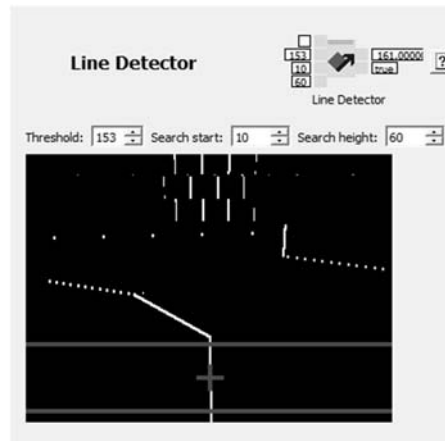


Рисунок 7.4 – Визуализация работы блока обнаружения линии (line detector)

Блок выделения области интереса (ROI/ Region Of Interest) позволяет задавать область интереса на входящем изображении в пикселях. Разрешение  $320 \times 240$  (рисунок 7.5). В данном случае областью интереса является всё изображение, приходящее с камеры.

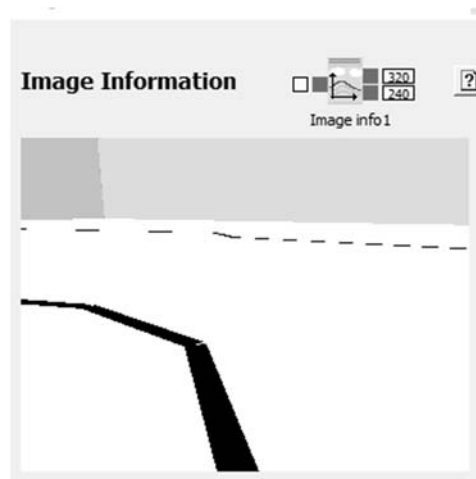


Рисунок 7.5 – Визуализация работы блока выделения области интереса (ROI)

По мере отклонения робота от движения по линии будет изменяться положение красного плюса блока обнаружения линии от центра, что будет приводить к изменению сигнала на блок управления движением робота, а именно в часть этого блока, отвечающую за поворот робота. В результате, когда робот отклоняется от движения по линии, блок управления будет выполнять поворот до тех пор, пока красный плюс детектора линий снова не окажется по центру камеры, что будет означать, что робот двигается прямо по линии. На рисунке 7.6 представлен пример работы данной программы, в которой робот движется по ближайшей обнаруженной линии.

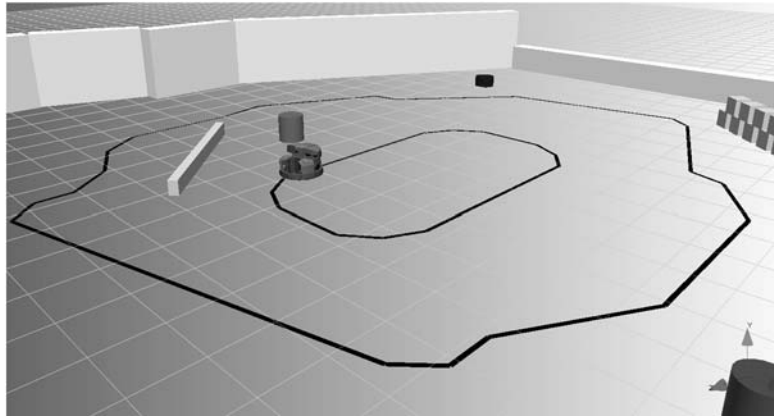


Рисунок 7.6 – Следование роботом по линии в среде Robotino SIM

### ***7.3 Задание***

1 Составить программу по примеру, проверить работоспособность в Robotino SIM или на реальном роботе.

2 Составить свою программу, в которой робот перед началом следования по линии переместился бы к внешней линии и начал бы двигаться по большому кругу.

### ***Контрольные вопросы***

1 Какие функции анализа изображений предусмотрены в среде Robotino View?

2 Для чего необходим оператор Line Detector?

3 Для чего необходим оператор Marker Detector?

## 8 Лабораторная работа № 8. Программирование мобильного робота Robotino в среде RobotinoView: функции перемещений

**Цель работы:** изучить новые функциональные блоки и элементы среды Robotino View. Составить программу для автоматического перемещения мобильного робота по заданному алгоритму.

### 8.1 Основные теоретические положения

Включение всенаправленного привода является достаточно сложной задачей. При использовании всенаправленного привода перемещение возможно только посредством трех управляемых колес (рисунок 8.1). В случае применения всенаправленного привода включаются три приводных двигателя. Если предположить, что существует три управляющие команды для включения двигателя (вперед, назад, выключен), то только эти три возможности позволяют составить 27 вариантов для трех двигателей. Эти 27 вариантов управления позволяют управлять движением робота в различных направлениях.

На данном этапе не рассматривается команда бокового управления, имеющая место в работе с Robotino.

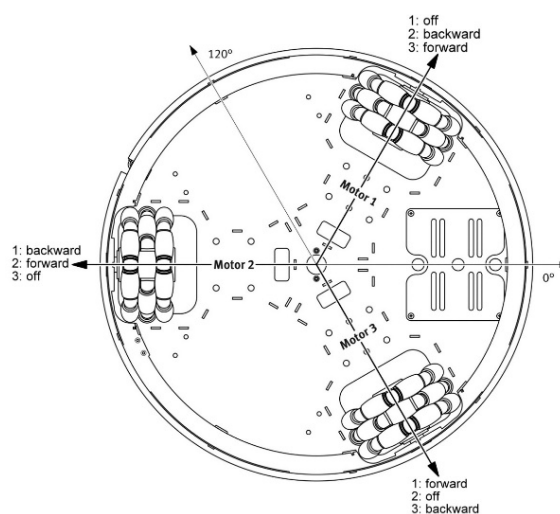


Рисунок 8.1 – Схема перемещения посредством трех управляемых колес

#### *Включение трех двигателей Robotino.*

Robotino оборудован тремя приводными двигателями для приведения в действие трех разнонаправленных колес. Скорость вращения отдельного колеса можно задать независимо от скорости вращения остальных колес. Чтобы двигать Robotino в одном направлении, надо согласовать вращение отдельных колес, т. к. только в случае правильного взаимодействия получается движение Robotino в необходимом направлении.

Двигатели можно включать или напрямую через Robotino View, или при помощи функционального блока Omnidrive. Однако для всех случаев рекомендуется применять функциональный блок Omnidrive (Robotino® View).

## 8.2 Порядок выполнения работы

Папка Drive system содержит функциональные блоки для управления системой привода Robotino.

Блоки двигателей (Motor). Каждый из блоков дает доступ к управлению одним из двигателей мобильного робота Robotino (рисунок 8.2). Номер двигателя, к которому предоставляется доступ, указан на самом блоке (рисунок 8.3).

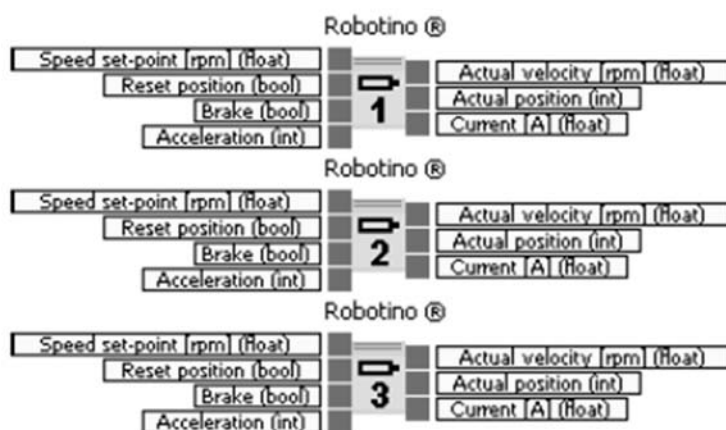


Рисунок 8.2 – Блоки двигателей Robotino

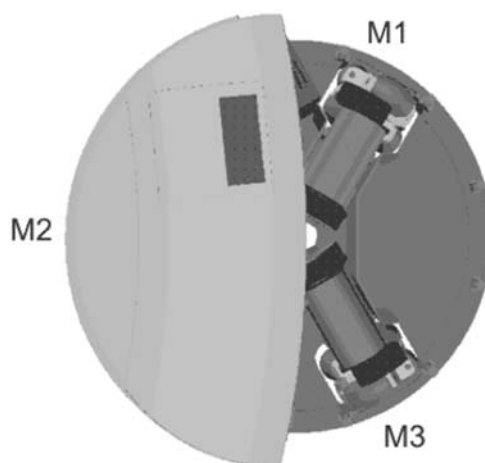


Рисунок 8.3 – Схема расположения двигателей

Управление скоростью каждого двигателя осуществляется с помощью ПИД-регулятора.

Функциональный блок Omnidrive.

В блоке Omnidrive рассчитываются заданные скорости вращения для двигателей 1, 2 и 3 в направлениях  $x$  и  $y$  на основании заданной скорости. Кроме того, фактическая скорость в направлении  $x$  и  $y$ , а также фактическая скорость вращения рассчитываются из фактической скорости (рисунок 8.4).

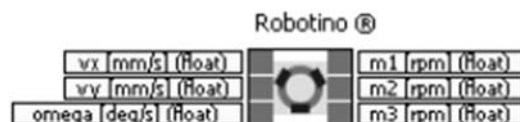


Рисунок 8.4 – Функциональный блок Omnidrive

У Robotino две поступательные степени свободы, движение по оси  $x$  и  $y$  для движения прямо вперед и боком (рисунок 8.5). Кроме того, возможно вращение вокруг оси  $z$ .

Входы и выходы модуля omni заняты соответствующим образом.

Входы:

$v_x\_float$  – заданная скорость в направлении  $x$  в локальной системе координат Robotino, мм/с;

$v_y\_float$  – заданная скорость в направлении  $y$  в локальной системе координат Robotino, мм/с;

$\omega\_float$  – заданная скорость вращения, град/с.

Выходы:

$m_1\_float$  – заданное значение скорости двигателя 1, мин<sup>-1</sup>;

$m_2\_float$  – заданное значение скорости двигателя 2, мин<sup>-1</sup>;

$m_3\_float$  – заданное значение скорости двигателя 3, мин<sup>-1</sup>.

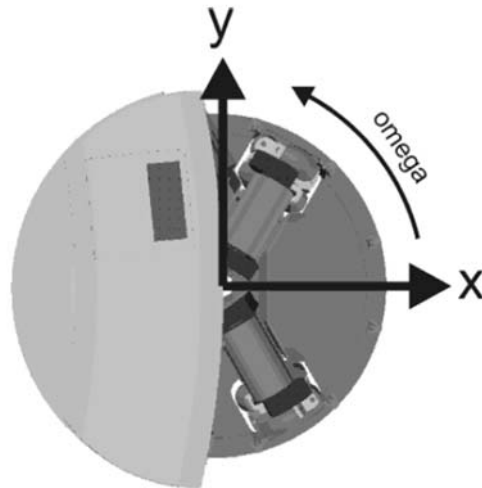


Рисунок 8.5 – Локальная система координат Robotino

Направление (координаты) и заданная скорость определяются, вводятся и рассчитываются на входе функционального блока omnidrive и преобразовываются в заданные значения для соответствующей скорости двигателя (см. рисунок 8.4).

Заданные скорости для трех двигателей выводятся на выходе функционального блока omnidrive. Эти заданные скорости позволяют

реализовать соответствующие фактические скорости в направлении x и y или скорость вращения.

По ключевому слову omnidrive в справке Robotino View можно найти информацию о том, как в результате преобразования вектора заданной скорости и заданной скорости вращения можно получить заданную скорость отдельных двигателей.

### ***8.3 Задание***

- 1 Составить и реализовать в среде программирования Robotino View алгоритм движения робота Robotino по заданной преподавателем траектории.
- 2 Проверить работоспособность в Robotino SIM или на реальном роботе.

### ***Контрольные вопросы***

- 1 Какие функции перемещений предусмотрены в среде Robotino View?
- 2 Для чего необходим оператор Motor #1?
- 3 Для чего необходим оператор Drive System?

## Список литературы

1 **Москвичев, А. А.** Захватные устройства промышленных роботов и манипуляторов / А. А. Москвичев, А. Р. Кварталов, Б. В. Устинов. – Москва: ФОРУМ; ИНФРА-М, 2015. – 176 с.

2 **Иванов, А. А.** Основы робототехники: учебное пособие / А. А. Иванов. – Москва: ФОРУМ, 2014. – 224 с.

3 **Выжигин, А. Ю.** Гибкие производственные системы: учебное пособие для вузов / А. Ю. Выжигин. – Москва : Машиностроение, 2009. – 288 с.

4 **Схиртладзе, А. Г.** Автоматизация производственных процессов в машиностроении: учебник для вузов / А. Г. Схиртладзе, В. Н. Воронов, В. П. Борискин. – Старый Оскол : ТНТ, 2011. – 612 с.

5 **Фельдштейн, Е. Э.** Автоматизация производственных процессов в машиностроении: учебное пособие / Е. Э. Фельдштейн, М. А. Корниевич. – Москва: ИНФРА-М ; Новое знание, 2016. – 264 с.

6 Роботизированные производственные комплексы / Под ред. Ю. Г. Козырева, А. А. Кудинова. – Москва: Машиностроение, 1987. – 272 с.

7 **Довбня, Н. М.** Роботизированные технологические комплексы в ГПС / Н. М. Довбня, А. Н. Кондратьев, Е. И. Юревич. – Ленинград: Машиностроение, 1990. – 303 с.