ТЕХНОЛАБ

ОБРАЗОВАТЕЛЬНЫЙ РОБОТОТЕХНИЧЕСКИЙ МОДУЛЬ

(ИССЛЕДОВАТЕЛЬСКИЙ УРОВЕНЬ)

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ

(ИССЛЕДОВАТЕЛЬСКИЙ УРОВЕНЬ)

К. В. Ермишин
М. А. Кольин
Д. Н. Каргин
А. О. Панфилов

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ

ОБРАЗОВАТЕЛЬНЫЙ
РОБОТОТЕХНИЧЕСКИЙ МОДУЛЬ

(ИССЛЕДОВАТЕЛЬСКИЙ УРОВЕНЬ)
ОТ 14 ЛЕТ

Учебно-методическое пособие

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

ЭКЗАМЕН ТЕХНОЛАБ
Издательство ЭКЗАМЕН®

МОСКВА
2014

УДК 372.8:004
ББК 32.816
Е73

Ермишин К. В.

Е73 Методические рекомендации для преподавателя: образовательный робототехнический модуль (исследовательский уровень): от 14 лет / К. В. Ермишин, М. А. Кольин, Д. Н. Каргин, А. О. Панфилов. — М. : Издательство «Экзамен», 2014. — 256 с.

ISBN 978-5-377-07627-8

Данное пособие предназначено для применения совместно с образовательным робототехническим модулем «Исследовательский уровень». В пособии описываются возможности робототехнического модуля и области его применения. Пособие содержит информацию о назначении робототехнического набора и описание работ по проектированию роботов и робототехнических устройств, которые можно провести совместно с учащимися среднего и старшего школьного возраста. Пособие демонстрирует возможности применения образовательного робототехнического модуля при решении исследовательских и технически сложных задач. Особенность данного образовательного робототехнического модуля заключается в решении нестандартных и технически сложных задач в рамках проектной и исследовательской деятельности учащихся в наглядной и игровой форме. Применение данного модуля позволяет затронуть в рамках образовательного процесса различные технически сложные проблемы из области робототехники, такие как управление и передача данных в технических системах, управление многозвенными механизмами, основы работ с техническим зрением и многое другое. Данное пособие носит рекомендательный характер и тем самым не ограничивает возможности применения робототехнических наборов в образовательной и учебной деятельности, но в свою очередь демонстрирует наиболее яркие примеры проектов и работ, которые возможно реализовать благодаря использованию образовательного робототехнического модуля «Исследовательский уровень».

УДК 372.8:004
ББК 32.816

Подписано в печать с диапозитивов 15.10.2013.
Формат 60x90/8. Гарнитура «Calibri». Бумага офсетная.
Усл. печ. л. 32. Тираж 1000 экз. Заказ №

ISBN 978-5-377-07627-8

© Ермишин К. В., Кольин М. А., Каргин Д. Н., Панфилов А. О., 2014
© Издательство ***«ЭКЗАМЕН»***, 2014
© ***«ЭКЗАМЕН-ТЕХНОЛАБ»***, 2014

Содержание

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Введение

Мировые тенденции развития инженерного образования свидетельствуют о глобальном внедрении информационных технологий в образовательный процесс. Робототехника является весьма перспективной областью для применения образовательных методик в процессе обучения за счет объединения в себе различных инженерных и естественнонаучных дисциплин. В результате такого подхода наблюдается рост эффективности восприятия информации учащимися за счет подкрепления изучаемых теоретических материалов экспериментом в междисциплинарной области.

Данное учебное пособие представляет собой методические рекомендации, раскрывающие возможности и особенности применения образовательного робототехнического модуля «Исследовательский уровень». Образовательный робототехнический модуль «Исследовательский уровень» предназначен для углубленного изучения принципов проектирования роботов и робототехнических устройств на примере эксперимента, который можно без особого труда выполнить в рамках индивидуальных или групповых занятий.

Образовательный модуль позволяет конструировать модели роботов, робототехнических устройств и производственных механизмов различной сложности. Каждая из моделей предназначена для проведения ряда лабораторных занятий по изучению различных приемов программирования и теоретических аспектов проектирования. Многообразие различных экспериментов позволяет анализировать и систематизировать полученные результаты, благодаря чему развиваются навыки работы с информацией и основы исследовательской деятельности.

Предлагаемые лабораторные работы и эксперименты, проводимые в их рамках, основываются на моделях роботов, которые можно сконструировать на базе робототехнического конструктора Bioloid и дополнительных компонентов, производимых корейской компанией ROBOTIS. Наборы данной серии отличает многообразие возможностей, позволяющих реализовать всевозможные задумки начинающих исследователей.

Образовательный робототехнический модуль «Исследовательский уровень» представляет собой специализированный комплект для развития навыков работы с робототехническими модулями, как в модуле «Профессиональный уровень», так и для разработки различных роботов под управлением сложных алгоритмов, требующих от разработчика творческого подхода, системного анализа и мышления.

Наличие программируемого блока управления и различных датчиков позволяет сделать полностью автономные модели роботов, которыми также можно управлять вручную с помощью пульта дистанционного управления и модулей беспроводной связи. Широкий спектр доступных компонент и возможностей позволяют пользователю на практике ознакомиться с принципом функционирования различных приводов, контроллеров и сенсорных устройств, а также разработать для них собственную систему управления.

Использование решений из области робототехники в рамках образовательного процесса позволяет формировать технологическую и проектную культуру учащихся, которые также не останутся равнодушными к столь увлекательному образовательному процессу.

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Основы изучения среды программирования RoboPlus

ЭКЗАМЕН ТЕХНОЛАБ

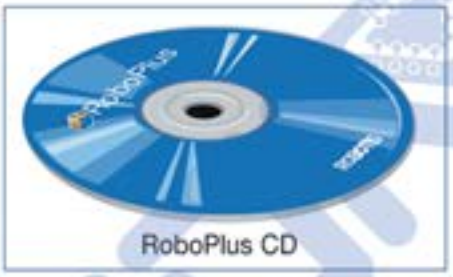
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ
EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXCHENGETECHNOLAB.RU

Основы изучения среды программирования RoboPlus

Разработка и программирование систем управления – это неотъемлемая часть процесса проектирования робототехнических систем. В настоящее время большинство устройств и механизмов управляются благодаря микропроцессорным устройствам, выполняющим какую-либо программу. Программирование подобных устройств осуществляется в специальных средах для разработки. Как правило, крупные производители микроэлектроники и контроллеров выпускают собственные среды разработки или поддерживают наиболее популярные среди разработчиков программные пакеты.

В нашем случае для программирования робототехнического модуля «Базовый уровень» применяется среда разработки RoboPlus.



RoboPlus CD

Установка RoboPlus

- Установка RoboPlus осуществляется с диска, входящего в состав модуля.
- Вставьте диск в дисковод и откройте корневую папку диска.
- Выберите и запустите файл Setup.exe и запустите процесс установки.
- Выберите в процессе установки английский язык (English).
- Для русификации RoboPlus скопируйте файлы из папки «Русификатор» в папку установки программы.
- В случае удачной установки на рабочем столе вашего ПК появится ярлык RoboPlus.

Запуск RoboPlus

Произведите запуск RoboPlus, использовав ярлык на рабочем столе. На экране появится окно, содержащее информацию о программах, входящих в состав среды разработки. Обратите внимание, что от версии ПО или года выпуска робототехнического модуля содержание окна может немного отличаться, но это ни каким образом не сказывается на работоспособности набора и возможностях среды разработки.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Вкладки стартового окна содержат информацию о наборах, которые можно программировать в RoboPlus. Среда разработки RoboPlus поддерживает все наборы компании ROBOTIS, на базе которых разработаны образовательные робототехнические модули «Предварительный уровень», «Начальный уровень», «Базовый уровень», «Профессиональный уровень», «Исследовательский уровень».

Состав RoboPlus

В состав среды разработки RoboPlus входят специальные программы, предназначенные для настройки различных устройств, входящих в состав робота; программирования и управления роботами.

RoboPlus Task

RoboPlus Manager

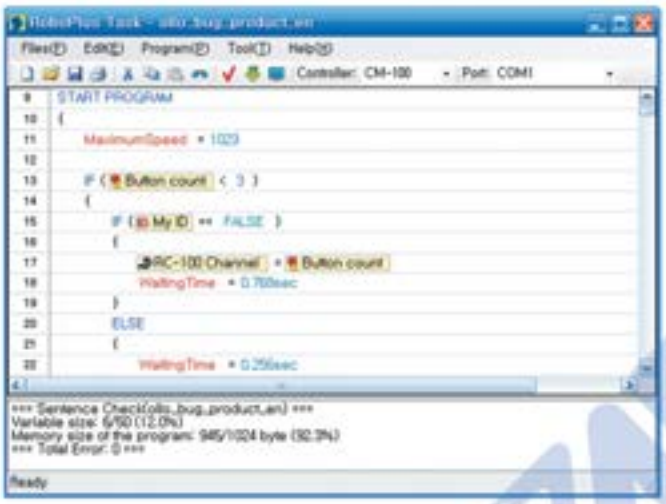
RoboPlus Motion

RoboPlus Terminal

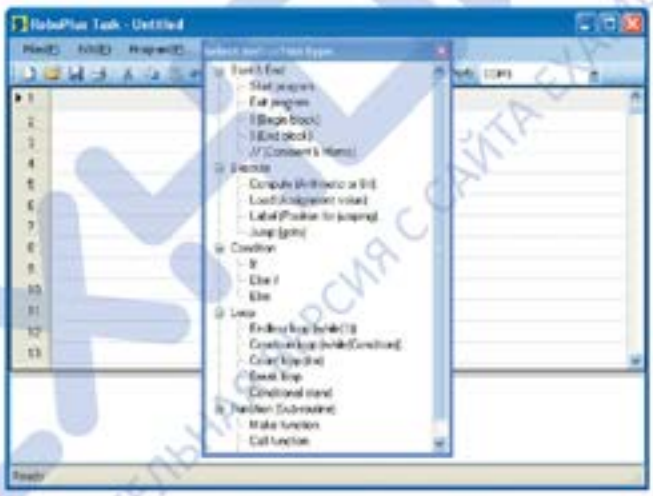
Dynamixel Wizard

RoboPlus Task – программная среда для написания и редактирования управляющих программ. Данная программа является основным инструментом для разработки программ для образовательных робототехнических модулей.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMENTECHNOLAB.RU



Программирование в RoboPlus Task осуществляется с помощью специализированного языка, подобного языку программирования С. Для удобства пользователя в RoboPlus в виде графических блоков реализованы базовые возможности набора, такие так таймеры, блоки обработки данных с датчиков, блоки передачи данных между устройствами и т.п.



В языке среды RoboPlus все служебные слова, программные блоки, команды и комментарии располагаются в строках. Для того чтобы активировать строку, по ней нужно нажать дважды. После нажатия появится диалоговое окно, позволяющее выбрать различные команды языка.

После выбора определенного набора команд необходимо заполнить пустующие элементы или условия выполнения команд. В пустующие места строк устанавливаются необходимые программируемые блоки. Программируемые блоки выбираются в зависимости от типа программируемого контроллера и числа подключенных внешних устройств. Перечень блоков определяется автоматически и обновляется при подключении новых устройств.

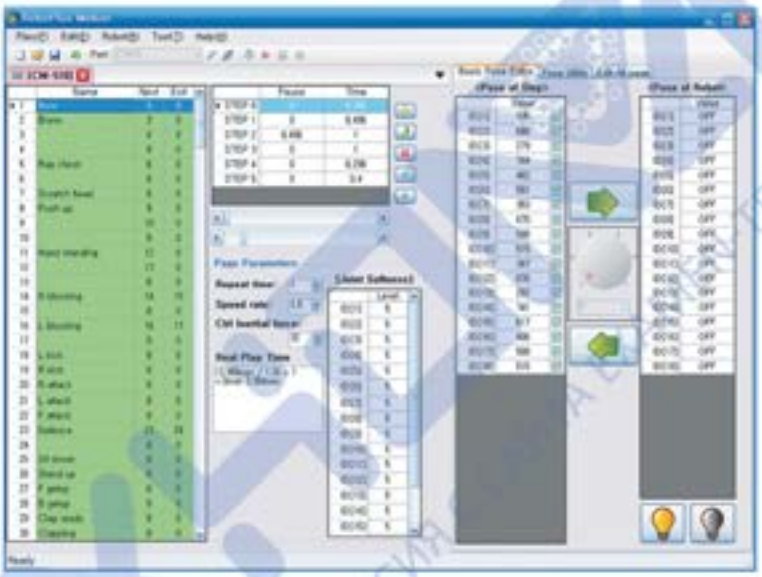
Весь процесс написания программы в RoboPlus Task сводится к дальнейшей загрузке полученных результатов в программируемый контроллер. Для того чтобы компьютер определил тип программируемого контроллера и порт, к которому он подключен, необходимо воспользоваться функцией автоматического поиска.

После того как программа определит тип программируемого контроллера, можно произвести компиляцию программы, тем самым проверить ее на наличие ошибок и подготовить к загрузке в программируемый контроллер.

Полученную программу можно загрузить в контроллер робота, и она запустится сразу же после подачи на него питания.

RoboPlus Manager – программа для настройки оборудования, входящего в состав робототехнических конструкторов ROBOTIS. С помощью данной программы RoboPlus обновляет собственные файлы и производит тестирование оборудования, подключенного к компьютеру в данный момент при помощи контроллера или специализированных переходников. Благодаря использованию RoboPlus Manager возможно изменять параметры контроллера, сервоприводов, производить настройку коммуникационных устройств и т.п.

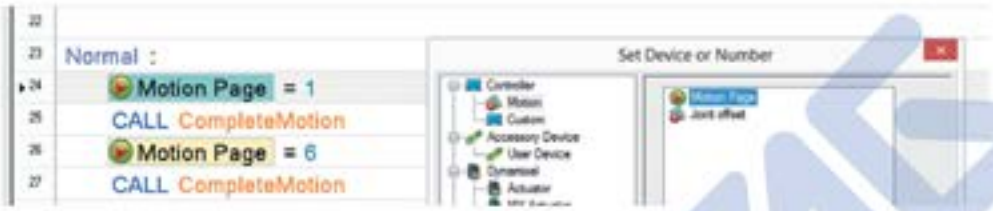
RoboPlus Motion – среда программирования сложных движений робота. Благодаря RoboPlus Motion можно запрограммировать различные действия робота, а после использовать их в основной программе. Зачастую в процессе движения робота участвует множество различных приводов и задать их скорости вращения и углы поворотов вслепую крайне затруднительно.



RoboPlus Motion позволяет промоделировать движение в процессе написания управляющей программы. В специальном окне RoboPlus Motion отображаются все приводы, подключенные в данный момент к роботу. Пользователь в режиме реального времени может задать для каждого из приводов скорость и угол поворота, а после запустить программу и увидеть результат ее работы на реальном роботе. Таким образом, можно разработать программу, реализующую сложное движение робота.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ
WWW.EXAMEN-TECHNOLAB.RU

Для того чтобы загрузить файл движений робота в память контроллера, воспользуйтесь меню Download Motion вкладки Robot. Очень важно помнить о том, что файл движений (Motion-файл) должен загружаться в память робота до загрузки файла управляющей программы (Task-файл). Нарушение данной последовательности может привести к сбою процесса компиляции основной программы.



Для того чтобы в основной программе использовать заранее разработанное движение, необходимо воспользоваться программируемым блоком Motion Page. В параметрах данного блока следует установить номер строки Motion-файла, отвечающей за необходимое движение.

RoboPlus Terminal – программа, предназначенная для получения и отправки данных посредством терминала операционной системы компьютера. Применяется для отладки управляющих алгоритмов, например для вывода на экран показаний датчиков и т.п., т.е. для отображения той информации, к которой пользователь обычно не имеет доступа в процессе выполнения программы.

Dynamixel Wizard – программа, предназначенная для настройки и калибровки сервоприводов Dynamixel. С помощью данной программы для каждого из приводов можно задать ограничения скоростей вращения и углов поворота, а также получить код ошибки, препятствующей работе устройства.

Среда разработки RoboPlus содержит в себе все необходимые инструменты для программирования робототехнических наборов на базе конструкторов фирмы ROBOTIS. С ее помощью можно программировать модели на базе наборов серии OLLO, Bioloid, а также отдельные устройства, например сервопривода Dynamixel и модули беспроводной связи.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА WWW.TECHNOLAB.RU

Лабораторная работа № 1

ЭКЗАМЕН ТЕХНОЛАБ

Управление положением вала сервопривода с помощью кнопок программируемого контроллера

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 1

«Управление положением вала сервопривода с помощью кнопок программируемого контроллера»

Управление роботами и робототехническими системами, а также различными производственными механизмами осуществляется за счет различных технических средств. Технические средства, приводящие в движение механизмы роботов, называются исполнительными устройствами, а механизмы на их основе, выполняющие определенные узкоспециализированные задачи, – исполнительными механизмами.

Сервопривод – наиболее часто встречающийся исполнительный механизм в робототехнических устройствах. Обычно сервопривод представляется собой устройство, состоящее из привода, редуктора или механической передачи, системы управления и датчиков обратной связи. Система управления и датчики обратной связи применяются в сервоприводах для регулирования основных рабочих параметров – скорости вращения вала и его положения.

Целью данной лабораторной работы является изучение принципов работы с сервоприводами. Для этого в рамках данной работы предлагается разработать модель автоматизированного шлагбаума с приводом качения на базе Dynamixel AX-12A. Управление данной моделью предлагается осуществлять с помощью кнопок программируемого контроллера.

Таким образом, алгоритм поставленной задачи сводится к двум операциям:

1) При нажатии кнопки U программируемого контроллера – осуществить открытие шлагбаума.

2) При нажатии кнопки D программируемого контроллера – осуществить закрытие шлагбаума.

Движение шлагбаума осуществляется от одного крайнего положения к другому, задаваемому пользователем в начале программы. Необходимо помнить, то положение шлагбаума определяется положением выходного вала сервопривода, которое в свою очередь рассчитывается в системе координат сервопривода относительно начального положения.

Управляющая программа начинается с инструкций базовой инициализации. В первую очередь проверяется правильность установки идентификационного номера сервопривода ID=1 и в случае неполадок вызывается функция AssemblyError. После чего задается величина ограничения угла поворота путем записи соответствующего значения по адресу 0x08.

```
4:  START PROGRAM
5:  {
6:      IF ( ID[1]: ADDR[3(b)] != 1 )
7:          CALL AssemblyError
8:      // Dynamixel ID check
9:      IF ( ID[1]: ADDR[8(w)] == 0 )
10:         ID[1]: ADDR[8(w)] = 1023
11:     // Dynamixel mode check (joint or wheel)
12:     IF ( Button == D )
13:         JUMP AssemblyCheckMode
14:     // Assembly check
15:     // Default values
16:     OpenInitial = 619
17:     ClosedInitial = 512
18:     ID[1]: Moving speed = 100
19:     CALL InitialPosition
```

Примечание: данный пример содержит функцию AssemblyCheckMode, предназначенную для проверки правильности сборки механизма и настройки приводов. Данная функция проверяет правильность настроек идентификационных адресов ID сервоприводов, отсутствие ошибок и аварийных сообщений.

Положения шлагбаума задаются с помощью переменных OpenInitial для открытого состояния и ClosedInitial для закрытого состояния. После этого задается скорость вращения привода и подается команда на переход механизма в начальное положение с помощью вызова функции InitialPosition.

Управление приводом подъема шлагбаума реализовано в бесконечном цикле. В цикле анализируется, какая из кнопок контроллера нажата в данный момент, и в зависимости от этого изменяется значение переменной, характеризующей целевое положение привода.

```
21:     ENDLESS LOOP
22:     {
23:         // Open when U is pressed
24:         IF ( Button == U )
25:             ID[1]: Goal position = OpenInitial
26:         // Closed when D is pressed
27:         IF ( Button == D )
28:             ID[1]: Goal position = ClosedInitial
```

В случае если нажата кнопка U, значению целевой позиции Goal Position сервопривода присваивается значение OpenInitial. Если же нажата кнопка D, то значению целевой позиции присваивается значение переменной ClosedInitial.

Управление кнопками программируемого контроллера осуществляется с помощью панели управления. Во вкладке Controller необходимо выбрать блок Button, который можно настроить на срабатывание одной из кнопок программируемого контроллера.

Отдельно стоит рассмотреть функцию InitialPosition, переводящую шлагбаум в начальное положение. При вызове функции шлагбаум переводится в открытое положение, а после в закрытое.

```
112: FUNCTION InitialPosition
113: {
114:     ID[1]: Goal position = OpenInitial
115:     WAIT WHILE ( ID[1]: Moving == TRUE )
116:     ID[1]: Goal position = ClosedInitial
117: }
```

Подобная последовательность операций вызвана необходимостью гарантированного закрытия шлагбаума. Рассмотрим работу функции более подробно – сервоприводу задается команда перемещения в положение, при котором шлагбаум открыт. Как только команда поступает на исполнение, сервопривод начинает вращать вал в течение определенного времени. Если во время вращения вала привода поступит новая команда на перемещение, это может нарушить процесс работы механизма и конечное положение не будет достигнуто. Для того чтобы во время движения шлагбаума никакие другие команды не препятствовали этому процессу, используется вынужденная задержка, реализуемая с помощью конструкции WAIT WHILE. Смысл подобной конструкции сводится к следующему – пока сервопривод находится в движении, управляющая программа находится в вынужденной остановке. Как только вал привода достигнет целевого положения и закончит свое движение, выполнение программы перейдет на следующую операцию, а именно: перевод шлагбаума в закрытое положение. Таким образом, становится возможным точно позиционировать привод в целевом положении.

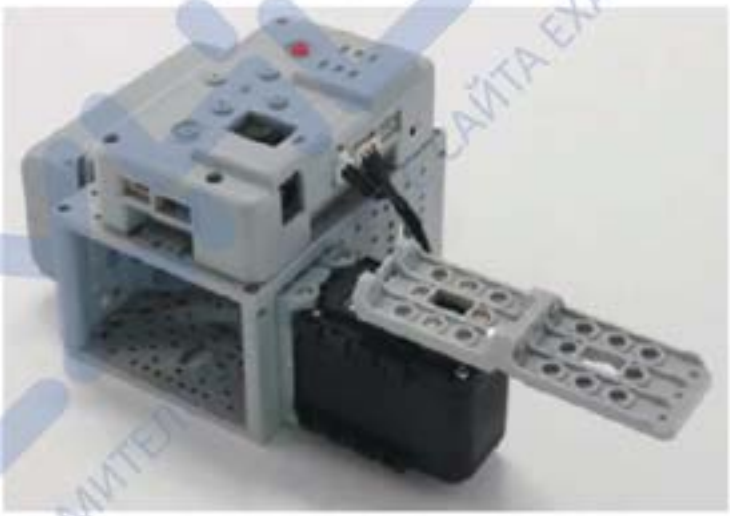
В рамках данной работы были рассмотрены основы управления сервоприводом Dynamixel AX-12A, была разработана программа, управляющая положением шлагбаума с помощью нажатий на кнопки программируемого контроллера.

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНСК.ТЕХНОЛАБ.RU

Лабораторная работа № 2

ЭКЗАМЕН ТЕХНОЛАБ

Управление скоростью вала сервопривода с помощью кнопок программируемого контроллера



ОЗНАКОМИТЕЛЬНЫЙ САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 2

«Управление скоростью вала сервопривода с помощью кнопок программируемого контроллера»

Зачастую различные производственные процессы нуждаются в поддержании ряда технологических параметров в пределах заданного диапазона. Поскольку значительная часть всех автоматизированных технологических процессов так или иначе находится под управлением электропривода, на первый план теории автоматического управления выходит вопрос регулирования скорости вращения вала электропривода, входящего в состав различных исполнительных механизмов.

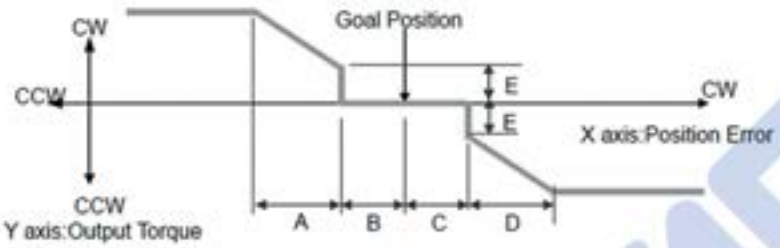
Регулирование скорости привода осуществляется благодаря использованию следящих систем управления, включающих в себя датчики обратной связи, измеряющие отклонение скорости от заданного значения. В результате оценки реальной скорости вращения вала привода система управления может компенсировать величину ошибки, ускорив или замедлив вращение.

Регулирование скорости дает возможность осуществлять плавный разгон и торможение робота как при постоянном вращении привода, так и при переходе в целевое положение. Также немаловажен тот факт, что плавный разгон и торможение существенно снижают нагрузку на исполнительные механизмы, что повышает их срок эксплуатации.

Целью данной работы является изучение процесса регулирования скорости вращения сервопривода Dynamixel. В рамках данной работы предлагается сконструировать поворотный механизм, вращающийся между двумя крайними положениями. При нажатии на кнопки программируемого контроллера скорость вращения должна изменяться следующим образом:

1) При нажатии кнопки U скорость вращения должна увеличиваться от текущего значения до максимального.

2) При нажатии кнопки D скорость вращения должна уменьшаться от текущего значения до минимального.



A : CCW Compliance Slope(Address0x1D)
B : CCW Compliance Margin(Address0x1B)
C : CW Compliance Margin(Address0x1A)
D : CW Compliance Slope (Address0x1C)
E : Punch(Address0x30,31)

Управление скоростью сервопривода возможно как в режиме постоянного вращения, так и в режиме перемещения в целевое положение. При перемещении сервопривода в целевое положение возможно изменять характер изменения выходной мощности на валу привода. В качестве программируемых величин используются зоны нарастания мощности «A» и зона запаса – «B» и зона ударного останова – «E». Изменяя данные параметры, становится возможным регулировать характер скорости вращения вала сервопривода.

Программирование системы управления начинается с настройки базовых переменных и проверки правильности сборки механизма.

```
4:  START PROGRAM
5:  {
6:      IF ( ID[1]: ADDR(3(0x)) != 1 )
7:          CALL AssemblyError
8:      // Dynamixel ID check
9:      IF ( ID[1]: ADDR(&(w)) == 0 )
10:         ID[1]: ADDR[&(w)] = 1023
11:     // Dynamixel mode check (joint or wheel)
12:     IF ( Button == D )
13:         JUMP AssemblyCheckMode
14:     // Assembly check
15:     MaxSpeed = 800
16:     LowSpeed = 100
17:     InitialPosition = 200
18:     FinalPosition = 800
19:     ToggleValue = 0
20:     ID[1]: CW slope = 64
21:     ID[1]: CCW slope = 64
22:     CALL InitialPosition
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.I-TECHNOLAB.RU

На стадии инициализации задаются начальные значения переменных: MaxSpeed – величина максимальной скорости, MinSpeed – величина минимальной скорости, InitialPosition – начальное положение механизма, FinalPosition – конечное положение механизма. Переменная ToggleValue определяет направление движения механизма и является переменной – флагом с состояниями «0» и «1».

Отдельно стоит обратить внимание на то, что режим торможения возле целевой позиции характеризуется наклоном «slope» и характер наклона определяется значением 64. Пользователь может выбрать один из 7 доступных наклонов характеристики, поэтому если введено какое-либо промежуточное значение, то оно приравнивается к ближайшему стандартному.

```
FUNCTION InitialPosition
{
    ID[1].Moving speed = LowSpeed
    CALL TogglePosition
}
```

Движение механизма начинается после вызова функции InitialPosition, задающей стартовую скорость движения. Направление движения определяется с помощью функции TogglePosition.

```
FUNCTION TogglePosition
{
    IF ( ToggleValue == 0 )
    {
        ID[1].Goal position = InitialPosition
        ToggleValue = 1
    }
    ELSE IF ( ToggleValue == 1 )
    {
        ID[1].Goal position = FinalPosition
        ToggleValue = 0
    }
}
```

В зависимости от значения флага выбирается одно из направлений движения механизма – либо к начальной точке, либо к конечной. Опрос нажатий кнопок программируемого контроллера осуществляется в бесконечном цикле.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
24:     ENDLESS LOOP
25:     {
26:         IF ( ID[1]: Moving == 0 )
27:             CALL TogglePosition
28:         IF ( Timer == 0.000sec )
29:         {
30:             IF ( Button == U )
31:                 CALL SpeedUp
32:             ELSE IF ( Button == D )
33:                 CALL SlowDown
34:             Timer = 2
35:         }
36:     }
```

В зависимости от того, какая из кнопок контроллера нажата, происходит уменьшение или увеличение скорости движения механизма.

```
139: FUNCTION SpeedUp
140: {
141:     IF ( ID[1]: Moving speed >= MaxSpeed )
142:         RETURN
143:     ID[1]: Moving speed = ID[1]: Moving speed + 40
144: }


146: FUNCTION SlowDown
147: {
148:     IF ( ID[1]: Moving speed <= MaxSpeed )
149:         RETURN
150:     ID[1]: Moving speed = ID[1]: Moving speed - 40
151: }
```

Изменение скорости движения механизма осуществляется в заданных пределах – от минимального значения до максимального.

Стоит обратить внимание на тот факт, что изменение наклона характеристики мощности привода происходит автоматически, без участия со стороны пользователя. В этом случае при приближении к целевому положению сервопривод развивает мощность в соответствии с заданным значением наклона характеристики. Таким образом, можно программировать плавные передвижения робота.

В рамках данной работы были рассмотрены основы регулирования скорости сервопривода при перемещении в целевую позицию. Была разработана программа, управляющая скоростью движения сервопривода, с помощью нажатий на кнопки программируемого контроллера.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Лабораторная работа № 3

ЭКЗАМЕН ТЕХНОЛАБ

Основы работы с ИК-датчиком и таймером

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 3

«Основы работы с ИК-датчиком и таймером»

Системы управления роботов и производственных механизмов должны не только работать по жестко заданной программе, но и иметь возможность реагировать на внешние сигналы от других устройств управления и датчиков.

В рамках данной работы предлагается исследовать принцип управления механизмом с использованием информации об объекте, находящемся вблизи. Для определения объекта применяется ИК-датчик, выдающий программируемому контроллеру аналоговый сигнал, пропорциональный отраженному ИК-излучению от объекта. ИК-датчик работает на основе принципа отраженного света от поверхности объекта.

В данной лабораторной работе необходимо разработать механизм – «пасть крокодила», срабатывающий при наличии вблизи объекта. При наличии объекта – «пасть крокодила» открывается, при отсутствии – закрывается, если объект не был обнаружен в течение более 10 секунд – воспроизводится мелодия.

Отсчет временного интервала производится с помощью таймера, настраиваемого на 10 секунд. Таймер – важнейший инструмент в программировании систем управления робототехнических систем. С помощью таймеров становится возможным отсчитывать различные промежутки времени с большой точностью.

ОФИЦИАЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

```
5:  START PROGRAM
6:  {
7:      IF ( ID[1]: ADDR[3(b)] != 1 )
8:          CALL AssemblyError
9:      // Dynamixel ID check
10:     IF ( ID[1]: ADDR[8(w)] == 0 )
11:         ID[1]: ADDR[8(w)] = 1023
12:     // Dynamixel mode check (joint or wheel)
13:     IF ( Button == D )
14:         JUMP AssemblyCheckMode
15:     // Assembly check
16:     OpenMouthInitial = 400
17:     ClosedMouthInitial = 690
```

Программа начинается со стандартной процедуры инициализации начальных значений – присваивается идентификационный номер привода, задаются начальное и конечное положения вала сервопривода, соответствующие закрытому и открытому положению «пасти крокодила».

```
19:     ID[1]: Moving speed = 80
20:     CALL InitialPosition
```

После задается скорость вращения и вызывается функция перехода в начальное положение.

```
123: FUNCTION InitialPosition
124: {
125:     ID[1]: Goal position = OpenMouthInitial
126:     CALL Movement
127:     ID[1]: Goal position = ClosedMouthInitial
128:     CALL Movement
```

Переход в начальное положение осуществляется в два этапа – сначала «пасть крокодила» открывается, а после закрывается. Движение механизма контролируется с помощью функции Movement. Каждый раз, когда сервоприводу дается команда на перемещение в целевое положение, функция Movement обеспечивает временную задержку на время работы привода, тем самым не давая программе перейти к выполнению другой операции и нарушить ход выполнения программы.

```
131: FUNCTION MouthMovement
132: {
133:     ID[1]: Goal position = OpenMouthInitial
134:     CALL Movement
135:     ID[1]: Goal position = ClosedMouthInitial
136: }
137:
138: FUNCTION Movement
139: {
140:     WAIT WHILE ( ID[1]: Moving == TRUE )
141: }
```

Управляющий алгоритм программы содержится в бесконечном цикле, который анализирует наличие объекта перед «пастью крокодила».

```
22:  Timer = 10.240sec
23:  ENDLESS LOOP
24:  {
25:      IF ( PORT[1] >= 200 || ID[1]: Present load >= CW:100 )
26:      {
27:          CALL MouthMovement
28:          Timer = 10.240sec
29:      }
30:      ELSE IF ( Timer == 0.000sec )
31:      {
32:          CALL PlayMelody
33:          Timer = 10.240sec
34:      }
35:  }
```

Еще до начала бесконечного цикла инициализируется таймер на 10 секунд, который используется на первом такте работы цикла, в дальнейшем таймер инициализируется заново на каждой итерации цикла.

```
25:      IF ( PORT[1] >= 200 || ID[1]: Present load >= CW:100 )
26:      {
27:          CALL MouthMovement
28:          Timer = 10.240sec
29:      }
```

В теле цикла анализируется условие, свидетельствующее о том, что сработал ИК-датчик или привод находится в движении. Если данное условие выполняется, то механизм совершает заданное движение и переходит на следующую итерацию цикла. Таймер в этом случае инициализируется заново.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
30:     ELSE IF ( Timer == 0.000sec )
31:     {
32:         CALL PlayMelody
33:         Timer = 10.240sec
34:     }
```

Если же вблизи механизма в течение более 10 секунд отсутствует объект, запускается новая ветвь цикла и робот воспроизводит мелодию. После чего таймер инициализируется заново, и программа переходит заново к выполнению бесконечного цикла.

Стоит обратить внимание на то, что для задания временного интервала таймер инициализируется каждый раз заново. Чтобы программа осуществила заданную задержку, необходимо сразу же после инициализации таймера использовать любой оператор условия – WHAIT WHILE или IF, которые приостанавливают выполнение программы до выполнения заданного в них условия.

В данной лабораторной работе был приведен пример работы с ИК-датчиком, а также исследован принцип работы с таймером. Применение подобных навыков на практике крайне важно, поскольку позволяет качественно повысить сложность решаемых задач.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNO.RU

Лабораторная работа № 4

ЭКЗАМЕН ТЕХНОЛАБ

Управление простейшими механизмами с помощью кнопок программируемого контроллера

ОЗНАКОМИТЕЛЬНЫЙ ВЕРСИЙ ЭКЗАМЕН-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЛАБОРАТОРИЯ-ТЕХНОЛАБ.РФ

Лабораторная работа № 4

«Управление простейшими механизмами с помощью кнопок программируемого контроллера»

Многие производственные механизмы, в том числе и робототехнические, состоят из нескольких степеней подвижности, в каждой из которых располагается привод, обеспечивающий ее движение. Для того чтобы обеспечивать управление многостепенным механизмом, необходимо согласованно управлять каждым из приводов, входящих в его состав.

В рамках данной работы предлагается сконструировать простейший манипуляционный механизм, состоящий из поворотного основания и привода качения. Управление механизмом предлагается осуществлять с помощью кнопок программируемого контроллера:

1) Если нажата кнопка U, то привод качения движется вверх.
2) Если нажата кнопка D, то привод качения движется вниз.
3) Если нажата кнопка R, то привод поворота движется направо.
4) Если нажата кнопка L, то привод поворота движется налево.

Программа робота представляет собой последовательность конструкций IF-ELSE, каждая из которых содержит условие нажатия на одну из кнопок контроллера. Для того чтобы управление роботом осуществлялось непрерывным образом, возможно применение комплексных условий, таких, при которых анализируется нажатие двух кнопок одновременно.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

В начале программы осуществляется традиционная процедура инициализации базовых переменных и устройств. В первую очередь задаются идентификационные номера для каждого из приводов и проверяется правильность сборки. После задается скорость движения и вызывается функция перехода в начальное положение.

```
START PROGRAM
{
    IF ( ID[1]: ADDR[3(b)] != 1 )
        CALL AssemblyError
    IF ( ID[2]: ADDR[3(b)] != 2 )
        CALL AssemblyError
    // Dynamixel ID check
    IF ( ID[1]: ADDR[8(w)] == 0 )
        ID[1]: ADDR[8(w)] = 1023
    IF ( ID[2]: ADDR[8(w)] == 0 )
        ID[2]: ADDR[8(w)] = 1023
    // Dynamixel mode check ( joint or wheel)
    IF ( Button == D )
        JUMP AssemblyCheckMode
    // Assembly check
    ID[All] Moving speed = 200
    CALL InitialPosition
```

Основная часть программы представляет собой бесконечный цикл, в котором анализируются условия нажатия той или иной комбинации кнопок программируемого контроллера.

```
ENDLESS LOOP
{
    IF ( Button == D )
    {
        // check up
        IF ( ID[2] Present position <= 700 )
            ID[2] Goal position = ID[2] Present position + 10
    }
    ELSE IF ( Button == U )
    {
        // check down
        IF ( ID[2] Present position >= 200 )
            ID[2] Goal position = ID[2] Present position - 10
    }
```

В случае движения отдельных частей манипуляционного механизма осуществляется нажатие на одну из кнопок в отдельности. Для каждого из приводов анализируется, какая из кнопок нажата, и в случае нажатия соответствующей кнопки привод начинает движение.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Движение привода осуществляется за счет изменения его целевого задания, а именно: при увеличении задания к текущему положению привода прибавляется фиксированная величина, а в случае уменьшения задания такая же величина отнимается. Текущее положение привода определяется с помощью функции Present position.

Аналогичным образом анализируется нажатие нескольких кнопок контроллера. В зависимости от того, какие кнопки нажаты, привода осуществляют одновременное движение в заданном направлении.

```
50:     ELSE IF ( Button == D+L )
51:     {
52:         // open left:
53:         IF ( ID[1]: Present position <= 700 )
54:             ID[1]: Goal position = ID[1] Present position + 10
55:         IF ( ID[2]: Present position <= 700 )
56:             ID[2]: Goal position = ID[2] Present position + 10
57:     }
```

В этом случае задание для обоих приводов устанавливается одновременно. Так же как и в предыдущем случае при установке задания анализируется условие достижения граничного положения для сервопривода.

В рамках данной работы был рассмотрен принцип одновременного управления группой приводов с помощью кнопок программируемого контроллера. В ходе выполнения работы был разработан алгоритм управления манипуляционным роботом, состоящим из двух степеней подвижности. Управляющая программа данного робота была разработана таким образом, благодаря чему предусматривается управление как отдельными приводами, так и синхронное управление двумя приводами одновременно. Синхронное управление приводами дает возможность осуществлять плавные и согласованные перемещения исполнительных механизмов роботов и робототехнических систем.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ МАТЕРИАЛОВ EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНСК.ТЕХНОЛАБ.RU

Лабораторная работа № 5

ЭКЗАМЕН ТЕХНОЛАБ

Определение нагрузки на сервопривод

ЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЛАБОРАТОРИЯ-ТЕХНОЛАБ.РФ

Лабораторная работа № 5

«Определение нагрузки на сервопривод»

При работе различных автоматизированных систем зачастую необходимо контролировать процесс работы отдельных исполнительных механизмов. Подобные меры позволяют избегать аварийных ситуаций, например при превышении допустимой нагрузки на элементы механики роботов или при заклинивании каких-либо механизмов. Как правило, о нагрузке на исполнительные механизмы роботов свидетельствует нагрузка приводов, т.е. значение рабочего тока. Также о повышении нагрузки на привод может сигнализировать резкое изменение скорости его вращения и т.п.

В рамках данной работы предлагается уделить внимание вопросам оценки текущей нагрузки на привод. В частности, предлагается разработать конструкцию автоматизированного шлагбаума, состоящего из двух приводов, управляющих движением механизма, а также ИК-датчика для обнаружения объекта вблизи.

Алгоритм управления робототехнической системой сводится к следующему:

1) При наличии препятствия перед автоматизированным шлагбаумом необходимо осуществить его подъем.

2) Если же механизм находится в движении, т.е. шлагбаум поднимается вверх и при этом обнаруживается препятствие, шлагбаум начинает поворачиваться в горизонтальном направлении.

3) В случае если объект не обнаружен, шлагбаум должен быть закрыт.

Управляющая программа начинается с традиционной инициализации идентификационных номеров сервоприводов и инициализации базовых переменных, характеризующих начальные и конечные положения каждого из сервоприводов. Переменная SensorValue задает значение показаний ИК-датчика, при котором засчитывается его срабатывание. Следует иметь в виду, что на работу ИК-датчика оказывает влияние тип отражающих поверхностей, освещенность в рабочей зоне и т.п.

```
7:   IF ( ID[1]: ADDR[3(b)] != 1 )
8:       CALL AssemblyError
9:   IF ( ID[2]: ADDR[3(b)] != 2 )
10:      CALL AssemblyError
11:  // Dynamixel ID check
12:  IF ( ID[1]: ADDR[8(w)] == 0 )
13:      ID[1]: ADDR[8(w)] = 1023
14:  IF ( ID[2]: ADDR[8(w)] == 0 )
15:      ID[2]: ADDR[8(w)] = 1023
16:  // Dynamixel mode check (joint or wheel)
17:  IF ( Button == D )
18:      JUMP AssemblyCheckMode
19:  // Assembly check
20:  VerticalInitialOpen = 205
21:  VerticalInitialClosed = 512
22:  HorizontalInitialOpen = 819
23:  HorizontalInitialClosed = 512
24:  SensorValue = 40
25:  ID[All]: Moving speed = 100
26:  CALL InitialPosition
```

После базовой инициализации осуществляется переход механизма в базовое положение с помощью функции InitialPosition. После этого выполнение программы осуществляется в бесконечном цикле.

```
28:  ENDLESS LOOP
29:  {
30:      WAIT WHILE ( PORT[6] < SensorValue )
31:      CALL Moving
32:      IF ( PORT[6] >= SensorValue && ID[1]: Present load > CCW:50 && ID[1]: Present load < CW:0 )
33:          CALL HorizontalOpening
34:      ELSE IF ( PORT[6] >= SensorValue )
35:          CALL VerticalOpening
36:  }
```

В бесконечном цикле анализируются показания ИК-датчика, подключенного к PORT[6]. В случае если ИК-датчик не обнаруживает объектов, шлагбаум находится в закрытом состоянии и осуществляется задержка на время выполнения предыдущих команд, т.е. программа ждет до тех пор, пока привода не окончат свое движение. Находится привод в движении или нет, определяется с помощью функции Moving, анализирующей флаг состояния «Moving» для каждого из сервоприводов.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Движется привод или нет, также определяется по показаниям функции Present load, которая возвращает значение нагрузки на привод в данный момент. С помощью данной функции можно определить факт того, что привод в движении, также можно определить поврежденный привод или то, что заклинило, в случае если полученное значение превышает предельно установленные величины.

Работа механизма описывается функциями HorizontalOpening и VerticalOpening, управляющих движением и поворотом шлагбаума.

```
139: FUNCTION HorizontalOpening
140: {
141:     Buzzer time = 255
142:     Buzzer index = 9
143:     ID[1]:Goal position = HorizontalInitialOpen
144:     CALL Moving
145:
146:     ENDLESS LOOP
147:     {
148:         WAIT WHILE ( PORT[6] >= SensorValue )
149:         CALL Wait2
150:         IF ( PORT[6] < SensorValue )
151:         {
152:             Buzzer time = 255
153:             Buzzer index = 10
154:             ID[1]:Goal position = HorizontalInitialClosed
155:             ID[2]:Goal position = VerticalInitialClosed
156:             CALL Moving
157:             BREAK LOOP
158:         }
159:     }
160: }
```

При вызове каждой из функций воспроизводится звуковой сигнал, свидетельствующий о начале работы. После механизм переводится в целевое положение и удерживается в нем до тех пор, пока вблизи находится обнаруженный ИК-датчиком объект. Как только объект исчезает из поля видимости ИК-датчика, механизм переходит в начальное (закрытое) положение.

Интервалы ожидания реализуются с помощью специально разрабатываемых функций Wait1, Wait2, Wait3, которые инициализируют таймер на заданный промежуток времени.

В рамках данной работы был рассмотрен принцип управления механизмом и определения нагрузки на его привода. Проделанная работа позволяет упрочнить полученные ранее знания в области управления приводами, обработке показаний таймеров и ИК-датчиков.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 6

ЭКЗАМЕН ТЕХНОЛАБ

Управление роботом и режимом его работы с помощью кнопок программируемого контроллера

ОЗНАКОМИТЕЛЬНЫЙ САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 6

«Управление роботом и режимом его работы с помощью кнопок программируемого контроллера»

Существует множество различных способов управления роботами и робототехническими системами. Большинство из них чрезвычайно похожи и дополняют друг друга, но несмотря на это требуют детального рассмотрения.

Рассмотрим процесс управления колесным мобильным роботом с двумя ведущими колесами. Данный тип роботов наиболее часто встречается, поэтому такой пример должен быть наиболее показательным.

На первый взгляд нет ничего проще управления подобным роботом – подав напряжения питания на каждый из приводов колес, можно привести их в движение. Таким образом, можно управлять различными механизмами, состоящими из множества различных приводов. Но подобный подход к управлению не может привести ни к чему хорошему. Отсутствие информации о состоянии каждого из приводов в отдельности, ошибки следящих систем и другое может привести к существенным погрешностям перемещения исполнительного механизма.

Для избежания подобных ситуаций при проектировании следящих систем в качестве наблюдаемого параметра выбирается величина наиболее емким образом описывающая работу робототехнической системы.

Например, при проектировании системы управления мобильным роботом, в качестве параметров, описывающих состояние системы, могут быть использованы линейная и угловая скорость.

В рамках данной работы предлагается спроектировать модель мобильного робота, управляемого с кнопок программируемого контроллера. В данном примере робот управляется в двух различных режимах работы:

1) При одновременном нажатии кнопок U и Start осуществляется управление роботом в ручном режиме. В этом случае по нажатию кнопки L робот движется вперед, по нажатию кнопки R – назад, нажатие кнопок U и D приводит к поворотам направо и налево соответственно.

2) При одновременном нажатии кнопок D и Start робот переходит в режим автономного перемещения. В этом случае робот при движении обнаруживает объекты на своем пути и объезжает их, с помощью ИК-датчика установленного на днище, робот обнаруживает обрывы и объезжает их.

Управляющая программа традиционно начинается с процедуры проверки правильности сборки. При желании эту часть программы можно исключить для простых механизмов, в этом случае пользователь должен самостоятельно удостовериться в работоспособности модели.

```
24:  ForwardSpeed  = 500
25:  ReverseSpeed  = 500
26:  RotatingSpeed  = 500
27:  // Initial Poisition : Forward
28:  CALL Forward
29:  Timer  = 1.408sec
30:  WAIT WHILE ( Timer  > 0.000sec  )
31:  CALL Stop
```

В самом начале программы инициализируются базовые переменные, описывающие скорости движения вперед и назад, а также разворота. Далее в бесконечном цикле выбирается один из режимов работы робота.

```
32:  ENDLESS LOOP
33:  {
34:      IF ( Button  ==  U+S  )
35:      {
36:          JUMP ControlMode
37:      }
38:      IF ( Button  ==  D+S  )
39:      {
40:          JUMP SmartMode
41:      }
42:  }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Режим ControlMode применяется для ручного управления движением робота с помощью кнопок программируемого контроллера. Переход к участку программы, отвечающему за данный режим работы, осуществляется с помощью метки и оператора перехода JUMP.

```
44: ControlMode :
45:     ENDLESS LOOP
46:     {
47:         IF ( Button == L )
48:         {
49:             CALL Advance
50:         }
51:         ELSE IF ( Button == R )
52:         {
53:             CALL Retreat
54:         }
55:         ELSE IF ( Button == D )
56:         {
57:             CALL ForwardLeft
58:         }
59:         ELSE IF ( Button == U )
60:         {
61:             CALL ForwardRight
62:         }
63:     }
```

В режиме ControlMode анализируется, какая из кнопок программируемого контроллера была нажата, и в зависимости от этого вызывается одна из функций, воспроизводящих движения робота.

```
232: FUNCTION Forward
233: {
234:     ID[1]: Moving speed = CW:0 + ForwardSpeed
235:     ID[2]: Moving speed = CCW:0 + ForwardSpeed
236:     ID[3]: Moving speed = CW:0 + ForwardSpeed
237:     ID[4]: Moving speed = CCW:0 + ForwardSpeed
238: }
```

Реализация движений робота осуществляется с помощью стандартных принципов управления приводами. Каждому из приводов задается скорость вращения в зависимости от типа движения – при движении вперед каждый из приводов синхронно вращается с одной скоростью, а при повороте – с разной.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
64: SmartMode ;
65:     ENDLESS LOOP
66:     {
67:         // Avoids cliffs or voids
68:         IF ( PORT[2] < 100 )
69:             CALL AvoidCliff
70:
71:         // Avoids objects
72:         IF ( PORT[1] >= 15 )
73:             CALL AvoidObject
74:         ELSE
75:         {
76:             CALL Forward
77:         }
78:     }
```

При работе в режиме автономного движения в первую очередь анализируются показания ИК-датчиков. При обнаружении обрыва снизу или препятствия на пути, вызывается соответствующая функция объезда AvoidCliff или AvoidObject. Каждая из этих функций описывает маневр робота справа, благодаря которому избегается столкновение.

В рамках данной работы был исследован принцип управления режимами работы мобильного робота. Помимо этого был рассмотрен комплексный подход к управлению движением робота, а именно его линейной и угловой скоростью. Благодаря этому подходу становится возможным разрабатывать алгоритмы плавных перемещений роботов.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ. САЙТ ДЛЯ ЕКАТЕРИНЫ TECHNOLAB.RU

Лабораторная работа № 7

ЭКЗАМЕН ТЕХНОЛАБ

Основы применения микрофона

ЗНАКОМИТЕСЬ С САЙТОМ EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 7

«Основы применения микрофона»

Существует множество различных типов сенсорных устройств, применяемых в робототехнике. Помимо традиционных датчиков в сервисной робототехнике применяются устройства для взаимодействия роботов с человеком. Одним из примеров подобного взаимодействия может быть голосовое управление. Технология голосового управления основывается на детектировании звуков, их обработке и распознавании.

Робот воспринимает окружающие его звуки с помощью микрофона. В зависимости от возлагаемых на робота задач производится дальнейшая обработка аудиоинформации. Простейшие системы управления детектируют наиболее интенсивные звуки, таким образом выявляя наличие посторонних звуков. Более сложные системы могут распознавать звуки и классифицировать их по различным категориям.

В рамках данной работы предлагается исследовать основные принципы работы со звуком. В качестве эксперимента предлагается разработать программу управления роботом, реагирующую на звуковые сигналы. Для примера предлагается разработать модель робота краба с подвижной клешней, управляющая программа которого должна подсчитывать хлопки и производить столько же хлопков в ответ.

Инициализация счетчика хлопков производится сразу же после инициализации приводов и установки их начального положения.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
17:     // Assembly check
18:     // Initial state at open arms
19:     ID[1]: Goal position = 612
20:     ID[2]: Goal position = 412
21:
22:     // Sound detection initialization
23:     Timer = 1.536sec
24:     WAIT WHILE ( Timer > 0.000sec )
25:     Sound count = 0
```

Подсчет хлопков производится в бесконечном цикле, в котором на первом этапе с помощью функции Sound count определяется общее количество звуков, а после осуществляется такое же количество хлопков клешней робота.

```
27:   ENDLESS LOOP
28:   {
29:       IF ( Sound count > 0 )
30:       {
31:           ClappingTimes = Sound count
32:           LOOP FOR ( n = 1 ~ ClappingTimes )
33:           {
34:               // The robot claps
35:               ID[1]: Goal position = 612
36:               ID[2]: Goal position = 412
37:               // Pause and waits until melody ends.
38:               Timer = 0.128sec
39:               WAIT WHILE ( Timer > 0.000sec )
40:               // clapping melody
41:               Buzzer time = Play Melody
42:               Buzzer index = Melody13
43:               ID[1]: Goal position = 692
44:               ID[2]: Goal position = 332
45:               Timer = 0.256sec
46:               WAIT WHILE ( Timer > 0.000sec )
47:               ID[1]: Goal position = 612
48:               ID[2]: Goal position = 412
49:               // Pause and wait until melody ends.
50:               Timer = 0.512sec
51:               WAIT WHILE ( Timer > 0.000sec || Buzzer time > 0.0sec )
52:           }
53:
54:           // Initial position before clapping
55:           Timer = 1.024sec
56:           WAIT WHILE ( Timer > 0.000sec )
57:           Sound count = 0
58:       }
59:   }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

С помощью конструкции Loop For образуется цикл, функционирующий столько раз, сколько задано определяющей его переменной ClappingTimes. В данном цикле на каждой из его итераций робот производит хлопок клешней и каждый раз воспроизводит мелодию, сигнализирующую о совершении хлопка.

По завершении цикла Loop For число хлопок обнуляется, и программа снова переходит к бесконечному циклу подсчета внешних звуков.

Данная программа демонстрирует базовые принципы работы со звуковыми командами. По аналогии с этим примером можно разработать различные робототехнические системы и механизмы, управляемые с помощью хлопков. Подобный принцип управления может применяться в различных ситуациях, например для дистанционного управления устройствами или механизмами, в системах «умного дома» и т.п., а также может служить первым шагом к совершенствованию навыков разработки интеллектуальных систем управления.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК.ТЕХНОЛАБ.РУ

Лабораторная работа № 8

ЭКЗАМЕН ТЕХНОЛАБ

Определение объектов с помощью ИК-датчиков

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 8

«Определение объектов с помощью ИК-датчиков»

Большинство роботов и робототехнических систем эксплуатируются в тесном контакте с людьми или любыми другими роботами или объектами. Для обеспечения безопасности работы необходимо знать положение окружающих объектов, чтобы избегать столкновения с ними. Существует множество различных технических средств для обнаружения объектов, в качестве подобных устройств применяются ИК-датчики, УЗ сонары, камеры или сканирующие лазерные дальномеры.

В рамках данной работы предлагается рассмотреть принцип обнаружения роботом посторонних объектов с помощью ИК-датчиков. В качестве робота необходимо разработать манипуляционную платформу, приводимую в движение с помощью двух сервоприводов. Для обнаружения объектов механизм оснащается парой ИК-датчиков, устанавливаемых по обе стороны робота.

Непрерывно поворачивая ИК-датчики, робот может обнаруживать различные объекты вокруг себя, в случае если робот обнаружил объект, он атакует его.

Управляющая программа начинается с перевода механизма в стартовое положение и задания скорости движения. После этого в бесконечном цикле робот начинает сканировать окружающее пространство в поисках объектов. Сканирование осуществляется за счет движения механизма между двумя концевыми положениями.

В зависимости от того, в каком положении находится сервопривод, выбирается направление движения. После того как были заданы конечное положение и скорость движения, привод начинает перемещение. Поскольку теперь движение привода осуществляется под управлением внутренней системы управления, программируемый контроллер переходит к выполнению следующей инструкции, а именно: вызову функции Detect.

```
21:   // Assembly check
22:   ID[1]: Moving speed = 50
23:   CALL InitialPosition
24:   ENDLESS LOOP
25:   {
26:
27:       LOOP WHILE ( ID[1]: Present position < 752 )
28:       {
29:           ID[1]: Moving speed = 50
30:           ID[1]: Goal position = 782
31:           CALL Detect
32:       }
33:       LOOP WHILE ( ID[1]: Present position > 272 )
34:       {
35:           ID[1]: Moving speed = 50
36:           ID[1]: Goal position = 252
37:           CALL Detect
38:       }
39:   }
```

Функция Detect управляет движением робота с целью поиска объекта, находящегося в радиусе действия робота. Сканируя пространство, робот одним из датчиков обнаруживает объект, после чего ускоряется и движется в данном направлении до тех пор, пока не сработает второй датчик.

```
128: FUNCTION Detect
129: {
130:     IF ( PORT[2] >= 50 )
131:     {
132:         LOOP WHILE ( PORT[2] >= 50 )
133:         {
134:             ID[1]: Moving speed = 100
135:             ID[1]: Goal position = ID[1]: Present position + 300
136:             IF ( PORT[2] >= 50 && PORT[6] >= 50 )
137:             {
138:                 ID[1]: Goal position = ID[1]: Present position
139:                 CALL BackAttack
140:             }
141:         }
142:     }
143:     IF ( PORT[6] >= 50 )
144:     {
145:         LOOP WHILE ( PORT[6] >= 50 )
146:         {
147:             ID[1]: Moving speed = 100
148:             ID[1]: Goal position = ID[1]: Present position - 300
149:             IF ( PORT[2] >= 50 && PORT[6] >= 50 )
150:             {
151:                 ID[1]: Goal position = ID[1]: Present position
152:                 CALL BackAttack
153:             }
154:         }
155:     }
156:     ELSE
157:     {
158:
159:     }
160: }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN.TECHNOLAB.RU

Срабатывание ИК-датчиков определяется по превышению их показаний заданных пороговых значений – порогов срабатывания датчиков. Данные значения выбираются исходя из различных условий – дальности срабатывания, отражающих способностей поверхности объекта и т.п.

Как только объект оказывается в зоне видимости двух датчиков, вызывается функция атаки на объект BeakAttack. Атака выполняется роботом за счет резкого выпрямления собственного механизма вперед. Для этого в теле функции изменяется скорость движения приводов и задаются новые целевые значения.

```
172: FUNCTION BeakAttack
173: {
174:     // attack
175:     ID[2]: Moving speed = 300
176:     ID[3]: Moving speed = 300
177:     ID[2]: Goal position = 812
178:     ID[3]: Goal position = 512
179:     CALL Movement
180:     // turning
181:     ID[2]: Moving speed = 300
182:     ID[3]: Moving speed = 300
183:     ID[2]: Goal position = 512
184:     ID[3]: Goal position = 812
185:     CALL Movement
186: }
```

Поскольку между атакующим движением робота и возвратом его в начальное положение должна быть задержка, используется функция Movement, приостанавливающая программу на время движения приводов.

```
188: FUNCTION Movement
189: {
190:     // Actuators come to a stull stop
191:     WAIT WHILE ( ID[2]: Moving == TRUE || ID[3] Moving == TRUE )
192: }
```

Функция Movement имеет очень важное значение, поскольку она оперирует с флагами состояния сервопривода, поэтому может давать реальную оценку текущего состояния механизма. Использование в подобной ситуации временной задержки, реализованной на основе таймера, может не привести к желаемому эффекту, поскольку для реального механизма иногда бывает сложно рассчитать требуемое время для движения.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Обратите внимание на то, что разные объекты имеют различные отражающие свойства, поэтому они могут по разному обнаруживаться роботом. Проведите ряд испытаний и выберите пороги срабатывания датчика на используемый в работе объект. Сравните полученные результаты и оцените влияние цвета и качества поверхности объекта на работу робота.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ARMEN-TECHNOLAB.RU

Лабораторная работа № 9

ЭКЗАМЕН ТЕХНОЛАБ

Определение расстояния до объектов

ЗНАКОМИТЕЛЬНЫЙ САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXCHEMTECHNOLAB.RU

Лабораторная работа № 9

«Определение расстояния до объектов»

Задача определения расстояния до объектов довольно часто встречается в повседневной жизни и в робототехнике. Существует множество различных способов определения расстояния до объектов, и простейший из них – использование ИК-датчиков был рассмотрен в предыдущих работах. Использование ИК-датчика дает пользователю не точную информацию о расстоянии, а лишь косвенное представление о наличии объекта вблизи или на удалении от робота. Это обусловлено тем, что ИК-датчики работают с показаниями интенсивности отраженного излучения от поверхности. Поскольку процесс работы ИК-датчиков зависит от свойств поверхности, они не могут дать точных показаний расстояния до объекта.

Для того чтобы величина измеренного расстояния не зависела от свойств окружающей среды, в качестве датчиков необходимо использовать ИК-дальномеры. Отличительная их особенность заключается в том, что их показания основываются на времени распространения излучения от устройства до объекта и обратно. Поскольку время распространения волны в окружающей среде зависит исключительно от расстояния до объекта, ИК-дальномеры могут выдавать достоверную информацию.

В данной работе предлагается рассмотреть процесс использования ИК-дальномера на примере управления моделью автоматизированного шлагбаума. Данная модель наглядно демонстрирует результаты измерения расстояния до объекта – чем ближе находится объект к ИК-дальномеру, тем выше поднимается шлагбаум. Высота поднятия шлагбаума вычисляется в зависимости от измеренной дистанции.

ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА С САЙТА EXAMEN-TECHNOLAB.RU

Данная программа достаточно проста и состоит из бесконечного цикла, определяющего расстояние до объекта. После получения результатов измерений вычисляется конечное положение, в которое должен перейти сервопривод, чтобы шлагбаум поднялся на необходимую высоту.

```
START PROGRAM
{
    IF ( ID[1].ADDR[3(b)] != 1 )
        CALL AssemblyError
    // Dynamixel ID check
    IF ( ID[1].ADDR[8(w)] == 0 )
        ID[1].ADDR[8(w)] = 1023
    // Dynamixel mode check (joint or wheel)
    IF ( Button == D )
        JUMP AssemblyCheckMode
    // Assembly check
    ENDLESS LOOP
    {
        CALL Measure
        ID[1].Goal position = Position
    }
```

Измерение расстояния до объекта осуществляется с помощью функции Measure, которая анализирует показания ИК-дальномера, подключенного к PORT[1].

Функция Measure обрабатывает показания ИК-дальномера и нормализует их для дальнейшего использования в вычислениях. Ранее заданное значение InitialValue предназначено для установки сервопривода в начальном положении при первом запуске программы.

```
FUNCTION Measure
{
    InitialValue = 180
    StepValue = 30
    InitialPosition = PORT[1]
    InitialPosition = InitialPosition / StepValue
    InitialPosition = InitialPosition * StepValue
    Position = InitialValue + InitialPosition
}
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА ЭКЗАМЕН.TECHNOLAB.RU

В процессе работы программы ИК-дальномер вычисляет расстояние до объекта, полученное значение нормализуется с помощью переменной StepValue и прибавляется к переменной Position, характеризующей положение сервопривода и шлагбаума соответственно.

Вычисленное значение переменной Position используется в бесконечном цикле работы программы для задания конечного положения сервопривода.

```
14:    ENDLESS LOOP
15:    {
16:        CALL Measure
17:        ID[1]: Goal position = Position
18:    }
```

При разработке программы необходимо учитывать особенности конструкции робота. Поскольку шлагбаум может подниматься на ограниченную высоту, величина переменной Position должна быть ограничена также. Ограничение обусловлено как величиной рабочего угла сервопривода Dynamixel, так и особенностью конструкции модели, не позволяющей осуществлять вращение во всем диапазоне рабочего угла.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

В рамках данной работы был рассмотрен процесс применения ИК-дальномера в роботах и робототехнических устройствах. Подобные устройства достаточно часто встречаются в различных проектах, поэтому навыки работы с ними крайне важны.

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК.ТЕХНОЛАБ.RU

Лабораторная работа № 10

ЭКЗАМЕН ТЕХНОЛАБ

Управление роботом, перемещающимся вдоль линии

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ЭКЗАМЕН-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXCHENGETECHNOLAB.RU

Лабораторная работа № 10

«Управление роботом, перемещающимся вдоль линии»

Управление мобильными роботами – одна из наиболее часто встречающихся задач в робототехнике. Существует множество методов управления мобильными роботами, один из которых – движение робота вдоль направляющей линии. Данный метод наиболее часто встречается на практике, потому что отличается крайней дешевизной и надежностью. В данный момент перемещающиеся вдоль направляющей линии мобильные роботы являются наиболее часто встречающимся решением в промышленности. Так называемые робокары передвигаются по производственным помещениям с целью транспортировки грузов или инструментов, доставки готовой продукции на склады или участки производственных линий.

Суть управления подобными роботами заключается в определении относительного положения робота и направляющей линии. Для обнаружения направляющей линии применяются либо системы технического зрения, либо ИК-датчики.

В рамках данной работы предлагается рассмотреть процесс управления мобильным роботом, движущимся вдоль направляющей линии. Конструкция мобильного робота представляет собой мобильное шасси с двумя ведущими колесами и двумя ИК-датчиками, установленными на днище робота.

В процессе движения мобильного робота то один, то другой датчик оказывается над черной линией, соответственно его показания изменяются. На этом и основывается закон управления роботом: если сработал правый датчик – робот поворачивается налево, если сработал левый датчик – робот поворачивается направо.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

```
4:  START PROGRAM
5:  {
6:      IF ( Button == D )
7:          JUMP AssemblyCheckMode
8:      LOOP FOR ( ID = 1 ~ 2 )
9:      {
10:         IF ( ID[ID]: ADDR[3(b)] != ID )
11:             CALL AssemblyError
12:         // Dynamixel ID check
13:         IF ( ID[1]: ADDR[8(w)] != 0 )
14:             ID[1]: ADDR[8(w)] = 0
15:         // Dynamixel mode check (joint or wheel)
16:     }
17:     // Assembly check
18:     WheelSpeed = 1023
19:     BlackValue = 250
```

Управляющая программа начинается с базовой инициализации приводов и переменных WheelSpeed и BlackValue, отвечающих за скорость движения и пороговое значение, на которое срабатывает робот при обнаружении черной линии.

```
20:  ENDLESS LOOP
21:  {
22:      IF ( PORT[0] < BlackValue && PORT[3] < BlackValue )
23:      {
24:          CALL TurnLeft
25:      }
26:      ELSE IF ( PORT[0] > BlackValue && PORT[3] < BlackValue )
27:      {
28:          CALL TurnRight
29:      }
30:      ELSE
31:      {
32:          CALL Move
33:      }
34:
35:  }
```

Алгоритм управления заключается в том, что робот едет в противоположную сторону от обнаруженной черной линии. Если же ИК-датчики не обнаруживают линию, то мобильный робот движется прямолинейно. Таким образом, мобильный робот перемещается вдоль линии, которая всегда находится между двумя ИК-датчиками.

Движения робота описываются с помощью функций поворотов направо и налево, каждая из которых изменяет скорость вращения колес робота.

```
153:  FUNCTION TurnLeft
154:  {
155:      ID[1]: Moving speed = CW 0 + WheelSpeed
156:      ID[2]: Moving speed = 200
157:  }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
147: FUNCTION TurnRight
148: {
149:     ID[1]: Moving speed = CW:0 + 200
150:     ID[2]: Moving speed = WheelSpeed
151: }
```

При прямолинейном движении робота скорость вращения его колес синхронная в отличие от поворотов. В случае если робот поворачивает в одну из сторон, колесо, расположенное с противоположной стороны, должно опережать другое.

```
140: FUNCTION Move
141: {
142:     ID[1]: Moving speed = CW:0 + WheelSpeed
143:     ID[2]: Moving speed = WheelSpeed
144: }
```

Чем будет больше разница скоростей вращения колес, тем будет меньше радиус разворота робота. В случае если колеса робота вращаются в противоположном направлении, но с одинаковой скоростью, робот будет разворачиваться на месте относительно собственного центра масс.

Данная работа иллюстрирует простейший способ управления мобильным роботом. Несмотря на кажущуюся простоту управление мобильными роботами, движущимися вдоль направляющей линии, не такая простая задача. Сложность ее заключается в многообразии препятствующих факторов, например наличие посторонних объектов на пути робота, загрязнение или разрывы направляющей линии и др. Также немаловажным фактором является скорость движения робота. Чем выше скорость движения робота, тем больше вероятность того, что робот успеет выехать за пределы линии, прежде чем система управления успеет отреагировать на это.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ЭЛЕКТРОННОЙ КНИГИ ЭКЗАМЕН-ТЕХНОЛАБ.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 11

ЭКЗАМЕН ТЕХНОЛАБ

Управление шагающим роботом

ЗНАКОМИТЕЛЬНЫЙ
EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 11

«Управление шагающим роботом»

Существует множество различных типов мобильных роботов, обладающих различными отличительными чертами. Одной из основных отличительных черт можно считать кинематическую схему робота, обобщающую в себе особенности механической конструкции и принципов управления.

Чем сложнее кинематическая схема робота, тем больше различных движений он может выполнять. Из-за того, что простейшие движения могут быть реализованы за счет достаточно сложных алгоритмов, зачастую при разработке систем управления применяют паттерны, каждый из которых реализует одно из движений. По сути паттерны представляют собой функции, описывающие движения робота, но в отличие от стандартных функций они описывают методологию выполнения каких-либо действий, без привязки к конкретному типу роботов.

В рамках данной работы предлагается рассмотреть процесс разработки системы управления шагающего робота. Конструкцию робота предлагается оснастить ИК-датчиком для обнаружения объектов на пути. Управляющий алгоритм робота может быть произвольным, но в данной работе акцент делается на разработку функций, реализующих перемещения робота.

ОБНАРУЖЕНО НАРУШЕНИЕ С САЙТА EXAMEN-TECHNOLAB.RU

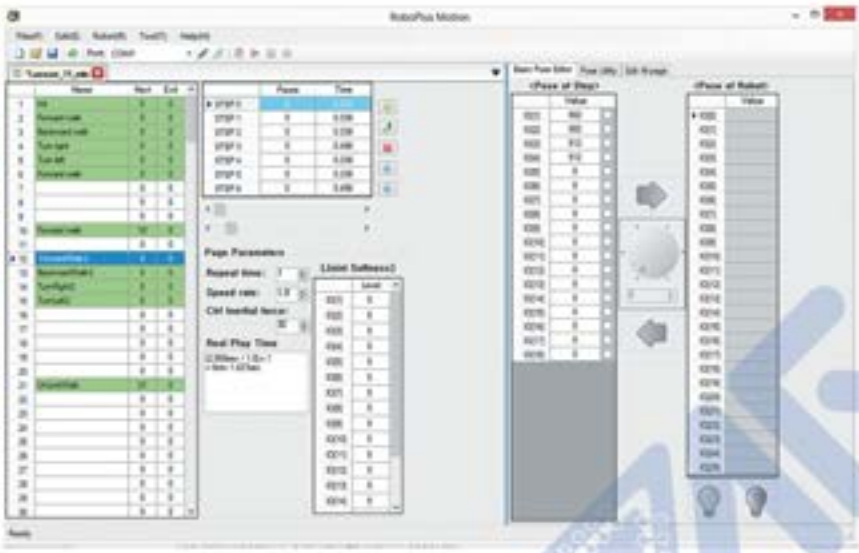
```
17:   CALL ReadyPosition
18:
19:   ENDLESS LOOP
20:   {
21:       CALL GoForward
22:       IF ( PORT[1] >= 30 )
23:       {
24:           CALL GoBackwards
25:           CALL GoLeft
26:           CALL GoLeft
27:           LOOP WHILE ( PORT[1] >= 30 )
28:               CALL GoLeft
29:       }
30:   }
```

Рассматриваемая в данной работе управляющая программа сводится к прямолинейному движению робота до обнаружения постороннего объекта на его пути. При обнаружении объекта робот отходит назад и после движется налево до тех пор, пока объект не исчезнет из зоны видимости ИК-датчика.

```
130: FUNCTION GoForward
131: {
132:     Motion Page = 12
133:     CALL WaitMotion
134: }
135:
136: FUNCTION GoBackwards
137: {
138:     Motion Page = 13
139:     CALL WaitMotion
140: }
141:
142: FUNCTION GoLeft
143: {
144:     Motion Page = 15
145:     CALL WaitMotion
146: }
```

Как видно, движения робота описываются рядом функций, представляющих собой паттерны, описывающие движения робота в том или ином направлении. Каждый из этих паттернов описывается в программной среде Roboplus Motion, предназначенной для программирования сложных движений роботов.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU



Каждый из паттернов описывается последовательностью положений сервопривода, которые достигаются в процессе выполнения данного действия. Так, любое из движений робота можно описать как набор положений его механизмов в разные моменты времени. При работе в Roboplus Motion любое движение робота разбивается на простейшие, описываемые шагами STEP. Каждый STEP определяется положением сервоприводов Pose of Step и временем, за которое данное положение должно быть достигнуто.

Таким образом, в Roboplus Motion можно разработать движения различной сложности, а после использовать их в управляющей программе. Вызов паттерна в управляющей программе осуществляется с помощью модуля Motion Page, ссылающегося на конкретную строку в списке паттернов Roboplus Motion.

Использование Roboplus Motion позволяет моделировать движения сразу же, в процессе их программирования. Применение паттернов при разработке систем управления роботов со сложной кинематикой позволяет облегчить процесс проектирования роботов и сосредоточить усилия разработчиков на совершенствовании системы в целом.

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 12

ЭКЗАМЕН ТЕХНОЛАБ

Управление роботом, определяющим положение окружающих объектов

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 12

«Управление роботом, определяющим положение окружающих объектов»

В современном мире существует множество различных применений мобильных роботов, и с каждым днем их число все больше растет. При использовании роботов в повседневной обстановке на их пути могут появляться различные препятствия. Зачастую нельзя разработать универсальное решение задачи планирования маршрута. Для того чтобы робот успешно миновал все препятствия, необходимо не только знать их положение относительно робота, но и их габариты.

В рамках данной работы предлагается разработать робота, оснащенного захватным устройством и датчиками для обнаружения объектов на своем пути. При движении вперед робот обозревает окружающее пространство с помощью ИК-дальномера, в случае обнаружения препятствия на своем пути робот останавливается и оценивает его габариты. Если габариты препятствия позволяют роботу переместить его, то с помощью захватного устройства робот смещает препятствие с пути и продолжает свое движение. Если же робот не способен переместить объект, ввиду того, что его габариты превышают допустимые для захватного устройства, робот объезжает препятствие сбоку. Помимо этого робот оснащен ИК-датчиком, с помощью которого он обнаруживает обрывы рабочей зоны и не заезжает за них.

Управляющая программа робота состоит двух взаимосвязанных алгоритмов – алгоритм прямолинейного перемещения и алгоритм оценки габаритов объекта. Оба алгоритма выполняются в бесконечном цикле работы мобильного робота.

Управляющая программа начинается с базовых настроек, определяющих скорость движения робота, характеристику работы привода, ограничения рабочего момента приводов захватного устройства. После задания базовых настроек вызывается функция инициализации, приводящая робота в рабочее положение.

```
26:   WheelSpeed = 400
27:   ObjectValue = 8
28:   ID[5]: Moving speed = CCW:100
29:   ID[6]: Moving speed = CCW:100
30:   ID[7]: Moving speed = CCW:100
31:   ID[6]: CW slope = 0000 0000 0100 0000
32:   ID[6]: CCW slope = 0000 0000 0100 0000
33:   ID[7]: CW slope = 0000 0000 0100 0000
34:   ID[7]: CCW slope = 0000 0000 0100 0000
35:   ID[6]: Torque limit = 380
36:   ID[7]: Torque limit = 380
37:   CALL Initial
```

Основные алгоритмы управления робота сосредоточены в бесконечном цикле, описывающем режим движения робота и манипулирования объектами.

```
39:   ENDLESS LOOP
40:   {
41:       IF ( PORT[2] < 80 )
42:       {
43:           // Avoids cliffs/voids
44:           CALL Reverse
45:           Timer = 1.792sec
46:           CALL WaitTimerCompletion
47:           CALL GoRight
48:           Timer = 3.200sec
49:           CALL WaitTimerCompletion
50:           CALL Forward
51:       }
52:       IF ( PORT[1] >= 630 )
53:       {
54:           // object detection
55:           CALL ObjectDetect
56:       }
57:   }
```

Каждый из рабочих режимов базируется на показаниях одного из датчиков. В случае прямолинейного движения определяющим фактором являются показания ИК-датчика, обнаруживающего кромку рабочей зоны. При обнаружении препятствия робот отъезжает назад, после чего поворачивает направо и продолжает свое прямолинейное движение.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
277: FUNCTION Reverse
278: {
279:     ID[1]: Moving speed = WheelSpeed
280:     ID[2]: Moving speed = CW.0 - WheelSpeed
281:     ID[3]: Moving speed = WheelSpeed
282:     ID[4]: Moving speed = CW.0 - WheelSpeed
283: }
```

```
285: FUNCTION GoRight
286: {
287:     ID[1]: Moving speed = WheelSpeed
288:     ID[2]: Moving speed = WheelSpeed
289:     ID[3]: Moving speed = WheelSpeed
290:     ID[4]: Moving speed = WheelSpeed
291: }
```

При обнаружении препятствия на пути мобильного робота запускается процесс оценки его габаритов с помощью функции ObjectDetect.

```
165: FUNCTION ObjectDetect
166: {
167:     Object = 1
168:     CALL GoRight
169:     // turns right
170:     ENDLESS LOOP
171:     {
172:         Timer = 0.128sec
173:         CALL WaitTimerCompletion
174:         IF ( PORT[1] < 300 )
175:             BREAK LOOP
176:         Object = Object + 1
177:         IF ( Object > ObjectValue )
178:         {
179:             // avoids large obstacles
180:             WAIT WHILE ( PORT[1] >= 300 )
181:             Timer = 1.024sec
182:             CALL WaitTimerCompletion
183:             CALL Forward
184:             RETURN
185:         }
186:     }
```

```
189:     Object = 3
190:     CALL GoLeft
191:     Timer = 0.256sec
192:     CALL WaitTimerCompletion
193:     // turns left
194:     ENDLESS LOOP
195:     {
196:         Timer = 0.128sec
197:         CALL WaitTimerCompletion
198:         IF ( PORT[1] < 300 )
199:             BREAK LOOP
200:         Object = Object + 1
201:         IF ( Object > ObjectValue )
202:         {
203:             // avoids large obstacles
204:             WAIT WHILE ( PORT[1] >= 300 )
205:             Timer = 1.024sec
206:             CALL WaitTimerCompletion
207:             CALL Forward
208:             RETURN
209:         }
210:     }
```

С помощью функции ObjectDetect робот поворачивает сначала направо и измеряет габариты объекта, а потом налево. Если габариты объекта оказываются больше допустимых, то робот осуществляет поворот до тех пор, пока объект не перестанет препятствовать прямолинейному движению. Если же объект обладает допустимыми габаритами, робот поворачивается в другую сторону и также оценивает габариты. В случае превышения допустимых габаритов робот объезжает объект с данной стороны, иначе же робот с помощью захватного устройства перемещает препятствие со своего пути.

```
212:     // compare object values
213:     CALL Stop
214:     CALL GoRight
215:     Timer = Object / 2
216:     CALL WaitTimerCompletion
217:     CALL Stop
218:     CALL SetAside
219:     CALL Forward
220: }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Перемещение объекта заключается в его подъеме, переносе направо и установке в стороне от маршрута робота.

```
228: FUNCTION SetAside
229: {
230:     CALL LiftObject
231:     CALL GoRight
232:     Timer = 1.792sec
233:     CALL WaitTimerCompletion
234:     CALL Forward
235:     CALL SetObject
236:     CALL Reverse
237:     Timer = 0.512sec
238:     CALL WaitTimerCompletion
239:     CALL GoLeft
240:     Timer = 1.792sec
241:     CALL WaitTimerCompletion
242:     CALL Stop
243: }
```

Функция SetAside, описывающая данный процесс, состоит из отдельных функций, реализующих ряд действий – поднятие объекта, поворот направо и т.п., которые в свою очередь также делятся на более простые.

Реализация данной функции должна восприниматься как образец написания управляющих программ для подобных роботов. Набор задач и действий робота должен быть классифицирован на ряд отдельных операций, каждая из которых должна описываться набором простейших движений. Таким образом, удается достичь возможности реализации управляющих программ роботов согласно принципу «от простого к сложному», когда любая из операций описывается в виде последовательного набора стандартных действий. Такой принцип позволяет упростить процесс разработки робототехнических систем в разы.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ОБРАЗОВАТЕЛЬНЫЙ РОБОТОТЕХНИЧЕСКИЙ МОДУЛЬ TECHNOLAB.RU

Лабораторная работа № 13

ЭКЗАМЕН ТЕХНОЛАБ

Управление роботом-экскаватором

EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЛАБОРАТОРИЯ-ТЕХНОЛАБ.РФ

Лабораторная работа № 13

«Управление роботом-экскаватором»

Задача манипулирования объектами является одной из часто встречающихся задач в робототехнике. Существует ряд традиционных методов решения подобной задачи, но несмотря на это теоретический подход к решению данной задачи каждый раз должен учитывать особенности конкретного механизма.

В рамках данной работы предлагается рассмотреть процесс разработки системы управления роботом-экскаватором. Модель данного робота оснащается ковшом для манипулирования различными объектами. Оснащенный ИК-датчиками и дальномером робот может обнаруживать препятствия в собственной рабочей зоне и точно позиционироваться относительно объекта, чтобы иметь возможность манипулировать им в процессе работы.

Система управления робота должна обнаруживать объекты на пути с помощью ИК-дальномера и останавливаться на расстоянии, достаточном для подъема объекта с помощью ковша. В случае если робот везет груз и обнаруживает препятствие справа или слева от себя, он должен развернуться в данную сторону и выгрузить груз вблизи обнаруженного объекта.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Управляющая программа робота начинается со стандартной процедуры настройки и инициализации переменных. Поскольку привода с идентификаторами ID[1]-ID[4] используются для управления колесами робота, они должны быть включены в режим постоянного вращения. Это можно осуществить с помощью отмены режима Torgue Enable, в этом случае отключается режим управления положением привода и Dynamixel работает в режиме постоянного вращения.

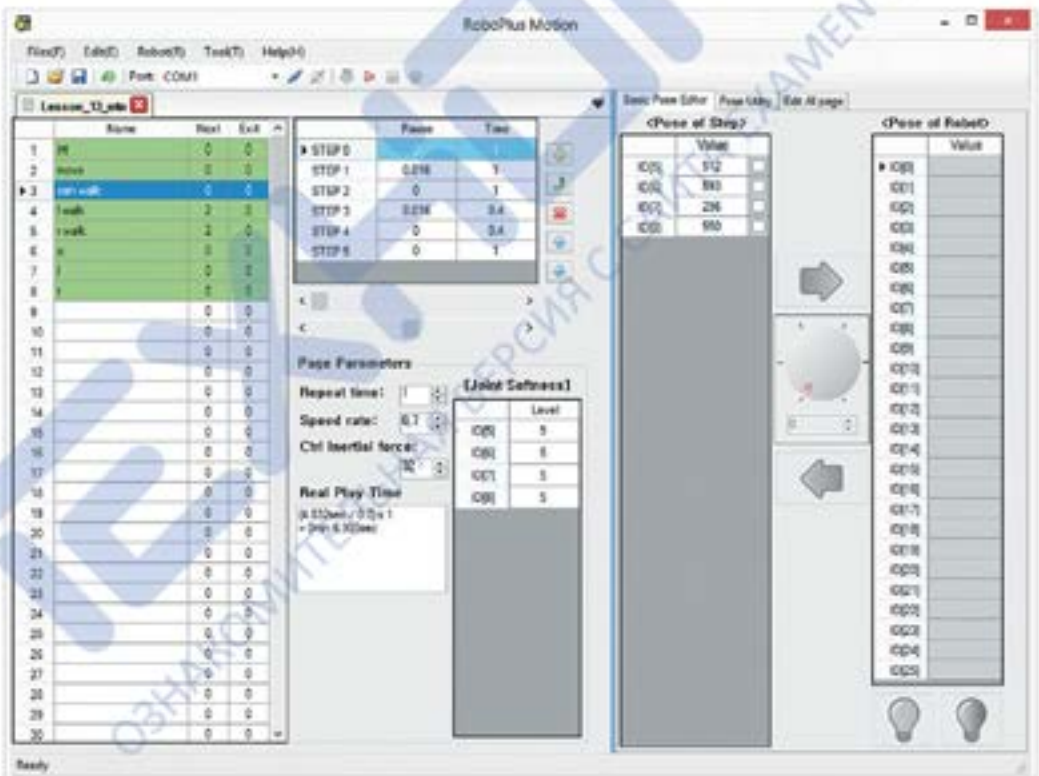

```
24:  // assembly check
25:  ID[1]: Torque Enable = FALSE
26:  ID[2]: Torque Enable = FALSE
27:  ID[3]: Torque Enable = FALSE
28:  ID[4]: Torque Enable = FALSE
29:  WheelSpeed = 400
30:  Dig = 0
31:  ID[6]: CW slope = 0000 0000 0100 0000
32:  ID[6]: CCW slope = 0000 0000 0100 0000
33:  ID[7]: CW slope = 0000 0000 0100 0000
34:  ID[7]: CCW slope = 0000 0000 0100 0000
35:  ID[8]: CW slope = 0000 0000 0100 0000
36:  ID[8]: CCW slope = 0000 0000 0100 0000
37:
38:  CALL InitialPosition
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ
TECHNOLAB.ROBOTEXAMEN-TECHNOLAB.RU

Для приводов, входящих в состав механизма подъема ковша экскаватора, применяется режим управления положением и настраивается рабочая характеристика. После проведения ряда настроек робот переводится в рабочее положение с помощью функции InitialPosition.

```
220: FUNCTION InitialPosition
221: {
222:     Motion Page = 2
223:     CALL WaitMotion
224:     Motion Page = 3
225:     CALL Move
226:     CALL Forward
227: }
```

Функция InitialPosition разработана с помощью паттернов, реализованных в Roboplus Motion, а также стандартных функций, написанных в Task-редакторе. С помощью паттернов реализуется закон управления ковшом экскаватора, а с помощью стандартных функций описываются движения робота.



Управляющий алгоритм реализован с помощью бесконечного цикла, анализирующего ряд возможных ситуаций, возникающих с мобильным роботом.

```
42:     IF ( PORT[1] >= 500 && Dig == 0 )
43:     {
44:         WAIT WHILE ( Timer > 0.000sec )
45:         Motion Page = 3
46:         CALL Move
47:         CALL GoPosition
48:         CALL TurnLeft
49:         Timer = 1.792sec
50:         WAIT WHILE ( Timer > 0.000sec )
51:         Dig = 1
52:     }
```

В первом случае робот, который еще не успел загрузить свой ковш, при обнаружении ИК-дальномером объекта на своем пути с помощью ковша поднимает объект, после чего продолжает свое движение и поворачивает налево. При этом система управления робота изменяет флаг состояния Dig, сигнализирующий о наличии объекта в ковше робота.

```
53:     IF ( PORT[1] >= 300 && Dig == 1 )
54:     {
55:         CALL Reverse
56:         Timer = 0.640sec
57:         WAIT WHILE ( Timer > 0.000sec )
58:         Motion Page = 6
59:         CALL Move
60:         CALL GoPosition
61:         CALL Reverse
62:         Timer = 1.792sec
63:         WAIT WHILE ( Timer > 0.000sec )
64:         CALL TurnLeft
65:         Timer = 1.792sec
66:         WAIT WHILE ( Timer > 0.000sec )
67:         Dig = 0
68:     }
```

В случае если робот движется с загруженным ковшом и обнаруживает препятствие на своем пути, он отъезжает назад, разгружает ковш и поворачивает налево. При этом система управления робота обнуляет флаг состояния Dig, сигнализирующий о наличии объекта в ковше робота.

Если в процессе движения робота с загруженным ковшом на пути обнаруживается объект справа или слева от робота, система управления робота выдает команду на остановку робота. После чего экскаватор выгружает груз из ковша вблизи с обнаруженным препятствием. После чего робот поворачивает налево и продолжает прямолинейное движение.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
87:         IF ( PORT[3] >= 20 && Dig == 1 )
88:         {
89:             WheelSpeed = 400
90:             CALL Stop
91:             Timer = 3.200sec
92:             WAIT WHILE ( Timer > 0.000sec )
93:             Motion Page = 8
94:             CALL Move
95:             CALL GoPosition
96:             WheelSpeed = 400
97:             CALL Reverse
98:             Timer = 1.792sec
99:             WAIT WHILE ( Timer > 0.000sec )
100:            CALL TurnLeft
101:            Timer = 1.792sec
102:            WAIT WHILE ( Timer > 0.000sec )
103:            Dig = 0
104:        }
105:        WheelSpeed = 400
106:        CALL Forward
```

Данная работа примечательна тем, что в ней демонстрируется процесс применения алгоритмов управления, реализованных в разных редакторах. Сложные движения ковша робота и процесс манипулирования объектами описывается с помощью RoboPlus Motion – редактора, обладающего инструментарием для моделирования различных движений. Простые движения робота реализуются с помощью Task-кода и описываются в виде привычных функций.

Благодаря комбинированию возможностей RoboPlus Motion и RoboPlus Task становится возможным реализовывать системы управления сложными механизмами, состоящими из множества приводов. В отличие от «слепого» программирования в RoboPlus Task процесс проектирования в RoboPlus Motion позволяет на каждом этапе ознакомиться с полученными результатами и отладить кусок программы на реальном роботе, подключенном к компьютеру.

ОЗНАКОМИТЕЛЬНЫЙ ДИЗАЙН САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 14

ЭКЗАМЕН ТЕХНОЛАБ

Управление роботами и механизмами с помощью звуковых команд

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATEHNOLAB.RU

Лабораторная работа № 14

«Управление роботами и механизмами с помощью звуковых команд»

Данная работа является продолжением цикла работ, посвященных управлению роботами с помощью звуков и работе с редактором Roboplus Motion. В рамках данной работы предлагается сконструировать модель робота-цветка, трепещущегося под дуновением ветра. Движения робота описываются паттернами, создаваемыми с помощью Roboplus Motion, которые проигрываются в бесконечном цикле.

Система управления робота постоянно оценивает уровень шумов вокруг и в случае обнаружения хлопка робот-цветок прекращает свое движение и смыкает лепестки. Если же система управления робота детектирует несколько хлопков, то робот-цветок закрывает свои лепестки быстрее.

```
18:     CALL Blossom
19:     Sound count = 0
20:     ENDLESS LOOP
21:     {
22:         IF ( Sound count == 1 )
23:         {
24:             CALL FlowerBend
25:             Timer = 35
26:             WAIT WHILE ( Timer > 0.000sec )
27:             Sound count = 0
28:         }
29:         IF ( Sound count >= 2 )
30:         {
31:             CALL Wonder
32:             Timer = 35
33:             WAIT WHILE ( Timer > 0.000sec )
34:             Sound count = 0
35:         }
36:         ELSE
37:         {
38:             CALL FlowerBend
39:         }
40:     }
```

Подсчет звуков осуществляется с помощью функции Sound count, которая автоматически выдает количество последовательных хлопков. Если зафиксирован один хлопок, вызывается функция FlowerBend, если число хлопков более двух, вызывается функция Wonder.

Все возможные движения робота описываются с помощью паттернов в RoboPlus Motion. Управляющая программа в необходимый момент ссылается на нужный паттерн, реализующий одно из движений робота. Таким образом, можно скомпоновать программу управления любым роботом.

```
134: FUNCTION Blossom
135: {
136:     Motion Page = 5
137: }
138:
139: FUNCTION Wonder
140: {
141:     Motion Page = -1
142:     WAIT WHILE ( Motion Status == TRUE )
143:     Motion Page = 9
144:     WAIT WHILE ( Motion Status == TRUE )
145: }
146:
147: FUNCTION FlowerBend
148: {
149:     Motion Page = -1
150:     WAIT WHILE ( Motion Status == TRUE )
151:     Motion Page = 1
152:     WAIT WHILE ( Motion Status == TRUE )
153: }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMPLE.TECHNOLAB.RU

В данной программе используется особый прием – обращение в теле функции к используемому в данный момент паттерну. Если реализовать равенство Motion Page = -1, то программа управления перейдет к выполнению паттерна, работающего в данный момент. Благодаря этому можно прервать выполнение какого-либо действия или скорректировать его в процессе работы робота.

Также данная работа предназначена для того чтобы обратить внимание разработчиков на функцию Motion Status, доступную в панели управления.

Функция Motion Status возвращает текущий статус выполнения паттерна, т.е. по сути является флагом, сигнализирующим о том, выполнено ли запрограммированное действие или нет.

В рамках данной работы описываются базовые приемы работы со средами программирования RoboPlus Task и RoboPlus Motion, применяемых для разработки систем управления роботами на базе наборов Bioloid. Рассматриваемый пример иллюстрирует возможность применения паттернов, разработанных в RoboPlus Motion, в программах, реализованных в RoboPlus Task, а также демонстрирует возможность изменять ход выполнения паттернов в зависимости от алгоритма управления робота.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ЭКЗАМЕН-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 15

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота, отслеживающего посторонние объекты

ЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ЭКЗАМЕНА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 15

«Разработка робота, отслеживающего посторонние объекты»

В зависимости от поставленных задач роботы и робототехнические системы могут оснащаться различными сенсорными устройствами. Чаще всего они применяются для обнаружения посторонних объектов в рабочей зоне, но возможны и другие применения. По мере роста числа применений робототехнических решений в повседневной жизни человека перед разработчиками встает проблема человеко-машинного взаимодействия.

Взаимодействие человека и робота осуществляется множеством различных способов, например с помощью различных пультов управления, как в ранее рассматриваемых работах или с помощью звуковых команд. В этом случае робот функционирует в полуавтоматическом режиме, выполняя команды человека. Подобное решение может быть не применимо для полностью автоматических систем управления. Робот, функционирующий в автоматическом режиме, должен самостоятельно обнаруживать человека и в зависимости от его действий принимать собственные решения. Естественно, многие

подобные задачи не решаемы на сегодняшний день, но в данный момент робототехника является одной из наиболее динамично развивающихся технических дисциплин и с каждым днем все больше и больше реальных задач решается роботами.

Одной из проблем автономного взаимодействия робота с человеком является обнаружение и распознавание роботом человека и его действий. Данная лабораторная работа предназначена для демонстрации простейшего решения обнаружения человека и его сопровождения.

В рамках данной работы предлагается разработать модель ходящего робота, который способен обозревать окружающее пространство с помощью ИК-датчика, установленного на поворачивающейся голове. Благодаря этому робот вращает головой и при обнаружении объекта движется за ним.

Программа управления робота начинается с базовой инициализации и вызова функции Initial, реализующей последовательность действий робота при запуске.

```
150: FUNCTION Initial
151: {
152:     Buzzer time = 255
153:     Buzzer index = 0
154:     CALL Ready
155:     CALL Forward
156:     CALL Reverse
157:     CALL SitDown
158: }
159:
160: FUNCTION Ready
161: {
162:     Motion Page = 2
163:     CALL WaitMotion
164: }
165:
166: FUNCTION Forward
167: {
168:     Motion Page = 3
169:     CALL WaitMotion
170: }
```

В момент запуска робот воспроизводит мелодию, делает несколько шагов вперед и назад, а после садится на задние лапы.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
21:     ENDLESS LOOP
22:     {
23:         IF ( PORT[1] >= 10 )
24:             CALL Follow
25:
26:         IF ( ID[1]: Moving != 1 )
27:         {
28:             IF ( Toggle == 0 )
29:             {
30:                 // turn head left
31:                 ID[1]: Moving speed = 30
32:                 ID[1]: Goal position = 712
33:                 Toggle = 1
34:             }
35:             ELSE IF ( Toggle == 1 )
36:             {
37:                 // turn head right
38:                 ID[1]: Moving speed = 30
39:                 ID[1]: Goal position = 312
40:                 Toggle = 0
41:             }
42:         }
```

Основной алгоритм управления реализуется в бесконечном цикле, который заключается во вращении головы робота справа налево и анализе показаний ИК-датчика. Направление вращения головы робота определяется с помощью специальной переменной флага Toggle, меняющего свое значение в зависимости положения головы робота.

Как только ИК-датчик обнаруживает объект, вызывается функция Follow, предназначенная для следования робота за обнаруженным объектом. Как только управление передается функции Follow, робот поднимается и начинает двигаться за обнаруженным объектом, если расстояние до объекта увеличивается – робот увеличивает скорость своего движения.

```
131: FUNCTION Follow
132: {
133:     CALL GetUp
134:     CALL Forward
135:     ENDLESS LOOP
136:     {
137:         IF ( PORT[1] >= 120 )
138:         {
139:             CALL Forward
140:         }
141:         ELSE IF ( PORT[1] >= 15 && PORT[1] < 120 )
142:             CALL FastForward
143:         ELSE
144:             BREAK LOOP
145:     }
146:     CALL SitDown
147: }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Все движения и походка робота описываются функциями GetUp, Forward, Fast-Forward, SitDown, реализованных в виде паттернов в RoboPlus Motion. Благодаря этому любое из движений робота может быть быстро изменено в случае необходимости.

Данная работа наглядно иллюстрирует простейший принцип следования робота за объектом. В качестве примера можно исследовать влияние на алгоритм отражающих свойств поверхности, как будет изменяться дальность восприятия роботом объекта и как от этого изменится алгоритм управления.

ТЕХНОЛАБ ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Лабораторная работа № 16

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота, маневрирующего среди препятствий

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 16

«Разработка робота, маневрирующего среди препятствий»

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Существует множество различных применений мобильных роботов, и все они так или иначе связаны с перемещением вблизи различных объектов. В связи с этим вопросы безопасности движения робота выходят на первый план. Для того чтобы движение робота в процессе работы не привело к столкновению с объектами, система управления робота должна непрерывно оценивать текущее положение робота и положение объектов, относительно него.

Чем сложнее робототехническая система, тем более ответственным должен быть подход к вопросам, касающимся безопасности людей и окружающих объектов. Для безопасного передвижения мобильный робот должен иметь возможность обозревать окружающее его пространство и строить локальную карту местности. С этой целью роботы и робототехнические системы оснащаются сенсорными системами, состоящими из различных датчиков.

В данной работе предлагается рассмотреть процесс функционирования мобильного робота, оснащенного сенсорной системой, состоящей из двух ИК-датчиков и ИК-дальномера. С помощью этих датчиков система управления робота анализирует наличие объектов, препятствующих движению робота как спереди, так и по бокам относительно робота. Двигаясь прямолинейно, робот анализирует расстояние до ближайших объектов и при обнаружении препятствия обходит его. Если препятствие было обнаружено слева, робот поворачивает направо, если препятствие было справа – робот движется налево. В случае если препятствие обнаружено перед роботом, оценивается расстояние до препятствий справа и слева и выбирается более простое направление для маневра, если направление выбрать невозможно – робот движется налево.

Программа управления робота достаточно проста и представляет бесконечный цикл, анализирующий условия срабатывания датчиков.

```
19:   ENDLESS LOOP
20:   {
21:       IF ( PORT[3] >= 400 )
22:           CALL Evade
23:       ELSE IF ( PORT[2] >= 15 )
24:           CALL EvadeRight
25:       ELSE IF ( PORT[1] >= 15 )
26:           CALL EvadeLeft
27:       ELSE
28:           CALL Forward
29:   }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САМ.TECHNOLAB.RU

В зависимости от показаний датчиков робот выполняет ряд маневров: движение направо с помощью функции EvadeRight, движение налево с помощью функции EvadeLeft и обход препятствия при движении прямо с помощью функции Evade.

```
175: FUNCTION EvadeLeft
176: {
177:     CALL Stop
178:     CALL GoLeft
179:     WAIT WHILE ( PORT[1] >= 10 )
180:     CALL Stop
181: }
```

При обнаружении препятствия сбоку робот движется в противоположную сторону до тех пор, пока препятствие не исчезнет из виду. После этого выполнение функции прекращается, и робот переходит к прямолинейному движению под управлением функции Forward.

При обнаружении препятствия перед роботом оценивается, на каком расстоянии от робота находится объект и с какой из сторон он ближе. В следствие этого выбирается предпочтительное направление для движения.

```
137: FUNCTION Evade
138: {
139:     CALL Stop
140:     CALL Reverse
141:     WAIT WHILE ( PORT[3] >= 300 )
142:     CALL Stop
143:     IF ( PORT[2] >= 15 )
144:         CALL TurnRight
145:     ELSE IF ( PORT[1] >= 15 )
146:         CALL TurnLeft
147:     ELSE
148:         CALL TurnLeft
149: }
```

Маневры робота реализуются с помощью функций Evade, EvadeRight и EvadeLeft, причем каждая из этих функций реализована с помощью редактора паттернов RoboPlus Motion, поэтому может быть видоизменена достаточно легко. Благодаря этому становится возможным использовать алгоритм работы сенсорной системы и системы управления робота без привязки к конкретной кинематике шасси робота. Это дает возможность реализовывать универсальные системы управления, применимые для роботов различного типа.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНТЕХНОЛАБ.RU

Лабораторная работа № 17

ЭКЗАМЕН ТЕХНОЛАБ

Управление шагающим роботом

ЭКЗАМЕН-ТЕХНОЛАБ.RU
ЗНАКОМИТЕЛЬНАЯ

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXCHENGETECHNOLAB.RU

Лабораторная работа № 17

«Управление шагающим роботом»

Данная лабораторная работа посвящена вопросам управления шагающим роботом, а в частности его движению в среде с различными препятствиями. Как и в предыдущей работе, робот оснащается сенсорной системой, состоящей из двух ИК-датчиков, установленных по бокам и ИК-дальномера, расположенного по центру робота.

В предыдущей работе было отмечено, что при правильном подходе к разработке системы управления, алгоритм обхода препятствий может быть полностью независим от алгоритмов перемещения робота.

```
19:  ENDLESS LOOP
20:  {
21:      IF ( PORT[1] >= 400 )
22:      {
23:          CALL Reverse
24:      }
25:      ELSE IF ( PORT[3] >= 20 )
26:      {
27:          LOOP WHILE ( PORT[3] >= 15 )
28:              CALL GoRight
29:      }
30:      ELSE IF ( PORT[4] >= 20 )
31:      {
32:          LOOP WHILE ( PORT[4] >= 15 )
33:              CALL GoLeft
34:      }
35:      ELSE
36:          CALL Forward
37:  }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Программа управления и ее алгоритм в целом идентичны рассмотренному алгоритму в предыдущей работе. Основные отличия заключаются только в алгоритме движения самого робота из-за различий в кинематике.

Поскольку за основу робота взято шагающее шасси, то одной из важнейших задач в процессе движения является сохранение устойчивого положения. В связи с этим выбрана модель походки робота, идентичная птичьей. Такая походка отличается тем, что каждый шаг сопровождается наклоном туловища в противоположную сторону. Программным образом походка задается с помощью группы паттернов в RoboPlus Motion, каждый из которых задает простейшее движение робота – вынос ноги вперед, постановку стопы, наклон туловища и т.п.

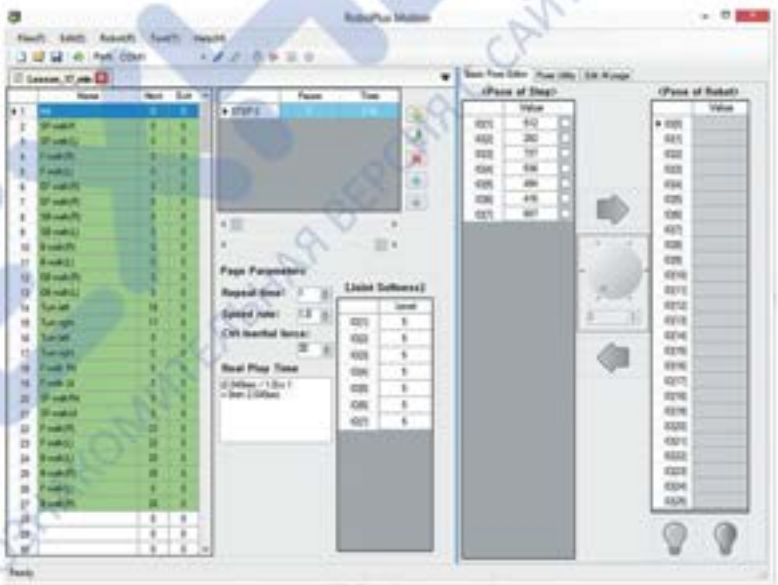
```
231: FUNCTION GoRight
232: {
233:     IF ( Motion Page == 2 )
234:         Motion Page = 4
235:     ELSE IF ( Motion Page == 4 )
236:         Motion Page = 4
237:     ELSE IF ( Motion Page == 5 )
238:         Motion Page = 18
239:     ELSE IF ( Motion Page == 6 )
240:         Motion Page = 20
241:     ELSE IF ( Motion Page == 7 )
242:         Motion Page = 20
243:     ELSE IF ( Motion Page == 12 )
244:         Motion Page = 20
245:     ELSE IF ( Motion Page == 13 )
246:         Motion Page = 20
247:     ELSE IF ( Motion Page == 20 )
248:         Motion Page = 4
249:     ELSE
250:         CALL Forward
251: }
```

```
205: FUNCTION Reverse
206: {
207:     IF ( Motion Page == 2 )
208:         Motion Page = 27
209:     ELSE IF ( Motion Page == 3 )
210:         Motion Page = 24
211:     ELSE IF ( Motion Page == 4 )
212:         Motion Page = 27
213:     ELSE IF ( Motion Page == 5 )
214:         Motion Page = 24
215:     ELSE IF ( Motion Page == 8 )
216:         Motion Page = 11
217:     ELSE IF ( Motion Page == 9 )
218:         Motion Page = 10
219:     ELSE IF ( Motion Page == 10 )
220:         Motion Page = 11
221:     ELSE IF ( Motion Page == 11 )
222:         Motion Page = 10
223:     ELSE IF ( Motion Page == 14 )
224:         Motion Page = 11
225:     ELSE IF ( Motion Page == 15 )
226:         Motion Page = 10
227:     ELSE
228:         Motion Page = 8
229: }
```

Выше приведены примеры реализации функций поворота направо и движения назад. Каждая из этих функций состоит из набора условий, соответствующих возможным положениям робота при движении. В зависимости от текущего положения робота вызывается определенный паттерн, реализующий следующее движение, допустимое на данном шаге.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.TECHNOLAB.RU

Проанализировав файл программы RoboPlus Motion, можно более детально ознакомиться с каждым из паттернов в отдельности. Всего в программе управления движением шагающего робота используется 27 различных паттернов, описывающих элементарные движения робота.



Данная программа может служить наглядным примером необходимости реализации систем управления роботами и робототехническими системами в виде отдельных функций и паттернов. Благодаря такому подходу системы управления получаются инвариантными к особенностям кинематики шасси, что дает возможность осуществлять их быструю переналадку.

Ознакомительная версия сайта EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКЗАМЕН-ТЕХНОЛАБ.РУ

Лабораторная работа № 18

ЭКЗАМЕН ТЕХНОЛАБ

Управление макетом боевого робота

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 18

«Управление макетом боевого робота»

Данная лабораторная работа посвящена разработке системы управления боевым шагающим роботом. Боевой робот обладает двумя руками, с помощью которых он может атаковать объекты, обнаруживаемые на его пути с помощью ИК-датчика.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Отличительная особенность данной работы заключается в том, что предлагаемая система управления позволяет роботу определять факт собственного падения. Причем помимо того, что робот упал, определяется как он упал – на спину или на грудь.

```
ENDLESS LOOP
{
    CALL Forward

    IF ( ID[3]: Present load <= 100 && falldown < 2 )
        falldown = falldown + 1
    ELSE IF ( ID[7]: Present load >= 50 && falldown < 2 )
        falldown = falldown + 1
    ELSE IF ( ID[8]: Present load >= 50 && falldown < 2 )
        falldown = falldown + 1
    ELSE IF ( ID[3]: Present load <= 100 && falldown >= 2 )
        CALL StandUp
    ELSE IF ( ID[7]: Present load >= 50 && falldown >= 2 )
        CALL StandUp
    ELSE IF ( ID[8]: Present load >= 50 && falldown >= 2 )
        CALL StandUp
    ELSE IF ( ID[3]: Present load >= 1000 )
        falldown = 0

    IF ( PORT[1] >= 10 )
        CALL Attack
}
```

Управляющая программа представляет собой бесконечный цикл, в котором осуществляется прямолинейное движение и анализ показаний ИК-датчика. При обнаружении ИК-датчиком препятствия на пути робота, вызывается функция Attack, которая приводит в атакующее движение обе руки робота.

```
FUNCTION Attack
{
    Motion Page = 9
    CALL WaitMotion
    Motion Page = 10
    CALL WaitMotion
    Motion Page = 11
    CALL WaitMotion
    Motion Page = 12
    CALL WaitMotion
    Motion Page = 2
    CALL WaitMotion
}
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Больший интерес представляет оставшаяся часть цикла ENDLESS LOOP, содержащая последовательность условий ELSE IF. Каждое из этих условий анализирует текущую нагрузку на привода робота с помощью функции Present load и значение флага falldown. Если нагрузка на приводы отсутствует или почти мала, это может означать то, что робот упал и его ноги движутся в воздухе. Для того чтобы робот вернулся в рабочее положение, вызывается функция StandUp, вызывающая последовательность движения робота, приводящих его в вертикальное положение.

```
158: FUNCTION StandUp
159: {
160:     CALL InitialPosition
161:     falldown = 0
162:
163:     IF ( PORT[1] >= 50 )
164:         CALL BackwardsGetUp
165:     ELSE
166:         CALL ForwardGetUp
167: }
```

Функция StandUp с помощью ИК-датчика определяет, каким образом робот упал – на грудь или на спину, и в зависимости от этого вызывает одну из функций BackwardsGetUp и ForwardGetUp.

```
220: FUNCTION ForwardGetUp
221: {
222:     Motion Page = 13
223:     CALL WaitMotion
224: }
225:
226: FUNCTION BackwardsGetUp
227: {
228:     Motion Page = 14
229:     CALL WaitMotion
230: }
231:
232: FUNCTION WaitMotion
233: {
234:     WAIT WHILE ( Motion Status == TRUE )
235: }
```

Данная работа примечательна тем, что иллюстрирует, каким образом по взаимосвязи параметров в сложной робототехнической системе косвенным образом можно определить состояние системы и выработать управляющее воздействие. Подобный подход позволяет решать поставленные задачи без усложнения системы путем добавления дополнительных устройств или усовершенствования алгоритмов управления.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 19

ЭКЗАМЕН ТЕХНОЛАБ

Управление четвероногим шагающим роботом

ЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ЭКЗАМЕНА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 19

«Управление четвероногим шагающим роботом»

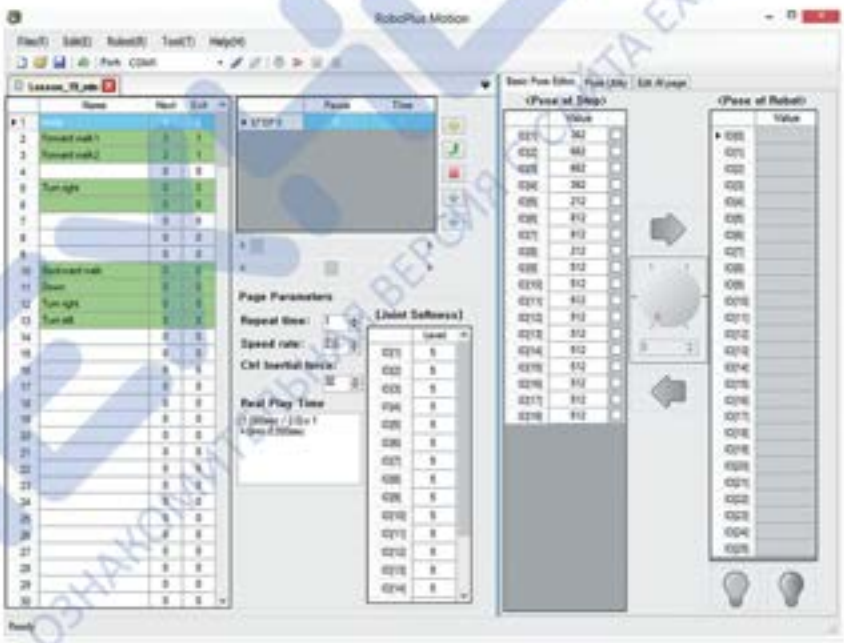
В предыдущих работах было рассмотрено множество роботов, обладающих различной кинематикой и принципами передвижения. Данная работа посвящена одному из часто встречающихся типов шагающих шасси – четвероногому шагающему роботу, в иностранной литературе также называемому «Quadruped». Подобный тип шасси достаточно часто применяется в исследовательской деятельности для моделирования походки живых существ, решения задач планирования движения в ограниченной местности и т.п.

Робот оснащен датчиками, с помощью которых он может обнаруживать объекты, препятствующие его движению. С помощью ИК-датчика робот обнаруживает объекты над собой и благодаря этому может во время пригнуться. Благодаря ИК-дальномеру робот обнаруживает препятствия на своем пути и может обходить их.

```
19:  ENDLESS LOOP
20:  {
21:      IF ( PORT[1] >= 250 )
22:      {
23:          CALL Stop
24:          CALL GoRight
25:      }
26:      IF ( PORT[2] >= 40 )
27:      {
28:          CALL Stop
29:          CALL Down
30:      }
31:      CALL Forward
32:  }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ – EXAMEN-TECHNOLAB.RU

В случае если никаких препятствий на пути робота не обнаружено, он продолжает собственное прямолинейное движение в заданном направлении.



Все движения робота реализуются в виде набора простейших паттернов, представляющих собой последовательность перемещений робота.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ПРОЕКТА EXAMEN-TECHNOLAB.RU

```
161 FUNCTION Reverse
162 {
163     Motion Page = 10
164 }
165
166 FUNCTION GoRight
167 {
168     Motion Page = 5
169     CALL WaitMotion
170     Motion Page = 6
171
172 }
173
174 FUNCTION GoLeft
175 {
176     Motion Page = 13
177 }
178 FUNCTION Down
179 {
180     Motion Page = 11
181 }
182 FUNCTION WaitMotion
183 {
184     WAIT WHILE ( Motion Status == TRUE )
185 }
```

Данная работа примечательна тем, что позволяет проанализировать различные возможные варианты походок робота. В рамках проведения эксперимента рекомендуется подробно ознакомиться с паттернами движения робота и модифицировать их. Например, можно разработать алгоритм движения боком, благодаря этому робот может обходить препятствия на своем пути, не изменяя направления движения. Также можно реализовать алгоритм движения на полусогнутых конечностях, в случае наличия какого-либо объекта над роботом. Благодаря этому робот может оценивать высоту препятствий и предпринимать попытку перемещаться под ними.

Подобные шагающие шасси обладают огромным потенциалом применения в исследовательской деятельности. Поэтому данная работа должна быть изучена крайне внимательно.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНТЕХНОЛАБ.RU

Лабораторная работа № 20

ЭКЗАМЕН ТЕХНОЛАБ

Управление шагающим роботом

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 20

«Управление шагающим роботом»

Данная лабораторная работа посвящена решению особенной задачи – на этот раз предлагается рассмотреть процесс разработки системы управления шагающего робота. Ранее в предыдущих работах нами уже рассматривались подобные роботы, но отличие данного робота заключается в походке. Пользователю предлагается разработать основание человекоподобного робота, передвигающегося привычными всем шагами.

Походка данного робота существенно отличается от ранее рассмотренной птичьей походки, которая основывалась на цикличном перемещении верхней части туловища робота в противоположную сторону от шагающей ноги.

Целью данной работы является разработка системы управления шагающим роботом, способным к обходу препятствий на своем пути. Робот обнаруживает препятствия с помощью ИК-дальномера, установленного на груди и двух ИК-датчиков по бокам.

В предыдущей работе уже рассматривался алгоритм движения робота, инвариантный к кинематической схеме шасси. Благодаря тому, что алгоритм движения робота не зависит от функций, реализующих его отдельные движения при походке, можно воспользоваться системой управления, используемой ранее в предыдущих работах.

ФУНКЦИОНАЛЬНАЯ ЕГЭ ИЗ САЙТА EXAMEN-TECHNOLAB.RU

```
ENDLESS LOOP
{
    IF ( PORT[1] >= 300 )
    {
        CALL Reverse
    }
    ELSE IF ( PORT[2] >= 20 )
    {
        LOOP WHILE ( PORT[2] >= 15 )
            CALL TurnRight
    }
    ELSE IF ( PORT[3] >= 20 )
    {
        LOOP WHILE ( PORT[3] >= 15 )
            CALL TurnLeft
    }
    ELSE
        CALL Forward
}
```

Управляющая программа представляет собой бесконечный цикл, состоящий из анализируемых условий. Каждое условие представляет собой анализ показаний датчиков: ИК-дальномера для обнаружения объектов спереди и ИК-датчиков для обнаружения объектов по бокам.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Стоит обратить внимание на то, что каждое из условий срабатывания датчика рассматривается дважды. Дублирование условий препятствует ложному срабатыванию датчика, а в данном случае система управления при обнаружении объекта будет ожидать изменения показаний ИК-датчика, вызванных приближением объекта к роботу.

```
FUNCTION TurnRight
{
    IF ( Motion Page == 6 )
        Motion Page = 18
    ELSE IF ( Motion Page == 7 )
        Motion Page = 22
    ELSE IF ( Motion Page == 8 )
        Motion Page = 18
    ELSE IF ( Motion Page == 9 )
        Motion Page = 22
    ELSE IF ( Motion Page == 12 )
        Motion Page = 21
    ELSE IF ( Motion Page == 13 )
        Motion Page = 25
    ELSE IF ( Motion Page == 14 )
        Motion Page = 25
    ELSE IF ( Motion Page == 15 )
        Motion Page = 21
    ELSE IF ( Motion Page == 25 )
        Motion Page = 21
    ELSE IF ( Motion Page == 21 )
        Motion Page = 21
    ELSE
        CALL Forward
}
```

```
FUNCTION TurnLeft
{
    IF ( Motion Page == 6 )
        Motion Page = 23
    ELSE IF ( Motion Page == 7 )
        Motion Page = 19
    ELSE IF ( Motion Page == 8 )
        Motion Page = 23
    ELSE IF ( Motion Page == 9 )
        Motion Page = 19
    ELSE IF ( Motion Page == 12 )
        Motion Page = 24
    ELSE IF ( Motion Page == 13 )
        Motion Page = 20
    ELSE IF ( Motion Page == 14 )
        Motion Page = 20
    ELSE IF ( Motion Page == 15 )
        Motion Page = 24
    ELSE IF ( Motion Page == 20 )
        Motion Page = 20
    ELSE IF ( Motion Page == 24 )
        Motion Page = 20
    ELSE
        CALL Forward
}
```

Каждое из движений робота описывается паттерном в программной среде RoboPlus Motion. Любое стандартное движение робота со сложной кинематикой описывается в виде последовательности перемещений множества приводов. Для более глубокого освоения основ работы с RoboPlus Motion и проектирования паттернов, рекомендуется уделить наибольшее внимание процессу разработки и моделирования движений. Видоизменяя паттерны управления, можно скорректировать различные системы управления, тем самым сократив время разработки.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 21

ЭКЗАМЕН ТЕХНОЛАБ

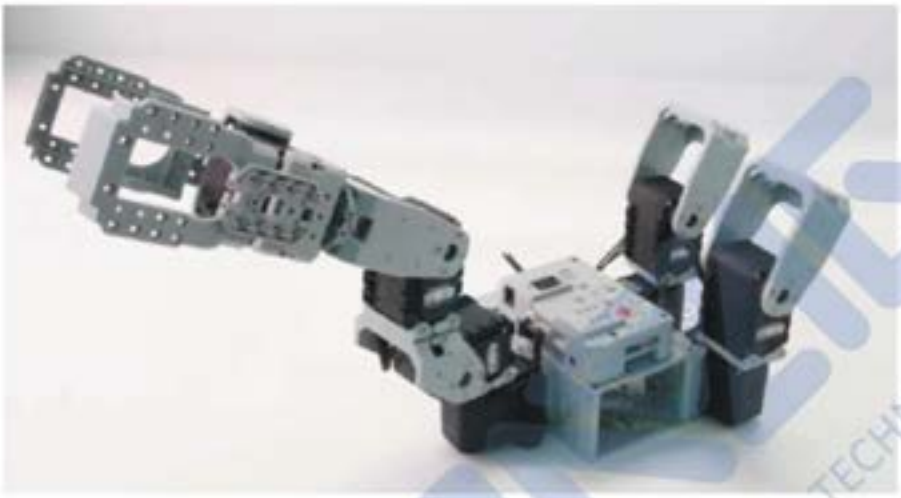
Управление манипулятором копирующего типа

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEKTEHTEXHOLAB.RU

Лабораторная работа № 21

«Управление манипулятором копирующего типа»



Существуют различные манипуляционные устройства, которые классифицируются в зависимости от решаемых задач. Некоторые манипуляторы могут работать полностью в автономном режиме, а некоторые – под управлением человека.

Существует отдельный тип манипуляторов, называемых копирующими. Данные манипуляторы представляют собой пару – рабочий манипулятор и его модель, управляемая оператором. Оператор вручную может управлять движением модели манипулятора, придавать его звеньям различное положение и манипулировать с его помощью объектами. В этом случае рабочий манипулятор полностью повторяет движения модели, перемещаясь в те же самые положения с той же скоростью и ускорением.

Подобные манипуляторы применялись в начале роботизации производств на особо опасных объектах и предназначались для манипулирования объектами, опасными для человека. На данный момент подобные системы не так часто встречаются, им на смену пришли полностью автономные робототехнические комплексы, а там, где применение автономных систем не возможно, используются пульты ручного управления.

В рамках данной работы предлагается разработать модель манипулятора, управляемого с помощью движения специальных рукояток. В качестве управляющих рукояток предлагается использовать подвижные механизмы на базе сервоприводов Dynamixel. Задавая положение рукояток можно управлять движением манипулятора.

Управляющая программа должна анализировать положение задающих приводов и на основе полученного значения их положения система управления должна рассчитывать положение исполнительного механизма.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Поскольку для движения механизма и приводов есть ряд ограничений, при базовой процедуре инициализации вводятся пределы углов поворота сервопривода.

```
155: FUNCTION ActuatorInitialization
156: {
157:     ID[1].ADDR[8(w)] = 0000 0011 1111 1111
158:     ID[2].ADDR[8(w)] = 0000 0011 1111 1111
159:     ID[3].ADDR[8(w)] = 0000 0011 1111 1111
160:     ID[4].ADDR[8(w)] = 0000 0011 1111 1111
161:     ID[5].ADDR[8(w)] = 0000 0011 1111 1111
162:     ID[6].ADDR[8(w)] = 0000 0011 1111 1111
163:     ID[7].ADDR[8(w)] = 0000 0011 1111 1111
164:     ID[8].ADDR[8(w)] = 0000 0011 1111 1111
165: }
```

Управляющая программа начинается с процедуры инициализации приводов и установки их стартовых значений. Для того чтобы каждый из приводов смог закончить свое движение к заданному положению, вводится специальная задержка, приостанавливающая выполнение программы.

В бесконечном цикле программы анализируется положение управляющих приводов с ID[5] – ID[8]. В зависимости от их положения рассчитывается положение приводов механизма манипулятора.

```
53      WHILE
54      {
55          IF ( ID[8].Present position >= 397 && ID[8].Present position <= 613 )
56          {
57              ID[1].Goal position = ID[8].Present position
58          }
59          IF ( ID[7].Present position >= 0 && ID[7].Present position <= 823 )
60          {
61              ID[2].Goal position = ID[7].Present position + 200
62          }
63          IF ( ID[5].Present position >= 154 && ID[5].Present position <= 858 )
64          {
65              ID[3].Goal position = ID[5].Present position
66          }
67          IF ( ID[6].Present position >= 113 && ID[6].Present position <= 900 )
68          {
69              ID[4].Goal position = ID[6].Present position
70          }
71      }
```

После каждого цикла задающие привода переключаются в режим свободного вращения, это крайне важно, поскольку при попытке провернуть их в режиме позиционирования можно повредить их.

Программа использует функцию Present position, с помощью которой определяет положение сервопривода. Данная функция возвращает значение из диапазона 0 – 1023, что соответствует углу вращения сервопривода.

Таким образом, данная работа демонстрирует возможность определения текущего положения сервопривода и применения полученного значения в качестве управляющего задания. Тем самым демонстрируется один из принципов следящих систем – наблюдение за объектом управления и выработка управляющего воздействия.

Лабораторная работа № 22

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-динозавра

ОЗНАКОМИТЕЛЬНАЯ

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 22

«Разработка робота-динозавра»

В данной работе предлагается сконструировать модель робота-динозавра, способного к самостоятельному передвижению, обнаружению объектов и атаке их. Для обнаружения объектов робот оснащается ИК-датчиком, устанавливаемым на голове.

Робот может выполнять множество различных движений, описываемых с помощью RoboPlus Motion. В управляющей программе переход между движениями осуществляется с помощью меток и оператора JUMP.

```
20: Normal :
21:     Motion Page = 1
22:     CALL CompleteMotion
23:     Motion Page = 6
24:     CALL CompleteMotion
25: Walk1 :
26:     Timer = 120
27: Walk2 :
28:     Motion Page = 11
29:     CALL CompleteMotion
30:     Motion Page = 10
31:     IF ( PORT[2] >= 10 )
32:         JUMP Threat
33:     ELSE IF ( Timer == 0 )
34:         JUMP Sleep
35:     JUMP Walk2
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

С помощью оператора JUMP реализуются условия перехода и циклы, но стоит помнить, что в программировании не рекомендуется применять данный оператор, поскольку его простота использования может привести к привычке применять его постоянно, что обычно отрицательно сказывается на качестве кода программы. Такой код достаточно тяжело воспринимать непосвященному в детали программисту, поэтому вместо оператора JUMP рекомендуется использовать традиционные средства языка программирования – операторы ветвлений, условного и безусловного перехода и т.п.

Почти вся программа построена на переходах от одного участка кода к другому, причем переход осуществляется сразу же к управляющим паттернам, реализующим одно из заданных движений. В отличие от предыдущих работ, каждое из движений описывается одним, максимум несколькими паттернами, поэтому их выполнение может занять достаточно много времени.

Для того чтобы новая команда не прервала выполнение паттерна, используется временная задержка на длительность его работы.

```
169: FUNCTION CompleteMotion
170: {
171:     WAIT WHILE (Motion Status == TRUE)
172: }
```

Выполняя последовательность паттернов, робот следит за показаниями таймера, который отсчитывает время, которое робот движется и не обнаруживает никаких объектов на пути. По истечении отсчета таймера робот переходит в спящий режим, из которого он может выйти по хлопку пользователя.

```
68: Sleep:
69:     Motion Page = 1
70:     CALL CompleteMotion
71:     Motion Page = 4
72:     CALL CompleteMotion
73:     Sound count = 0
74:     WAIT WHILE (Sound count < 2)
75:     JUMP Normal
```

Данный робот может выполнять еще много интересных движений и различных действий, поскольку его программа достаточно объемна, но в свою очередь в принципе проста, предлагается ознакомиться более подробно, усовершенствовав и добавив различные режимы, например ходьба на двух или четырех лапах, атака хвостом и многое другое.

Подобные роботы могут применяться в соревнованиях, проводимых среди учащихся, например можно организовать бои роботов или соревнования по прохождению маршрута на пересеченной местности. Диапазон различных применений подобных роботов в образовательном процессе велик, поэтому они могут удовлетворить требованиям как взыскательного педагога, так и любознательных учащихся.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Лабораторная работа № 23

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-динозавра. Использование универсального сенсорного модуля

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 23

«Разработка робота-динозавра. Использование универсального сенсорного модуля»

Данная работа служит продолжением предыдущей, поэтому для сборки модели можно воспользоваться предыдущей инструкцией и рекомендациями по установке универсального сенсорного модуля.

Универсальный сенсорный модуль AX-S1 позволяет расширить функционал любого робота, а в данном случае он выполняет функции ИК-датчика и микрофона.

Отличия в применении модуля AX-S1 заключаются в том, что он является сетевым устройством с ID[100], таким образом, он работает аналогичным образом как сервопривода Dynamixel.

```
9:  Usual :
10:     Motion Page = 1
11:     CALL MotionComplete
12:     Motion Page = 6
13:     CALL MotionComplete
14: Walk1 :
15:     Timer = 120
16: Walk2 :
17:     Motion Page = 10
18:     IF ( ID[100]: IR Center >= 30 )
19:         JUMP Menace
20:     ELSE IF ( Timer == 0 )
21:         JUMP Sleep
22:     JUMP Walk2
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Получение показаний с ИК-датчиков, встроенных в модуль AX-S1, осуществляется с помощью команд ID[100]: ID Center, ID[100]: ID Left, ID[100]: ID Right. Вызов каждой из функций осуществляется из панели управления.

```
Usual :
    Motion Page = 1
    CALL MotionComplete
    Motion Page = 6
    CALL MotionComplete
Walk1 :
    Timer = 120
Walk2 :
    Motion Page = 10
    IF ( ID[100]: IR Center >= 30 then )
        JUMP Menace
    ELSE IF ( Timer == 0 )
        JUMP Sleep
    JUMP Walk2

Menace :
    CALL MotionComplete
    Timer = 2
```

Аналогичным образом осуществляется работа со встроенным в AX-S1 микрофоном, распознающим и производящим подсчет различных звуков.

```
Sleep :
    Motion Page = 1
    CALL MotionComplete
    Motion Page = 4
    CALL MotionComplete
    ID[100]: Sound count = 0
    WAIT WHILE ( ID[100]: Sound count < 2 )
    JUMP Usual
}
```

Универсальный сенсорный модуль AX-S1 позволяет расширить область применения роботов и робототехнических систем, проектируемых на базе робототехнических конструкторов Bioloid. Благодаря тому, что AX-S1 является сетевым устройством, его применение не требует использования дополнительных портов управления программируемого контроллера. Поскольку в связи с этим к роботу можно подключить дополнительные сенсорные устройства, можно существенно усовершенствовать робота и придать ему множество дополнительных возможностей, например обнаружение препятствий по бокам, дистанционное управление с помощью ИК-джойстика и т.п.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Лабораторная работа № 24

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-собачки

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 24

«Разработка робота-собачки»

В данной работе предлагается сконструировать робота-собачку, обнаруживающую окружающие объекты, реагирующую на хлопки и выполняющую множество различных акробатических трюков.

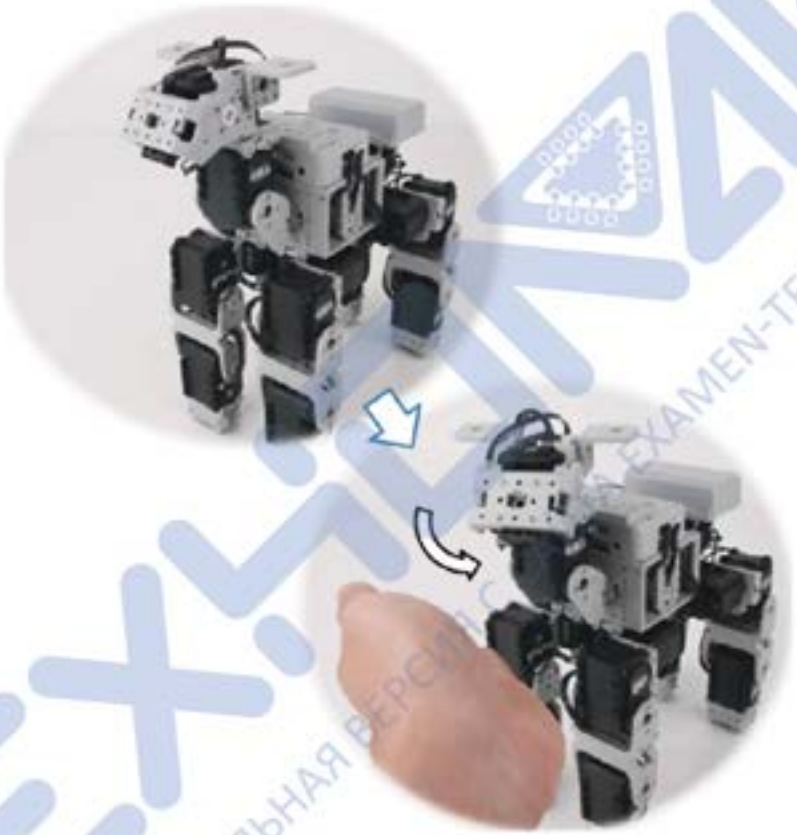
Робот может засыпать в случае длительного бездействия и просыпаться по хлопку, после чего начинать движение или выполнять различные трюки.

```
25: Sleep :
26:     CALL Stop
27:     Motion Page = 10
28:     CALL CompleteMotion
29:     Sound count = 0
30: SleepAgain :
31:     IF ( Sound count >= 1 )
32:         JUMP Awake
33:     JUMP SleepAgain
```

При движении робота в прямолинейном направлении с помощью ИК-датчика обнаруживаются препятствия на его пути, благодаря чему робот может корректировать собственный маршрут.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
75: Walk :
76:     Timer = 6.400sec
77:     Motion Page = 35
78: ForwardAgain :
79:     IF ( Timer <= 0 )
80:         JUMP RestAfterWalk
81:     ELSE IF ( PORT[1] >= 10 )
82:         JUMP AvoidObstacle
83:     JUMP ForwardAgain
```



Данный робот может выполнять еще много интересных движений и различных действий, поскольку его программа достаточно объемна, но в свою очередь в принципе проста, предлагается ознакомиться более подробно, усовершенствовав и добавив различные режимы.

Подобные роботы могут применяться в соревнованиях, проводимых среди учащихся. Диапазон различных применений подобных роботов в образовательном процессе велик, поэтому они могут удовлетворить требованиям как взыскательного педагога, так и любознательных учащихся.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ EXAMEN-TECHNOLAB.RU

Лабораторная работа № 25

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-собачки. Использование универсального сенсорного модуля

ЭКЗАМЕН-ТЕХНОЛАБ.РУ
ЗНАКОМИТЕЛЬНЫЙ ДОСТУП С САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 25

«Разработка робота-собачки. Использование универсального сенсорного модуля»

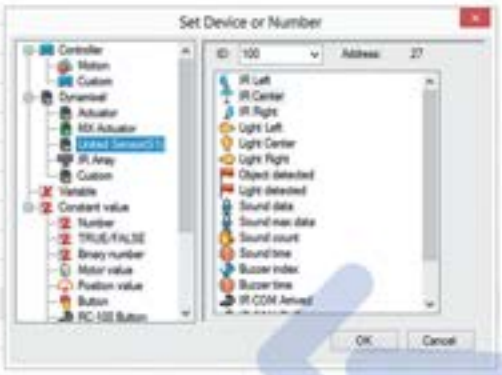
Данная работа служит продолжением предыдущей, поэтому для сборки модели можно воспользоваться предыдущей инструкцией и рекомендациями по установке универсального сенсорного модуля.

Универсальный сенсорный модуль AX-S1 позволяет расширить функционал любого робота, а в данном случае он выполняет функции ИК-датчика и микрофона.

Отличия в применении модуля AX-S1 заключаются в том, что он является сетевым устройством с ID[100], таким образом, он работает аналогичным образом как сервопривода Dynamixel.

```
SleepWake :
    Motion Page = 11
    CALL MotionComplete
    Motion Page = 24
    CALL MotionComplete

    ID[100]: Sound count = 0
WaitCommand :
    IF ( ID[100]: Sound count >= 2 )
        JUMP ReadyWalk
    ELSE IF ( ID[100]: IR Center >= 200 then )
        JUMP Cute
    JUMP WaitCommand
```



Получение показаний с ИК-датчиков, встроенных в модуль AX-S1 осуществляется с помощью команд ID[100]: ID Center, ID[100]: ID Left, ID[100]: ID Right. Вызов каждой из функций осуществляется из панели управления.

Аналогичным образом осуществляется работа со встроенным в AX-S1 микрофоном, распознающим и производящим подсчет различных звуков.

Универсальный сенсорный модуль AX-S1 позволяет расширить область применения роботов и робототехнических систем, проектируемых на базе робототехнических конструкторов Bioloid. Благодаря тому, что AX-S1 является сетевым устройством, его применение не требует использования дополнительных портов управления программируемого контроллера. Поскольку в связи с этим к роботу можно подключить дополнительные сенсорные устройства, можно существенно усовершенствовать робота и придать ему множество дополнительных возможностей, например обнаружение препятствий по бокам, дистанционное управление с помощью ИК-джойстика и т.п.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ © ТЕХНОЛАБ EDUCATIONAL TECHNOLAB.RU

Лабораторная работа № 26

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-паука

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ
EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEKTEXHOLAB.RU

Лабораторная работа № 26

«Разработка робота-паука»

В данной работе предлагается сконструировать модель робота-паука способного автономно перемещаться, обнаруживать препятствия и избегать столкновений с ними. Во время своего движения робот может обнаруживать препятствия на своем пути и атаковать их.

Робот оснащается двумя ИК-датчиками, направленными вперед и вверх, благодаря им он может обозревать рабочую зону на своем пути и над собой, тем самым планируя маршрут движения.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

```
29: Normal :
30:     CALL Step
31:     Motion Page = 50
32:     // 30 seconds
33:     Timer = 240
34: NormalAgain :
35:     IF ( PORT[3] >= 100 )
36:         JUMP Alert
37:     IF ( PORT[4] >= 20 )
38:         JUMP ObstacleReaction
39:     IF ( Timer == 0 )
40:         JUMP Sleep
41:     JUMP NormalAgain
```

Движения робота описываются с помощью Motion-файла, разработанного в Roboplus Motion. Переход от одного движения к другому осуществляется с помощью функции JUMP, перенаправляющей программу на место, где установлена метка.

```
69: Attack :
70:     CALL CompleteMotion
71:     Motion Page = 12
72:     CALL CompleteMotion
73:     // 10 seconds
74:     Timer = 80
75: AttackAgain :
76:     IF ( PORT[4] >= 20 )
77:         JUMP ObjectReaction
78:     IF ( PORT[3] >= 100 )
79:         JUMP Attack
80:     IF ( Timer == 0 )
81:         JUMP Normal
82:     JUMP AttackAgain
```

Один и тот же кусок программы можно реализовать с использованием либо меток, либо функций и строгих рекомендаций на этот счет не существует. Существует правило хорошего тона, говорящее о том, что использование меток при программировании не рекомендуется. Поэтому следует избегать применения меток и операторов JUMP, а вместо них использовать стандартные операторы условий, циклов и ветвления, вызывающие на исполнение функции.

Примечание: в среде программирования RoboPlus ссылки обозначаются фиолетовым цветом, а функции – оранжевым.

Данный робот может выполнять еще много интересных движений и различных действий, поскольку его программа достаточно объемна, но в свою очередь в принципе проста, предлагается ознакомиться более подробно, усовершенствовав и добавив различные режимы.

Подобные роботы могут применяться в соревнованиях, проводимых среди учащихся. Диапазон различных применений подобных роботов в образовательном процессе велик, поэтому они могут удовлетворить требованиям как взыскательного педагога, так и любознательных учащихся.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Лабораторная работа № 27

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-паука. Использование универсального сенсорного модуля

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКСМО-ТЕХНОЛАБ.РФ

Лабораторная работа № 27

«Разработка робота-паука. Использование универсального сенсорного модуля»

Данная работа служит продолжением предыдущей, поэтому для сборки модели можно воспользоваться предыдущей инструкцией и рекомендациями по установке универсального сенсорного модуля.

Универсальный сенсорный модуль AX-S1 позволяет расширить функционал любого робота, а в данном случае он выполняет функции ИК-датчика и микрофона.

Отличия в применении модуля AX-S1 заключаются в том, что он является сетевым устройством с ID[100], таким образом, он работает аналогичным образом как сервопривод Dynamixel.

```
Usual :
    CALL MotionStop
    Motion Page = 50
    // 30 Second
    Timer = 240
UsualRepeat :
    IF ( ID[100]: IR Left >= 200 then )
        JUMP Alert
    IF ( ID[100]: IR Center >= 20 )
        JUMP ResponseObstacle
    IF ( Timer == 0 )
        JUMP SleepState
    JUMP UsualRepeat
```

Получение показаний с ИК-датчиков, встроенных в модуль AX-S1 осуществляется с помощью команд ID[100]: ID Center, ID[100]: ID Left, ID[100]: ID Right. Вызов каждой из функций осуществляется из панели управления.

Аналогичным образом осуществляется работа со встроенным в AX-S1 микрофоном, распознающим и производящим подсчет различных звуков.

```
SleepState :
    CALL MotionStop
    Motion Page = 10
    CALL MotionComplete
    ID[100]: Sound count = 0
SleepRepeat :
    IF ( ID[100]: Sound count > 0 )
        JUMP Usual
    JUMP SleepRepeat

Usual :
    CALL MotionStop
    Motion Page = 50
    // 30 Second
    Timer = 240
```

Универсальный сенсорный модуль AX-S1 позволяет расширить область применения роботов и робототехнических систем, проектируемых на базе робототехнических конструкторов Bioloid. Благодаря тому, что AX-S1 является сетевым устройством, его применение не требует использования дополнительных портов управления программируемого контроллера. Поскольку в связи с этим к роботу можно подключить дополнительные сенсорные устройства, можно существенно усовершенствовать робота и придать ему множество дополнительных возможностей, например обнаружение препятствий по бокам, дистанционное управление с помощью ИК-джойстика и т.п.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ © ТЕХНОЛАБ
WWW.ROBOT-TECHNOLAB.RU

Лабораторная работа № 28

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-скорпиона

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ
WWW.EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Лабораторная работа № 28

«Разработка робота-скорпиона»

В рамках данной работы предлагается сконструировать модель робота-скорпиона, автономно передвигающегося, реагирующего на внешние звуки, обнаруживающего препятствия и атакующего их.

Реагирование на внешние звуки осуществляется с помощью встроенного в программируемый контроллер СМ-530 микрофона, а обнаружение препятствий с помощью ИК-дальномера.

Для того чтобы робот мог обнаруживать различные объекты в более широком диапазоне, ИК-дальномер устанавливается на вращающийся механизм, приводимый в движение с помощью сервопривода.

```
31: Normal :
32:     CALL Rise
33:     // sleeps after 10 seconds idle
34:     Timer = 10.240sec
35: NormalAgain :
36:     IF ( PORT[3] >= 100 && PORT[3] < 200 )
37:         JUMP Alert
38:     ELSE IF ( PORT[3] >= 200 && PORT[3] < 450 )
39:         JUMP ReadyAttack
40:     ELSE IF ( PORT[3] >= 450 )
41:         JUMP Attack
42:     ELSE IF ( Sound count >= 2 )
43:         JUMP Escape
44:     ELSE IF ( Timer == 0.000sec )
45:         JUMP Sleep
46:     JUMP NormalAgain
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

С помощью ИК-дальномера робот определяет расстояние до объекта, тем самым выбирая необходимое расстояние для атаки. Если объект находится на удалении от робота, то скорпион производит пугающие движения, угрожает жалом и клацает клешнями. В случае если объект приближается к роботу, робот-скорпион атакует.

```
70: Attack :
71:     CALL Stop
72:     Motion Page = 7
73:     CALL Completemotion
74:     IF ( PORT[3] < 100 )
75:         CALL Basic
76:     // sleeps after 10 seconds idle
77:     Timer = 10.240sec
78:     Sound count = 0
79:     JUMP NormalAgain
```

Сложные движения робота, например атакующие действия или передвижения описываются с помощью паттернов в RoboPlus Motion.

Робот может распознавать посторонние звуки, например: если вблизи спящего робота произвести хлопок, он проснется и приступит к работе. Если же произвести пару хлопков, робот попытается убежать от источника звука.

```
81: Escape :
82:     LOOP FOR ( Reverse = 0 ~ 3 )
83:     {
84:         CALL Reverse
85:     }
86:     LOOP FOR ( Right = 0 ~ 7 )
87:     {
88:         CALL GoRight
89:     }
90:     LOOP FOR ( Forward = 0 ~ 7 )
91:     {
92:         CALL FastForward
93:     }
94:     Timer = 10.240sec
95:     Sound count = 0
96:     JUMP NormalAgain
```

```
195: FUNCTION Forward
196: {
197:     Motion Page = 11
198:     CALL Completemotion
199: }
200:
201: FUNCTION Reverse
202: {
203:     Motion Page = 16
204:     CALL Completemotion
205: }
206:
207: FUNCTION GoLeft
208: {
209:     Motion Page = 21
210:     CALL Completemotion
211: }
```

Функции Forward, Reverse, GoLeft, описывающие движения робота-скорпиона, реализуются с помощью Motion-файла в RoboPlus Motion. Движение робота определяется как последовательность действий, реализованных на базе циклически выполняющихся функций.

Робот-скорпион может реагировать на внешние звуковые сигналы, фиксируемые с помощью микрофона. Благодаря этому он может считать хлопки пользователя и выполнять запрограммированные действия.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКЗАМЕН-ТЕХНОЛАБ.RU

```
20: Sleep :
21:     CALL PlayDead
22:     Sound count = 0
23: SleepAgain :
24:     IF ( Sound count > 0 )
25:     {
26:         Sound count = 0
27:         JUMP Normal
28:     }
29:     JUMP SleepAgain
```

Данный робот может выполнять еще много интересных движений и различных действий, поскольку его программа достаточно объемна, но в свою очередь в принципе проста, предлагается ознакомиться более подробно, усовершенствовав и добавив различные режимы.

Подобные роботы могут применяться в соревнованиях, проводимых среди учащихся. Диапазон различных применений подобных роботов в образовательном процессе велик, поэтому они могут удовлетворить требованиям как взыскательного педагога, так и любознательных учащихся.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ОТ EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКЗАМЕН-ТЕХНОЛАБ.РУ

Лабораторная работа № 29

ЭКЗАМЕН ТЕХНОЛАБ

Разработка робота-ящерицы

ЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЛАБОРАТОРИЯТЕХНОЛАБ.РУ

Лабораторная работа № 29

«Разработка робота-ящерицы»

В рамках данной лабораторной работы предлагается сконструировать модель робота-ящерицы, свободно передвигающейся в пространстве. При обнаружении объектов на своем пути робот останавливается и выполняет атакующие действия: угрожает хвостом и клацает пастью.

Робот оснащен ИК-дальномером, установленным спереди робота, что позволяет ему определять расстояние до стоящих впереди объектов. На хвосте робота располагается ИК-датчик, который позволяет обнаруживать препятствия позади робота.

```
18:   CALL Ready
19:   Scared = 0
20:   ENDLESS LOOP
21:   {
22:       IF (PORT[1] >= 15 )
23:           CALL Escape
24:       ELSE IF (PORT[0] >= 300 && Scared >= 2 )
25:       {
26:           CALL Evade
27:           Scared = 0
28:       }
29:       ELSE IF (PORT[0] >= 300 )
30:       {
31:           CALL Scared
32:           Scared = Scared + 1
33:       }
34:       ELSE
35:           CALL Forward
36:   }
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Робот запрограммирован таким образом, что при обнаружении объекта вблизи себя он должен попытаться убежать или каким-либо образом отреагировать на это. За поведение робота при обнаружении препятствий отвечает функция Scared, которая некоторое время воспроизводит устрашающие действия. После двукратного вызова функции Scared управление передается функции Evade, отвечающей за атакующие действия робота.

```
138: FUNCTION Scared
139: {
140:     CALL EvilSlapWithoutRun
141:     Motion Page = 11
142:     CALL CompleteMotion
143: }
144:
145: FUNCTION Evade
146: {
147:     IF ( Motion Page == 3 )
148:     {
149:         CALL EvilSlapWithoutRun
150:         Motion Page = 14
151:         CALL CompleteMotion
152:     }
153:     ELSE IF ( Motion Page == 4 )
154:     {
155:         CALL EvilSlapWithoutRun
156:         Motion Page = 13
157:         CALL CompleteMotion
158:     }
159:     ELSE
160:     {
161:         CALL EvilSlapWithoutRun
162:         Motion Page = 14
163:         CALL CompleteMotion
164:     }
165: }
```

С помощью функции Evade робот делает выпад в сторону объекта, препятствующего движению, а после убегает в противоположном направлении.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Если в процессе движения робота срабатывает ИК-датчик, установленный на хвосте, вызывается функция Escape, с помощью которой робот убегает от обнаруженного объекта.

Данный робот может выполнять еще много интересных движений и различных действий, поскольку его программа достаточно объемна, но в свою очередь в принципе проста, предлагается ознакомиться более подробно, усовершенствовав и добавив различные режимы.

Подобные роботы могут применяться в соревнованиях, проводимых среди учащихся. Диапазон различных применений подобных роботов в образовательном процессе велик, поэтому они могут удовлетворить требованиям как взыскательного педагога, так и любознательных учащихся.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНЫЙ ВАРИАНТ С САЙТА ЕКАМЕНСК-ТЕХНОЛАБ.РУ

Лабораторная работа № 30

ЭКЗАМЕН ТЕХНОЛАБ

Разработка человекоподобного робота

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ЭКЗАМЕН ТЕХНОЛАБ EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЛАБОРАТОРИЯ-ТЕХНОЛАБ.РФ

Лабораторная работа № 30

«Разработка человекоподобного робота»

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ ОБЪЕКТА EXAMEN-TECHNOLAB.RU

В рамках данной работы предлагается сконструировать человекоподобного робота, состоящего из 18 сервоприводов Dynamixel, ИК-дальномера, гироскопа для стабилизации при движении.

Робот обладает инфракрасным приемопередатчиком и может дистанционно управляться с помощью джойстика. Управление роботом происходит путем нажатия различных комбинаций клавиш.

Ниже приводится перечень возможных движений и действий, производимых роботом. Робот может выполнять различные движения – базовые перемещения, спортивные движения, боевые и социальные.

Движение (U / L / D / R)			
U	Вперед	**D**	Назад
L	Налево	**R**	Направо
U + L	Прямо и налево	**D + L**	Движение влево вбок
U + R	Прямо и направо	**D + R**	Движение вправо вбок

Изменение положения (1 + U / L / D / R)			
1 + U	Подъем из положения «лежа на животе»	**1 + D**	Подъем из положения «лежа на спине»
1 + L	Отжимания	**1 + R**	Стойка на руках

Изменение положения (2 + U / L / D / R)			
2 + U	Удар в грудь	**2 + D**	Размахивание рукой
2 + L	Приветствие	**2 + R**	Поклон

Футбольные движения (2 + U / L / D / R)			
3 + U	Блок вправо	**3 + D**	Блок влево
3 + L	Удар левой ногой	**3 + R**	Удар правой ногой

ЭКСКЛЮЗИВНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Атакующие движения (2 + U / L / D / R)			
4 + U	Атака	**4 + D**	Защитный блок
4 + L	Атака влево	**4 + R**	Атака вправо

Каждое из движений может быть запрограммировано пользователем и настроено на различные кнопки джойстика. Помимо этого робот может быть дополнительно оснащен ИК-датчиками, с помощью которых он может обнаруживать препятствия и избегать столкновения с ними.

Данный робот может выполнять еще много интересных движений и различных действий, поскольку его программа достаточно объемна, но в свою очередь в принципе проста, предлагается ознакомиться более подробно, усовершенствовав и добавив различные режимы.

Подобные роботы могут применяться в соревнованиях, проводимых среди учащихся. Диапазон различных применений подобных роботов в образовательном процессе велик, поэтому они могут удовлетворить требованиям как взыскательного педагога, так и любознательных учащихся.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЕКАМЕНСК.ТЕХНОЛАБ.RU

Основы работы с модулем на базе CMOS камеры

ЭКЗАМЕН ТЕХНОЛАБ

Основы работы с модулем на базе CMOS камеры

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATECHNOLAB.RU

Основы работы с модулем на базе CMOS камеры

Проблемы классификации и определения типа объекта давно уже будоражат умы исследователей. С развитием систем технического зрения стало возможным применять визуальную информацию для определения параметров исследуемого объекта или определения состояния рабочей обстановки. На данный момент камеры массово применяются в робототехнических системах для распознавания каких-либо объектов, обнаружения дефектов в продукции, построения карты окружающего пространства и др. Применение систем технического зрения обусловлено в первую очередь простотой и очевидностью получаемой информации. Благодаря применению камеры становится возможным определить габариты, цвет и положение объекта, оценить его положение относительно окружающих объектов.

В состав робототехнического модуля входит модуль на базе CMOS камеры, позволяющий реализовывать простейшие функции технического зрения на моделях, разработанных из робототехнического конструктора серии Bioloid.

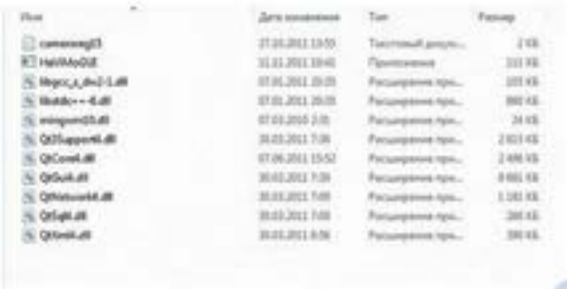
Технические характеристики:

- Разрешение 160x120 пикселей
- Глубина цвета 12 бит
- Частота кадров 19 кадров в секунду
- Возможность сохранения настроек камеры в EEPROM.
- Автоматическая и ручная экспозиция и баланс белого.
- Возможность распознавания до 256 различных объектов и 15 различных объектов или сопредельных областей в кадре.
- Совместимость с последовательными интерфейсами контроллера CM-530.
- Интерфейс TTL (уровень RS-232).
- 115200 BAUD в полнодуплексном режиме.
- 1 MBAUD в полудуплексном режиме.

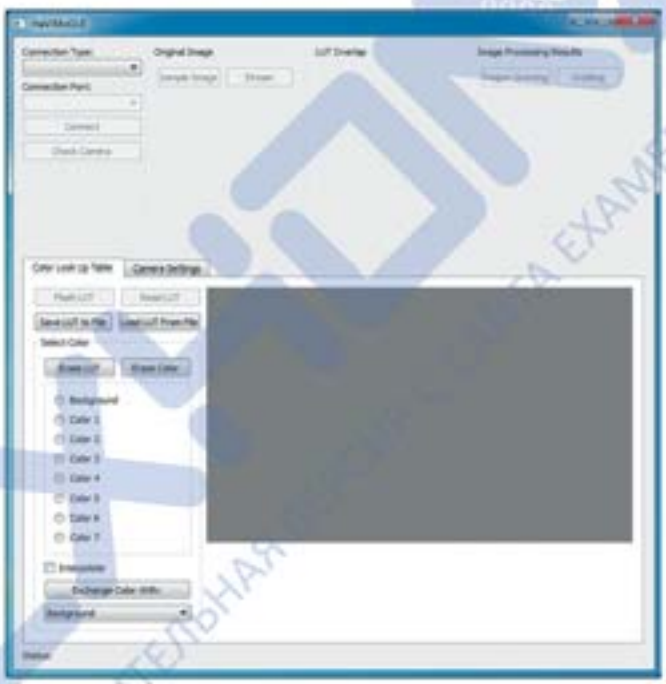
Модуль камеры HaViMo является уникальным устройством для малопроизводительных программируемых контроллеров. Обладая собственным встроенным вычислителем, данный модуль производит предварительную обработку информации, что позволяет использовать функции технического зрения в робототехнических системах невысокой вычислительной мощности.

Универсальный модуль камеры HaViMo подключается к одному из последовательных портов программируемого контроллера CM-530, устройству задается ID, и HaViMo может использоваться как сетевое устройство, наряду с сервоприводами, датчиками и т.п.

Настройка камеры осуществляется с помощью программного обеспечения HaViMoGUI. Для работы с программным обеспечением необходимо запустить HaViMoGUI на компьютере, желательно от имени администратора.



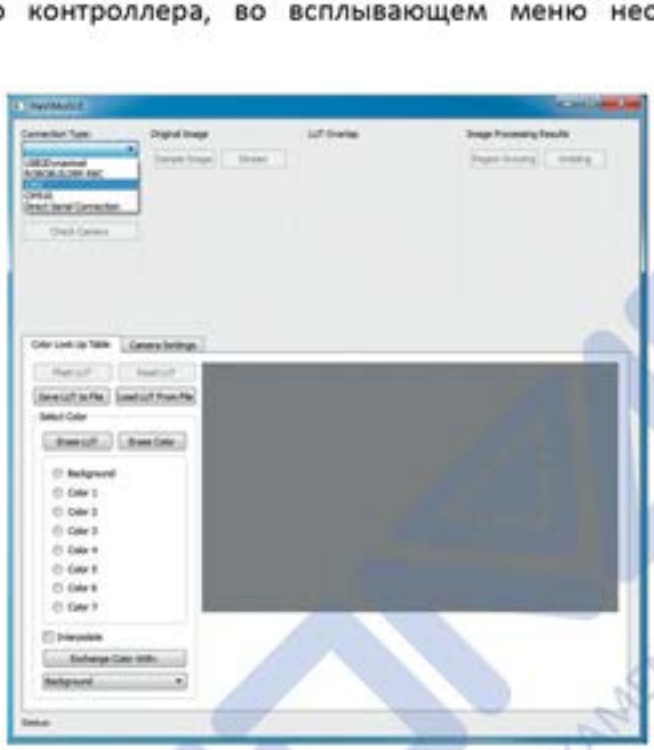
После запуска на экране компьютера должен появиться рабочий интерфейс и активироваться меню Connection Type, предназначенное для настройки типа соединения.



Для дальнейшей работы с камерой необходимо в меню Connection Type выбрать тип используемого соединения, а именно указать номер COM-порта к которому осуществлено подключение устройства.

Примечание: устройство HaViMo рекомендуется подключать к программируемому контроллеру, к одному из его последовательных портов. В свою очередь программируемый контроллер подключается к персональному компьютеру с помощью интерфейса USB. Тип подключения контроллера CM-530 с персональным компьютером распознается как виртуальный COM-порт с определенным номером. Данный номер является определяющим значением Connection Port.

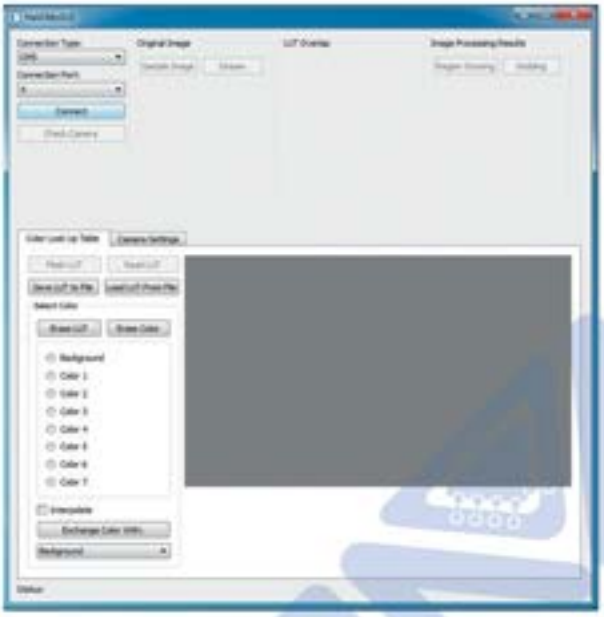
В меню Connection Type необходимо указать способ соединения камеры с персональным компьютером. Поскольку соединение осуществляется с помощью программируемого контроллера, во всплывающем меню необходимо выбрать требуемый тип.



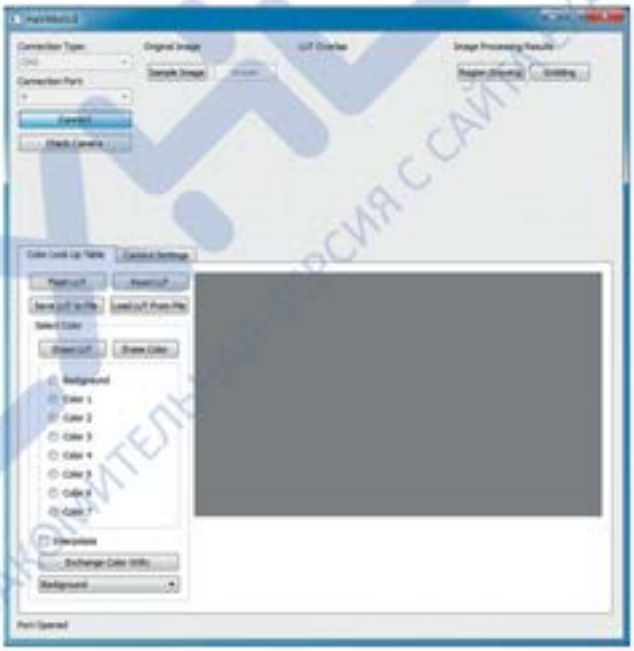
На рисунке ниже иллюстрируется подключение камеры с помощью программируемого контроллера CM-5 к персональному компьютеру к COM-порту 4.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ - СКИДКА EXAMEN-TECHNOLAB.RU

После осуществления настроек необходимо подключить устройства, чтобы установить соединение, необходимо нажать кнопку Connect.

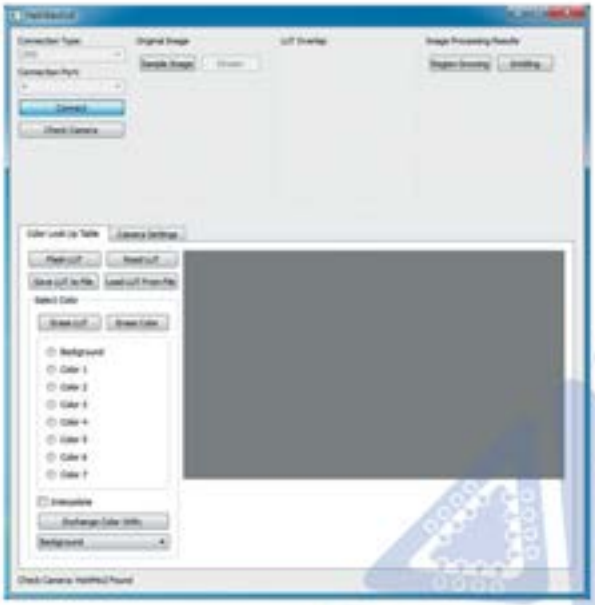


Если соединение было установлено успешно, в низу окна в статусной строке должно появиться сообщение Port Opened.

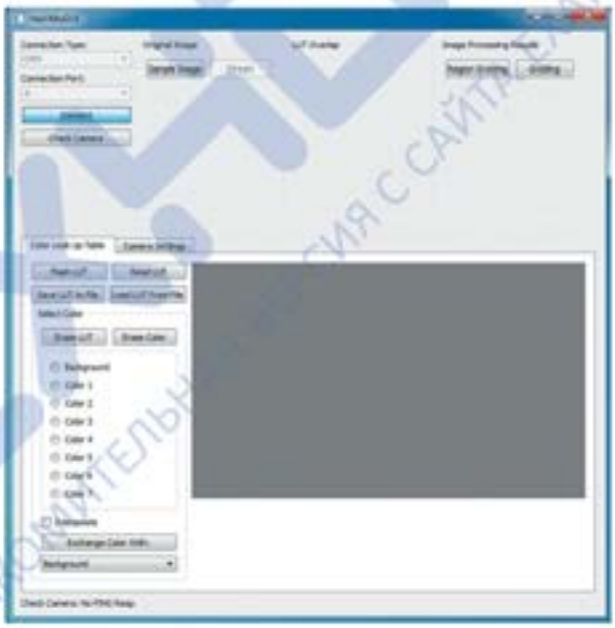


В случае неполадок с установлением соединения рекомендуется осуществить подключение устройств заново и повторить операцию Connect.

Подключение к модулю камеры осуществляется при нажатии на кнопку Check Camera. В случае корректного подключения в статусной строке должно появиться сообщение Check Camera: HaViMo2 Found.



Если в процессе установления соединения возникают неполадки, в статусной строке выводится сообщение No PING Resp.



Для устранения данной ошибки необходимо заново подключиться к контроллеру и повторить процедуру установки оборудования.

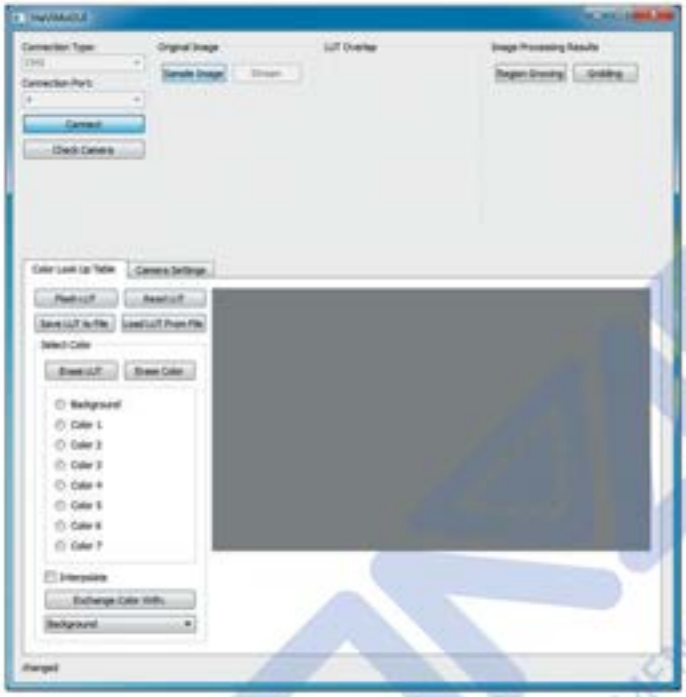
Примечание: для установки соединения крайне важно осуществлять правильное подключение устройств.

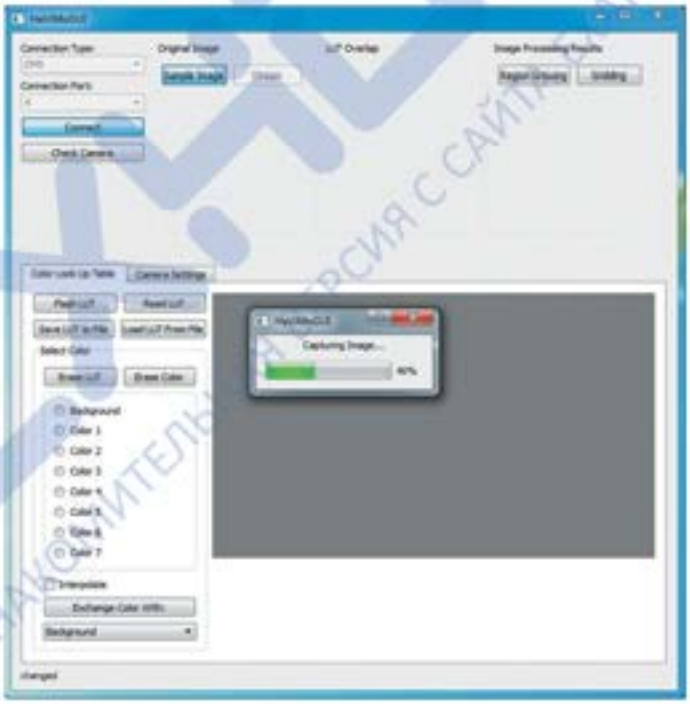
Подключившись к камере, необходимо перейти во вкладку Camera Settings и активировать режим Auto Update Settings, который позволяет камере автоматически выбирать яркость/контрастность (и другие настройки) изображения в зависимости от освещенности кадра.

После завершения настроек необходимо вернуться в раздел Color Look Up Table для дальнейшей работы с изображением.

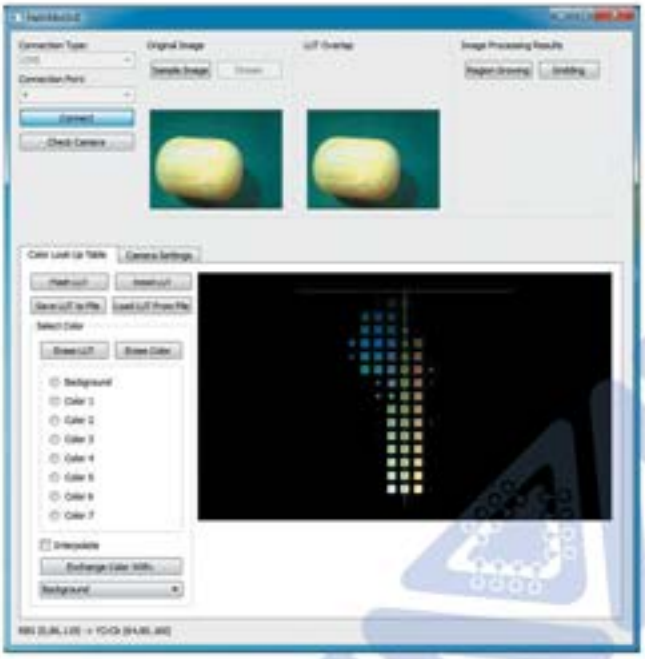
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.AMPERKA.RU

В меню Color Look Up Table необходимо нажать кнопку Sample Image для получения изображения с камеры.

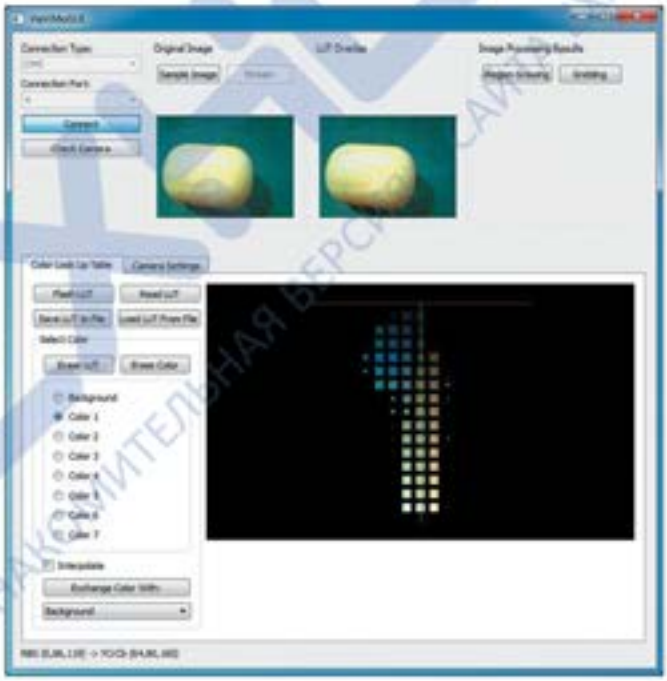




Полученное с камеры изображение автоматически отображается в окне Original Image.

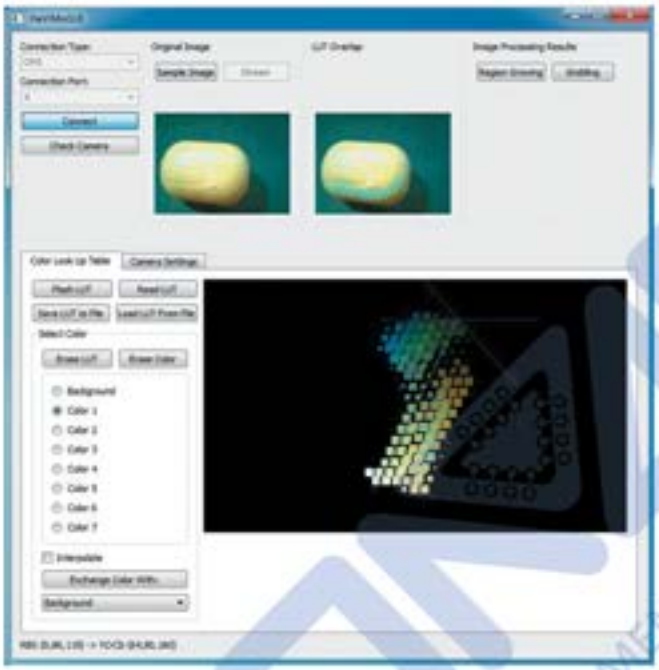


Во вкладке Color Look Up Table справа должны появиться распознанные камерой оттенки в виде 3D-градиента, поскольку камера распознает объект по его цвету.

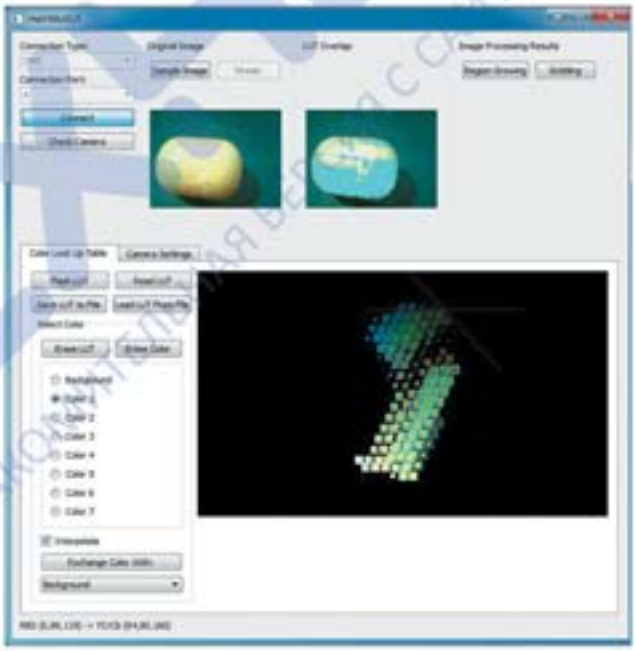


Для настройки камеры на распознавание данного объекта необходимо задать области одного цвета. Одновременно можно запомнить 7 различных объектов (+фон). Для того чтобы запомнить цвет, во вкладке Color Look Up Table необходимо выбрать один из цветов, например Color 1.

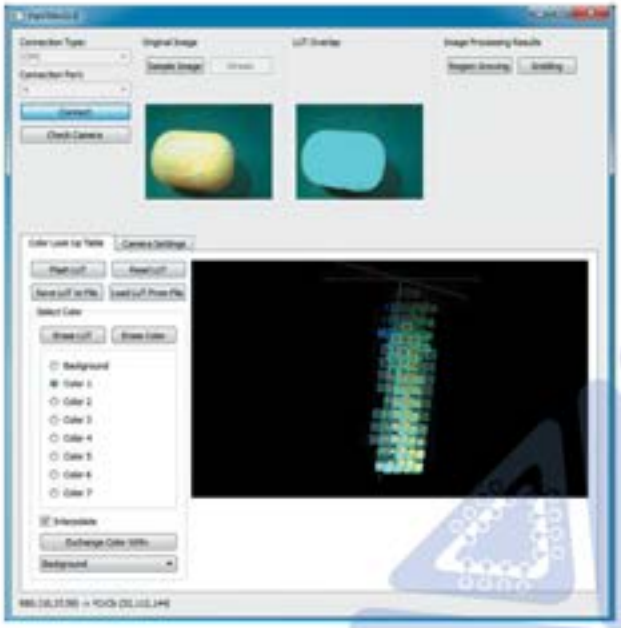
Оттенок цвета выбирается вручную при помощи 3D-градиента (необходимо выбрать один из элементов 3D-градиента) или при помощи Original Image, на котором участок заданного цвета выбирается вручную и отмечается прямо на изображении.



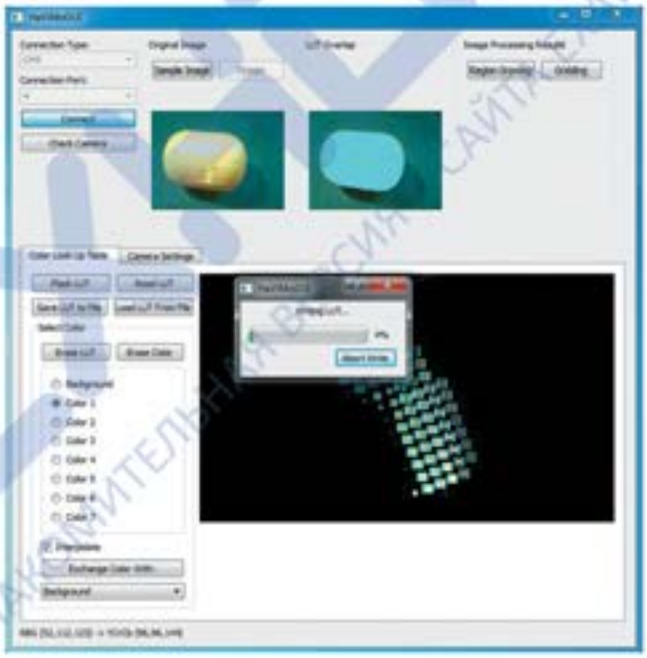
Для ускорения процесса задания цветов можно выбрать функцию Interpolate, которая выделяет в общую зону участки с одинаковым цветом.



Таким образом, поочередно задавая цвета, в кадре выделяется целиком весь объект. Для чистоты эксперимента можно поэкспериментировать с освещением, заново снять Sample Image и добавить недостающие оттенки, но это необязательно.

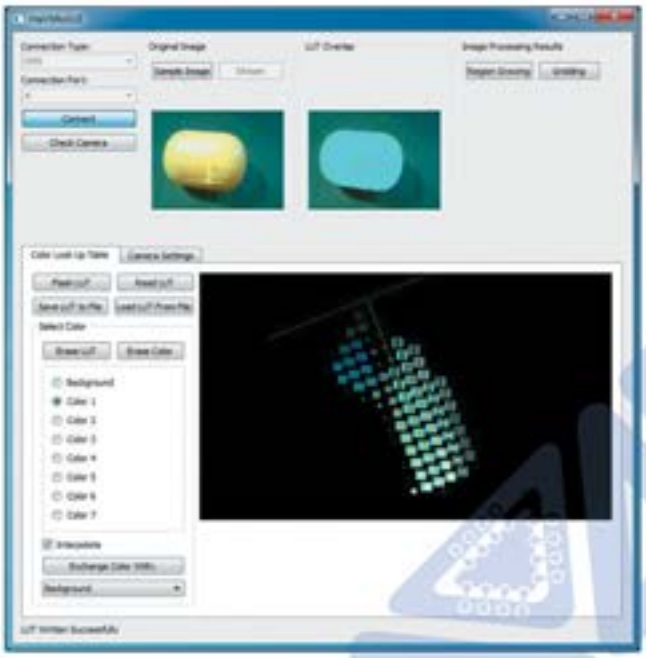


Полученные настройки системы технического зрения необходимо сохранить в ПЗУ модуля камеры. Для этого необходимо нажать на кнопку Flash LUT.



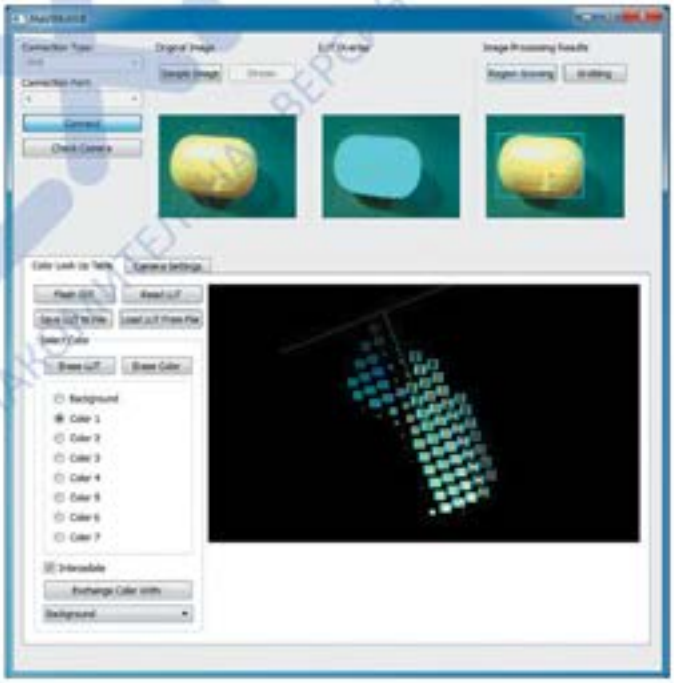
В результате должно появиться меню со строкой иллюстрирующей статус загрузки данных в память модуля камеры.

Если в статусной строке появилась надпись LUT Written Successfully, значит сохранение настроек успешно.

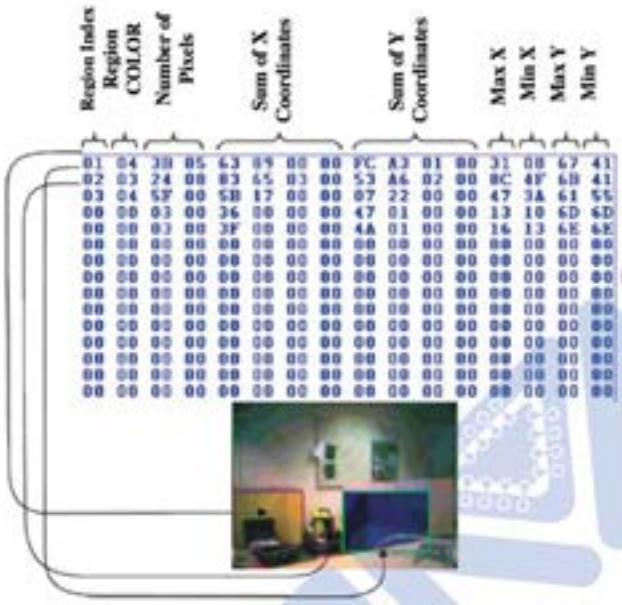


Также можно сохранить все настройки камеры в отдельный файл, чтобы в будущем не производить процедуру настройки и распознавания изображения заново. Для этого необходимо нажать кнопку Save LUT to File, для чтения настроек из камеры необходимо нажать Read LUT.

Полностью настроенная камера может самостоятельно распознавать изображения, для этого необходимо задать команду Region Growing, которая выделяет в кадре область на распознавание, которой была обучена камера.



Взаимодействие камеры и управляющего контроллера осуществляется с помощью определенного формата передачи данных.



Region Index – определяет тип области в кадре. Параметр равен 0, если область не относится к числу распознаваемых, в обратном случае значение параметра носит произвольный характер.

Region Color – параметр, определяющий цвет области.

Number of Pixels – количество пикселей, обнаруженных в данной области.

Sum of X Coordinates – результат сложения координат всех точек области по оси X. Используется для вычисления средней координаты.

Sum of Y Coordinates – результат сложения координат всех точек области по оси Y. Используется для вычисления средней координаты.

Max X – величина границы области справа.

Min X – величина границы области слева.

Max Y – величина границы области снизу.

Min Y – величина границы области сверху.

В качестве примера работы универсального модуля камеры HaViMo предлагается рассмотреть систему управления самонаводящегося на объект манипулятора. За основу модели предлагается использовать конструкцию манипуляционного механизма из лабораторной работы №4.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Для работы с камерой требуется усовершенствовать данную конструкцию, установив на качающейся скобе сверху угловой кронштейн с закрепленной камерой. Оптимальным вариантом должна быть ситуация, при которой камера направлена направо согласно данному рисунку.

Для того чтобы система технического зрения взаимодействовала с системой управления механизма, необходимо, чтобы данные от нее поступали в программируемый контроллер и использовались программой.

Управляющая программа начинается с базовой инициализации переменных и оборудования.

```
1: START PROGRAM
2: {
3:     ID[19] Moving speed = 150
4:     ID[20] Moving speed = 50
5:     Buzzer time = Play Melody
6:     Buzzer index = Melody0
7:     LOOP WHILE ( Button != 0 )
8:     {
9:         CamID = 100
10:        Color = 1
11:        ID[CamID] ADDR[0(0)] = 0
12:        Timer = 0.256sec
13:        WAIT WHILE ( Timer > 0.000sec )
14:        CALL Get_Bounding_Box
15:        Cx = Minx + Maxx
16:        Cy = Miny + Maxy
17:        Cx = Cx / 2
18:        Cy = Cy / 2
```

В начале программы модулю камеры присваивается идентификационный номер CamID=100. А также вычисляются координаты центра распознаваемой области Cx и Cy.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

После чего в бесконечном цикле вычисляется положение центра распознаваемой области и сравнивается с полученным ранее значением. Вычисляется отклонение по осям X и Y, которые преобразуются в величины отклонений приводов вращения и качения.

```
19:     IF ( Max != 0 )
20:     {
21:         Print = ID[CamID].ADDR[0(b)]
22:         Print = Max
23:         Print = Cx
24:         Print with Line = Cy
25:         IF ( Cx > 80 )
26:         {
27:             Delta = Cx - 80
28:             Pan = ID[19].Present position - Delta
29:             ID[19].Goal position = Pan
30:         }
31:         ELSE IF ( Cx < 80 )
32:         {
33:             Delta = Cx - 80
34:             Pan = ID[19].Present position - Delta
35:             ID[19].Goal position = Pan
36:         }
37:         IF ( Cy > 60 )
38:         {
39:             Delta = Cy - 60
40:             Tilt = ID[20].Present position - Delta
41:             ID[20].Goal position = Tilt
42:         }
43:         ELSE IF ( Cy < 60 )
44:         {
45:             Delta = Cy - 60
46:             Tilt = ID[20].Present position - Delta
47:             ID[20].Goal position = Tilt
48:         }
49:     }
```

Данный пример иллюстрирует один из простейших вариантов работы с системой технического зрения, входящей в образовательный робототехнический модуль. В случае усложнения решаемых задач, данная программа может претерпевать различные изменения. Например, с помощью камеры можно распознавать как рабочий элемент, так и элементы окружающей среды, например можно разработать модель робота футболиста, обнаруживающего мяч и наводящегося на ворота и др.

Универсальный модуль камеры является уникальным устройством, существенно улучшающим функциональные возможности роботов на базе конструкторов Bioloid.

ТЕХНОЛАБ ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Беспроводное управление роботами с помощью ZigBee

ЭКЗАМЕН ТЕХНОЛАБ

Беспроводное управление роботами с помощью **ZigBee**

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА ЭКЗАМЕН-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MECHATRONIKA.RU

Беспроводное управление роботами с помощью ZigBee

Беспроводное управление по радиоканалу – наиболее часто встречающееся решение в робототехнике. Для гарантированной передачи данных между устройствами применяются различные протоколы передачи информации. В случае если совместно работающих устройств достаточно много, их объединяют в беспроводные сети и используют сетевые протоколы передачи данных. ZigBee – стандарт высокоуровневых протоколов беспроводной связи, применяемый массово при автоматизации промышленного оборудования, в системах автоматизации зданий и жилых помещений, в медицинском и телекоммуникационном оборудовании. Отличительная особенность стандарта ZigBee заключается в его высокой помехозащищенности, низком энергопотреблении и отсутствии необходимости получения частотного разрешения. Благодаря стандартизации и открытой спецификации различные производители электронных устройств могут разрабатывать собственные устройства, совместимые с устройствами, использующими протокол ZigBee.

На сегодняшний день радиомодули ZigBee используются в системах с невысокой скоростью передачи данных на небольших расстояниях, но требующих гарантированной безопасности канала связи и точности доставки информации при крайне низком энергопотреблении. Низкое энергопотребление обусловлено функцией «спящий режим» во время простоев, но в отличие от модулей Bluetooth, время пробуждения радиомодулей ZigBee в разы меньше. Поэтому реакция устройств на передаваемые сигналы намного быстрее.

Для управления моделями роботов с помощью радиоканала на базе протокола ZigBee в робототехнических конструкторах ROBOTIS применяются специальные модули: ZIG- 100/110A, ZIG2Serial, USB2Dynamixel.

Радиомодуль ZIG-110A предназначен для последовательной передачи данных с помощью ZigBee интерфейса от внешнего устройства к контроллеру робота и обратно. Данный модуль полностью совместим с программируемыми контроллерами CM-100 для наборов на базе конструкторов OLLO и CM-530 для наборов на базе конструкторов Bioloid.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

Для того чтобы компьютер и робот могли осуществлять взаимный обмен информацией используются устройства ZIG2Serial и USB2Dynamixel.

Модуль ZIG-100 представляет собой радиомодуль ZigBee, предназначенный для встраивания в различные устройства, например ZIG-100 может быть встроен в пульт ДУ RC-100, а также использоваться для беспроводного управления роботами с помощью ПК.

Устройство USB2Dynamixel – это универсальный преобразователь интерфейсов USB персонального компьютера в последовательный интерфейс, применяемый в сервоприводах, контроллерах и беспроводных модулях ROBOTIS.

Настройка и установка соединения по интерфейсу ZigBee

Для настройки и установки соединения между компьютером и роботом необходимо подключить к контроллеру и компьютеру модули ZIG-100 и ZIG-110A, а после их настроить.

Для этого необходимо выполнить следующие шаги:

1. Подключить модуль ZIG-110A к порту Communication Jack на контроллере CM-530 и к порту программирования для контроллера CM-100.

2. Подключить контроллер робота к компьютеру с помощью USB-кабеля.

3. Включить питание контроллера.

4. Запустить RoboPlus и выбрать программу для настройки оборудования RoboPlus Manager.

5. Выбрать порт, к которому подключен контроллер (по умолчанию COM3) и нажать Connect для подключения к контроллеру.

6. Во вкладке Controller должны появиться все подключенные к контроллеру устройства. Каждое устройство, с которым работает контроллер, обладает ID номером. Модуль ZIG-110A распознается контроллером как устройство «My Remote ID» с ID 36. Модуль ZIG-100 распознается контроллером как устройство с именем «Remote ID» и идентификатором ID 34.

Примечание: в случае применения контроллера CM-100 ID номера явным образом не доступны пользователю, поскольку к данному контроллеру подключается ограниченное количество устройств с заранее известными идентификационными номерами. Если же используется контроллер CM-530 необходимо указывать ID номера подключаемых к нему устройств.

7. Каждое из устройств имеет собственное значение Value. В рассматриваемом случае модуль с ID 34 имеет значение 63560, а значение модуля с ID 36 равно 63561.

8. Для того чтобы радиомодули с разными ID работали в паре, необходимо, чтобы значение внешнего модуля ZIG-100 равнялось значению модуля ZIG-110A, подключенного к контроллеру. Если выявлено несовпадение, то необходимо в графу Value внести требуемое значение и нажать кнопку Apply.

9. Подключение модуля ZIG2Serial к компьютеру осуществляется с помощью устройства USB2Dynamixel. Для совместной работы двух устройств необходимо установить ZIG2Serial в разъем COM-порта USB2Dynamixel и перевести переключатель выбора режимов в положение соответствующее RS232.

10. Подключите модуль USB2Dynamixel в USB-порт компьютера. Дождитесь того, чтобы компьютер распознал подключенное внешнее устройство.

11. Запустите RoboPlus и RoboPlus Manager.

12. Выберите номер порта, к которому подключен USB2Dynamixel (по умолчанию COM3) и нажмите на кнопку Zig2Serial Management.

13. Войдите в меню ZigBee Setting (удерживайте кнопку в нажатом состоянии в течение 3 секунд). В появившемся меню необходимо ввести значение Value, соответствующее значению My Remote ID (в рассматриваемом примере значение соответствующее ZIG-110A равно 63561).

14. Проверьте соединение радиомодулей. Включите устройства, если настройки были произведены верно, то светодиоды модулей Zig2Serial и ZIG-110A, которые мигали, начнут гореть ровным светом.

15. Проверьте соединение радиомодулей. Запустите виртуальный пульт управления RC-100 (данная функция доступна для контроллеров CM-530).

16. Переведите контроллер CM-530 в режим PLAY с помощью кнопки MODE и нажмите на кнопку START.

17. С помощью открывшегося виртуального пульта управления можно путем нажатия кнопок U, L, R и D управлять движением робота. Для этого необходимо, чтобы в контроллер робота была записана одна из тестовых программ, которую можно скачать с официального сайта ROBOTIS или скачать из Help среды разработки RoboPlus.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА ЭКАМЕН-TECHNOLAB.RU

Управление роботами с помощью программной среды LabView

ЭКЗАМЕН ТЕХНОЛАБ

Управление роботами с помощью программной среды **LabView**

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXCHEMTECHNOLAB.RU

Управление роботами с помощью программной среды LabView

Дистанционное управление роботами или даже производственными механизмами довольно часто встречается в наше время. Несмотря на то что автономные системы управления все больше и больше занимают нишу управления робототехническими системами, системы ручного управления все еще играют значимую роль.

Дистанционное управление робототехническими системами в зависимости от поставленной задачи может сводиться к кардинально разным по реализации системам. Самое часто встречающееся в мобильной робототехнике решение – применение пультов дистанционного управления, в этом случае человек-оператор вручную с помощью пульта или джойстика управляет движением робота. В этом случае перемещение робота контролируется также вручную: оператор следует за роботом на некотором расстоянии или же оператор дистанционно наблюдает за окружающим пространством вокруг с помощью видеокамер, установленных на борту робота.

Также помимо ручного управления человеку-оператору обычно предоставляют функции наблюдения за ходом функционирования робототехнической системы. Такие решения наиболее часто встречаются в промышленности, в этом случае на рабочее место оператора поступает информация о состоянии технологического оборудования, информация и показания от датчиков, количество выпущенной продукции и многое другое. В подобных системах человеку-оператору очень редко предоставляется возможность управлять каким-либо механизмом, и все его функции сводятся к наблюдению за работой и остановке системы управления в аварийной ситуации.

В некоторых случаях системы управления роботом или робототехнической системой реализуют на удаленном высокопроизводительном сервере, который анализирует показания бортовых систем робота, производит необходимые расчеты и передает роботу управляющий сигнал, по которому тот осуществляют свою дальнейшую работу.

Примером последней системы может быть лабораторный стенд для управления роботами на базе программной среды LabView. С помощью программной среды LabView на удаленном сервере или обычном компьютере можно создать программу управления роботом, которая возьмет на себя самые ресурсоемкие функции, например анализ показаний датчиков, планирование маршрута, распознавание графической и видеоинформации. Благодаря этому можно существенно упростить бортовую систему управления роботом, оставив за ней простейшие функции реагирования на поступающие от удаленного компьютера команды.

Такой подход позволяет создавать сложные системы управления для достаточно простых роботов, разрабатываемых даже на базе робототехнических конструкторов. Данное пособие содержит рекомендации по основам разработки систем управления роботами с использованием программной среды LabView. В качестве модели для отработки алгоритмов управления предлагается использовать роботов из образовательных модулей «Профессиональный уровень» или «Исследовательский уровень». Этих роботов объединяет общее свойство, они принадлежат к одному модельному ряду Bioloid корейской компании Robotis, поэтому в их состав входят одинаковые аппаратные средства: программируемый контроллер CM-530, сервопривода Dynamixel, ИК-датчики и многое другое.

Благодаря применению программной среды LabView для роботов серии Bioloid можно разработать сложные системы управления, использующие все ресурсы удаленного компьютера. Связь робота с управляющим компьютером осуществляется с помощью беспроводного канала ZigBee. Применение беспроводного интерфейса дает возможность разрабатывать роботов, не привязанных к пульту управления с ИК-каналом, ограничивающим их мобильность. Роботы с беспроводным радио интерфейсом могут работать на удаленном расстоянии от управляющей станции, в других помещениях и при наличии каких-либо преград между ними. Таким образом, совместное использование возможностей робототехнических наборов и программной среды LabView открывают перед разработчиками новые возможности по проектированию и эксплуатации роботов в образовательном и соревновательном процессе.

Примечание: предлагаемый цикл работ идентичен циклу, входящему в программу образовательный робототехнический модуль «Профессиональный уровень».

Управление базовым робототехническим набором

ЭКЗАМЕН ТЕХНОЛАБ

Управление базовым робототехническим набором с помощью программной среды LabView

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.TECHNOLAB.RU

Управление базовым робототехническим набором с помощью программной среды LabView

Робототехнический модуль обладает достаточно широким функционалом – на базе данного руководства можно проводить лабораторные работы по изучению основ проектирования роботов и разработки для них систем управления. Роботы, разрабатываемые из образовательного робототехнического модуля, могут применяться в робототехнических соревнованиях и исследовательской деятельности.

Одной из основ исследовательской деятельности является проведение экспериментов, сбор и анализ данных. Процесс исследовательской деятельности неотъемлем от процесса проектирования и разработки каких-либо новых и инновационных решений, поэтому он крайне важен. Программная среда LabView является одним из универсальных инструментов автоматизации исследовательской деятельности и лабораторных экспериментов. С помощью LabView можно анализировать результаты измерений сенсорных устройств, производить сложные преобразования и вычисления. Поэтому знакомство с программной средой LabView является важным этапом процесса изучения робототехники.

Для ознакомления с процессом применения LabView для управления роботами предлагается воспользоваться специальной базовой программой, представляющей собой дистанционный пульт управления человека-оператора мобильного робота. Данная программа позволяет управлять мобильным роботом на базе робототехнического набора аналогичным образом как с помощью ИК-пульта управления RC-100, производить сбор данных и многое другое. В данном разделе мы кратко остановимся на описании данной программы и возможностей по ее применению.

Базовая программа представляет собой проект в программной среде LabView и располагается на диске для преподавателя в разделе «LabView/Базовая программа». Данная программа может применяться в качестве эмулятора ИК-пульта для всех роботов серии Bioloid.

Требования:

1) Операционная система ПК Windows XP/Vista/7 (32/64bit)

2) Наличие установленной программной среды LabView (версия не позднее 9.0)

3) Наличие установленного программного пакета Microsoft Visual C++ 2005 re-distribution package

4) Наличие установленной программной среды RoboPlus

5) Наличие интегрированных в программную среду LabView библиотек для управления оборудованием Dynamixel SDK и ZigBee SDK

6) Наличие установленных драйверов LabView NI VISA (версия не позднее 5.3)

7) Наличие устройств USB2Dynamixel, Zig2Serial, комплекта ZigBee для контроллера CM-530.

Описание базовой программы

Данная программа представляет собой библиотеку виртуальных инструментов на базе LabView 2012, адаптированную для работы с внешним устройством, в нашем случае роботом серии Bioloid. Данная программа разработана для работы с Bioloid STEM, но применима для работы с любыми роботами серии Bioloid.

Для работы с базовой программой необходимо осуществить запуск проекта, в результате чего на экране компьютера появится лицевая панель программы.

Лицевая панель содержит ряд элементов управления, предназначенных для перехода к различным режимам работы программы.

«Перейти к работе с набором» – меню, переводящее к средствам управления роботом.

«Руководство пользователя» – меню, вызывающее руководство по работе с базовой программой.

«Описание набора» – меню, вызывающее руководство по работе с базовым робототехническим набором.

«Выход» – завершение работы программы.

Работа с базовой программой начинается с перехода в меню «Перейти к работе с набором». В результате этого пользователю предлагается ознакомиться с различными способами работы с устройствами, подключенными к компьютеру. На выбор предлагается два варианта – управление контроллером CM-530 и сервоприводами Dynamixel посредством USB интерфейса, а также беспроводное управление роботом с помощью ZigBee интерфейса.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

При переходе в меню «Знакомство с USB2Dynamixel и Dynamixel SDK» пользователю предлагается воспользоваться одним из режимов работы – «Проводной режим» для управления сервоприводами с помощью проводного канала связи или «Беспроводной режим» для управления сервоприводами с помощью интерфейса ZigBee.

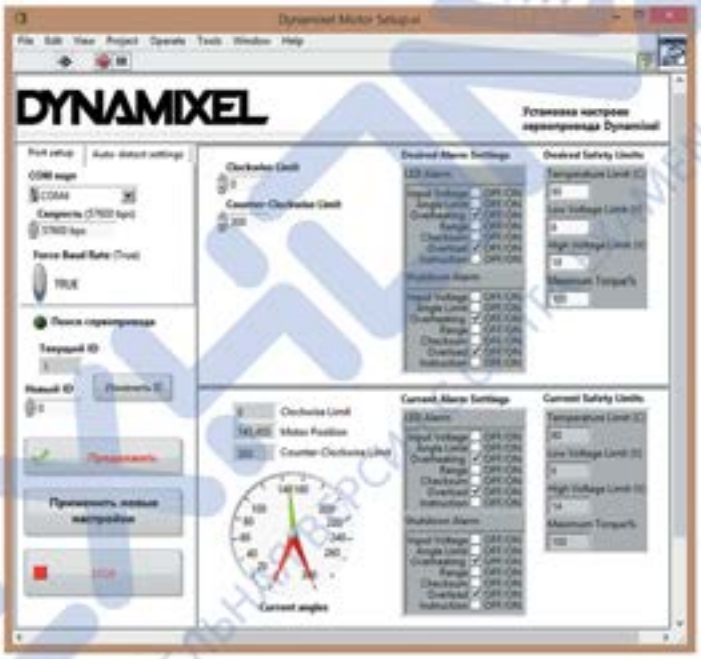
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ - САЙТ ЭКЗАМЕН-ТЕХНОЛАБ.RU

Работа в режиме «Проводной режим»

Для взаимодействия с сервоприводами в проводном режиме необходимо подключить сервопривод (или несколько сервоприводов) к адаптеру USB2Dynamixel, а сам адаптер необходимо перевести в положение TTL. Для подключения адаптера необходимо использовать кабель с ответвлением для питания сервоприводов (+9.6 В) или же питание приводов можно осуществить от контроллера CM-530.

1) Установка настроек сервопривода

Данный виртуальный инструмент позволяет произвести настройку сервопривода перед использованием. В настройку сервопривода входят параметры инициализации (первоначального состояния/положения), а также параметры сигналов и критических состояний.



Порядок работы:

1. Убедиться в правильности подключения сервопривода Dynamixel к адаптеру USB2Dynamixel

2. Выбрать соответствующий COM-порт, указать скорость, а во вкладке Auto-detect settings указать параметры автоматического поиска сервопривода: диапазон ID и время поиска. Для продолжения работы нажать «Продолжить».

3. Дождаться того, что загорится индикатор «Поиск сервопривода». Как только сервопривод будет найден, на индикаторе «Текущий ID» отобразится значение ID сервопривода, что свидетельствует о том, что сервопривод готов к настройке.

4. Установить необходимые настройки, а после нажать на «Применить новые настройки», после чего выбранные настройки будут применены к сервоприводу.

5. Нажатие кнопки STOP осуществляет возврат к окну выбора виртуального элемента (vi).

2) Обмен данными с сервоприводом

С помощью данного виртуального элемента осуществляется обмен данными с сервоприводом по последовательному интерфейсу. Данный виртуальный элемент предназначен для отправки команд сервоприводу и получения ответных сообщений.

<u>Порядок работы:</u>

1. Необходимо убедиться в правильности подключения сервопривода Dynamixel к адаптеру USB2Dynamixel

2. Выбрать соответствующий COM-порт и указать скорость, после чего нажать кнопку «Продолжить».

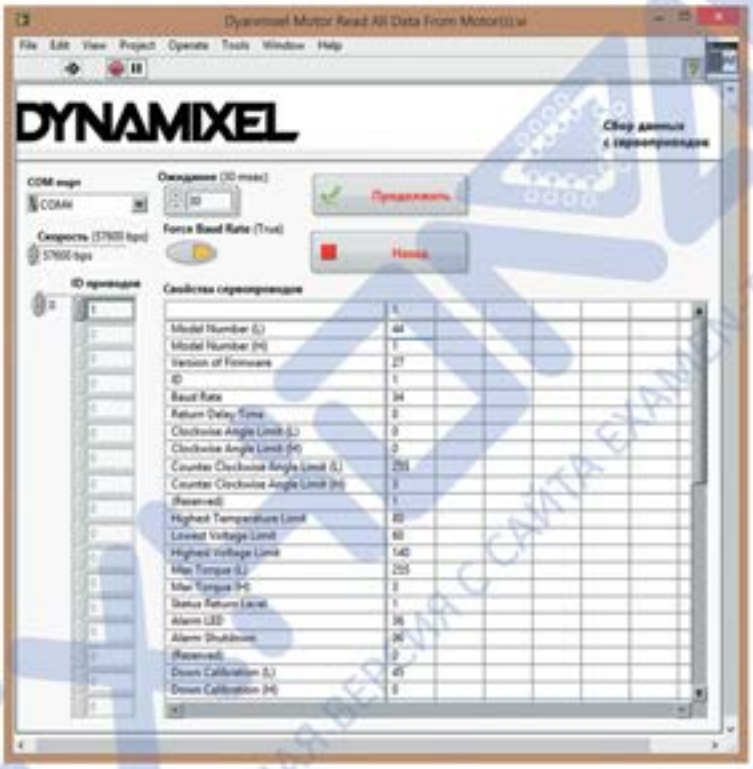
3. Указать ID сервопривода, адрес и выбранное действие нажать «Отправить команду». В окне «Результаты» отобразится результат выполнения заданной команды.

4. В случае завершения работы с данным виртуальным элементом нажать кнопку STOP.

Примечание: подробное описание команд и их адресация содержатся в руководстве пользователя для каждого из типов сервоприводов Dynamixel.

3) Сбор данных от сервоприводов

Данный виртуальный элемент предназначен для запроса данных, описывающих состояние сервопривода и вывода их на экран пользователя. Программа предназначена для работы как с одним сервоприводом, так и с группой сервоприводов.



Порядок работы:

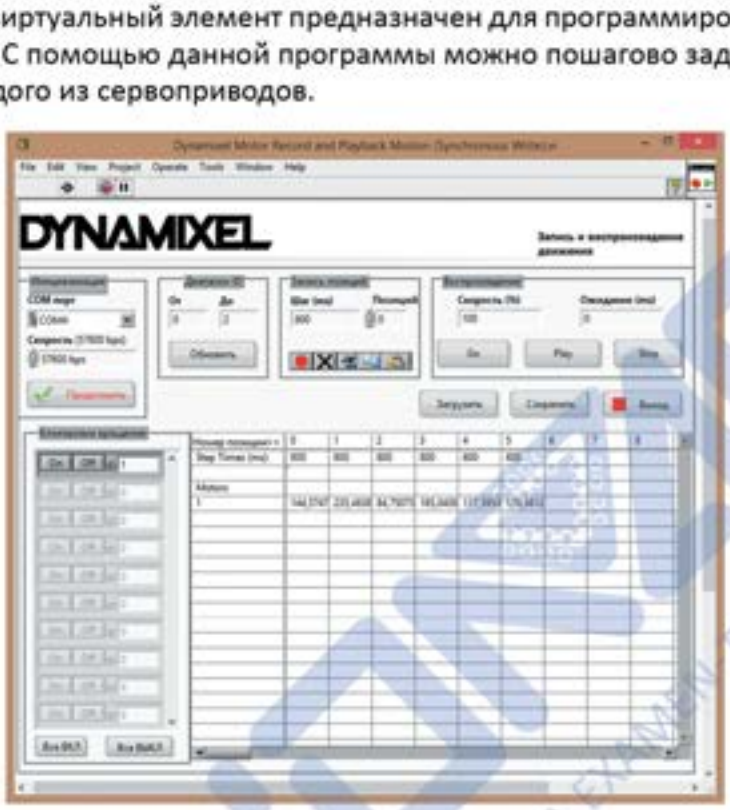
1. Необходимо убедиться в правильности подключения сервопривода Dynamixel к адаптеру USB2Dynamixel

2. Выбрать соответствующий COM-порт и указать скорость, а также в массив ID приводов необходимо ввести ID необходимых сервоприводов. После чего нажать кнопку «Продолжить». В окне «Свойства сервоприводов» отображаются все свойства выбранных сервоприводов и их текущие значения.

3. В случае завершения работы с данным виртуальным элементом нажать кнопку «Назад».

4) Запись и воспроизведение движения

Данный виртуальный элемент предназначен для программирования вращения сервоприводов. С помощью данной программы можно пошагово задать очередность положений каждого из сервоприводов.



Порядок работы:

1. Необходимо убедиться в правильности подключения сервопривода Dynamixel к адаптеру USB2Dynamixel

2. Выбрать соответствующий COM-порт, указать скорость обмена данными, после чего нажать кнопку «Продолжить».

3. Указать диапазон ID для поиска подключенных сервоприводов. Нажать «Обновить».

4. После обнаружения сервопривода его ID отобразится в списке «Блокировка вращения», а также в таблице.

5. С помощью нажатия кнопки «Record» фиксируется текущее положение сервопривода. Изменив положение сервопривода, снова нажмите Record. После чего установите индикатор позиций в положение 0 и нажмите Go, а затем Play. Сервопривод повторит запрограммированные движения.

6. Можно сохранять файлы с записью движения в формате dsq, а затем снова их воспроизводить. Для этого воспользуйтесь кнопками «Загрузить» и «Сохранить».

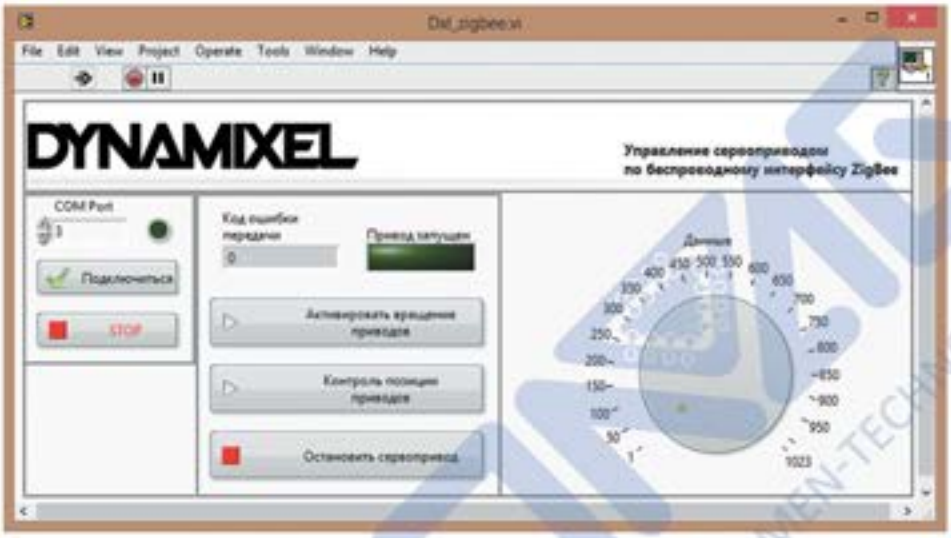
7. В случае завершения работы нажмите кнопку «Выход».

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN.TECHNOLAB.RU

Работа в режиме «Беспроводной режим»

Данный виртуальный элемент применяется для управления сервоприводом с помощью беспроводного интерфейса ZigBee.

Для работы с данной программой необходимо, чтобы в программируемый контроллер CM-530 была установлена программа CM_530_zigbee_dxl.tsk.



Порядок работы с программой:

1. Необходимо установить требуемый режим работы сервоприводов с помощью программной среды RoboPlus.
2. Необходимо включить робота Bioloid, подключить модуль ZIG-100 к компьютеру. Убедиться, что модули ZIG-100 и ZIG-110A установили между собой соединение (красные лампочки на них должны начать гореть ровным светом)
3. Запустить программу, выбрать COM-порт (по умолчанию COM3) и нажать кнопку «Подключиться». В случае успешного подключения к порту загорится лампочка.
4. Выбрать либо режим постоянного вращения сервоприводов или режим контроля положения (в зависимости от режима работы).
5. С помощью устройства управления «Данные» можно изменять значение скорости вращения или положения сервоприводов.
6. В случае завершения работы необходимо нажать кнопку «Остановить сервопривод» и затем Stop.

На этом знакомство с устройством USB2Dynamixel и Dynamixel SDK можно считать законченным. Вернувшись к начальному окну программы, можно перейти к знакомству с Zig2Serial и ZigBee SDK.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

При переходе в меню «Знакомство с ZIG2Serial и ZigBee SDK» пользователю предоставляется возможность дистанционного управления роботом и сбора показаний его датчиков.

Для работы с данной программой необходимо, чтобы в программируемый контроллер CM-530 была установлена программа CM_530_zigbee_main.tsk.

В появившемся окне на выбор предлагаются следующие меню:

«Управление роботом» – программа эмуляции ИК-пульта дистанционного управления RC-100.

«Опрос датчиков» – программа для сбора показаний датчиков, установленных на роботе.

«Выход» – меню выхода из программы.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

При переходе к программе «Опрос датчиков» появляется окно, позволяющее выбрать одну из программ опроса датчиков.

В зависимости от выбранной программы появляется один из виртуальных элементов, отображающих данные, поступающие от различных сенсорных устройств.

Программа «Управление роботом» полностью повторяет ИК-пульт RC-100 и дает возможность управлять роботом так, как будто у пользователя в руках реальный джойстик. Для управления роботом нужно подключить устройства ZigBee к контроллеру CM-530 и персональному компьютеру. Подключение устройства ZigBee к контроллеру CM-530 осуществляется путем включения модуля ZIG100 в разъем Communication Jack программируемого контроллера. Подключение этого же модуля к персональному компьютеру осуществляется с помощью устройств USB2Dynamixel и ZIG2Serial. Необходимо установить модуль ZIG100 в устройство ZIG2Serial и с помощью USB2Dynamixel подключить его к персональному компьютеру. Настройку данного соединения необходимо выполнить в соответствие с рекомендациями раздела «Беспроводное управление роботами с помощью ZigBee» данного учебного пособия.

Для работы с роботом включите контроллер CM-530 и с помощью кнопки MODE переведите его в режим PLAY. Запустите контроллер путем нажатия кнопку START.

На виртуальном пульте управления выставьте значение COM-порта, соответствующее соединению по интерфейсу ZigBee, далее нажмите Connect. В случае удачного подключения устройств друг к другу загорится зеленая лампочка.

С помощью виртуального пульта управления можно управлять роботом в ручном режиме. Управление возможно с помощью компьютерной мышки, в этом случае пользователь должен нажимать на соответствующие кнопки виртуального пульта. Также возможно управление с помощью клавиатуры, когда пользователь управляет направлением движения робота с помощью нажатий клавиш на клавиатуре персонального компьютера.

Для завершения работы с набором необходимо нажать на кнопку STOP, выключить программируемый контроллер CM-530 и, в случае необходимости отключить модули беспроводного интерфейса ZigBee.

Данный раздел демонстрирует возможности применения программной среды LabView для управления роботами. С помощью базовой программы пользователь может наглядно ознакомиться с функционалом программной среды LabView, оценить ее возможности и, быть может, придумать собственное решение одной из робототехнических задач. Для дальнейшего освоения технологии разработки систем управления роботами пользователю предлагается ознакомиться с руководством по разработке базовых программ управления в программной среде LabView.

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EKAMEN-TECHNOLAB.RU

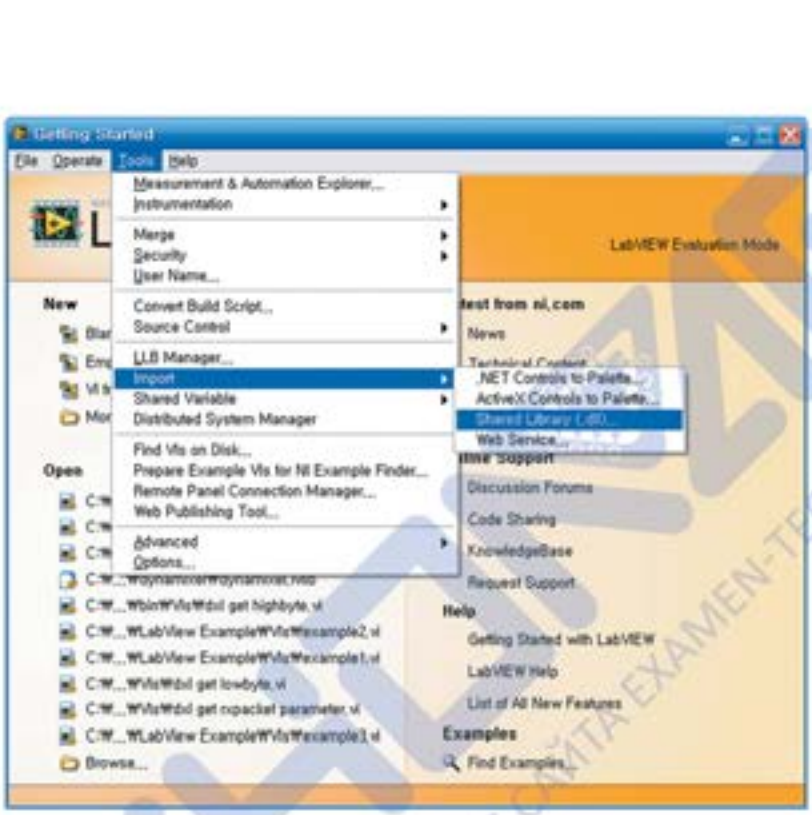
Разработка систем управления роботами и устройствами

ЭКЗАМЕН ТЕХНОЛАБ

Разработка систем управления роботами и устройствами с помощью **LabView**

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.ЛАБОРАТОРИЯ-ТЕХНОЛАБ.РФ

Разработка систем управления роботами и устройствами с помощью LabView



Применение различных подходов к решению поставленных задач, а также использования различных инструментов для их реализации, является отличительной особенностью профессионального подхода к проектной и исследовательской деятельности. Использование программной среды LabView позволяет расширить горизонты применения образовательных робототехнических модулей в процессе обучения учащихся робототехнике.

Данный раздел предназначен для демонстрации основ применения LabView при управлении роботами серии Bioloid. В данном разделе описывается процесс разработки программ управления в среде LabView и программ для управляющего контроллера CM-530. Программированию контроллера CM-530 уделяется отдельное внимание, поскольку в данном случае его программы качественно отличаются от рассматриваемых ранее. Теперь на программу контроллера CM-530 возлагается задача приема команд управления от удаленного компьютера и передачи ему требуемой информации, когда как все алгоритмические задачи решаются в программной среде LabView.

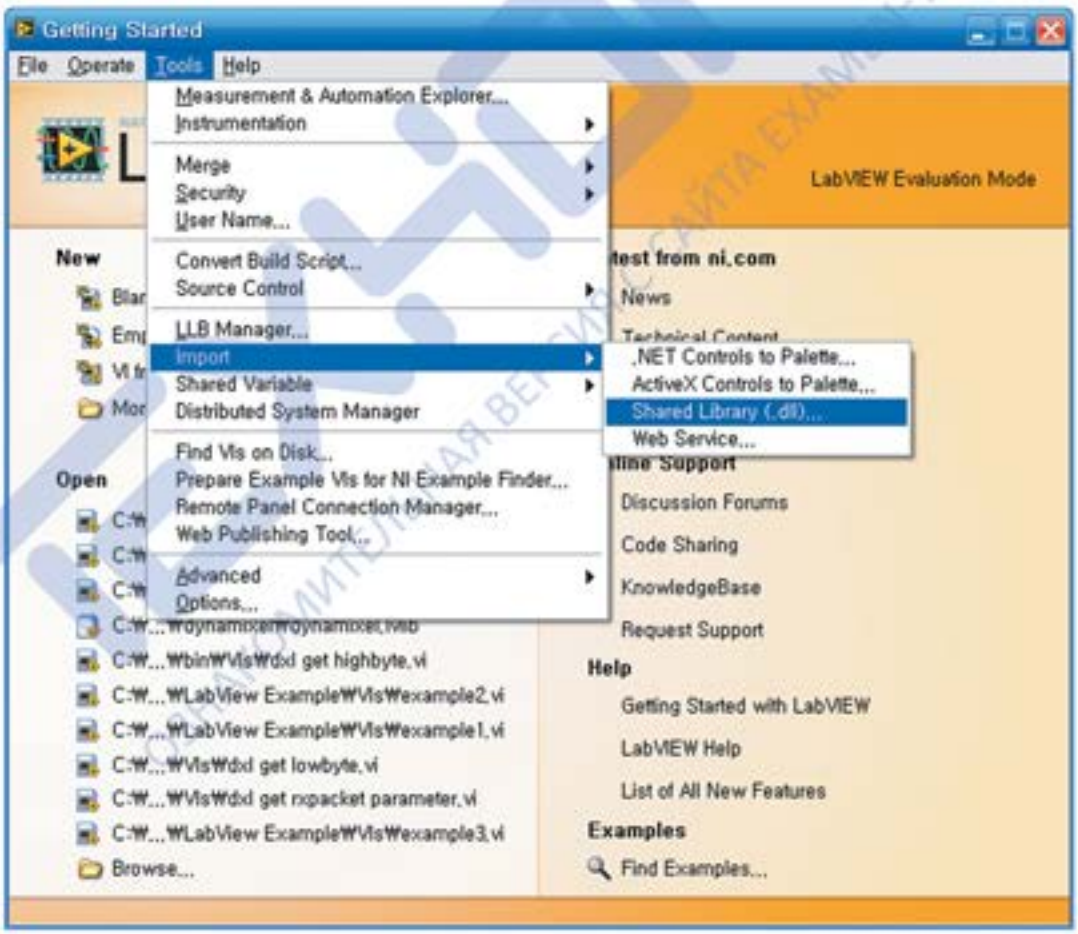
Описание библиотеки Dynamixel SDK

Библиотека Dynamixel SDK представляет собой стандартную библиотеку для работы с различными сервоприводами серии Dynamixel. В данной библиотеке реализован инструментарий для работы с сервоприводами, используя который пользователь может разрабатывать собственные системы управления.

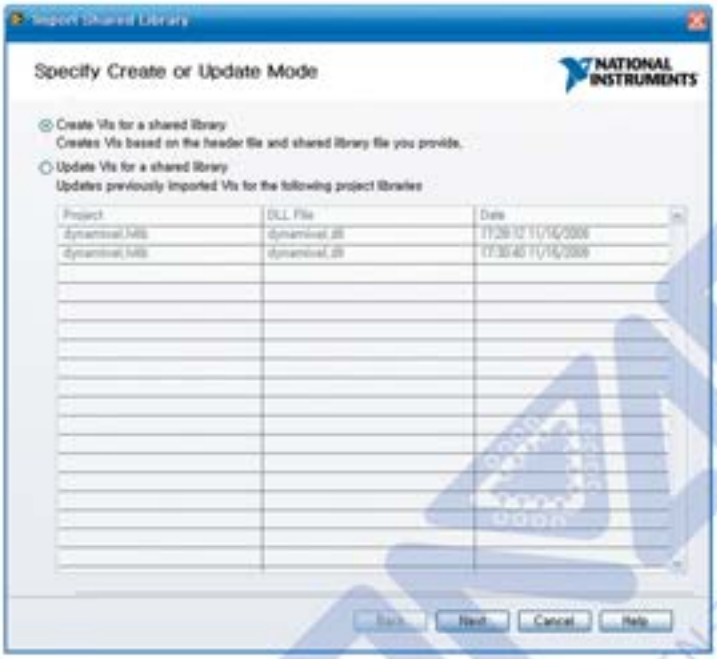
Импортирование библиотеки в LabView

Для того чтобы использовать стандартные функции управления сервоприводами Dynamixel, необходимо импортировать библиотеку Dynamixel SDK в программную среду LabView. Данную библиотеку можно скачать с сайта корейского производителя ROBOTIS, также она расположена в каталоге LabView диска для преподавателя.

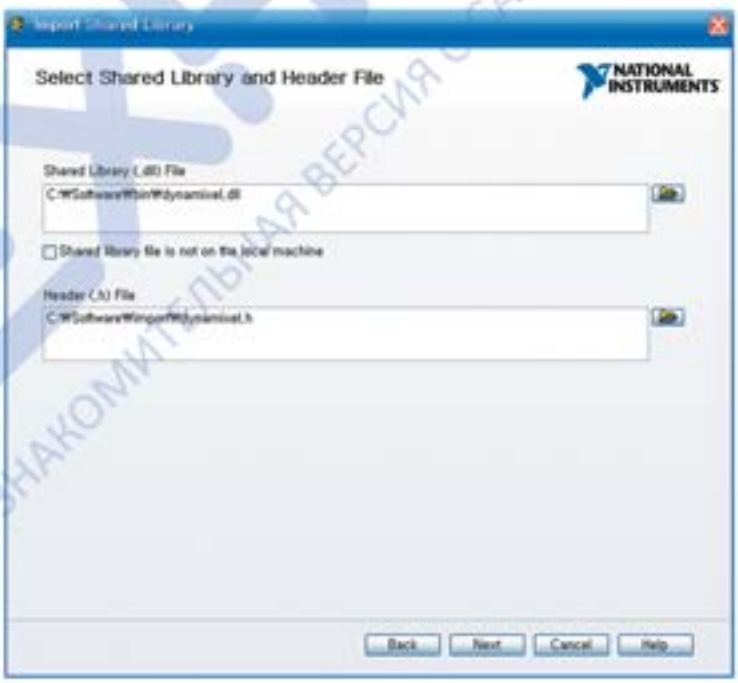
1) Выберите Tools – Import – Shared Library (.dll) в стартовом окне LabView.



2) В появившемся окне выберите Create VIs for a shared library и нажмите кнопку Next.

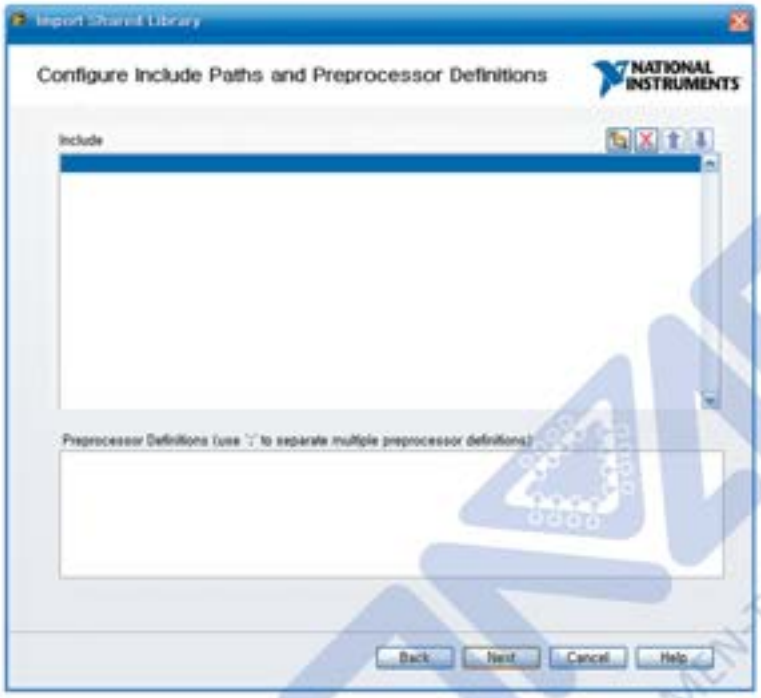


3) Выберите dynamixel.dll и dynamixel.h в каталоге Dynamixel SDK и нажмите Next.

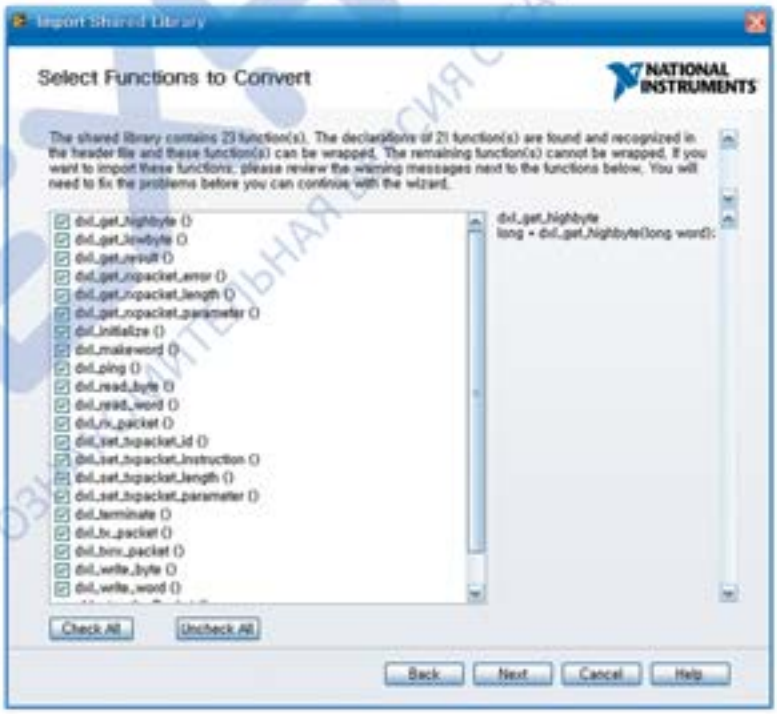


ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

4) Появившееся окно оставить без изменений и для перехода далее нажать Next.

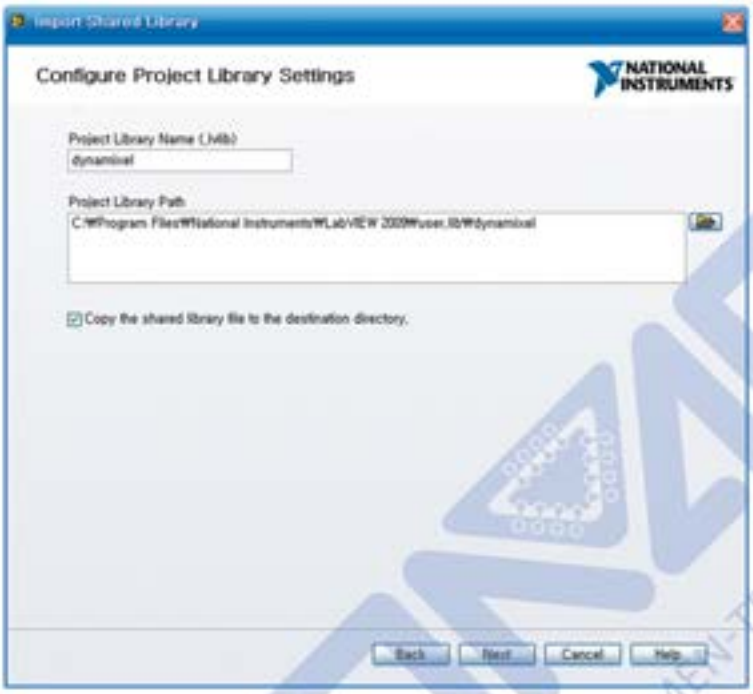


5) В появившемся окне проверить все ли выбраны функции. Нажать Next для перехода к следующему этапу.

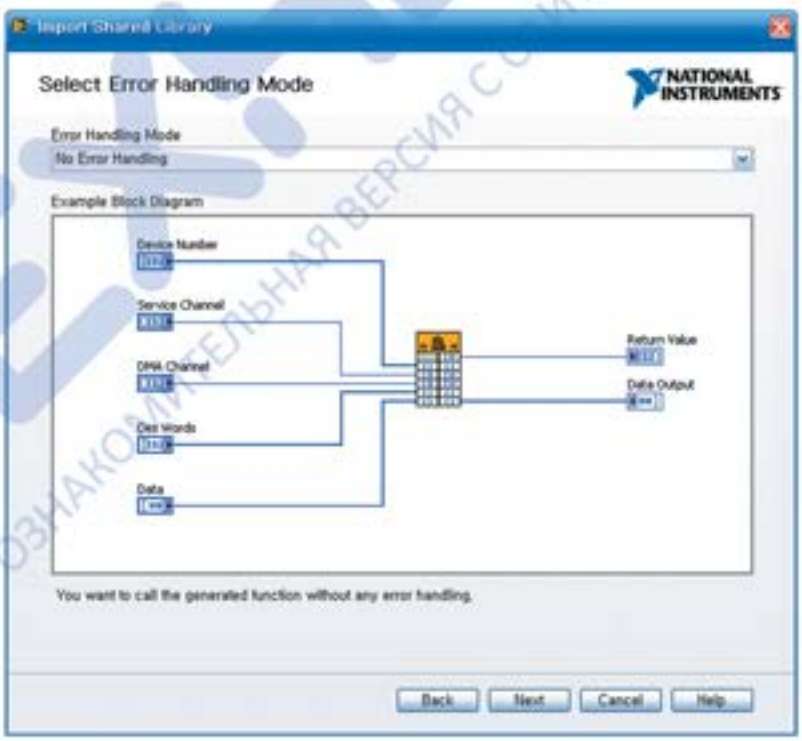


ОБУЧАЮЩАЯ ПЛАТФОРМА С САЙТА EXAMEN-TECHNOLAB.RU

6) Ничего не изменяя, перейти к следующему меню, нажав Next.

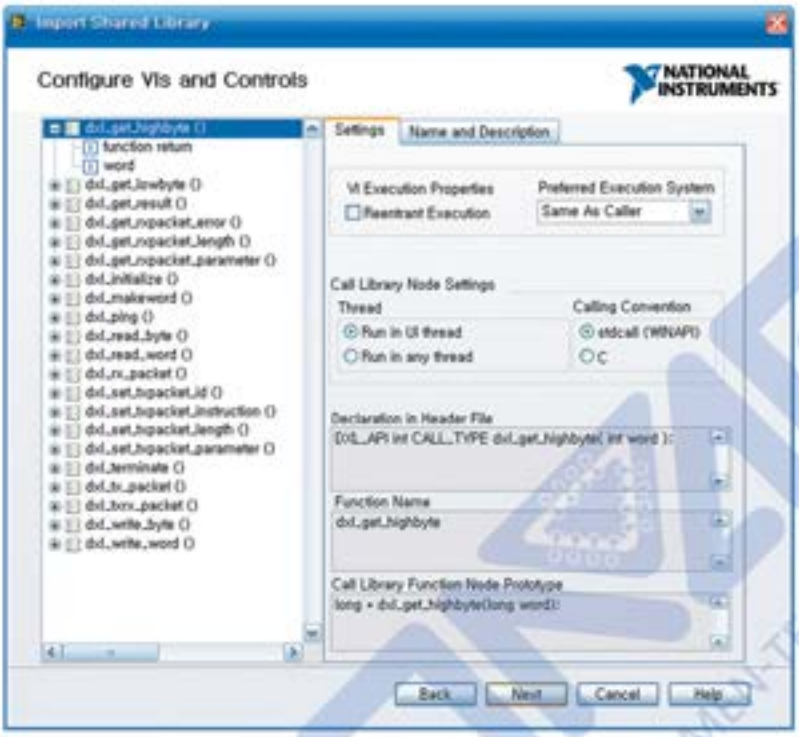


7) Ничего не изменяя, перейти к следующему меню, нажав Next.

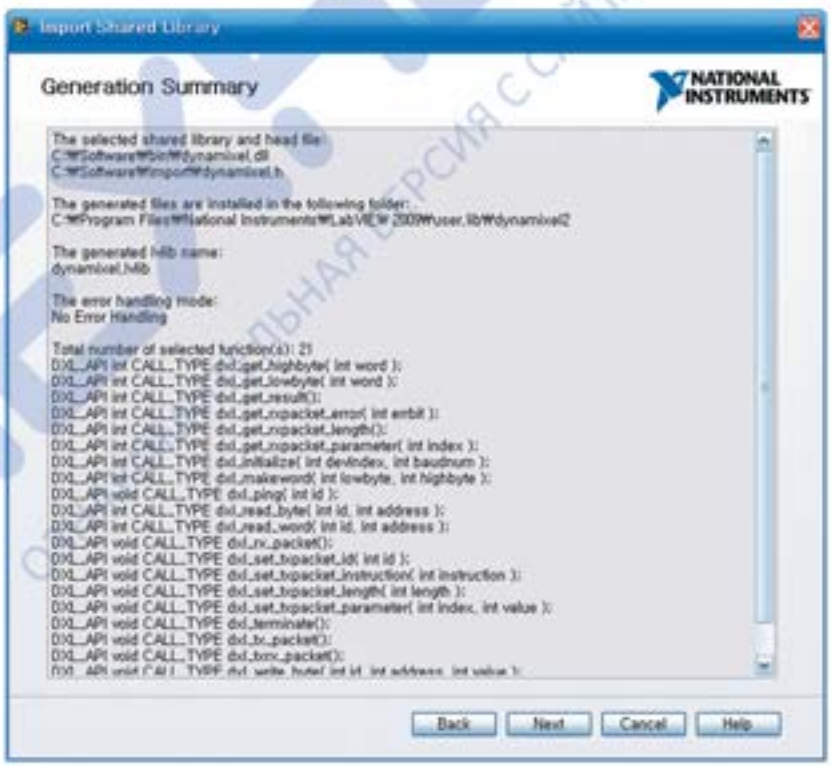


ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

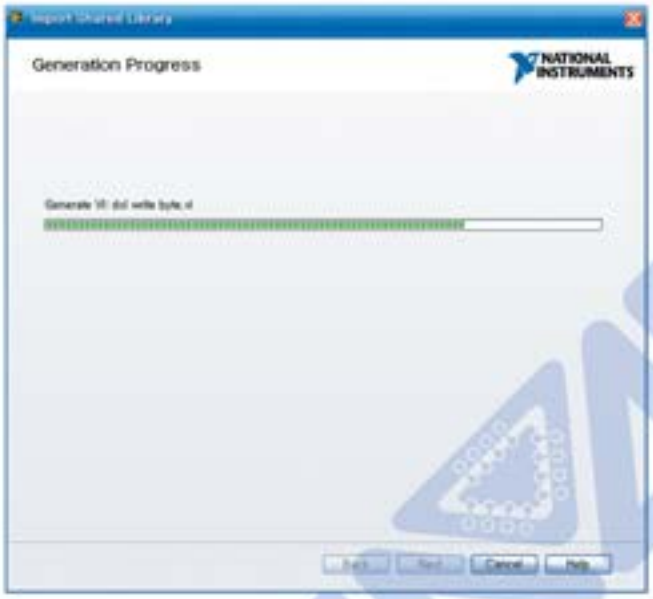
8) Ничего не изменяя, перейти к следующему меню, нажав Next.



9) Ничего не изменяя, перейти к следующему меню, нажав Next.



10) Дождаться завершения процесса создания виртуальных инструментов (vi). По завершению процесса перейти к следующему меню, нажав Next.



По завершении процесса создания виртуальных элементов появляется палитра виртуальных инструментов в разделе User Libraries.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Описание функций палитры виртуальных инструментов Dynamixel

Функция	Описание	Параметры входные	Параметры выходные
dxl_initialize	Инициализация и установка связи с подключенными устройствами	**devIndex** – число подключенных в настоящее время устройств **baudnum – значение скорости в бодах (см. Таблица 2).**	1 – ОК 0 – Ошибка
dxl_terminate	Отключение сеанса связи с устройствами	-	-
dxl_set_txpacket_id	ID сервопривода	ID – идентиф. номер сервопривода	-
dxl_set_txpacket_instruction	Входные командные коды в пакет инструкций	**instruction – одна из команд** (см. Таблица 3)	-
dxl_set_txpacket_instruction	Вводимый параметр в пакет инструкций	**index – номер параметра** **value – значение параметра в диапазоне** 0-255	-
dxl_set_txpacket_length	Установка длины пакета инструкций	**length – длина пакета инструкций**	-
dxl_get_rxpacket_length	Проверка длины пакета статуса	-	Значение длины пакета статуса
dxl_get_rxpacket_parameter	Проверка параметра пакета статуса	**index – номер параметра для проверки**	Значение заданного параметра
dxl_get_rxpacket_error	Фиксирует ошибки, включенные в пакет статуса	**errbit** – флаг бита для фиксации ошибки (см. Таблица 4)	1 – есть ошибка 0 – нет ошибки
dxl_tx_packet	Передача пакета инструкций на Dynamixel	-	-
dxl_rx_packet	Получение пакета статуса из буфера драйвера. После вызова результат должен быть проверен с помощью dxl_get_result	-	-
dxl_txrx_packet	Одновременное выполнение функций dxl_tx_packet dxl_tx_packet	-	-
dxl_get_result	Выдает результат проверки пакетов	-	(см. Таблица 5)
dxl_ping	Проверка Dynamixel с заданным ID. Результат проверяется функцией dxl_get_result	**id** – ID **сервопривода для проверки**	-

dxl_write_byte	Запись информационного байта в Dynamixel. Результат может быть получен с помощью dxl_get_result	**id** – ID Dynamixel **для записи информации** **address** – значение адреса **value** – записываемое значение	-
dxl_write_word	Запись двух информационных байтов в Dynamixel. Результат может быть получен с помощью dxl_get_result	**id** – ID Dynamixel **для записи информации** -**address** – значение адреса -**value** – записываемое значение	-
dxl_read_byte	Считывание одного информационного байта в Dynamixel. Результат может быть получен с помощью dxl_get_result	**id** – ID Dynamixel **для записи информации** **address** – значение адреса	-
dxl_read_word	Считывание двух информационных байтов в Dynamixel. Результат может быть получен с помощью dxl_get_result	**id** – ID Dynamixel **для записи информации** **address** – значение адреса	-
dxl_makeword	Преобразует двухбитный тип данных в тип WORD	**Lowbyte** – младший байт **Highbyte** – старший байт	Слово данных в формате WORD
dxl_get_highbyte	Извлечение старшего байта из переменной типа WORD	**word – данные для извлечения старшего бита**	Возвращает старший байт
dxl_get_lowbyte	Извлечение младшего байта из переменной типа WORD	**word – данные для извлечения младшего бита**	Возвращает младший байт

Вышеуказанная таблица палитры виртуальных инструментов демонстрирует собой реализацию протокола управления сервоприводами Dynamixel.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN.TECHNOLAB.RU

Таблица 2. Соответствие передаваемых значений и скоростей

Address	Set BPS	Goal BPS	Error
1	1000000.0	1000000.0	0.000 %
3	500000.0	500000.0	0.000 %
4	400000.0	400000.0	0.000 %
7	250000.0	250000.0	0.000 %
9	200000.0	200000.0	0.000 %
16	117647.1	115200.0	-2.124 %
34	57142.9	57600.0	0.794 %
103	19230.8	19200.0	-0.160 %
207	9615.4	9600.0	-0.160 %

Вышеуказанная таблица задает соответствие скоростей передачи данных по последовательному интерфейсу, задаваемых значений и значений погрешности.

Таблица 3. Командные коды

№	Имя	Содержание
1	INST_PING	Не выполняется. Используется, когда контроллер готов к передаче статусного пакета
2	INST_READ	Команда чтения данных из Dynamixel
3	INST_WRITE	Команда записи данных в Dynamixel
4	INST_REG_WRITE	Команда, идентичная команде WRITE_DATA, но для функционирования необходимо получение команды ACTION.
5	INST_ACTION	Команда активации работы
6	INST_RESET	Команда возврата Dynamixel в начальное состояние, идентичное заводскому
131	INST_SYNC_WRITE	Команда используется для одновременного контроля нескольких Dynamixel.

В данной таблице задается перечень командных кодов пакетов инструкций. С помощью данных кодов определяется назначение пакета.

Таблица 4. Описание ошибок и соответствующие им значения команд

№	Имя	Содержание
1	ERRBIT_VOLTAGE	If the Error-impressed voltage is deviated from the set movement voltage range in the control table, it is set to 1.
2	ERRBIT_ANGLE	If the writing of Goal Position value is deviated from the range between CW Angle Limit ~ CCW Angle Limit, it is set to 1.
4	ERRBIT_ OVERHEAT	If the internal temperature of Dynamixel is deviated from the set movement temperature range in the control table, it is set to 1.
8	ERRBIT_RANGE	When the command is deviated from the using range, it is set to 1.
16	ERRBIT_ CHECKSUM	When Checksum of the transmitted Instruction Packet is not matching, it is set to 1.
32	ERRBIT_ OVERLOAD	When the set torque cannot control the current load, it is set to 1.
64	ERRBIT_ INSTRUCTION	If an undefined instruction is transmitted, or action command is transmitted without reg_write command, it is set to 1.

Данная таблица содержит перечень сообщений об ошибках в процессе управления сервоприводами.

Таблица 5. Расшифровка результатов проверки

Значение	Сообщение
COMM_TXSUCCESS	Success of Instruction packet Transmission
COMM_RXSUCCESS	Success of Status Packet Reception
COMM_TXFAIL	Failure of Instruction Packet Transmission by Error
COMM_RXFAIL	Failure of Status Packet Reception by Error
COMM_TXERROR	Problems in Instruction packet
COMM_RXWAITING	Status packet is not arrived yet.
COMM_RXTIMEOUT	Relevant Dynamixel is not responding.
COMM_RXCORRUPT	Problems in Status packet

Данная таблица содержит перечень сообщений об ошибках в процессе управления сервоприводами.

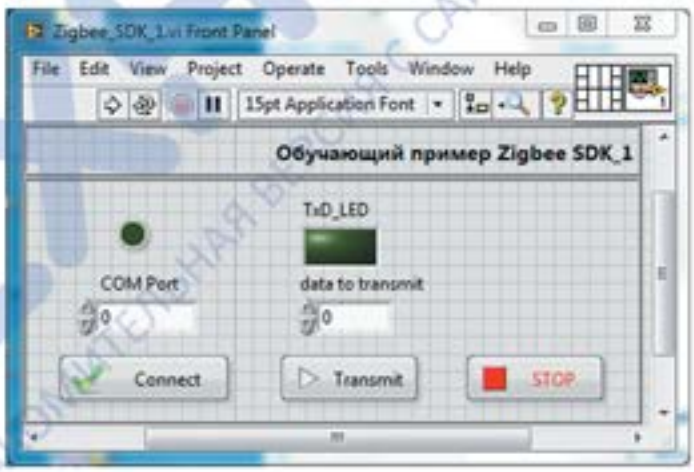
ОБНАРУЖЕНА ОШИБКА В ЗАГРУЗКЕ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
© ЭЛЕКТРОННЫЙ ВАРИАНТ С САЙТА ЕКЛИМЕНКО-ТЕХНОЛАБ.РУ

Разработка систем управления роботами и устройствами

ЭКЗАМЕН ТЕХНОЛАБ

Программирование и управление роботами серии **Bioloid** с помощью базовых средств **RoboPlus** и программной среды **LabView**



ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА WWW.MEGATEHNOLAB.RU

Программирование и управление роботами серии Bioloid с помощью базовых средств RoboPlus и программной среды LabView

Данный раздел содержит ряд примеров, направленных на наглядную демонстрацию процесса разработки программ управления роботами серии Bioloid в базовой для данных наборов среде программирования RoboPlus. Для демонстрации возможностей программной среды LabView, наряду с программированием роботов, разрабатывается программа управления для персонального компьютера, предназначенная для дистанционного управления роботами и отображения на экране компьютера рабочей информации.

Информация в данном разделе предназначена для предварительного ознакомления пользователями, в совершенстве освоившими процесс разработки роботов на базе оборудования из робототехнических наборов серии Bioloid, знакомых с программными средами RoboPlus и LabView.

Часть 1. Беспроводная передача данных от персонального компьютера к роботу с помощью интерфейса ZigBee

Основная задача данного примера – иллюстрация процесса передачи данных между персональным компьютером, с установленной программной средой LabView и программируемым контроллером CM-530.

Для работы с данным примером необходимо наличие установленной на персональный компьютер программной среды LabView и установленной библиотеки ZigBee SDK. Данная библиотека должна располагаться в разделе functions – User libraries – ZigBee – Vis.

Программа контроллера CM-530 должна быть настроена на прием данных по последовательному интерфейсу. В качестве передаваемых данных в примере используется набор команд, передаваемых от пульта управления.

Разработка программы в среде программирования LabView

Принцип работы данной программы заключается в установлении соединения по беспроводному последовательному интерфейсу между персональным компьютером и роботом, подключении и отключении COM-портов, инициализации оборудования в соответствии с выбранным портом и передачи текстовых кодов команд.

1) Создать новую блок-диаграмму в рабочем пространстве LabView. Поместить в нее две структуры Case Structure, Round LED – индикатор с именем ZGB_Connected, Square LED – индикатор с именем TxD_LED.

2) Создать кнопку OK Button с названием ZGB_Connect и соединить ее со входом Case Selector первой структуры. Установить настройку этой кнопки Mechanical Action в позицию Switch Until Released. Поместить внутрь первой структуры в окне True структуру Flat Sequence с тремя окнами (Frame).

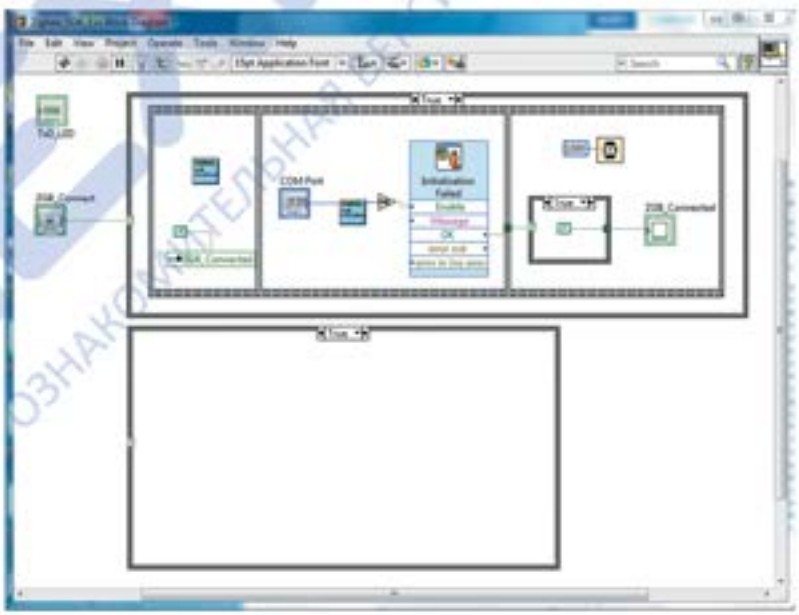
3) Поместить внутрь первого окна функцию zgb_terminate.vi из библиотеки Zigbee. Данная функция необходима для закрытия сессии работы с COM-портом, в случае когда предыдущая сессия не была корректно завершена.

4) Поместить локальную переменную ZGB_Connected внутрь первого окна и присвоить ей значение FALSE.

5) Поместить во второе окно функцию zgb_initialize.vi, вход которой необходимо соединить с numeric control COM-Port. Выход данной функции сравнить со значением «0», отправив на вход Enable express-vi Display Msg.

6) Установить в окне Message to Display параметр Initialization Failed. Данное окно предназначено для инициализации соединения с COM-портом, к которому подключен модуль ZIG-100 с помощью ZIG2Serial и USB2Dynamixel. В случае успешной инициализации соединения функция zgb_initialize.vi возвращает значение «1», а в случае неуспешной – значение «0» и сообщение о неуспешной инициализации.

7) В последнем окне поместить структуру Case Structure и соединить ее вход с выходом OK express-vi Display Msg. В окно True поместить константу False, а в окно False – константу True и соединить их с индикатором ZGB_Connected. При завершении данного этапа блок-диаграмма должна выглядеть как на рисунке.



8) Создать кнопку OK Button и дать ей название Transmit. Установить настройку этой кнопки Mechanical Action в позицию Switch Until Released.

9) На вход второй структуры Case Structure, созданной в п.1, необходимо подать значение переменной ZGB_Connected с помощью локальной переменной.

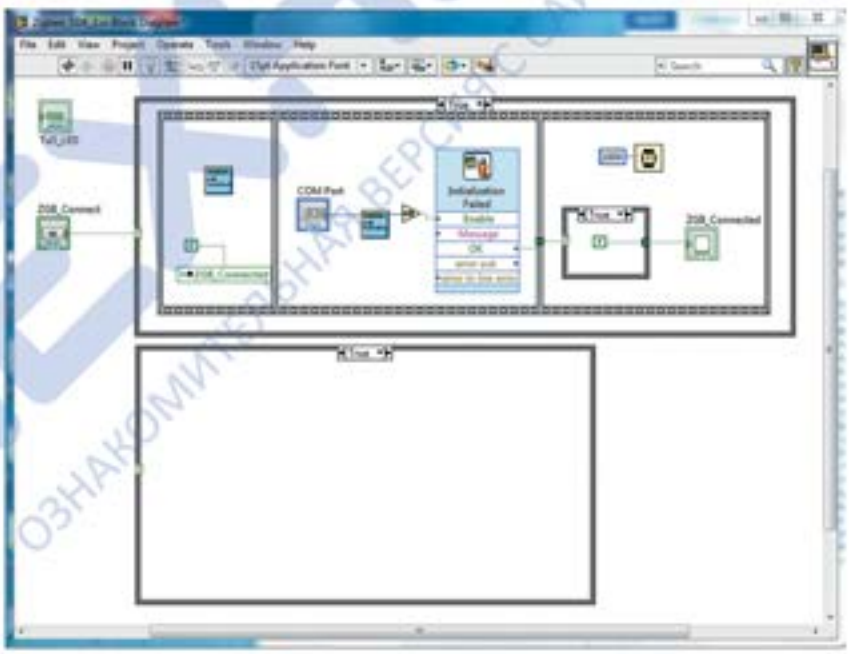
10) Поместить внутрь окна True цикл While Loop, а внутрь данного цикла поместить структуру Case Structure, вход которой необходимо соединить с кнопкой Transmit.

11) Внутрь структуры Case Structure поместить функцию zgb_tx_data.vi, вход которой необходимо соединить с numeric control data to transmit. Выход структуры необходимо сравнить со значением «0», отправив на вход Enable express-vi Display Msg, расположенный вне структуры.

12) В окне Message to Display необходимо установить сообщение Transmission Failed. Также внутри окна True текущей структуры Case Structure необходимо присвоить локальной переменной TxD_LED значение true. Данный этап реализует механизм отправки сообщений с помощью интерфейса ZigBee.

13) В окне False текущей структуры Case Structure необходимо поместить еще одну структуру Case Structure и подать ей на вход значение переменной TxD_LED.

14) Поместить в окне True структуры Case Structure функцию zgb_tx_data.vi, с поданным на вход значением «0». Реализация данного этапа необходима для фильтрации повторной отправки команд. При завершении данного этапа блок-диаграмма должна выглядеть как на рисунке.



15) Полученную блок-диаграмму необходимо целиком поместить в цикл While Loop.

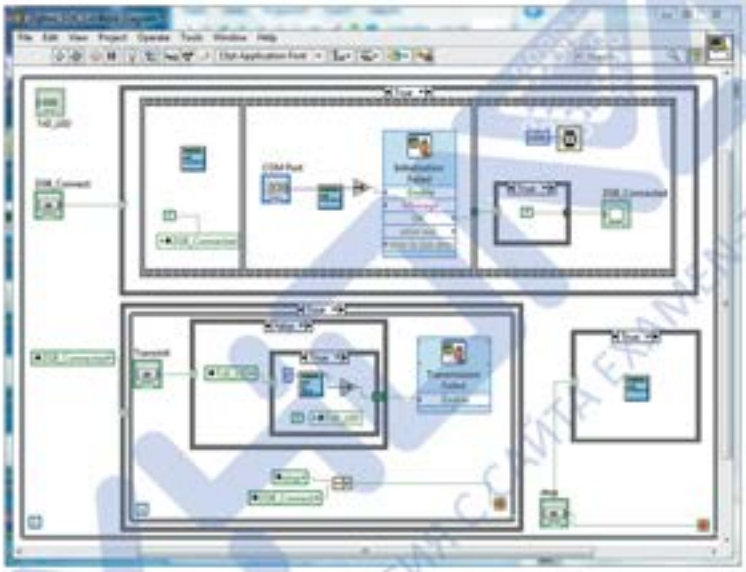
16) Создать кнопку Stop и установить ее настройку в позицию Switch When Released.

17) Соединить кнопку Stop и Loop Condition последнего созданного цикла While Loop. Также внутри текущего цикла While Loop расположить структуру Case Structure, а в ее окне True разместить функцию zgb_terminate.vi.

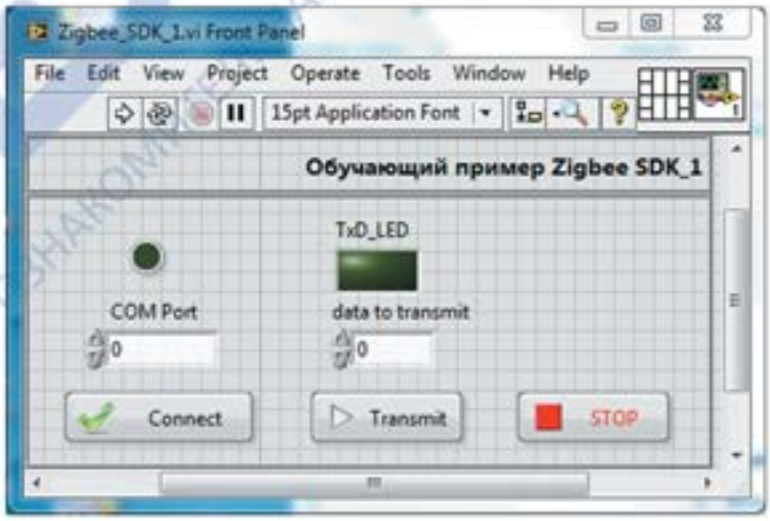
18) Соединяем вход структуры Case Structure с кнопкой Stop.

19) В созданном цикле While Loop необходимо объединить локальные переменные stop и ZGB_Connect с помощью логической операции OR. Полученный результат необходимо соединить с Loop Condition цикла While Loop.

20) Разрабатываемая блок-диаграмма готова. Результаты работы должны выглядеть как на рисунке ниже.



Для удобства применения данной программы необходимо настроить ее лицевую панель на подобии того, как это сделано на рисунке ниже.



Порядок работы с программой

1) Включить робота Bioloid и настроить беспроводное соединение по интерфейсу ZigBee.

2) Убедиться в том, что программируемый контроллер CM-530 запрограммирован соответствующим образом.

3) Запустить программу и осуществить подключение к COM-порту. В случае необходимости номер COM-порт можно задавать в соответствующем окне.

4) В окне data to transmit ввести код команды в текстовом формате.

Если в контроллере робота установлена соответствующая программа и робот собран верно, то в зависимости от поданной команды робот будет выполнять одно из заданных действий.

Данная программа дублирует пульт дистанционного управления RC-100, поэтому для ее использования совместно с роботом необходимо всего лишь модифицировать программу ручного управления роботом так, чтобы она воспринимала команды, поступающие по последовательному интерфейсу.

Модификация программы контроллера CM-530

Для того чтобы роботом серии Bioloid можно было управлять дистанционным образом, необходимо модифицировать его программу управления. Суть всех модификаций сводится к тому, что каждое из действий робота должно выполняться вследствие вызова с помощью команды от удаленного компьютера.

```
26:   ENDLESS LOOP
27:   {
28:       WAIT WHILE ( Remocon Arrived == FALSE )
29:       ReceiveData = Remocon RXD
```

Передаваемое значение ReceivedData сравнивается с всеми возможными командами, запрограммированными в контроллере CM-530. В результате определяется код заданной операции KeyValue.

```
31:       KeyValue = ReceiveData & U+D+L+R
```

В зависимости от текущего кода операции, т.е. от значения переменной KeyValue, выполняется одно из условий.

ОБРАЗОВАТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

```
32:     IF ( KeyValue == U )
33:     {
34:         CALL standby
35:         Buzzer time = Play Melody
36:         Buzzer index = Melody0
37:         CALL forward
38:     }
39:
40:     ELSE IF ( KeyValue == D )
41:     {
42:         CALL standby
43:         Buzzer time = Play Melody
44:         Buzzer index = Melody0
45:         CALL reverse
46:     }
47:
48:     ELSE IF ( KeyValue == L )
49:     {
50:         CALL standby
51:         Buzzer time = Play Melody
52:         Buzzer index = Melody1
53:         CALL pivot_left
54:     }
```

Поскольку программа LabView пересылает текстовые команды, в виде кода операции, необходимо определить коды операций, задаваемых нажатиями клавиш пульта RC-100.

Код операции можно определить в меню панели управления, в настройках пульта RC-100. При выборе какой-либо кнопки пульта управления, а также их комбинаций, в окне ниже отображается текстовое значение кода операции.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Часть 2. Сбор показаний массива ИК-датчиков

Основная задача данного примера – иллюстрация процесса сбора и анализа данных с помощью программы, разработанной в среде программирования LabView. В данной работе предлагается рассмотреть процесс сбора данных с массива ИК-дальномеров, часто применяемого в наборах серии Bioloid.

Данная работа демонстрирует одно из основных предназначений LabView – сбор и анализ результатов в рамках проведения эксперимента.

Разработка программы в среде программирования LabView.

Программа, разрабатываемая в данной части, основывается на блок-диаграмме из предыдущей, поскольку они содержат общую часть, касающуюся передачи данных по COM-порту.

Принцип работы программы заключается в передаче на удаленный компьютер по беспроводному каналу показаний одного из ИК-датчиков из массива.

1) Для разработки программы необходимо открыть блок-диаграмму, созданную в предыдущем разделе.

2) Очистить цикл While Loop от содержимого.

3) Внутри цикла While Loop необходимо создать структуру Case Structure и элемент управления (control) с именем «Запрос».

4) Соединить элемент управления «Запрос» со входом структуры Case Structure.

5) В окне True используемой структуры расположить элемент Flat Sequence с тремя окнами (Frame).

6) Поместить внутрь первого окна функцию zgb_tx_data.vi из библиотеки ZigBee. Данная функция предназначена для передачи контроллеру CM-530 номера ИК-датчика, показания которого необходимо передать.

7) Для задания номера ИК-датчика необходимо создать элемент управления с именем «Номер сенсора» и индикатор с именем «Код ошибки запроса». Соединить оба этих элемента со входом и выходом функции zgb_tx_data.vi.

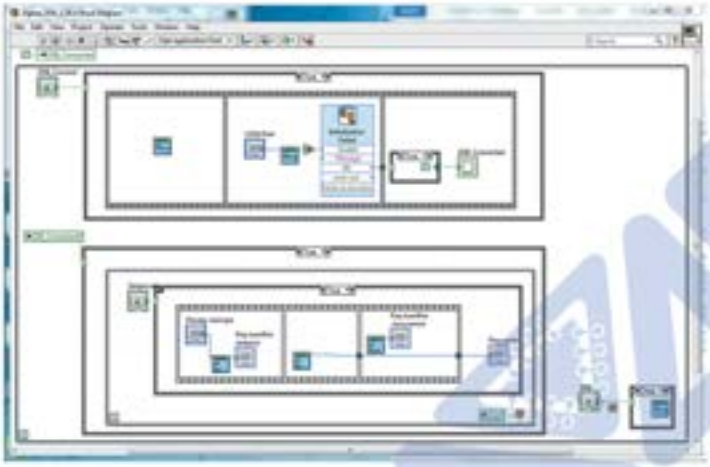
8) Поместить внутрь второго окна функцию zgb_rx_data.vi из библиотеки ZigBee. Данная функция предназначена для получения данных от контроллера CM-530.

9) Создать индикатор с именем «Результат» и соединить его с выходом функции, проведя соединение через третье окно Flat Sequence.

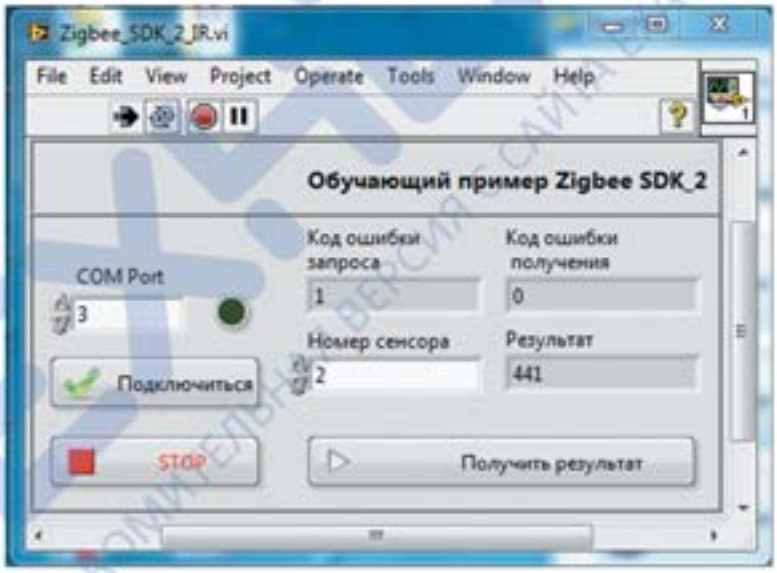
10) В третье окно элемента Flat Sequence необходимо поместить функцию zgb_rx_check.vi, предназначенную для вывода результатов.

11) Создать индикатор с именем «Код ошибки получения» и соединить его с функцией zgb_rx_check.vi.

12) В результате блок-диаграмма должна приобрести вид идентичный рисунку ниже.



Для удобства применения данной программы необходимо настроить ее лицевую панель на подобии того, как это сделано на рисунке ниже.



Порядок работы с программой

1) Включить робота Bioloid и настроить беспроводное соединение по интерфейсу ZigBee.

2) Убедиться в том, что программируемый контроллер CM-530 запрограммирован соответствующим образом.

3) Запустить программу и осуществить подключение к COM-порту. В случае необходимости номер COM-порт можно задавать в соответствующем окне.

4) В окне «Номер сенсора» ввести номер ИК-датчика из массива и нажать кнопку «Получить результат».

Модификация программы контроллера CM-530

Для того чтобы программируемый контроллер робота реагировал на запросы, поступающие от удаленного компьютера, его программа управления должна быть скорректирована соответствующим образом.

```
START PROGRAM
{
    ENDLESS LOOP
    {
        WAIT WHILE ( Remocon Arrived == FALSE )
        ReceiveData = Remocon RXD

        IF ( ReceiveData == 1 )
        {
            Remocon TXD = ID[100] IR Fire Data 1
        }
        IF ( ReceiveData == 2 )
        {
            Remocon TXD = ID[100] IR Fire Data 2
        }
        IF ( ReceiveData == 3 )
        {
            Remocon TXD = ID[100] IR Fire Data 3
        }
        IF ( ReceiveData == 4 )
        {
            Remocon TXD = ID[100] IR Fire Data 4
        }
        IF ( ReceiveData == 5 )
        {
            Remocon TXD = ID[100] IR Fire Data 5
        }
        IF ( ReceiveData == 6 )
        {
            Remocon TXD = ID[100] IR Fire Data 6
        }
        IF ( ReceiveData == 7 )
        {
            Remocon TXD = ID[100] IR Fire Data 7
        }


    }
}
```

Код управляющей программы должен содержать перечень условий, определяющих номер одного из датчиков, с которого необходимо считать значения. Номер каждого из условий должен соответствовать одному из номеров, передаваемых по команде от удаленного компьютера.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Часть 3. Сбор показаний массива ИК-датчиков

Основная задача данного примера – иллюстрация возможностей автоматического сбора данных с мобильного робота и передачи их на удаленный компьютер. Подобные решения очень востребованы в реальных задачах, т.к. обычно процесс сбора и архивации информации происходит автоматически или циклично, а не по отдельным запросам.

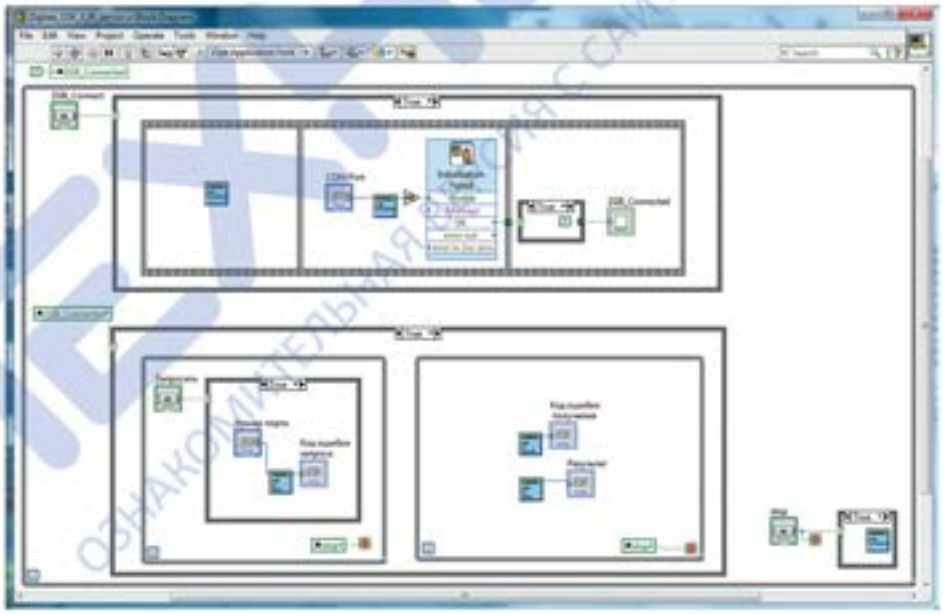
Получение своевременной информации о состоянии технической системы дает возможность системе управления скорректировать рабочий режим и выбрать оптимальный алгоритм работы.

Разработка программы в среде программирования LabView.

Программа, разрабатываемая в данной части, основывается на блок-диаграмме из предыдущей, поскольку они содержат общую часть, касающуюся передачи данных по COM-порту.

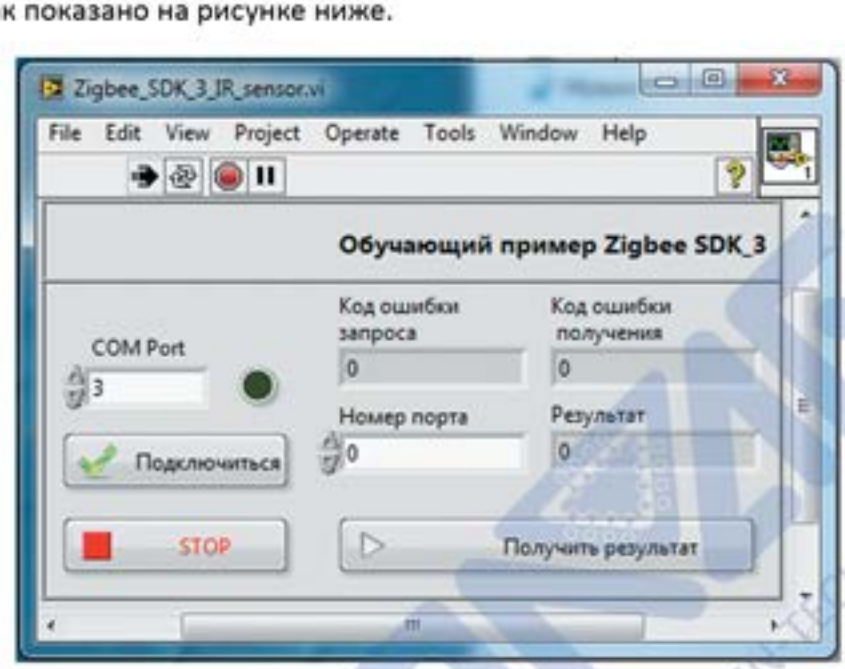
Принцип работы программы заключается в передаче на удаленный компьютер по беспроводному каналу показаний одного из ИК-датчиков, подключенных к одному из портов контроллера CM-530.

Отличие данной программы от предыдущей заключается в том, что в данной программе осуществляется непрерывный объем данными между контроллером CM-530 и удаленным компьютером. Для этого необходимо внести изменения в разработанную ранее блок-диаграмму.



Для автоматизации процесса считывания данных необходимо поместить функцию zgb_rx_data.vi в параллельный циклу запроса цикл While Loop. В итоге при нажатии на кнопку «Получить результат» будет проводиться опрос выбранного порта до тех пор, пока не будет выбран другой порт.

Лицевая панель программы не претерпела существенных изменений по сравнению с предыдущей частью. Рекомендуется настроить лицевую панель аналогичным образом, как показано на рисунке ниже.



Порядок работы с программой

1) Включить робота Bioloid и настроить беспроводное соединение по интерфейсу ZigBee.

2) Убедиться в том, что программируемый контроллер CM-530 запрограммирован соответствующим образом.

3) Запустить программу и осуществить подключение к COM-порту. В случае необходимости номер COM-порт можно задавать в соответствующем окне.

4) В окне «Номер порта» ввести номер порта, к которому подключен ИК-датчик и нажать кнопку «Получить результат».

5) Результаты измерений датчика будут отображаться в окне «Результат».

Модификация программы контроллера CM-530

Для того чтобы программируемый контроллер робота реагировал на запросы, поступающие от удаленного компьютера, его программа управления должна быть скорректирована соответствующим образом.

ПОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

```
START PROGRAM
{
    ENDLESS LOOP
    {
        WAIT WHILE ( Remocon Arrived == FALSE )
        ReceiveData = Remocon RXD

        IF ( ReceiveData == 1 )
        {
            LOOP WHILE ( Remocon RXD == 1 )
            {
                Remocon TXD = PORT[1]
            }
        }
        IF ( ReceiveData == 2 )
        {
            LOOP WHILE ( Remocon RXD == 2 )
            {
                Remocon TXD = PORT[2]
            }
        }
        IF ( ReceiveData == 3 )
        {
            LOOP WHILE ( Remocon RXD == 3 )
            {
                Remocon TXD = PORT[3]
            }
        }
        IF ( ReceiveData == 4 )
        {
            LOOP WHILE ( Remocon RXD == 4 )
            {
                Remocon TXD = PORT[4]
            }
        }
        IF ( ReceiveData == 5 )
        {
            LOOP WHILE ( Remocon RXD == 5 )
            {
                Remocon TXD = PORT[5]
            }
        }
        IF ( ReceiveData == 6 )
        {
            LOOP WHILE ( Remocon RXD == 6 )
            {
                Remocon TXD = PORT[6]
            }
        }
    }
}
```

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Код управляющей программы должен содержать перечень условий, определяющих номер одного из портов, с которого необходимо считать значения. Номер каждого из условий должен соответствовать одному из номеров, передаваемых по команде от удаленного компьютера.

Часть 4. Управление сервоприводами Dynamixel

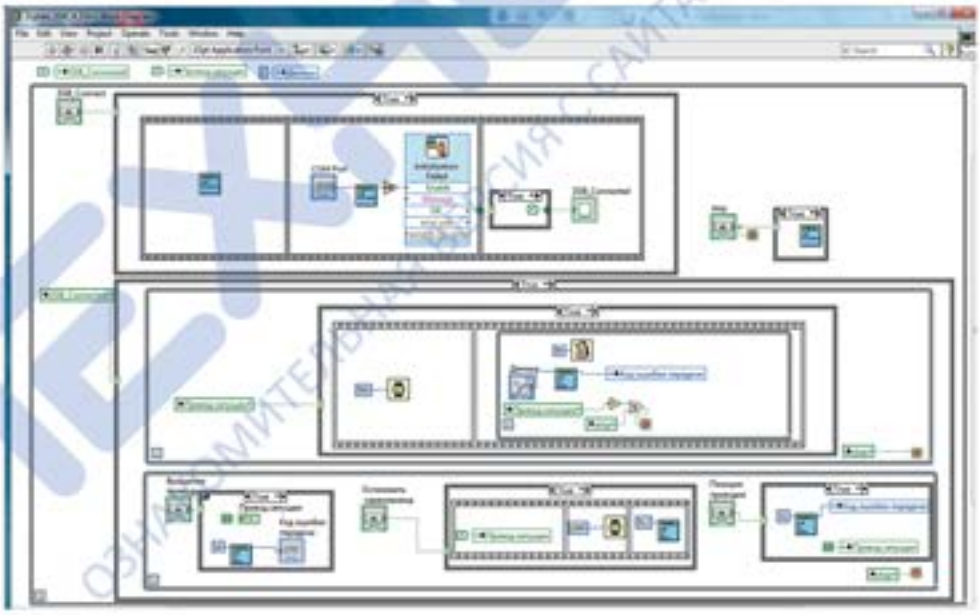
Основная задача данного примера – иллюстрация процесса управления сервоприводами с помощью последовательного протокола. В данной работе предлагается рассмотреть процесс управления сервоприводами Dynamixel с помощью программы, разработанной в программной среде LabView.

Данный пример демонстрирует процесс управления сервоприводами в различных режимах – в режиме управления скоростью вращения выходного вала и в режиме отработки и удержания заданного положения.

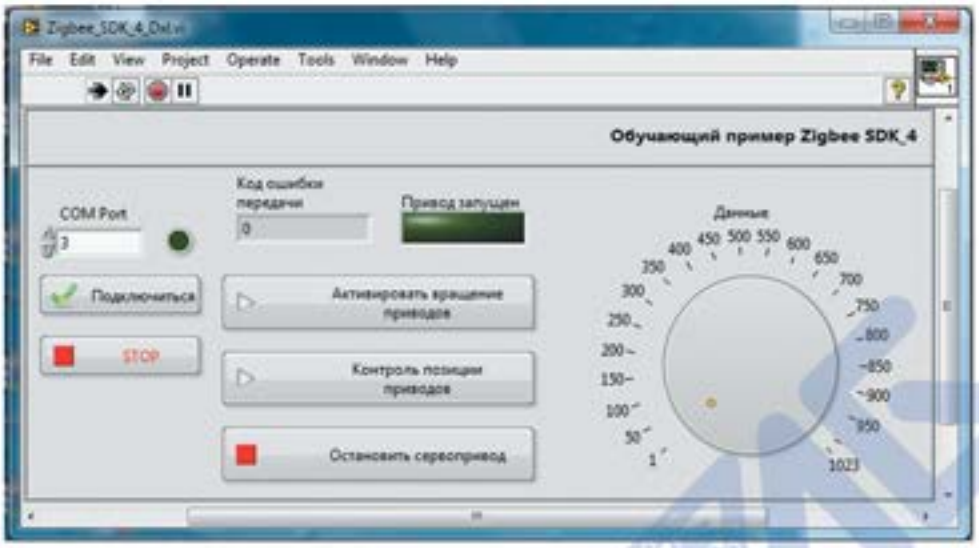
Разработка программы в среде программирования LabView

Данная программа является завершающей для цикла из четырех работ и обобщает в себе наработки каждой из предыдущих. Так же как и в предыдущих частях, программа разрабатывается на основе предыдущей, путем ее усовершенствования. Особенность данной работы в том, что она обобщает в себе два различных режима управления сервоприводами – режим постоянного вращения с заданной скоростью (Wheel), а также режим позиционного управления (Joint).

Поскольку сервопривод может работать в одном из двух вышеуказанных режимов, в блок-диаграмме необходимо добавить два параллельно работающих цикла While Loop. В результате выбора одного из режимов работы на привод поступают данные в диапазоне от 0 до 1023. В режиме Wheel данные представляют собой скорость вращения привода. В режиме Joint данные задают угол поворота привода.



Поступающие данные на привод задаются с помощью специального control элемента, располагаемого на лицевой панели программы.



Порядок работы с программой

1. Установить требуемый режим работы сервоприводов с помощью программы, написанной для контроллера CM-530 в среде RoboPlus.
2. Включить робота Bioloid и наладить канал связи по интерфейсу ZigBee. Убедиться в правильности настроек по свечению светодиодов модулей.
3. Запустить программу, выбрать COM-порт (по умолчанию COM3) и нажать кнопку «Подключиться». В случае успешного подключения к COM-порту загорится лампочка.
4. Выбрать один из режимов работы приводов.
5. Изменяя положение модуля управления «Данные», задать требуемое значение.
6. При завершении работы нажать кнопку «Остановить сервопривод» и затем Stop.

Модификация программы контроллера CM-530

В отличие от предыдущих примеров в данном случае программа контроллера должна не только реагировать на изменяющиеся входные значения, но и сама функционировать в двух различных режимах.

Программа алгоритмически представляет собой бесконечный цикл, в котором анализируются приходящие по COM-порту данные. Первые пришедшие данные классифицируются как команды, задающие один из режимов работы сервопривода – режим постоянного вращения с помощью функции forward или режим управления положением с помощью функции position.

```
1: START PROGRAM
2: {
3:     Buzzer time = Play Melody
4:     Buzzer index = Melody0
5:     ENDLESS LOOP
6:     {
7:         WAIT WHILE ( Remocon Arrived == FALSE )
8:         ReceiveData = Remocon RXD
9:
10:        KeyValue = ReceiveData & U+D+L+R+1+2+3+4+5+6
11:        IF ( KeyValue == 1 )
12:        {
13:            Buzzer time = Play Melody
14:            Buzzer index = Melody0
15:            CALL forward
16:        }
17:        IF ( KeyValue == 2 )
18:        {
19:            Buzzer time = Play Melody
20:            Buzzer index = Melody6
21:            CALL position
22:        }
23:    }
24: }
```

Данные, поступающие в контроллер CM-530, распознаются функциями forward и position как управляющие значение, т.е. как величины задания скорости вращения сервопривода и его целевое положение.

```
25: FUNCTION forward
26: {
27:     LOOP WHILE ( Remocon RXD != 6 )
28:     {
29:         l_wheel_speed = Remocon RXD
30:         r_wheel_speed = Remocon RXD
31:         ID[1]: Moving speed = CCW:0 + l_wheel_speed
32:         ID[2]: Moving speed = CW:0 + r_wheel_speed
33:     }
34:     ID[1]: Moving speed = CCW:0
35:     ID[2]: Moving speed = CW:0
36: }
```

```
37: FUNCTION position
38: {
39:     LOOP WHILE ( Remocon RXD != 6 )
40:     {
41:
42:         ID[1]: Goal position = Remocon RXD
43:         ID[2]: Goal position = Remocon RXD
44:     }
45:     ID[1]: Goal position = 1
46:     ID[2]: Goal position = 1
47: }
```

Аналогичным образом можно задавать различные скорости вращения и целевые положения для любого из сервоприводов, подключенных к контроллеру CM-530.

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ
ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Для заметок

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

Учебно-методическое издание

Ермишин Константин Владимирович
Кольин Максим Анатольевич
Каргин Дмитрий Николаевич
Панфилов Алексей Олегович

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЯ

ОБРАЗОВАТЕЛЬНЫЙ
РОБОТОТЕХНИЧЕСКИЙ МОДУЛЬ

(ИССЛЕДОВАТЕЛЬСКИЙ УРОВЕНЬ)
от 14 ЛЕТ

Издательство ***«ЭКЗАМЕН»***
«ЭКЗАМЕН-ТЕХНОЛАБ»

Гигиенический сертификат
№ РОСС RU. АЕ51. Н 16466 от 25.03.2013 г.

Главный редактор *Л. Д. Лаппо*
Корректоры *Н. С. Садовникова, С. С. Гаврилова, Е. В. Григорьева*
Дизайн обложки
и компьютерная верстка *А. А. Винокуров*

107045, Москва, Луков пер., д. 8.
E-mail: по общим вопросам: robo@examen-technolab.ru;
www.examen-technolab.ru
по вопросам реализации: sale@examen-technolab.ru
тел./факс +7 (495) 641-00-19 (многоканальный)

Отпечатано в соответствии с предоставленными материалами
в ООО «ИПК Парето-Принт», г. Тверь, www.pareto-print.ru

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ САЙТА EXAMEN-TECHNOLAB.RU

ТЕХНОЛАБ

ОБРАЗОВАТЕЛЬНЫЙ РОБОТОТЕХНИЧЕСКИЙ МОДУЛЬ

(ИССЛЕДОВАТЕЛЬСКИЙ УРОВЕНЬ)

ТЕХНОЛАБ

ОЗНАКОМИТЕЛЬНАЯ ВЕРСИЯ С САЙТА EXAMEN-TECHNOLAB.RU

www.examen-technolab.ru

Артикул ТР-0621-МП

ISBN 978-5-377-07627-8

14+
ЛЕТ