

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Тобольский педагогический институт им. Д.И. Менделеева (филиал) Тюменского
государственного университета

УТВЕРЖДАЮ

Директор

Шилов С.П.



ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ПО ДИСЦИПЛИНЕ

**МОБИЛЬНАЯ РОБОТОТЕХНИКА И ОСНОВЫ
МАШИННОГО ЗРЕНИЯ**

44.03.04 Профессиональное обучение (по отраслям)
Профиль: Сервис мехатронных систем
Форма обучения очная

1. Паспорт оценочных материалов по дисциплине

1.1. Перечень компетенций

Код и наименование компетенции	Планируемые результаты обучения: (знаниевые/функциональные)
ПК-1 – Способен реализовывать программы профессионального обучения СПО и (или)ДПП по учебным предметам, курсам, дисциплинам (модулям), практикам	Знает теоретические основы механики мобильных роботов, специфику дистанционного и автономного управления с обратной связью
	Может разработать технические требования и регламенты для создания мобильного робота под конкретную задачу
	Может участвовать в подготовке и организации итоговой государственной аттестации выпускников СПО по специальности «Мобильная робототехника» в формате демонстрационного экзамена
ПК-2 - Способен проводить учебно-производственный процесс при реализации образовательных программ различного уровня и направленности	Знает основы конструирования содержания учебного материала по техническим дисциплинам
	Может подготовить дидактический материал и методические рекомендации в рамках предметной области для подготовки специалистов среднего звена по мобильной робототехнике

1.2. Паспорт оценочных средств по дисциплине

№ п/п	Темы дисциплины (модуля) / Разделы (этапы) практики* в ходе текущего контроля, вид промежуточной аттестации (зачет, экзамен, с указанием семестра)	Код и содержание компетенции (или ее части)	Оценочные материалы (виды и количество)
1	2	3	4
1	История развития мобильной робототехники. Основные понятия	ПК-1, ПК-2	Задания для самостоятельной работы Практическая работа 1
2	Соревновательные мероприятия и компетенции WSR/WSI по мобильной робототехнике	ПК-1, ПК-2	Практические работы 2 Задания для самостоятельной работы
3	Цикл технического проектирования	ПК-1, ПК-2	Задания для самостоятельной работы
4	Механика и приводы мобильных роботов	ПК-1, ПК-2	Практические работы 3 - 4 Решение проектных задач Тестирование
5	Основы потокового программирования в среде LabVIEW МК myRIO	ПК-1, ПК-2	Практические работы 5 Разработка проектного задания
6	Программирование движения мобильного робота	ПК-1, ПК-2	Практические работы 6-8 Разработка проектного задания
7	Реализация обратной связи	ПК-1, ПК-2	Практические работы 9-10 Разработка проектного задания
8	Применение ТАУ для управления мобильным роботом	ПК-1, ПК-2	Практические работы 11 Разработка проектного задания

9	Основы технического зрения	ПК-1, ПК-2	Практические работы 12-14 Разработка проектного задания
10	Методика организации демонстрационного экзамена по мобильной робототехнике	ПК-1, ПК-2	Практические работы 15-16 Разработка проектного задания Подготовка организационной и технической документации для демонстрационного экзамена

2. Виды и характеристика оценочных средств

Текущий контроль осуществляется проверкой наличия конспектов лекций, выполнения заданий в ходе практических занятий, тестовых заданий для робота и самостоятельной работы.

2.1. Практические работы

Практические работы используются для формирования практико-ориентированных знаний, оценки умений по отдельным темам дисциплины. Выполнение практических работ включает в себя 3 этапа:

1) **Изучение/повторение необходимой теории** проходит в виде интерактивной беседы, рассказа, объяснения для понимания и уяснения студентами теоретической информации по данной теме, необходимой для эффективного выполнения практических заданий.

2) **Выполнение практических заданий** во время занятий и самостоятельной работы студентов.

3) **Защита заданий практической работы** проводится в виде соревнований роботов, демонстрации их конструкции и управляющего кода, в виде представления дидактических и учебно-методических проектов.

Содержание заданий и критерии оценки результата доводятся до сведения обучающихся в начале семестра. Оценка объявляется непосредственно после демонстрации решения. В зависимости от уровня сложности задания баллы могут распределяться от 0 до 5.

Балл	Критерий оценивания заданий
4-5	Задания выполнены правильно в полном объеме. Оформление соответствует всем требованиям. Может ответить на уточняющие вопросы. Используются наиболее эффективные методы и средства.
3	Задания выполнены правильно и практически полностью. Оформление в основном соответствует всем требованиям. Может ответить на некоторые уточняющие вопросы. Используются в основном эффективные методы и средства.
2	Задания выполнены частично правильно и не полностью. Оформление соответствует отдельным требованиям. С трудом может ответить на некоторые уточняющие вопросы. Используются не совсем подходящие методы и средства.
0 - 1	Результаты не достигли пороговых критериев или выполнен их незначительный объем

2.2. Тестовые задания

Критерии оценивания текстовых заданий

При составлении/подборе тестовых заданий заранее проектируется необходимый уровень сложности теста. Сложность теста определяется пятью уровнями:

2. Репродуктивный, основными операциями которого являются воспроизведение информации и ее преобразования алгоритмического характера.
3. Базовый, требующий от испытуемого понимания существенных сторон учебной информации, владения общими принципами поиска алгоритмов.
4. Повышенный, уровень сложности задания, требующий от испытуемого умения преобразовывать алгоритмы к условиям, отличающимся от стандартных, умение вести эвристический поиск.

5. Творческий, предполагающий наличие самостоятельного, критического оценивания учебной информации, умение решать нестандартные задания, владение элементами исследовательской деятельности.

Каждому из заданий в соответствии с его сложностью приписывается определенное число, например: информационного характера - 1; репродуктивного - 1,5; базового уровня - 2; повышенной сложности - 2,5; творческого – 3 (или другое количество баллов). Таким образом, получается измерительное устройство в виде шкалы, достаточно понятной и наглядной, которую можно предлагать ученикам или использовать при выставлении баллов за работу над тестом.

Измерительная шкала

Задание	Информационное	Репродуктивное	Базовое	Повышенного уровня	Творческое
Балл	1	1,5	2	2,5	3

Сложность теста определяется как среднее арифметическое сложностей всех заданий, входящих в рассматриваемый тест: $CT = \frac{\sum_{i=1}^n CZ_i}{n}$, где CT - сложность теста; CZ_i - сложность i -го задания теста; n - число заданий в тесте.

Для определения, каким будет тест по вычисленной сложности, следует воспользоваться специальной таблицей:

Определение вида теста по его сложности

Тест	Информативный (ТИ)	Репродуктивный (ТР)	Базовый (ТБ)	Повышенной сложности (ТП)	Творческий (ТТ)
СТ	1 - 1,3	1,4 – 1,6	1,7 – 2,1	2,2 – 2,4	> 2,5

Результаты выполнения различных тестов следует оценивать в зависимости от их сложности, при помощи специальной нормировочной таблицы:

Оценка результатов выполнения тестов различной сложности

СТ %	100	90	80	70	60	50	40	30	20	10	0
ТР	«5»		«4»		«3»		«2»		«1»		
ТБ	«5»			«4»		«3»		«2»		«1»	
ТП	«5»				«4»		«3»		«2»		

2.3. Контрольная работа

Подготовка к Контрольной работе проводится во время самостоятельной работы в конце семестра. Результат - работа мобильного робота на поле/полигоне представляется во время Практической работы 20. Контрольная работа является итоговой формой контроля результатов освоения знаний, умений и навыков, формирующих в рамках данного предмета компетенции ПК-13, ПК-14.

Форма контрольной работы: демонстрация работы мобильного робота на поле в соответствии с требованиями организации демонстрационного экзамена по компетенции «Мобильная робототехника». Оценивается контрольная работа по 10-балльной шкале.

Критерии оценки контрольной работы

Баллы	Показатели оценки
1	Узнавание отдельных объектов изучения программного учебного материала, предъявленных в готовом виде (фактов, терминов, инструктивных указаний, действий и т.д.)
2	Различение объектов изучения программного учебного материала, предъявленных в готовом виде, и осуществление соответствующих практических действий
3	Воспроизведение части программного учебного материала по памяти (фрагментарный пересказ и перечисление объектов изучения), осуществление умственных и практических действий по образцу

4	Воспроизведение большей части программного учебного материала по памяти (определений, описание в устной или письменной формах объектов изучения с указанием общих и отличительных внешних признаков без их объяснения), осуществление умственных и практических действий по образцу
5	Осознанное воспроизведение значительной части программного учебного материала (описание объектов изучения с указанием общих и отличительных существенных признаков без их объяснения), осуществление умственных и практических действий по известным правилам или образцу
6	Осознанное воспроизведение в полном объеме программного учебного материала (описание объектов изучения с элементами объяснения, раскрывающими структурные связи и отношения), применение знаний в знакомой ситуации по образцу путем выполнения практических упражнений, задач, заданий
7	Владение программным учебным материалом в знакомой ситуации (описание и объяснение объектов изучения, выявление и обоснование закономерных связей, приведение примеров из практики, выполнение упражнений задач и заданий по образцу, на основе предписаний)
8	Владение и оперирование программным учебным материалом в знакомой ситуации (развернутое описание и объяснение объектов изучения, раскрытие сущности, обоснование и доказательство, подтверждение аргументами и фактами, формулирование выводов, самостоятельное выполнение заданий)
9	Оперирование программным учебным материалом в частично измененной ситуации (применение учебного материала как на основе известных правил, предписаний, так и поиск нового знания, способа решения учебных задач, выдвижение предположений и гипотез, наличие действий и операций творческого характера при выполнении заданий)
10	Свободное оперирование программным учебным материалом, применение знаний и умений в незнакомой ситуации (самостоятельные действия по описанию, объяснению объектов изучения, формулированию правил, построению алгоритмов для выполнения заданий, демонстрация рациональных способов решения задач, выполнение творческих работ и заданий)

2.4. Промежуточная аттестация

Промежуточная аттестация может быть выставлена двумя способами: в соответствии с результатами балльно-рейтинговой аттестации в течение семестра или по результатам сдачи экзамена. Экзамен представляет собой устный ответ по вопросам с демонстрацией практических навыков в области создания и программирования мобильных роботов.

Балльно-рейтинговая аттестация

Экзамен выставляется автоматически по результатам балльно-рейтинговой аттестации. Содержание оцениваемой работы студентов приведено выше в пункте 3. Рубежные баллы рейтинговой системы оценки успеваемости студентов:

Вид аттестации	Соответствие рейтинговых баллов и академических оценок		
	<i>Удовлетворительно</i>	<i>Хорошо</i>	<i>Отлично</i>
Экзамен	61-75 баллов	76-90 баллов	91-100 баллов

Экзамен может быть получен и при устном собеседовании по вопросам билета

Критерии выставления оценки за экзамен

Оценка «отлично»:

- Результаты освоения программы дисциплины соответствуют повышенному уровню в соответствии с установленными критериями

- Свободно отвечает на дополнительные вопросы.
- Практическое задание выполнено правильно

Оценка «хорошо»:

- Результаты освоения программы дисциплины соответствуют базовому уровню в соответствии с установленными критериями.
- Частично отвечает на дополнительные вопросы.
- Практическое задание выполнено с небольшими ошибками

Оценка «удовлетворительно»:

- Результаты освоения программы дисциплины соответствуют пороговому уровню в соответствии с установленными критериями.
- Затрудняется отвечать на дополнительные вопросы.
- Затрудняется в разработке практического задания

3. Оценочные средства

3.1. Содержание практических работ

Практическое занятие 2. Стандарт компетенции WSI «Мобильная робототехника»

Вопросы для проработки:

1. Техническое описание компетенции
2. Оборудование рабочих мест
3. Дорожная карта организаторов
4. Конкурсные задания
5. Оценка конкурсных заданий

Источники информации:

<https://worldskills2019.com/ru/event/skills/mobilnaya-robototehnika/index.html>

<https://urpc.ru/institution/speciality/standarty-ps->

[ws/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D0%BA%D0%BE%D0%B5%20%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BC%D0%BF%D0%B5%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B8%20%D0%92%D0%A1%D0%A0%20%D0%9C%D0%BE%D0%B1%D0%B8%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0.pdf](https://urpc.ru/institution/speciality/standarty-ps-ws/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D0%BA%D0%BE%D0%B5%20%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BC%D0%BF%D0%B5%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B8%20%D0%92%D0%A1%D0%A0%20%D0%9C%D0%BE%D0%B1%D0%B8%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0.pdf)

<https://nationalteam.worldskills.ru/upload/iblock/08e/%D0%A2%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5%20%D0%BE%D0%BF%D0%B8%D1%81%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BA%D0%BE%D0%BC%D0%BF%D0%B5%D1%82%D0%B5%D0%BD%D1%86%D0%B8%D0%B8%20%D0%9C%D0%BE%D0%B1%D0%B8%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0.pdf>

http://worldskills-ekb.ru/documents/competition/WSR_2014_TD23_RUS_Robot.pdf

Практическое занятие 3. Разработка проекта мобильного робота под поставленные задачи

Вопросы для проработки:

1. Основные сведения о проектировании мобильных роботов
2. Этапы проектирования
3. Разработка математического описания
4. Синтез управления на упрощённой модели

Пример технического проекта

Задание WSR 2020

1. Рама и общая конструкция

1.1 Выбор шасси

Шасси робота должно обеспечивать следующие функции:

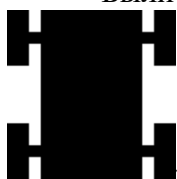
- Перемещение вперёд и назад на заданное расстояние с точностью до 5 мм
- Повороты с точностью до 2 градусов
- Способность въезжать на 3 ступени по 25x19 мм каждая
- Способность перемещаться по труднопроходимой поверхности (песок/газон)
- Способность маневрировать в квадрате 600x600мм
- Возможность сборки шасси из комплекта WorldSkills
- По возможности, использование минимального количества двигателей

Для того, что бы робот мог передвигаться по ступеням и труднопроходимой поверхности необходимо:

- постоянный контакт хотя бы одного ведущего колеса с поверхностью
- оси вращения всех колёс должны быть перпендикулярны направлению движения

Исходя из вышесказанного, необходима полноприводная схема шасси, НЕ предназначенная для omni-передвижения.

Были разработаны следующие варианты шасси:



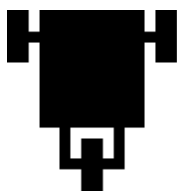
«Классическая» полноприводная схема:

+

Простая в реализации схема

-

Требуется 4 двигателя, сложное управление



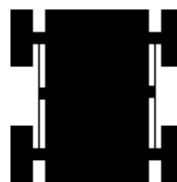
Трёхколёсная полноприводная схема:

+

Требуется 3 двигателя

-

Усложнённая реализация. Увеличенный продольный габарит/ Требуется omni-колесо



4-х колёсное полноприводное шасси с цепным приводом

+

Требуется 2 двигателя

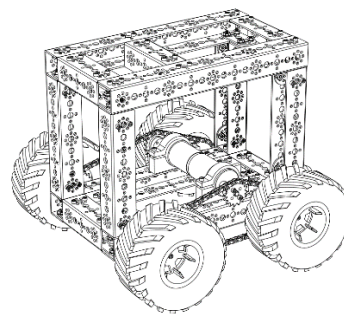
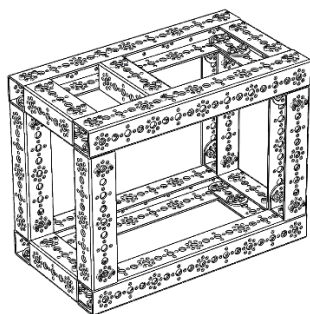
-

Усложнённая реализация
Большая нагрузка на двигатели

Из представленных схем нами была выбрана схема 4-х колёсного полноприводного шасси с цепным приводом колёс.

1.2.Рама

Рама работа представляет собой параллелепипед, собранный из 4-х балок длиной 160мм, 4-х балок длиной 288мм и, 2-х балок длиной 96мм. В качестве соединителей используются п-образные коннекторы и планки длиной 64 мм. В верхней части рамы установлена дополнительная поперечная 96мм балка для крепления СМО. Габариты рамы: 288 x 160 x 224.



а. Шасси

Шасси робота – четырёхколёсное полноприводное с цепным приводом колёс. Колёса, которые расположены на одном борту и двигатель соединены между собой цепью. Привод колёс осуществляется двигателями постоянного тока 12В. Данная схема позволяет использовать всего 2 двигателя для движения.

2. Электрика

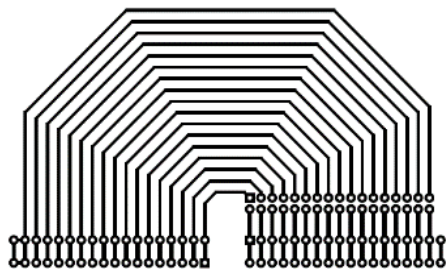
2.1 Список оборудования

На роботе установлено следующее электронное оборудование:

1. Контроллер myRIO
2. Драйверы двигателей (2 шт.)
3. Двигатели DC 12В (4 шт.)
4. DC-DC преобразователь с 12В на 5В (Voltage regulator module)
5. Сервоприводы HS-485HB (3 шт.)
6. Датчики:
 - Оптические энкодеры (4 шт.)
 - Гироскоп navX micro
 - Ультразвуковой датчик PING))) (1 шт.)
 - ИК датчики Sharp IR range sensor (2 шт.)
 - USB камера Microsoft LifeCam
 - FPV камера
7. Стартовая кнопка
8. Кнопка перезагрузки гироскопа
9. Устройства индикации:
 - Лампа накаливания 12В (повторитель поворота ВА3-2106)
 - Зелёные сигнальные светодиоды (3 шт.)
10. Трёхпозиционный переключатель выбора аккумулятора

2.2 Плата

Для удобного и надёжного подключения к контроллеру myRIO были сделаны две одинаковые специальные платы (для портов А и В). Плата подключается к контроллеру плоским кабелем на 34 провода с IDC-разъёмами. На плате установлены нажимные клеммники для подключения проводов и IDC-разъём для подключения к контроллеру.



Подключение доп. Оборудования

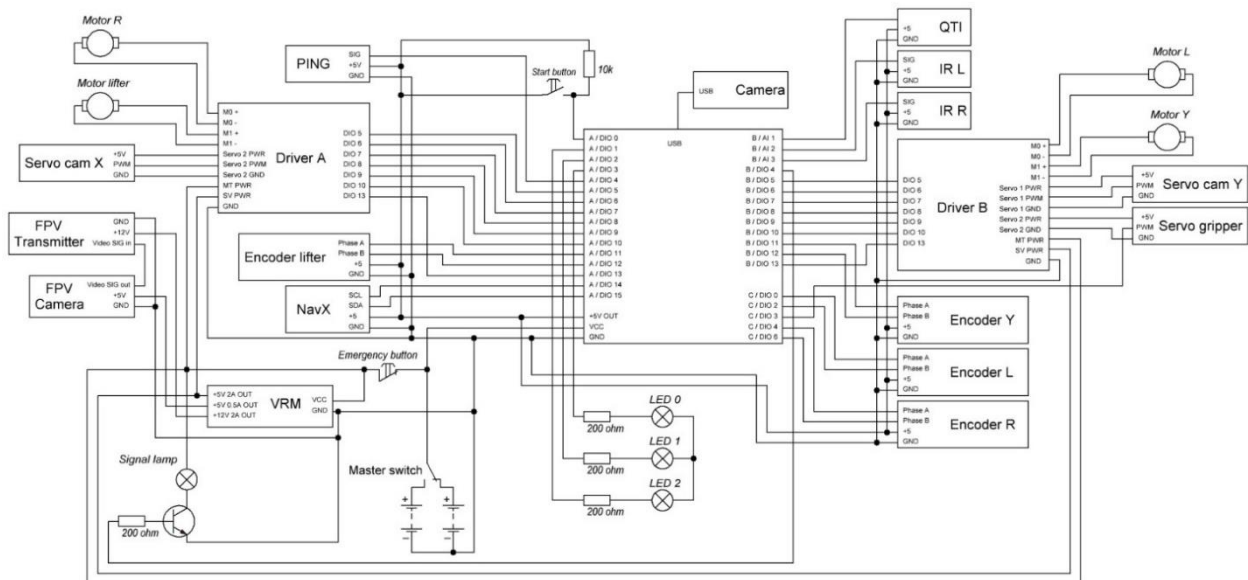


Таблица 1. Подключение проводов

На стороне ведущего устройства			На стороне ведомого устройства	
Устройство/ порт	Пин или разъём	Метка на проводе	Устройство	Пин или разъём
myRIO/A	DIO 0	Start button	Start button	SIG
	DIO 1	LED 1	LED 1	-
	DIO 2	LED 0	LED 0	-
	DIO 3	LED 2	LED 2	-
	DIO 4	PING	Start button	SIG
	DIO 5	DR A DIR M0 0	Driver A	DIO 5
	DIO 6	DR A DIR M0 1		DIO 6
	DIO 7	DR A DIR M1 0		DIO 7
	DIO 8	DR A PWM 0		PWM 0
	DIO 9	DR A PWM 1		PWM 1
	DIO 10	DR A PWM 2	PWM 2	
	DIO 13	DR A DIR M1 1	DIO 13	
	DIO 11	Enc lifter A	Encoder lifter	Phase A
	DIO 12	Enc lifter B		Phase B
	DIO 14	Gyro SCL	navX micro	SCL
DIO 15	Gyro SDA	SDA		
myRIO/B	AI 3	QTI	QTI	R
	DIO 4	LAMP	Signal lamp	SIG
	DIO 5	DR B DIR M0 0	Driver A	DIO 5
	DIO 6	DR B DIR M0 1		DIO 6
	DIO 7	DR B DIR M1 0		DIO 7
	DIO 8	DR B PWM 0		PWM 0
	DIO 9	DR B PWM 1		PWM 1
	DIO 10	DR B PWM 2	PWM 2	
	DIO 13	DR B DIR M1 1	DIO 13	
	DIO 11	Enc lifter A	Encoder Y	Phase A
	DIO 12	Enc lifter B		Phase B
AI 1	IR R	IR R	SIG	
AI 2	IR L	IR L	SIG	
myRIO/C	DIO 0	Enc L A	Encoder L	Phase A
	DIO 2	Enc L B		Phase B
	DIO 4	Enc R A	Encoder R	Phase A
	DIO 6	Enc R B		Phase B

	DIO 3	Servo gripper	Servo gripper	SIG
Diver A	M0 +	M0 +	Motor R	+
	M0 -	M0 -		-
	M1 +	M1 +	Motor lifter	+
	M1 -	M1 -		-
	Servo 2	Servo cam X	Servo cam X	SIG
Diver B	M0 +	M0 +	Motor L	+
	M0 -	M0 -		-
	M1 +	M1 +	Motor Y	+
	M1 -	M1 -		-
	Servo 2	Servo cam Y	Servo cam Y	SIG

3. Управление мобильностью

3.1 Управление драйвером

Для того, что бы управлять скоростью вращения колёс, нужно регулировать напряжение, которое подаётся на двигатель. Но контроллер не может выдать ток и напряжение, которые необходимы двигателю (ток - до 2А, напряжение – до 12 В). Поэтому на роботе установлены 2 двухканальных драйвера двигателя. В момент, когда на вход PWM0 или PWM1 приходит цифровой сигнал высокого уровня, драйвер подаёт питание на клеммы M0 или M1, куда подключаются двигатели. Напряжение на выходе соответствует напряжению на клеммах MTRSPWR. Для регулирования мощности на входы PWM0 и PWM1 подаётся ШИМ-сигнал.

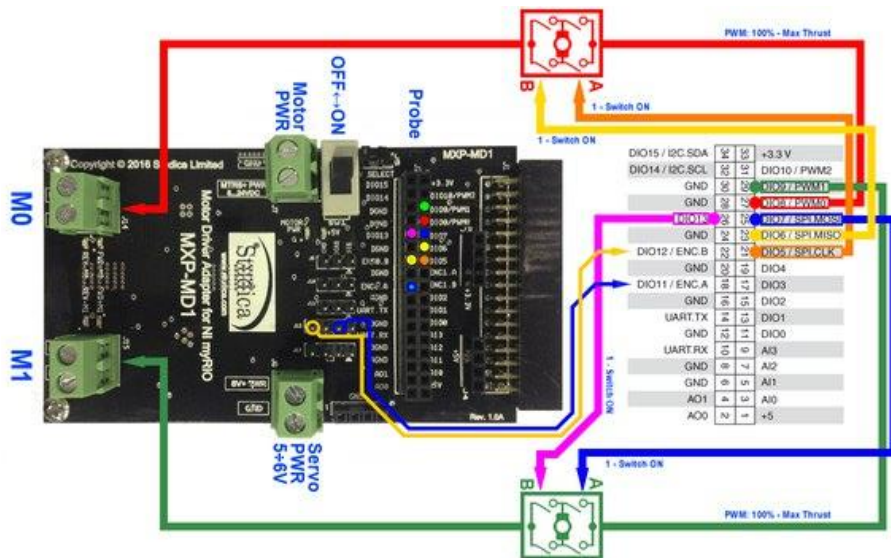
Направление вращения двигателей подачей цифрового сигнала 4 цифровых входа. Двигателю M0 соответствуют DIR0-0 и DIR0-1, двигателю M1 - DIR1-0 и DIR1-1.

Таблица 2. Режимы работы драйвера

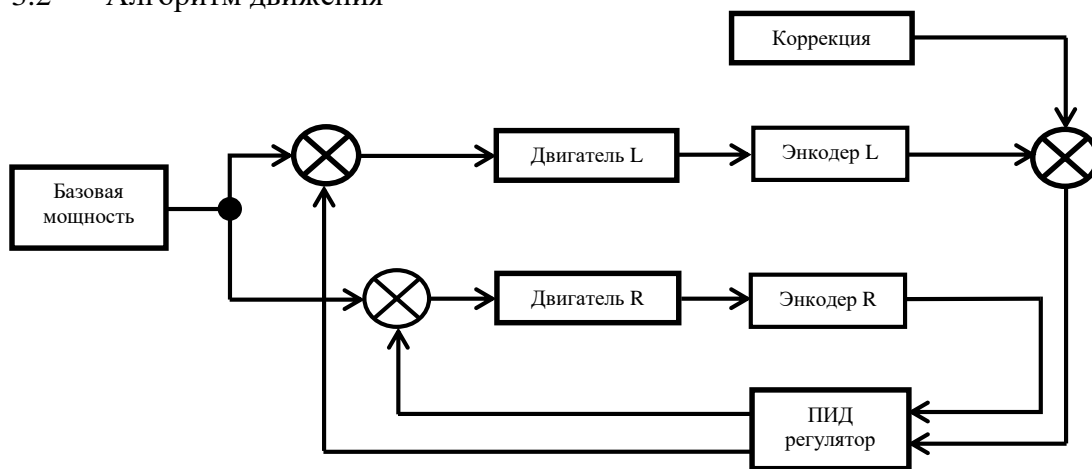
DIR 0	DIR 1	Двигатель
LOW	LOW	Режим торможения
HIGH	LOW	Вращение вперёд
LOW	HIGH	Вращение назад
HIGH	HIGH	Режим торможения

Таблица 3. Подключения к драйверам

Двигатель	Наименование драйвера	Пин драйвера	Подключенное устройство
Left	B	PWM0	myRIO B PWM0
		DIR0-0	myRIO B DIO 5
		DIR0-1	myRIO B DIO 6
		M0 +	Motor left +
		M0 -	Motor left -
Right	A	PWM0	myRIO A PWM0
		DIR0-0	myRIO A DIO 5
		DIR0-1	myRIO A DIO 6
		M0 +	Motor right +
		M0 -	Motor right -
Lifter	A	PWM1	myRIO A PWM1
		DIR1-0	myRIO A DIO 7
		DIR1-1	myRIO A DIO 13
		M0 +	Motor lifter +
		M0 -	Motor lifter -
Y	B	PWM1	myRIO B PWM1
		DIR1-0	myRIO B DIO 7
		DIR1-1	myRIO B DIO 13
		M1 +	Motor Y +
		M1 -	Motor Y -



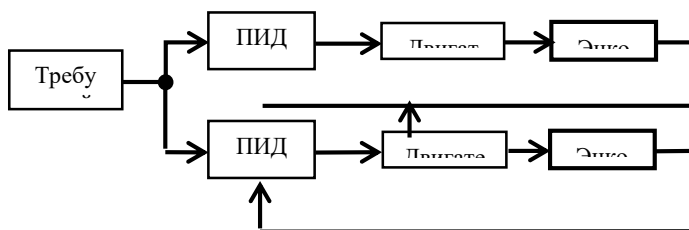
3.2 Алгоритм движения



Для движения используется информация с оптических энкодеров, расположенных на левом и правом двигателях. Для движения вперёд или назад необходимо, чтобы левое колесо вращалось одновременно с правым, то есть, показания энкодеров левого и правого колеса должны быть равны.

3.3 Алгоритм вращения

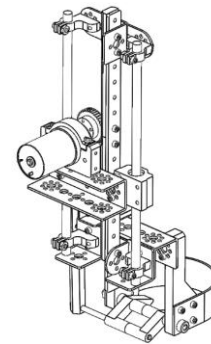
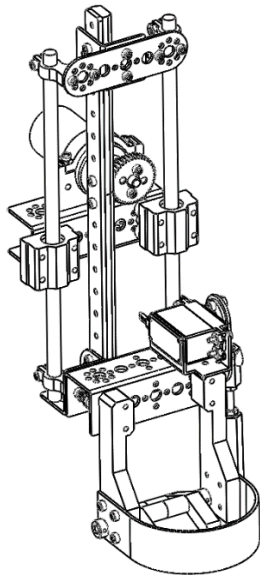
Для поворота на определённый угол необходимо, чтобы разница между левым и правым энкодером была равна какому-то значению. Это зависит от энкодера, диаметра колёс и ширины базы. У нас – 24,3 тика на один градус.



- Опорная балка
- Балка захватного механизма
- Линейные подшипники
- Валы 8мм
- Планка 144мм
- Зубчатые рейки
- Держатели валов
- Двигатель
- Приводная шестерня

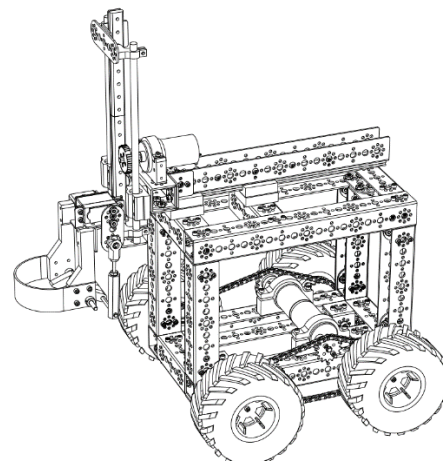
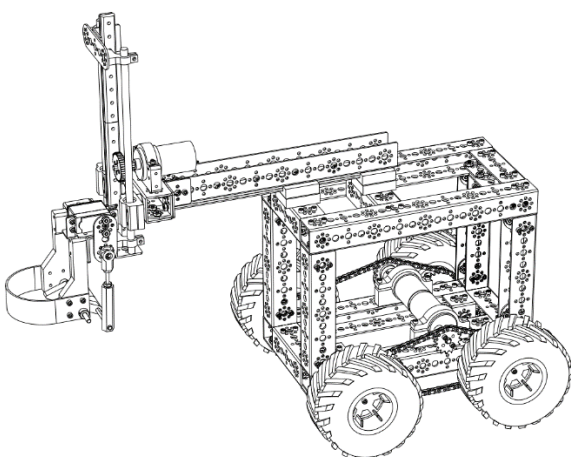
Привод подъёмного механизма представляет собой шестерённо-реечный механизм. Приводная шестерня закреплена непосредственно на валу двигателя. При вращении шестерни приводится в движение зубчатая планка, которая жёстко соединена с балкой захватного механизма.

На верхней опорной балке установлен QTI-датчик, который выполняет функции концевика сразу для двух осей. Он срабатывает при полном задвижении захвата и при полном подъёме подъёмного механизма.



4.3. Механизм задвигания/выдвигания СМО (ось Y)

Принцип работы механизма оси Y похож на механизм подъёма, здесь так же применяется шестерённо-реечный механизм. В качестве линейной направляющей используется рельсовая направляющая шириной 12мм и 2 каретки длиной 36 мм. Из-за того, что двигатель расположен слишком низко, вал двигателя пришлось удлинить при помощи переходника 6/4,7 мм оси 4,7 x 60 мм.



4.4.Преимущества и недостатки СМО

Данный захват позволяет произвести захват нужного шара, даже если он будет находится в углу поля и будет закрыт другими шарами. Так же данный захват позволяет произвести захват шара на газоне, не заезжая на газон. Система управления позволяет позиционировать захват с точностью до 2 мм. Однако, данный захват является довольно тяжёлым, что плохо сказывается на подвижности робота, а процедура захвата шара занимает много времени. Исходя из этого можно сделать вывод, что данный захват ориентирован в основном на выполнение автономных задач.

Знакомство с техническим конструктором WSR/WSI Tetrix.

Робототехнические комплекты TETRIX.

TETRIX – это линейка алюминиевых конструкторов. Наборы предназначены от 14 лет и старше. На его основе можно построить робота с дистанционным управлением или, используя микрокомпьютер и датчики, создать автономного робота.

TETRIX - основной конструктор международных соревнований FIRST Tech Challenge, на базе конструктора TETRIX MAX разрабатываются наборы для соревнований WSR/WSI

Конструктор TETRIX совместим с конструктором LEGO, позволяет использовать контроллер, любые совместимые датчики и исполнительные устройства.

Набор Studica – комплект конструкционных и электронных компонент, предназначенный для решения заданий WSR/WSI

Задания:

Изучить линейки робототехнических Tetrix.

Изучить название деталей и принципы их соединения.



Практическое занятие 4. Конструирование и расчет механических передач (зубчатая).

Задание: Зубчатая передача

1. Изучить и законспектировать особенности зубчатой механической передачи, ее виды и особенности.
2. Вычислить передаточное число указанных зубчатых передач
3. Сконструировать ступенчатую зубчатую передачу с передаточным числом 27, 45, 135.
4. Сконструировать ступенчатую передачу с передаточным числом 15 и противоположным направлением вращения ведущего и ведомого вала
5. Сконструировать коронную, реечную, червячную передачу с заданными характеристиками.

Задание: ременная, цепная передача

1. Изучить и законспектировать особенности ременной механической передачи, ее виды и особенности.
2. Вычислить передаточное число указанных ременных передач
3. Сконструировать ступенчатую ременную передачу с передаточным числом 4; числом 16 и противоположным направлением вращения ведущего и ведомого вала.
4. Изучить и законспектировать особенности цепной механической передачи, ее виды и особенности.
5. Вычислить передаточное число указанных цепных передач

6. Сконструировать цепную передачу с передаточным числом 3.

Задание: полный привод

1. Изучить и законспектировать особенности фрикционных передач.
2. Сконструировать два вида фрикционных передач
3. Изучить и законспектировать общетехнические сведения о редукторах
4. Сконструировать мобильную платформу с полноприводным механизмом

7. Сборка мобильного робота

Задание: согласно разработанного проекта собрать мобильного робота

Практическое занятие 7. Среда программирования учебного робота LabVIEW для МК myRIO

Базовые элементы: окно проекта, VI, лицевая панель, блок-диаграмма. Поточный принцип данных, типы данных. Методы отладки и поиска ошибок. Программные структуры в LabVIEW. Узлы свойств. Структуры данных, массивы, функции работы с массивами, кластеры, определители типа. Модульность в программировании. Переменные, очереди. Программирование последовательностей, конечный автомат

Задание 1: Необходимо собрать конечный автомат, который реализует следующие действия:

1 – инициализация: Инициализируется массив из Восьми булевых индикаторов.

2 – выполнение: Выполняется гирлянда от 0-ого элемента к 7-ому (перед включением следующего элемента, выключается предыдущий)



Необходимые элементы:

1. Delete from array
- 1 Insert into array
- 2 For loop
- 3 Wait (ms)
- 4 While Loop
- 5 Shift Register
- 6 Decrement
- 7 Increment

3 – закрытие: Гаснут все «огоньки»

Основы работы с myRIO.

Основы работы с myRIO начало работы с myRIO. Подключение по Wi-Fi. Подключение периферии: инфракрасный датчик расстояния Sharp. ультразвуковой датчик расстояния Parallax PING, сервопривод фиксированного вращения, сервопривод постоянного вращения, двигатель Tetrax DC Motor, драйвер Digilent Motor Adapter, драйвер Studica MXP-MD1, инкрементный датчик угла поворота. Работа с джойстиком. Гироскоп 3-Axis L3G4200D, датчиком линии QTI Sensor, вебкамерой LIFECAM CINEMA, внешним акселерометром MMA7455, датчиком внешнего освещения Digilent PmodALS.

Что такое myRIO? Учебный прибор разработчика NI MyRIO был создан чтобы студенты могли в течение одного семестра решать "настоящие" инженерные задачи. Он содержит двухъядерный программируемый процессор ARM Cortex-A9 с тактовой частотой 667 МГц. И кастомизируемую программируемую логическую интегральную схему (ПЛИС) Xilinx, которую студенты смогут использовать для начала разработки систем и быстрого решения встающих перед ними проблем разработчика, в компактном, простом и красивом форм-факторе. NI MyRIO содержит программируемый чип Zynq-7010, позволяющий в полную силу использовать возможности LabVIEW, как для приложений реального времени, так и для ПЛИС. Вместо долгих

часов, затраченных на отладку кода или разработку пользовательского интерфейса, студенты смогут воспользоваться графическим подходом LabVIEW к программированию и сконцентрироваться на разработке, без дополнительного давления необходимости изучить среду разработки.

NI MyRIO — реконфигурируемый инструмент, подходящий для многократного использования. С его помощью студенты могут освоить широкий круг инженерных концепций и даже полный цикл разработки. Возможности работы с ПЛИС и реальным временем, а так же встроенный Wi-Fi модуль позволяют запускать приложения удаленно и без подключения к компьютеру. Три разъема (2 порта расширения NI MyRIO (MXP) и один порт NI miniSystems (MSP), идентичный разъему NI myDAQ) передают и получают



сигналы от датчиков и электрических схем, используемых студентами в их системах. NI MyRIO содержит в общей сложности 40 цифровых линий ввода/вывода с поддержкой SPI, PWN выхода, входного импульсного датчика, UART и I²C; восемь односторонних аналоговых входов; два дифференциальных аналоговых входа; четыре односторонних аналоговых выхода; и два общих аналоговых выхода, позволяющие подключать бесчисленное количество сенсоров, устройств и программируемых контроллеров системы. Вся необходимая функциональность встроена и предварительно настроена в базовом функционале ПЛИС, что устраняет необходимость в платах расширения или "щитах" для добавления нужных возможностей. В конечном счете, NI MyRIO позволит студентам решать инженерные задачи реального времени прямо сейчас — от радиоуправляемых машин до одиночных медицинских приборов.

Легкий в освоении, и обладающий полной функциональностью в базовой поставке NI MyRIO прост для запуска, а студенты легко могут определять его рабочий статус. Полностью настроенный ПЛИС внедрен в устройство ещё на стадии производства, так что новички могут перейти к функциональным основам без программирования ПЛИС для нормального функционирования системы. Тем не менее мощь ПЛИС становится очевидной, когда студенты дойдут до работы с ПЛИС и настройки поведения устройства под нужды текущей задачи. С учетом гибких возможностей устройства, студенты могут использовать его в течение всего года, начиная с введения во встраиваемые устройства, закачивая курсами разработки в конце года.

Познакомившись ранее с LabVIEW, читатель понимает, что LabVIEW это графический, работающий на потоке данных, ситуационный язык программирования, многоцелевой и мультиплатформенный. Помимо этого он обеспечивает гибкость объектно-ориентированного программирования, многопоточность и параллельность. Виртуальные инструменты LabVIEW могут использоваться в задачах реального времени и ПЛИС.

Диалог LabVIEW Getting Started

После успешного запуска LabVIEW и загрузки всех необходимых ресурсов, открывается диалог Getting Started.

Если чекбокс "Show on lunch" отмечен, то появится и диалог Set Up and Explore. Для прибора NI MyRIO этот диалог скомпонован таким образом, чтобы отвечать нуждам студентов, работающих с NI MyRIO. С его помощью они смогут запустить Getting Started Wizard для NI MyRIO, почти так же, как это делает автозапуск при подключении устройства по USB, получить доступ к руководствам по началу работы, справочным материалам LabVIEW, и настроить Wi-Fi на устройстве NI MyRIO. Но пока что, закроем этот диалог.

С помощью диалога Getting Started, вы можете создать новые проекты и VI, или выбрать одну из множества опций продолжения работы над уже существующими проектами, или найти подходящий демонстрационный VI с помощью NI Example Finder. Вы можете открыть этот диалог когда угодно с помощью меню **LabVIEW»Getting Started Window** в любом VI. В нижней

части окна доступно ещё несколько опций. Каждая из них ведет к полезной информации, обучающим материалам, упражнениями, и другими средствами поддержки изысканий с NI MyRIO



Рис. 1 Диалог Getting Started

Частью философии NI MyRIO является создание моста между реальной проблемой при минимально возможном количестве промежуточных проблем. Избегайте опасностей "пустого VI" (или подавленности от вида чистой блок-схемы) находя с его помощью, шаблоны и примеры, или воспользуйтесь помощью интернет-форума или технической поддержки. А если дело доходит до простых систем или задач — многие пользователи или сотрудники NI уже создали что-то, решающее в точности нужную задачу или нечто очень близкое. Изобретать велосипед заново может быть тяжело, но благодаря мощности и гибкости LabVIEW и широкому кругу ученых и инженеров, работающих с ним, вы можете избежать этого.

Браузер проектов LabVIEW

Браузер проектов LabVIEW (Project Explorer) помогает вам контролировать связанные с приложением ресурсы. Они могут включать в себя несколько VI, изображения, текстовые документы, информация о конфигурации, сборке и внедрении. Структура проекта позволяет легко и быстро управлять ресурсами, вы можете распределять ресурсы по нескольким носителям.

Тем, кто уже знаком с другими интегрированными средами разработки (IDE), браузер проектов LabVIEW покажется знакомым, а принцип его использования по большому счету такой же. LabVIEW — это кросс-платформенный язык, с всеобъемлющей поддержкой всевозможных дополнений и библиотек. Вы можете написать большинство VI в то время, как закрепляете их целевыми устройствами, и легко переназначить их на другие устройства NI, такие как NI MyRIO, поскольку они управляются операционной системой, которая тоже работает с LabVIEW и дополнительными библиотеками. Иерархия проекта в браузер проектов LabVIEW не зависит от файлов с исходным кодом на компьютере. Обычно, проект создается в определенной директории, и все сопутствующие файлы с кодом находятся в той же директории без какой-либо систематизации, даже если отображение в браузере проектов хорошо структурировано. Перемещение файла в браузере проектов не затронет сам файл на диске — только его отображение.

Задание 2: Воспользуйтесь этой инструкцией для создания нового проекта:

1. В диалоге LabVIEW Getting Started, нажмите кнопку **Create a New Project**.
2. Откроется окно Create Project, предлагая несколько опций. Исследуйте все возможные опции либо создав проект с их помощью, либо воспользуйтесь ссылкой *More Information* для интересующего вас шаблона.
3. Выберите опцию **Blank Project**.
4. Нажмите кнопку Finish. Вы создали новый пустой проект. Откроется окно браузера проектов.

Чтобы сохранить новый проект:

1. В браузере проектов LabVIEW выберите пункт **File»Save**.

2. Откроется диалог сохранения, запрашивая имя проекта и директорию для его сохранения. Выберите подходящую директорию, и сохраните проект под именем, позволяющем идентифицировать проект и его назначение.

Теперь браузер проектов открыт и готов к заполнению ресурсами и исходным кодом. Меню браузера проектов содержит набор стандартных опций (File, Edit, View и т.д.) и несколько функций специфичных для LabVIEW. Браузер содержит две вкладки — Items и Files. Вкладка Items отображает сущности, используемые в проекте в иерархическом виде, а вкладка Files показывает файлы на диске, задействованные в проекте.

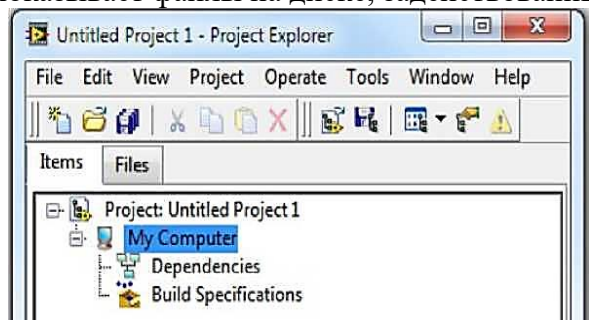


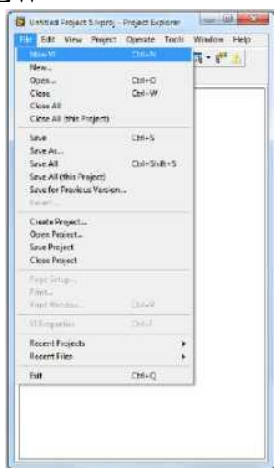
Рис 2. Браузер проектов LabVIEW.

На вкладке Items вы можете увидеть структуру проекта. Корень проекта, первая сущность в списке, показывает, над каким проектом вы сейчас работаете. Вторым уровнем находятся целевые устройства проекта. В пустом проекте целевым устройством по умолчанию является "мой компьютер". Следующим уровнем являются характеристики сборки целевого устройства. Они включают в себя конфигурации для исходников и других наборов инструментов и модулей, содержащихся в LabVIEW. Вы можете использовать их при внедрении приложений на встраиваемых системах. Хотя этот курс рассматривает простые VI, вам потребуется понимание работы с браузером проектов — это потребуется при разработке более сложных проектов. Пока наш проект слишком пуст, и не служит никаким целям. Для более подробного ознакомления с программированием в LabVIEW. Не закрывайте проект, чтобы использовать его в дальнейшей

Задание 3: Создание виртуального инструмента VI в LabVIEW в Windows

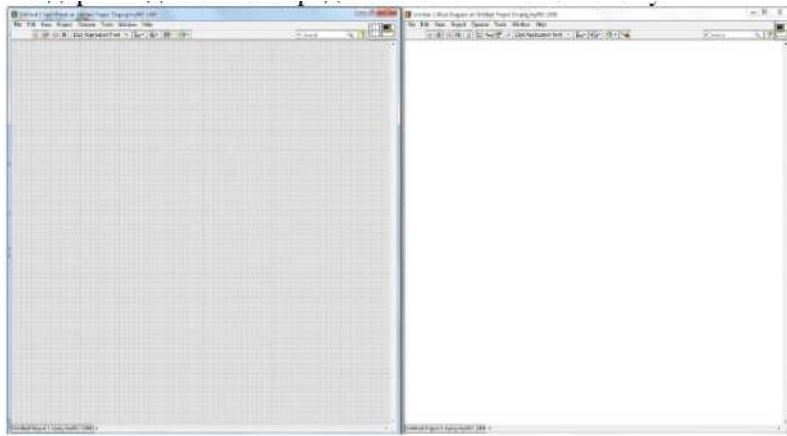
Чтобы познакомиться с графической средой разработки LabVIEW, создайте базовый код, иллюстрирующий некоторые свойства программирования потока данных. В этом упражнении мы создадим простой VI, конвертирующий температуру из шкалы Фаренгейта в шкалу Цельсия.

Для начала надо создать новый VI, с помощью меню **File»New VI** в браузере проектов LabVIEW



Создание пустого VI

Эта операция создаст пустой VI в целевом устройстве My Computer в проекте, который вы только что создали. Он задержит два окна: переднюю панель и блок-схему.



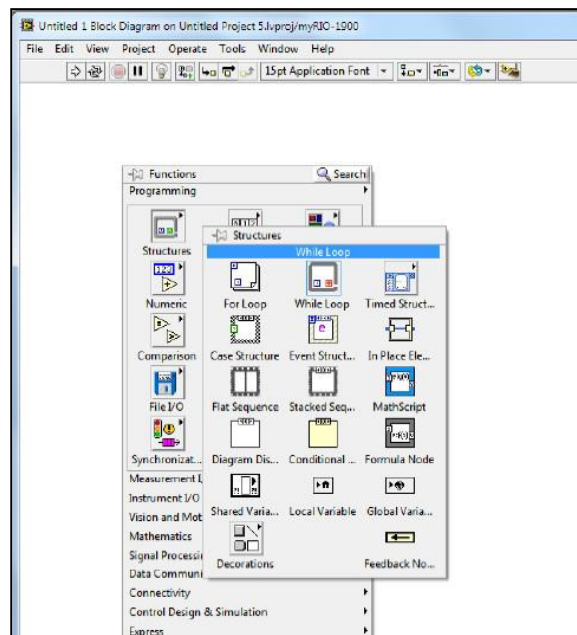
Сохраните VI, выбрав команду **File»Save** в любом из двух окон. В диалоге сохранения задайте для VI имя со смыслом. Сохраняйте VI в той же директории, что и проект, к в котором они используются. Вы можете сохранять VI, которые планируете использовать в нескольких проектах, в других директориях. Браузер проектов будет отслеживать их и предупредит об отсутствующем или неправильно расположенном VI

Теперь, когда вы сохранили VI, вы можете создать код, конвертирующий температуру. Для этого:

1. Этот код выполняется непрерывно, и как в текстовых языках программирования, LabVIEW использует заикливание для обеспечения непрерывной работы кода. Вы можете остановить выполнение либо с помощью кнопки на передней панели, либо воспользовавшись традиционной логикой, чтобы задать точку, где выполнение должно быть остановлено. В этом упражнении используйте цикл While, для постоянного отслеживания температуры по шкале Фаренгейта и непрерывного преобразования в шкалу Цельсия и кнопку на передней панели для остановки.

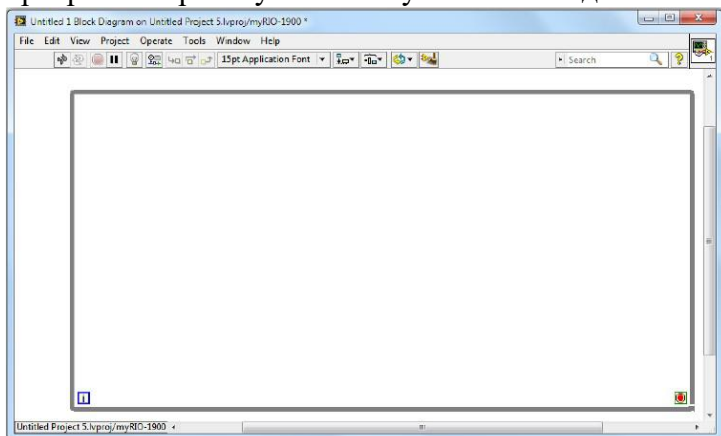
А. Переключитесь на окно блок-схемы, либо стандартными средствами Windows, либо воспользовавшись сочетанием клавиш Ctrl+E (это сочетание позволяет быстро переключаться между передней панелью и блок-схемой).

В. Щелкните правой кнопкой по блок-схеме, чтобы открыть палитру Functions. Она содержит все узлы и VI, используемые при программировании в LabVIEW. Вы можете настроить её отображение, но по умолчанию доступны все дополнения и VI. Ознакомьтесь с расположением всех подпанелей. Выберите пункт Programming»Structures»While Loop и выберите цикл While.



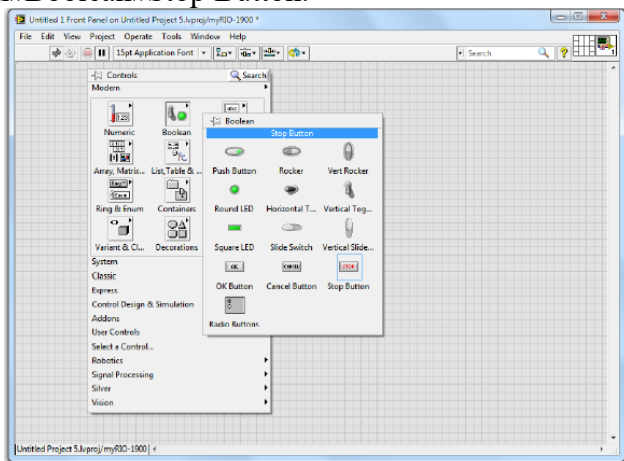
Выбор цикла While

С. Нарисуйте петлю в блок-схеме. Кликните левой кнопкой мыши, растяните окно на нужную площадь и кликните ещё раз для завершения. Вы можете изменять размеры петли, наведя курсор на границу петли, кликнув левой кнопкой мыши, схватив мышью один из восьми синих маркеров и перетянув его на нужное место для изменений размера.



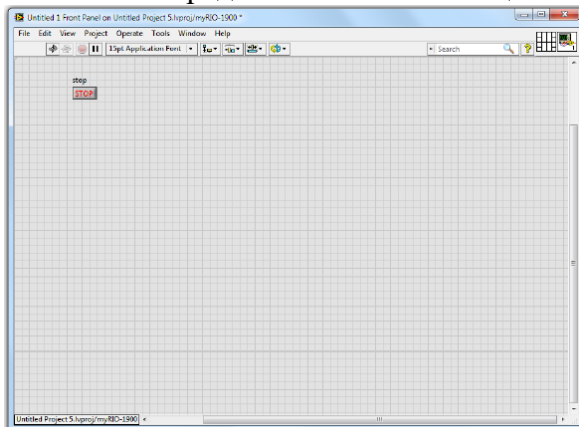
Блок-схема с циклом While

Д. Переключитесь на окно передней панели и щелкните правой кнопкой мыши для открытия палитры Controls. Она содержит все необходимые элементы для создания внешнего интерфейса и элементов управления, имитирующих реальный инструмент. Выберите пункт Modern»Boolean»Stop Button.



Выбор кнопки Stop

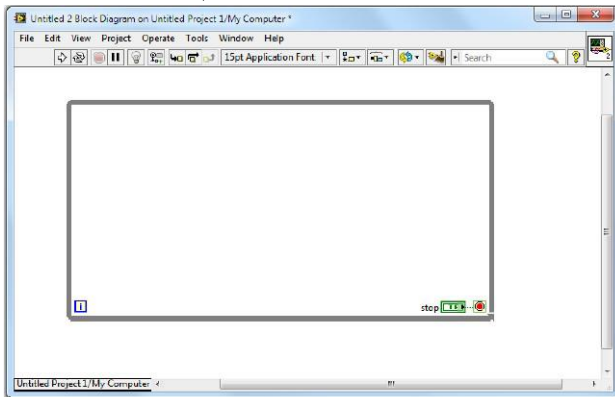
Е. Выберите кнопку Stop с помощью левого щелчка мышью и перетащите её на нужное место на передней панели. Вы можете отслеживать положение контроллера, который собираетесь расположить по его контуру, отображаемому на передней панели. Поместите кнопку в любом месте на передней панели с помощью левого щелчка.



Передняя панель с кнопкой Stop

Ф. Теперь переключитесь обратно на блок-схему и найдите на ней новую кнопку Stop. Если она находится вне цикла While, перетащите её туда с помощью мыши. При наведении курсора на кнопку подсветится её выходной разъем (зелёный кружочек с правой стороны кнопки). Наведите курсор на кружочек, и он примет вид мотка провода.

Г. Соедините кнопку Stop и иконку состояния выхода из цикла в правом нижнем углу поля цикла, кликнув по разъему кнопки и подтянув провод к разъему состояния выхода из цикла (тень провода будет отображаться при протягивании). Для закрепления, щелкните по разъему состояния выхода из цикла. Между двумя элементами должен появиться зеленый проводок. Это означает два правильно соединенных компонента логического (boolean) типа. Другие типы данных имеют свои цвета.



Цикл While с подключенным условием выхода

2. Теперь создадим контроллер для значений по Фаренгейту и индикатор для значений по Цельсию.

А. Переключитесь на переднюю панель, и щелкните правой кнопкой мыши для открытия палитры Controls. Выберите пункт Modern»Numeric»Numeric Control.

В. Перетащите контроллер на переднюю панель.

С. Обратите внимание на маркировку Numeric Control у контроллера. Не очень информативное имя. Переименуйте контроллер, двойным щелчком по нему и напечатайте новое имя — Fahrenheit.

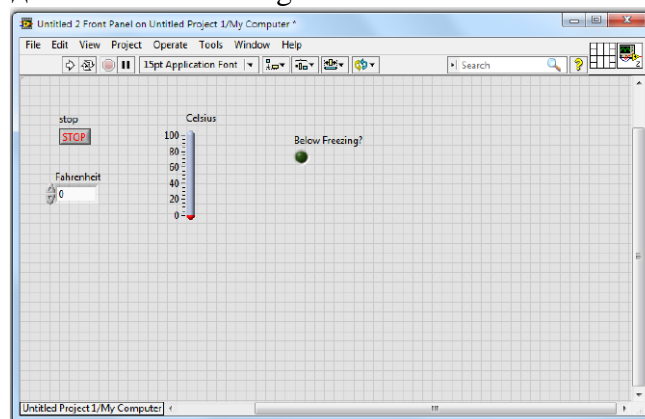
Д. На передней панели, щелкните правой кнопкой мыши для открытия палитры Controls. Выберите пункт Modern»Numeric»Thermometer. Поместите термометр на переднюю панель.

Е. Как и для контроллера Fahrenheit, переименуйте индикатор в Celsius.

Ф. Добавим функциональности, разместив на передней панели виртуальный светодиод, загорающийся, когда температура падает ниже нуля по Цельсию. Светодиод можно найти в палитре Controls, пункт Modern»Boolean»Round LED.

Г. Поместите светодиод на переднюю панель.

Н. Переименуйте светодиод — "Below Freezing"



Передняя панель с контроллером и индикаторами

3. Теперь, когда контроллеры и индикаторы расположены на передней панели, нужно добавить в проект необходимый код и логику для конвертации температуры из шкалы Фаренгейта в шкалу Цельсия.

А. Переключитесь на блок-схему и найдите иконки для контроллеров и индикаторов Fahrenheit, Celsius и Below Freezing. Перетащите иконку Fahrenheit в левую часть цикла While, а иконки Celsius и Below Freezing — в правую. Оставьте между иконками достаточно места для написания кода.

В. Поместите вычитающий модуль на блок-схеме. Щелкните правой кнопкой для открытия палитры Functions и выберите пункт Programming»Numeric»Subtract. Поместите модуль сразу справа от иконки Fahrenheit.

С. Соедините иконку Fahrenheit и верхний разъем вычитающего модуля.

Д. Щелкните правой кнопкой по нижнему разъему вычитающего модуля и добавьте константу, выбрав пункт Create»Constant.

Е. Задайте для константы значение 32.

Ф. Поместите умножающий модуль на блок-схеме. Щелкните правой кнопкой для открытия палитры Functions и выберите пункт Programming»Numeric»Multiply. Поместите модуль сразу справа от вычитающего. G. Соедините выходной разъем вычитающего модуля и верхний разъем умножающего.

Н. Поместите модуль для деления на блок-схеме. Щелкните правой кнопкой для открытия палитры Functions и выберите пункт Programming»Numeric»Divide. Поместите модуль снизу от вычитающего узла.

И. Щелкните правой кнопкой по верхнему разъему модуля для деления и добавьте константу, выбрав пункт Create»Constant. Задайте ей значение 5.

Ж. Щелкните правой кнопкой по нижнему разъему модуля для деления и добавьте константу, выбрав пункт Create»Constant. Задайте ей значение 9.

К. Соедините выходной разъем модуля для деления с нижним разъемом умножающего модуля.

Л. Соедините выходной разъем умножающего модуля с входным разъемом индикатора Celsius.

М. Добавьте сравнительный модуль "меньше нуля?" на блок-схему щелкнув правой кнопкой и выбрав в палитре Programming»Comparison»Less Than Zero? Поместите модуль справа от умножающего.

Н. Соедините выходной разъем умножающего модуля с модулем "меньше нуля?"

О. Соедините выходной разъем модуля "меньше нуля?" с индикатором Below Freezing?

4. Теперь, когда создан весь необходимый код, запустите VI. Убедитесь, что все соединения выполнены верно, сравнив свой VI с рисунком ниже и проверив, активна ли кнопка Run.

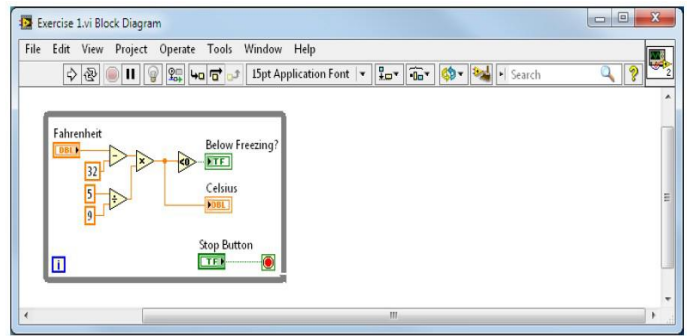
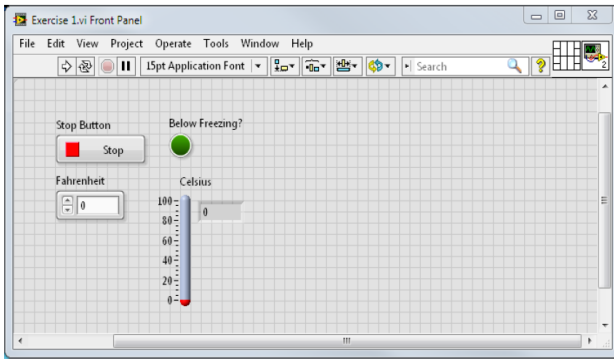
А. Переключитесь на переднюю панель (можете воспользоваться сочетанием клавиш Ctrl+E). Убедитесь, что все контроллеры и индикаторы присутствуют на экране.

В. Щелкните кнопку Run на приборной панели для запуска программы.

С. По умолчанию термометр показывает значения только в промежутке от 0 до 100. Вы можете изменить диапазон, дважды щелкнув по шкале и задав нужные значения. В этом упражнении, вводите значения температуры по шкале Фаренгейта и наблюдайте их преобразования на шкале Цельсия.

Д. По окончании, прервите выполнение с помощью кнопки Stop.

5. Сохраните и закройте VI и проект преобразования температуры. Выполнив это упражнение, вы освоитесь с графическим подходом к программированию. Кросс-платформенная модель LabVIEW делает разработку приложений для многих поддерживаемых устройств. В следующих упражнениях мы будем использовать ту же среду для создания программ для устройства NI MyRIO, с такой же структурой, контроллерами, индикаторами и функциональностью как в примере, который мы только что рассмотрели. Тем не менее, для начала вам надо установить и сконфигурировать устройство для работы с компьютером на котором ведется разработка. Инструментарий, поставляемый с NI MyRIO помогает сделать этот процесс легким и простым. Следующая глава описывает настройку устройства и мастер Getting Started для устройства NI MyRIO. VI из упражнения 2.



Настройка оборудования: подключения устройства NI MyRIO

Одна из целей разработки NI MyRIO — упростить настройку оборудования. Для достижения этой цели программное обеспечение NI MyRIO имеет собственную утилиту для подключения и конфигурирования, отличную от браузера измерений и автоматизации NI (NI MAX). Вы по-прежнему можете использовать MAX для установки и настройки оборудования и программного обеспечения, если эта среда кажется вам более комфортной. Устройство NI MyRIO имеет приложения USB мониторинга, которое запускается, когда вы подключаете устройство к компьютеру. В ближайших разделах мы изучим как использовать NI MyRIO USB монитор и мастер Getting Started.

NI MyRIO USB монитор

Убедитесь, что вы запитываете NI MyRIO с помощью кабеля, идущего в комплекте поставки. Подключите к устройству кабель USB тип B. Другой конец кабеля подключите к USB-порту компьютера.

Если устройство включено, то без запуска LabVIEW или NI MAX, операционная система распознает устройство NI MyRIO, и установит необходимые драйверы. По завершении установки, в операционной системе Windows откроется окно NI MyRIO USB монитора, показанное на рисунке ниже.



Окно NI MyRIO USB монитора

Вы можете выбрать одну из четырех доступных опций:

1. Запустить мастер Getting Started

С его помощью вы сможете быстро узнать функциональный статус устройства NI MyRIO. Он проверяет подключенные устройства NI MyRIO, подключается к выбранному устройству, проверяет актуальность программного обеспечения, в случае необходимости предлагает обновить программного обеспечение, может предложить переименовать устройство, и затем отображает экран, аналогичный передней панели NI MyRIO, который вы можете использовать для наблюдения за функцией акселерометра, включать и выключать встроенные светодиоды и тестировать кнопку с пользовательской настройкой функциональности.

Последний экран мастера имеет две опции:

Запустить мой первый проект (Start my first project now)

Эта опция запускает веб-инструктаж, аналогичный упражнению 2 в этом руководстве.

Запустить LabVIEW (Go Straight to LabVIEW)

Эта опция запускает диалог LabVIEW Getting Started.

2. Перейти к LabVIEW (Go to LabVIEW)

Эта опция запускает диалог LabVIEW Getting Started.

3. Настроить NI MyRIO (Configure NI MyRIO)

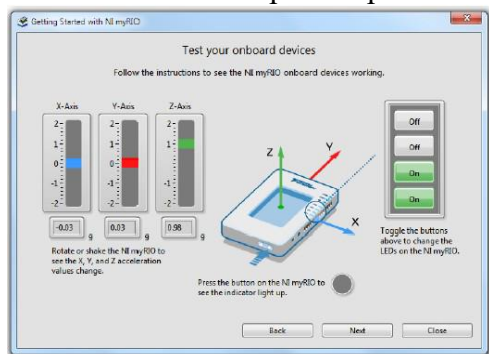
Эта опция открывает настроенную утилиту NI MyRIO с веб-интерфейсом.

4. Ничего не делать (Do nothing)

Если LabVIEW уже открыт и проект сконфигурирован, с NI MyRIO в качестве целевого устройства, вы можете использовать эту опцию чтобы закрыть NI MyRIO USB монитор, когда модуль был отключен и вновь подключен к компьютеру.

Запуск мастера Getting Started

В окне USB монитора NI MyRIO, выберите **Launch the Getting Started Wizard**. Выберите Next и мастер подключится к устройству NI MyRIO, проверит его программное обеспечение и предложит переименовать устройство. Если модуль NI MyRIO не имеет установленного программного обеспечения, мастер автоматически установит последнюю версию ПО. После этого откроется окно диагностики. В нем вы можете просмотреть показания встроенного трехосного акселерометра, проверить назначение кнопки с пользовательской настройкой функциональности и четырех встроенных светодиодов.



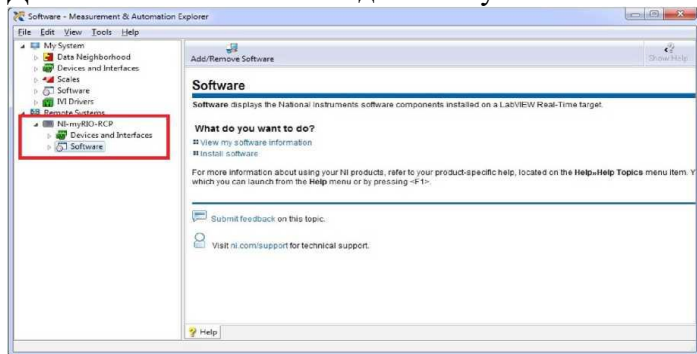
Мастер Getting Started

Теперь, когда вы установили и настроили устройство NI MyRIO, вы можете создавать VI реального времени, и выполнять их в процессоре (аналогично с Windows VI), а так же ПЛИС VI, для знакомства с возможностями параллельной обработки. Затем, создайте обычный проект, и исследуйте функциональность базовой архитектуры.

Установка программного обеспечения через NI MAX

Другой вариант установки программного ПО, более профессиональный подход – установка программного обеспечения через NI MAX (Measurement & Automation Explorer).

Для этого сначала необходимо запустить Ni MAX



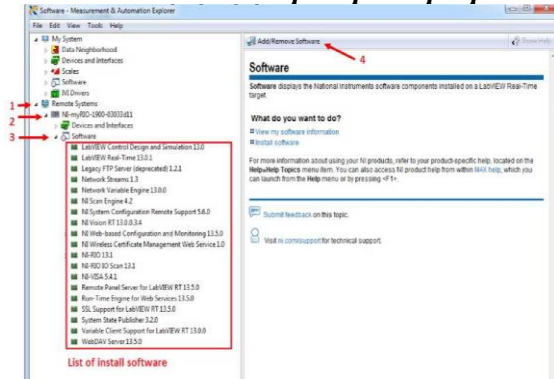
NI Max

Если категория Software пустая, необходимо установить ПО. В комплекте поставки устройства myRIO идет два DVD диска. Установите с первого DVD обязательно 3 модуля, без которых вы не сможете работать с myRIO:

- LabVIEW
- LabVIEW Real-Time
- myRIO Toolkit

Второй DVD содержит ПЛИС, установите его, если вы планируете разрабатывать свою собственную прошивку под ПЛИС. Как только все ПО будет установлено, вы будете готовы подключить myRIO и начать работу.

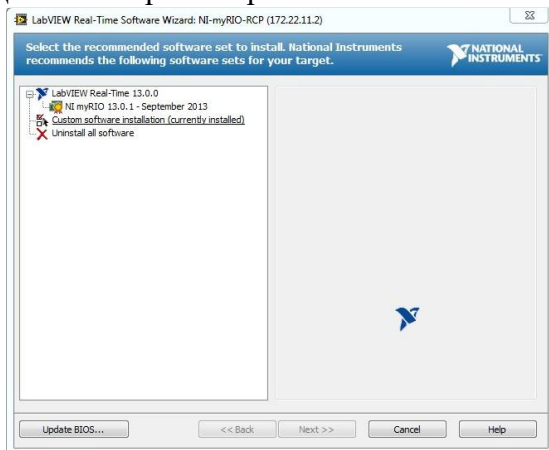
Проверка программного обеспечения (ПО)



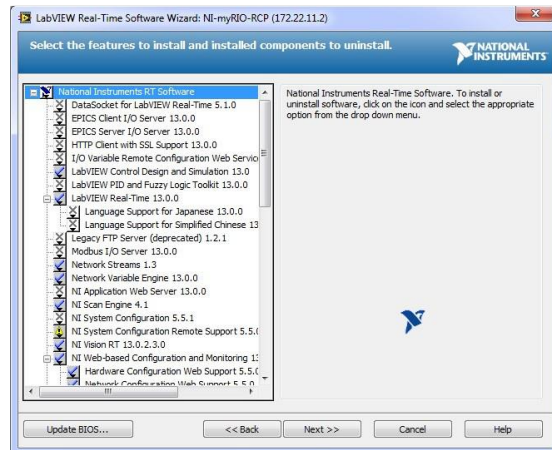
Проверка программного обеспечения

1. Устройство myRIO, подключенное к ПК
2. Перечень доступных интерфейсов и подключенных устройств к myRIO
3. Список программных пакетов на устройстве
4. Установка\удаление ПО на устройстве

Если необходимо пополнить список установленного ПО на устройстве или программное обеспечение отсутствует полностью необходимо выбрать Установка\удаление ПО на устройстве. Далее выбираем версию



Установка программного обеспечения



Выбор пакетов

Установка необходимого ПО. Выбираем пакеты. После выбора необходимых компонентов ПО переходим к самой установке

Создание беспроводной сети при помощи NI myRIO

Вы можете использовать NI myRIO как точку доступа для создания беспроводных сетей. После того, как вы создадите беспроводную сеть при помощи NI myRIO, вы можете подключать сторонние устройства к данной беспроводной сети.

Вы можете настроить создание беспроводной сети при помощи NI myRIO в любом из следующих приложений:

- NI Web-based Configuration & Monitoring
- Measurement & Automation Explorer (MAX)

В данном разделе используется Measurement & Automation Explorer (MAX) как пример.

Выполните следующие шаги для создания беспроводной сети при помощи NI myRIO:

1. Используйте USB кабель для подключения NI myRIO к вашему компьютеру.
2. Откройте Measurement & Automation Explorer
3. Раскройте элемент Remote Systems, выберите ваш NI myRIO
4. Перейдите на вкладку Network Settings.
5. В разделе Wireless Adapter wlan0, выберите Create wireless network из выпадающего списка Wireless Mode.
6. Выберите страну, в которой вы находитесь из выпадающего списка Country.
7. Обозначьте название вашей беспроводной сети в текстовом поле SSID.
8. (Опционально) Выберите канал из выпадающего списка Channel. Возможные значения Channel находятся в диапазоне [1, 11]. National Instruments рекомендует выбрать

значение 1.

9. Выберите одно из следующих значений в выпадающем списке Security:
10. Open—Создает открытую беспроводную сеть.
11. WPA2 Personal—Создает закрытую беспроводную сеть.
12. Если вы выбрали WPA2 Personal в предыдущем шаге, введите пароль в поле WPA Passphrase. Убедитесь, что длина символов в вашем пароле находится в диапазоне [8, 63]. Вы можете включить пункт Show characters для отображения пароля.
13. Выберите DHCP Only из выпадающего списка Configure IPv4 Address.
14. Нажмите Save для того, чтобы сохранить настройки беспроводной сети.
15. Убедитесь, что вы можете подключить пользователей к беспроводной сети, которую вы создали. Индикатор Wireless на контроллере NI myRIO загорится красным цветом, когда вы попытаетесь подключить пользователя к сети. После того, как вы успешно подключили пользователя к сети, индикатор Wireless продолжит моргать.

Выполнение кода реального времени на устройстве NI MyRIO

Теперь запустите проект NI MyRIO из шаблонов, входящих в LabVIEW 2013 для NI MyRIO. После создания проекта, откройте файл "Main.vi" и внимательно изучите его. затем выполните упражнение по его функциональности.

Код реального времени выполняется на процессоре, встроенном в NI MyRIO. Он может принимать и отправлять данные через ПЛИС с помощью его узлов входа/выхода, входов/выходов с прямым доступом к памяти (DMA) и экспресс-VI, которые используют "сущность" ПЛИС. Входы и выходы разъемов NI MyRIO и порт NIminiSystems взаимодействуют с процессором через ПЛИС. Мы подробно рассмотрим ПЛИС позднее, пока что запомните, что NI MyRIO поставляется с модулем ПЛИС, который по умолчанию настроен на прием и передачу данных между разъемами и встроенными устройствами (кнопки, акселерометр) и процессором, выполняющим код реального времени.

Теперь, когда вы подключили и настроили устройство NI MyRIO, вы можете создать проект для него и начать разработку кода. Использование ПЛИС по умолчанию — это быстрый способ начать и создать простой автономный код или даже код проверки концепции для более сложных проектов, требующих использования ПЛИС. Для кода, требующего кастомизации способа, которым NI MyRIO управляет входом/выходом, требуется собственный ПЛИС-проект, чтобы обеспечить наилучшую производительность кода. Повторимся, что для простых проектов пользовательская настройка ПЛИС не требуется, так что в этой главе мы рассматриваем стандартный ПЛИС-проект

Задание 4: Исследование виртуального инструмента Main.vi

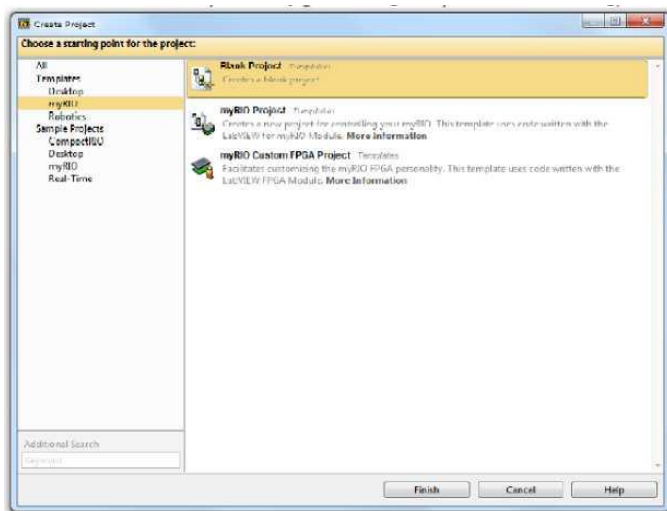
Следуйте этой инструкции, чтобы начать NI MyRIO проект из шаблона:

1. В диалоге LabVIEW Getting Started, выберите кнопку **Create Project**
2. В левой части диалога "Create Project", в дереве шаблонов выберите пункт **myRIO**
3. В правой части доступны три опции: стандартный пустой проект (Blank Project), проект myRIO (myRIO Project) и пользовательский myRIO ПЛИС проект (Custom myRIO FPGA Project). Стандартный пустой проект мы рассматривали в упражнении 1.

а. Используйте шаблон проекта myRIO для создания проекта с стандартной работой ПЛИС. Он используется в проектах, не требующих расширенной функциональности и дополнительной настройки ПЛИС



Создание проекта

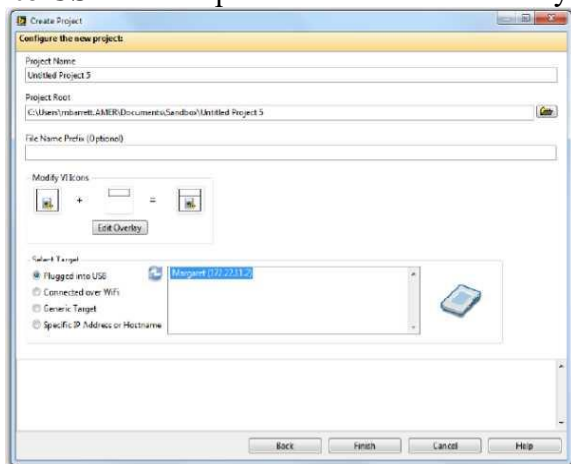


Шаблоны проектов NI MyRIO

б. Пользовательский ПЛИС проект используется для самостоятельной настройки ПЛИС в устройстве NI MyRIO. Например, объединение всех трех разъемов дает восемь цифровых линий ШИМ-входа/выхода. Но, если вам нужно подключить большей таких устройств или сигналов, вы можете настроить ПЛИС таким образом, чтобы поддержку ШИМ имело большее количество линий цифрового входа/выхода. Так же можно поступить с другими протоколами взаимодействия.

4. Выберите пункт **myRIO Project** и нажмите кнопку **Next**

5. Задайте проекту имя со смыслом, позволяющее идентифицировать его в дальнейшем, и сохраните его в подходящей директории. Убедитесь, что выбрана селективная кнопка **Plugged into USB** и что правильно озаглавленное устройство NI MyRIO отображается в листе справа.



Настройка LabVIEW проекта

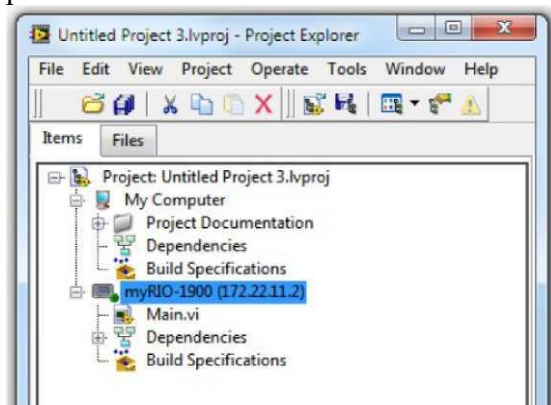
б. Если все настроено верно, нажмите кнопку **Finish**.

Этот шаблон задает NI MyRIO в качестве целевого устройства для проекта LabVIEW. Следует помнить очень важную вещь в концепции целевого устройства — если NI MyRIO

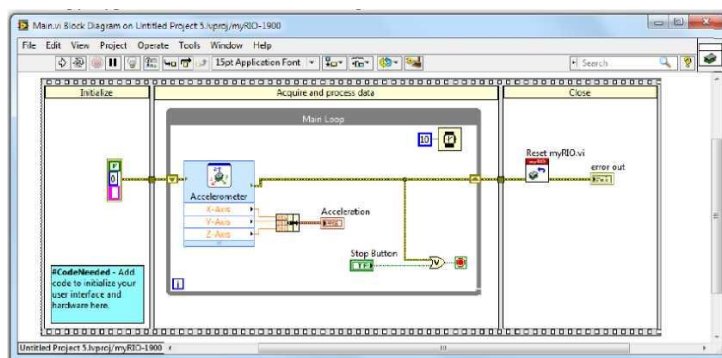
выбрано в качестве целевого устройства для VI, то код VI будет выполняться на нём, даже если передняя панель VI отображается на компьютере. Такой режим работы называется *режим интерактивной передней панели* (interactive front panel mode) и предназначен только для разработки и отладки. В конечном приложении любые контроллеры и индикаторы с передней панели VI будут недоступны. В конечном итоге, код развертывается на самом устройстве и работает даже без подключения к компьютеру через USB. Вы можете использовать сетевые совместные переменные или какую-либо форму сетевого потока в VI на компьютере для получения и обработки данных с устройства NI MyRIO. Теперь возможности становятся действительно впечатляющими. Даже без перенастройки ПЛИС вы можете собирать данные и принимать решения по управлению прямо в процессоре реального времени устройства NI MyRIO. Пока что, компьютер, на котором ведется разработка VI будем использовать для сохранения и/или анализа данных и даже для отправки управляющих воздействий высокого уровня на устройство NI MyRIO.

В этом упражнении, NI MyRIO привязан к компьютеру, но вы все равно можете захотеть выполнить полученный код на его процессоре. Это значит, вы сможете взаимодействовать с передней панелью VI, выполняющегося на процессоре реального времени, в то время как движок NI заботится о сетевой передаче данных. Для некоторых задач тестирования и настройки режим интерактивной передней панели вполне адекватен и устраняет необходимость установки более сложного обмена данным между хостом и целевым устройством реального времени.

В браузере проектов LabVIEW теперь отображаются два целевых устройства — "Мой компьютер" и "myRIO-1900 (xxx.xx.xx.x)". Целевое устройство NI MyRIO уже содержит в проекте VI под названием "Main.vi". Этот VI содержит код для того, чтобы облегчить вам начало работы.



Иерархия проекта LabVIEW



Структура последовательности

Практическое занятие 6. Программное управление мобильной платформой

Задание:

1. Повторить/изучить основы структурированного программирования и базовые алгоритмические конструкции (ветвление, циклы)
2. Изучить основы работы с динамиком микроконтроллера и светодиодной индикацией на передней панели корпуса, блоком ожидания
3. Разработать и протестировать программу, смены цвета индикатора через каждые 2 сек.
4. Разработать и протестировать программу, постоянной подачи звукового сигнала с промежутком в 1 сек.
5. Разработать и протестировать программу, постоянной подачи звукового сигнала с промежутком в 1 сек.

Теоретический материал

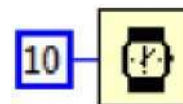
1. Двойным щелчком откройте Main.vi из браузера проектов LabVIEW.
2. Откроется передняя панель VI. Она должна содержать синусоидальный график и кнопку стоп.
3. Перед запуском VI изучите блок-схему. Переключитесь на неё с помощью сочетания клавиш Ctrl+E.

4. Структура, окружающая цикл While и имеющая вид трех кадров с фотопленки называется структурой последовательности (Sequence Structure).

Есть два варианта такой структуры — плоская и сферическая. Структура в Main.vi — плоская. Структура последовательности заводит выполняемый код в последовательность. Весь код из каждого кадра (слева направо) должен быть выполнен перед тем, как сможет начаться исполнение следующего кадра. Вы можете использовать *туннели* (tunnels) для передачи данных между кадрами. Использование последовательной структуры в таком ключе — это простейший способ гарантировать выполнение кода нужным образом, хотя вы можете достичь этого используя технику программирования потока данных, без захода на территорию структур последовательности.

а. Кадр "initialize" будет выполнен первым. Единственное действие, выполняемое в нем — создание кластера для константы ошибки и переход к следующему кадру через туннель.

б. Кадр "acquire and process data" получает кластер ошибки, и начинает выполнение цикла While, названного "Main Loop". В этом цикле экспресс-VI (синий VI, названный "Accelerometer") используется для получения данных с встроенного акселерометра. Затем данные от осей x, y и z передается через выходные разъемы по оранжевым проводам (тип данных с двойной точностью) и соединяются в кластер, для подачи на индикатор, соответствующий синусоидальному графику на передней панели.



Обратите внимание на VI "wait(ms)" в правом верхнем углу.

Он заставляет код цикла выполняться каждые 10 мсек, если код выполнен менее чем за 10 мсек; в противном случае цикл выполняется настолько быстро, насколько это возможно, без дополнительных 10 мсек ожидания. По существу это дает цикл с частотой 100 Гц (опять-таки, если код в цикле способен выполняться настолько быстро). Как только график получит 100 измерений с акселерометра, вы целиком увидите секунду наблюдений. Как и в большинстве структур с программным управлением временем, неизбежны некоторые дрожания. Для задач, требующих более точного отсчета времени, вы можете улучшить его в LabVIEW^С с помощью инструментов Timed Loops и программирования ПЛИС.

Вы можете выбрать два условия остановки основного цикла: кнопка Stop, можете нажать в любое время, чтобы прервать цикл While и продолжить последовательности, или назначить выход из цикла по срабатыванию ошибки, который можно создать подключив кластер ошибки из экспресс-VI "Accelerometer" к модулю ИЛИ с кнопкой стоп. Срабатывание приведет к остановке цикла While и продолжит выполнение последовательности при возникновении любой ошибки в VI, подключенном к модулю. Вы можете комбинировать ошибки от нескольких VI, чтобы создавать более безопасные протоколы выполнения, что позволяет избежать повреждений устройств, подключенных к системе из-за ошибок в выполнении кода или некорректной информации от датчиков.

с. Когда цикл While заканчивает выполнение, создается ошибка и передается сквозь структуру в финальный кадр. Этот кадр используется для закрытия всех переменных и ссылок на устройство NI myRIO перед тем, как программа закроется. Когда вы добавите больше функциональности, вы все равно можете использовать этот кадр для закрытия линий ввода/вывода

и сохранения или удаления данных.

5. Теперь вы изучили структуру кода, и как он ведет себя при исполнении. Переключитесь обратно на переднюю панель, нажав Ctrl+E. Кликните кнопку запуска или нажмите Ctrl+R для запуска этого VI на устройстве NI myRIO.

6. После того, как вы успешно внедрили VI на NI myRIO, график начинает отображать данные с акселерометра. Встряхните NI myRIO чтобы отследить изменение показаний акселерометра в реальном времени.

7. Нажмите кнопку стоп, чтобы позволить VI выйти из цикла While и завершить выполнение структуры.

8. Оставьте VI и проект открытыми.

Настройка проекта для NI myRIO аналогичная настройке любого другого проекта с целевым устройством реального времени. Вы можете переделать любой проект под реальное время, добавив устройство реального времени в качестве целевого. Вы также можете создавать проекты реального времени из шаблонов по описанному в последней главе процессу. Разбор стандартного виртуального инструмента "Main.vi", его архитектура и функциональности теперь не должны вызывать проблем. Вы сможете расширить функциональность существующего процесса (цикла While) или даже добавить другие циклы для параллельной обработки данных. Тем не менее, процессор реального времени может обрабатывать параллельно только столько процессов, сколько у него имеется ядер. При наличии большего количества циклов, процессор будет обрабатывать параллельность аналогично другим языкам — с использованием потоков и переключением между задачами при выполнении.

Задание 5: Создание кода реального времени для исполнения на устройстве NI myRIO

В этом упражнении мы, с помощью шаблона проекта NI myRIO, напишем в LabVIEW виртуальный инструмент для исследования встроенных функций устройства. В нашем случае, приложение будет использовать кнопку и встроенные светодиоды.

В проекте, оставшемся открытым в предыдущем упражнении, выполните нижеприведенные инструкции для создания VI реального времени. Если вы закрыли проект из прошлого упражнения, повторите его для создания проекта NI myRIO.

1. Откройте "Main.vi" под целевым устройством myRIO-1900 (xxx.xx.xx.x) в браузере проектов LabVIEW.

2. Откройте блок-схему.

3. Увеличьте центральный кадр последовательности и цикл While, расположенный внутри.

a. Поместите курсор мыши на нижнюю грань последовательности, для появления синего узла изменения размера. Схватите его и растяните последовательность, обеспечив больше места для иконок.

b. Аналогичным образом увеличьте центральный кадр последовательности, растянув его вправо.

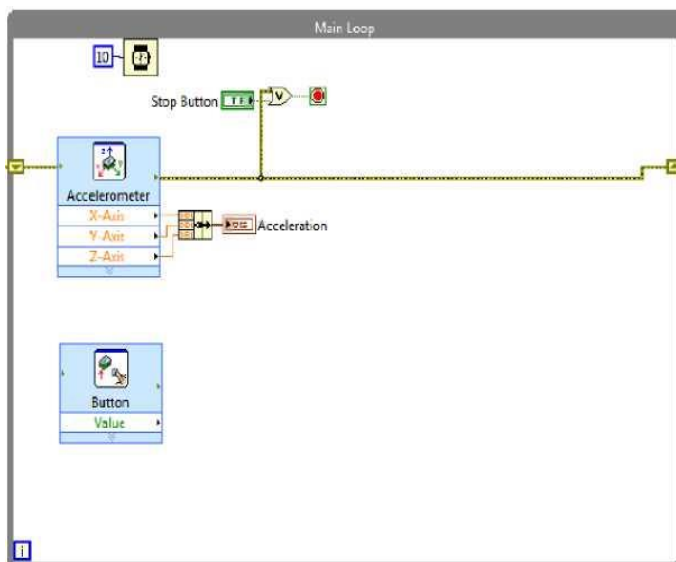
c. Увеличьте площадь цикла While, потянув за правый нижний угол.

4. Создайте триггер нарастающего фронта для детектирования нажатия на кнопку устройства NI myRIO (игнорируя выход, при удержании клавиши).

a. Поместите экспресс-VI Button Express VI на блок-схему из палитры NI myRIO.

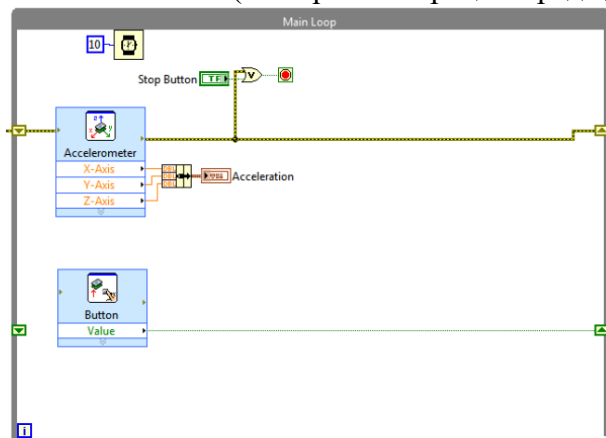
i. Щелкните правой кнопкой мыши и выберите пункт **myRIO»Onboard»Button**. Поместите экспресс-VI внутри цикла While прямо под VI акселерометра.

ii. Нажмите **OK** в диалоге конфигурации оставив стандартные настройки экспресс-VI. Он обрабатывает ссылку на физическое оборудование через стандартные настройки ПЛИС на борту NI myRIO и отслеживает текущее состояние кнопки. Поскольку экспресс-vi находится в теле цикла While, его код выполняется на каждой итерации цикла. Тем не менее, его в его коде имеется функция "smart open", которая позволяет избежать открытия соединения на каждой итерации. Таким образом, после первой итерации кода, экспресс-vi выполняется быстрее, поскольку ему требуется только считывать данные с кнопки



b. Соедините выход экспресс-ви кнопки с правой гранью цикла While.

c. Щелкните правой кнопкой по созданному тоннелю переменной логического типа (Boolean) и выберите **Replace with Shift Register**. Это изменит иконку тоннеля и создаст новую на том же месте с левой стороны цикла While. Два этих тоннеля создают регистр сдвига. Правая часть регистра хранит данные с текущей итерации, в то время как левая — с предыдущей итерации. С помощью этого регистра, вы можете сравнить текущее значение переменной кнопки с прошлым значением. (на первой итерации предыдущее значение по умолчанию FALSE).



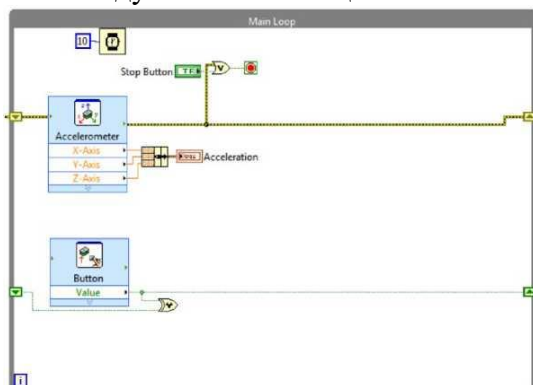
d. Поместите модуль **Exclusive OR** (исключающее ИЛИ) на блок-схему.

i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»Exclusive Or**. Поместите модуль внизу справа от экспресс-ви кнопки.

ii. Подключите разъем Value экспресс-ви кнопки с верхним разъемом модуля исключающего ИЛИ.

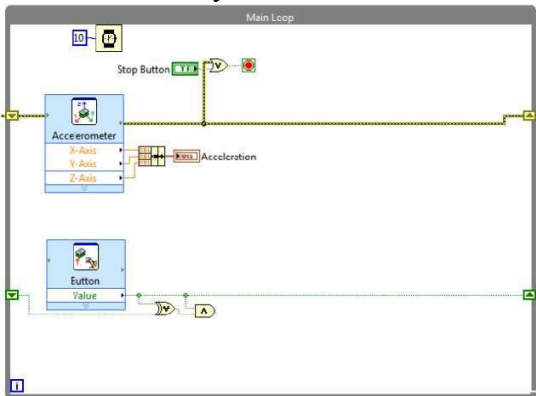
Подключите регистр сдвига, хранящий значение кнопки с нижним разъемом

iii. модуля исключающего ИЛИ



e. Разместите на блок-схеме модуль **And** (И).

- i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»And**. Поместите модуль справа от модуля исключающего ИЛИ.
- ii. Подключите выход экспресс-vi кнопки с верхним разъемом модуля И. Подключите выход модуля исключающего ИЛИ с нижним разъемом модуля И



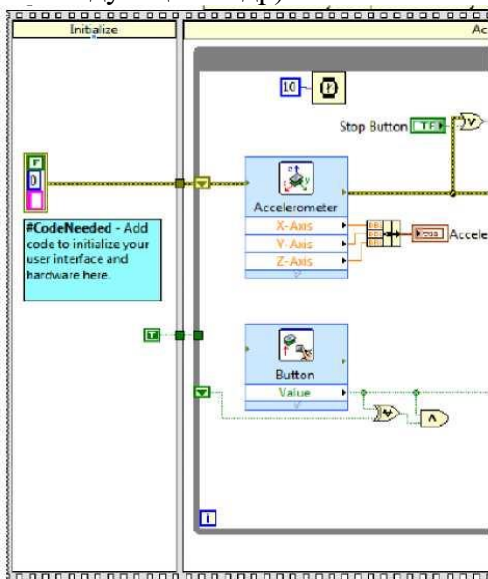
f. Используя сочетание клавиш *Ctrl+H*, откройте контекстную справку для изучения функционирования модулей **Exclusive OR** и **And**. Теперь, когда создана логика для детектирования нажатия на кнопку устройства NI myRIO, вы можете использовать её в функциональности.

2. Создайте цикл мигания светодиодов встроенных светодиодов при нажатии кнопки.

a. Поместите константу **True Constant** на блок-схему.

i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»True Constant**. Поместите константу в кадр *Initialize frame*.

ii. Подключите константу к левой грани цикла While (она автоматически передается через тоннель в следующий кадр).



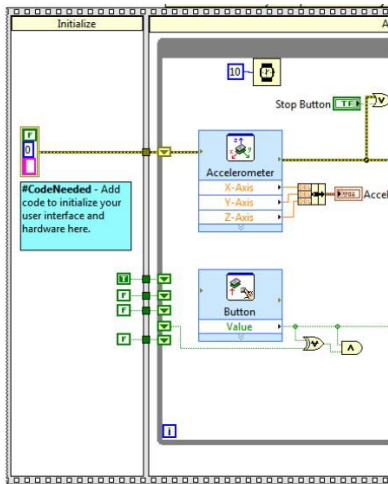
b. Поместите константу **False Constant** на блок-схему.

i. Щелкните правой кнопкой и выберите пункт **Programming»Boolean»False Constant**. Поместите три константы в кадр *Initialize frame* — первую прямо под константой **True**, остальные друг под другом.

ii. Подключите константы к левой грани цикла While.

c. Замените тоннели констант регистрами сдвига.

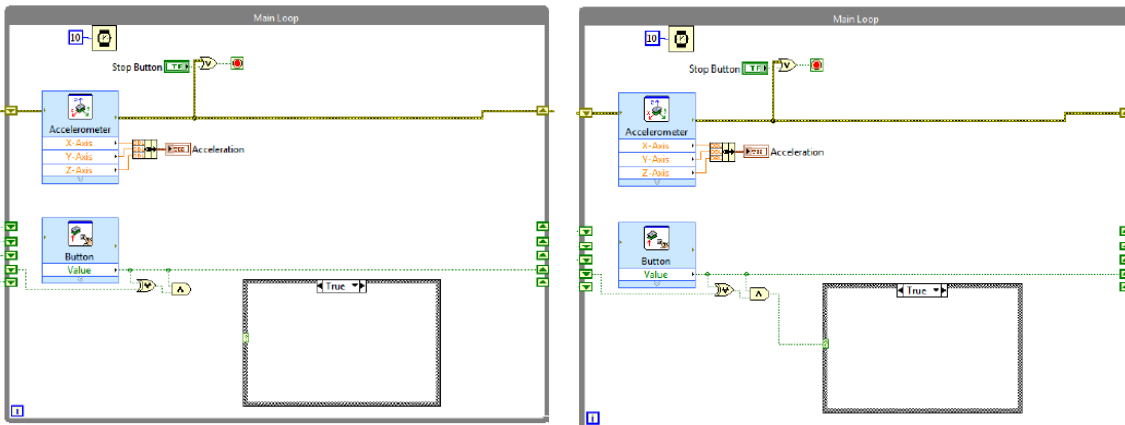
i. Щелкните правой кнопкой по каждому тоннелю логических данных только что созданных слева от цикла и выберите **Replace with Shift Register**.



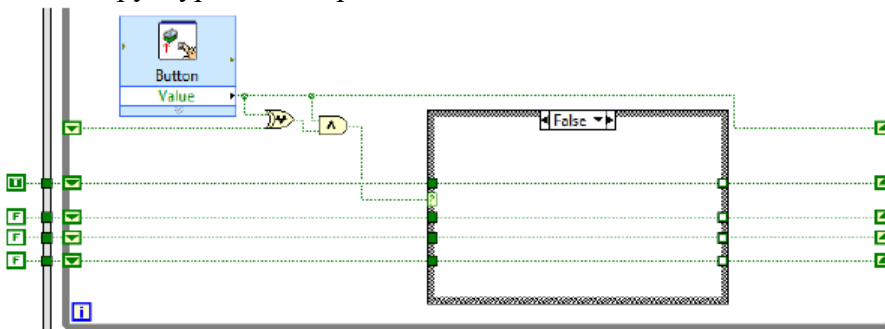
- a. Каждая из четырех логических констант только что помещенных на блок-схеме используется для отслеживания состояния четырех светодиодов на борту устройства NI myRIO. Этот код — очень простая и прямолинейная реализация сдвига горения светодиода на последующий светодиод. Другой метод использует счетчик и модульную функцию для вычисления остатка от деления отсчета и сравнения к конкретному светодиоду (0, 1, 2,3). Такой подход позволяет использовать только один регистр сдвига, но требует размещения большего количества функций на блок-схеме.
- b. Поместите структуру **case structure** (выбирающая структура) на блок-схему.

Щелкните правой кнопкой мыши и выберите пункт **Programming»Structure»Case**

- i. **Structure.** Поместите структуру в цикл While справа внизу от триггера (модулей **Exclusive OR** и **AND**).

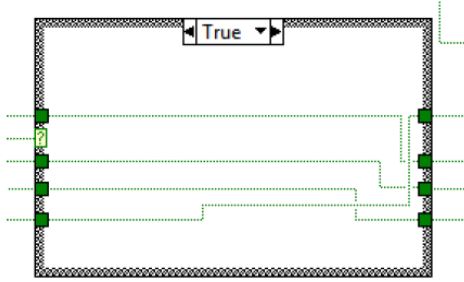


- ii. Соедините выход модуля AND и селектор на структуре
- iii. Переключите структуру на состояние False с помощью стрелочек возле индикатора.
- i. Пустите регистры сдвига через структуру, а затем напрямую к правой грани цикла While. Имейте в виду, что если они изначально не были подключены к левой грани структуры, автоматический инструмент соединения скорее пустит провода в обход структуры, чем через неё.



- v. Переключите структуру в состояние True.

vi. Подключите тоннель для каждой логической переменной к тоннелю, проходящему ниже, соединив последний тоннель слева с первым тоннелем справа, и т.д.



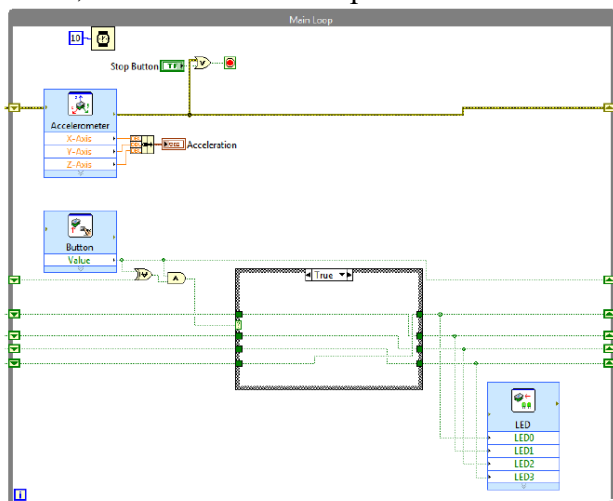
f. Логика работы такова, что когда детектируется восходящий фронт сигнала с кнопки (другими словами — когда кто-то нажимает кнопку), состояние структуры true переключает светодиод на следующий. Загорается следующий светодиод, а предыдущий гаснет. Такое происходит при каждом нажатии на кнопку. А если нажать кнопку при горящем четвертом светодиоде, он погаснет, а загорится первый светодиод. Тем не менее, этот код не будет работать, если не объяснить NI myRIO что логические переменные надо интерпретировать как состояния светодиодов.

б. Привяжите созданную логику к состоянию светодиодов.

а. Поместите на блок-схему экспресс-ви **LED**.

i. Щелкните правой кнопкой мыши и выберите пункт **myRIO»Onboard»LED**. Поместите экспресс-ви LED в правом нижнем углу цикла While. Нажмите **OK** в диалоге настройки экспресс-ви для сохранения стандартных настроек. По умолчанию этот экспресс-ви позволяет управлять всеми четырьмя светодиодами.

ii. Подключите каждую из логических переменных созданных ранее в входам LED0, LED1, LED2 и LED3 экспресс-ви.



7. Запустите VI нажав кнопку запуска на приборной панели. Проверьте работу кода, нажимая кнопку на NI myRIO и наблюдая за работой светодиодов.

8. Чтобы убедиться, что код выполняется на устройстве NI myRIO, а не на компьютере, щелкните правой кнопкой на NI myRIO в проекте LabVIEW и выберите пункт **Disconnect**.

9. Извлеките кабель USB из устройства NI myRIO и убедитесь, что код продолжает выполняться.

10. Заново подключите USB кабель.

11. В проекте LabVIEW выберите **Connect**.

12. Щелкните кнопку стоп на передней панели LabVIEW для остановки кода.

Программируемая логическая интегральная схема

ПЛИС — это кремниевые чипы, работающие в матричной структуре логических блоков, соединенных программируемыми переключателями. Вы можете настраивать эти блоки для обработки стандартных логических операций и, во многих случаях, более сложной логики. Первые коммерческие ПЛИС были изобретены со-основателями Xilinx Россом Фриманом и

Бернардов Фондершмиттом в 1985 году. Главное преимущество ПЛИС состоит в том, что вы можете изменять логику на аппаратном уровне без физических изменений оборудования. Это значит, что вы можете создавать собственную логику для своих систем и реконфигурировать ПЛИС для выполнения логики в работе. Вы программируете "сущность" ПЛИС на программном уровне, а затем воплощаете её в кремнии. В силу природы ПЛИС, отдельные секции ПЛИС на чипе независимы и могут выполняться в истинном параллелизме.

Истинный параллелизм означает, что задачи, выполняемые на ПЛИС, по-настоящему независимы и высоко детерминированы. Детерминизм критически важен в задачах контроля, роботизации и других задачах механотроники (типичная ПЛИС система может быть создана для взаимодействия с цифровым входом длительностью 25 нсек (40 МГц) а иногда и быстрее). Некоторые примеры ПЛИС-приложений LabVIEW включают интеллектуальный сбор данных, контроль на сверх-высоких скоростях, специализированные протоколы взаимодействия, настройка последовательности задач процессора, для сохранения процессорного времени для более сложного анализа, комплексного тактирования и синхронизации и программно-аппаратного тестирования. Благодаря возможности реагировать так быстро, ПЛИС позволяет вам создавать системы и эксперименты промышленного класса без крупных инвестиций и промышленного оборудования.

С модулем LabVIEW FPGA процесс программирования ПЛИС полностью графический и поддерживает полностью автоматическую компиляцию. Вы программируете ПЛИС с использованием модуля LabVIEW FPGA. Затем, когда вы готовы, LabVIEW генерирует промежуточные VHDL файлы, требуемые компилятором Xilinx, запускает компилятор и передает ему файлы. В результате получается побитовый файл, загружаемый во флэш-память, и считывающийся в работе. При работе ПЛИС, он считывает файл, и перенастраивается в соответствии с его инструкциями.

Задание 5: Исследование NI myRIO ПЛИС

1. Откройте диалог LabVIEW Getting Started и выберите **Create Project**.
2. В диалоге выберите стартовую точку для проекта из дерева в левой части. В категориях шаблонов, выберите **myRIO**.
3. В правой рамке, выберите myRIO Custom FPGA Project. Информацию по настройке проекта можно найти по ссылке [more information](#).
4. Выберите **Next** для настройки проекта. Задайте проекту подходящее имя, выберите директорию для сохранения и выберите устройство NI myRIO, подключенное по USB.
5. Выберите **Finish** для создания проекта.
6. Когда браузер проектов LabVIEW загрузит проект, "Мой компьютер" и myRIO-1900 (xxx.xx.xx.x) появляются как целевые устройства.
7. Раскройте целевое устройство NI myRIO и обратите внимание на новое дерево "chassis" (шасси). Такое дерево имеется в устройствах, содержащих ПЛИС, который можно задать целевым. Логически, целевой ПЛИС содержится в дереве шасси. Первоочередная задача ПЛИС — обрабатывать вход/выход на устройстве NI myRIO, так что вы можете обнаружить вход/выход под ПЛИС в дереве. Иерархия ПЛИС построена на папках, так что пользователь может сказать, где физически расположен каждый узел входа/выхода. Два MXP разъема, MSP разъем и встроенный вход/выход имеют каждый уникальную папку. Папки в свою очередь делятся по типу входа/выхода (цифровой/аналоговый) и физические банки. Вы можете перетащить их на ПЛИС-VI для чтения/записи соответствующего разъема. Каждый контроллер или индикатор с передней панели может быть прочитан/записан в VI реального времени.
8. Откройте "myRIO-1900 Customized FPGA.vi" для просмотра кода в ПЛИС.
9. Исследуйте код.

Целевое устройство ПЛИС имеет таймер 40МГц, настроенный в проекте LabVIEW. Любой VI в этой ветке автоматически становится ПЛИС-VI, и LabVIEW автоматически настраивает функции и тип данных, допустимые в VI. Вы можете создать новый ПЛИС-VI или модифицировать стандартный ПЛИС-VI.

Стандартный ПЛИС-VI подтверждает безопасность обработки входных/выходных данных через ПЛИС и подготовку к передаче в VI реального времени. NI myRIO поставляется со стандартной настройкой ПЛИС, которая обрабатывает все входные и выходные данные через оба MXP-разъема и MSP-разъем. ПЛИС.

Для упрощения архитектуры проектов, используйте стандартную настройку ПЛИС и только программируйте хост-VI реального времени (который выполняется на процессоре NI myRIO) и Windows VI (который выполняется на компьютере)

Практическое занятие 6. Управление движением мобильной платформы с пульта

Задание:

1. Изучить устройства и принцип работы пульта радиуправления
2. Установить на борт мобильного робота интерфейсную плату связи по радиосигналу
3. Разобраться с системой управления мобильной платформы с помощью пульта
4. Выполнить контрольные задания на полигоне.

Практическое занятие 7-8: Программное управление мобильной платформой

Задание:

1. Запрограммировать мобильную платформу на выполнение следующих действий:
 - a. Движение вперед
 - b. Движение назад
 - c. Движение на заданное расстояние
2. Изучить особенности маневрирования мобильных платформ с системой мотор-колесо.
3. Запрограммировать мобильную платформу на выполнение следующих действий:
 - a. Разворот на колесе
 - b. Разворот на места
 - c. Поворот по дуге
4. Запрограммировать робота так, чтобы он совершал движение по квадрату
5. Запрограммировать робота так, чтобы он совершал движение по «восьмерке»

Теоретический материал:

Команды для управления через драйвер двигателей. Остановка и удерживание вала двигателя. Движение колеса с установленной скоростью. Обработка показаний датчика угла поворота, подсчет скорости вращения. Плавные разгон и торможение двигателя. Определение пути прохождения. Управление мобильным роботом с дифференциальным приводом. Кинематика мобильного робота с дифференциальным приводом. Прямолинейное движение. Поворот шасси мобильного робота на необходимый угол. Плавные начало и остановка движения.

Подключение периферийных устройств

Драйвер для моторов позволяет вам легко подключить 2 DC мотора и 1 сервомотор или 3 сервомотора на коннектор MXP на MyRIO. Серво моторы подключаются через стандартные 3-пиновые насадки, тогда как DC моторы крепятся через терминалы подключений. ШИМ сигналы со стандартного ПЛИСа MyRIO поданы на соответствующие входы для DC моторов и серво – моторов. Вход энкодера для мотора 0 соответственно подведен к разъему энкодер 0, а энкодер для мотора 0 подведен на соответствующие цифровые сигналы входа/вывода.

Основные особенности:

- Используется выход MXP с MyRIO
- Можно использовать одновременно 2 DC мотора и 1 сервомотор или 3 сервомотора
- Защита от перегруза
- 34-пина для подключения периферии
- 2 интерфейса для подключения энкодеров
- Выключатель питания моторов
- LED индикация работы моторов
- LED индикация питания

- Зеленый и красный LED индикатор для указания направления сигнала

DC моторы

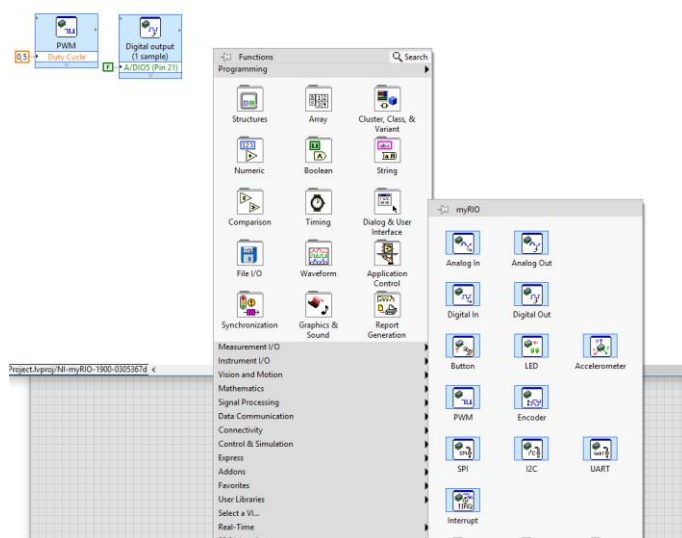
Для контроля DC моторов необходимо подключить их к терминалам J14 или J15, а для контроля энкодеров, их следует подключить к разъемам J12/J13. Подвести питание к моторам в разъем J7(максимум 24В). Контролировать направление вращения моторов возможно с помощью переключения сигналов на цифровых выходах DIO5/DIO6 для мотора 0, DIO7/DIO13 для мотора 1. Что бы изменить скорость вращения, на моторах для PWM0/1 следует задать 1000 Гц. Скорость изменяется в диапазоне от 0 до 1, где 0 – полная остановка, 1 – 100% скорость.

Сервомоторы

Для контроля сервомоторов используются выходы J9/J10/J11, что бы подать на них питания, следует подвести питание в 5В на вход J8. Что бы изменить положение серво – мотора следует задать 50Гц, значения подбирать в диапазоне от 0.01 до 0.02

Задание 1: Необходимо придать вращение DC – мотору в обе стороны.

Необходимые элементы: Express VI – PWM Express VI – Digital Output Wait (ms) While Loop



Задание 3: Подавая сигнал на мотор, вращать его до тех пор, пока значения энкодера не станет больше 5000

Необходимые элементы:

Express VI – PWM

Express VI – Digital Output

Express VI – Encoder

Задание 4: Изучить блок In Range and Coerce (для нахождения блока использовать Ctrl+Space). Написать логику на возвращение колеса к значению 5000 по значению энкодера. Погрешность в 100 тиков.

Необходимые элементы:

Express VI – PWM

Express VI – Digital Output

Express VI – Encoder

In Range and Coerce

Задание 5: Вычислить изменение пройденного пути(S) колесом (перевести в сантиметры) между текущей и предыдущей итерацией.

Необходимые элементы:

Express VI – Encoder

Shift Register

Wait (ms)

While Loop

Задание 6: Вычислить изменение прошедшего времени (T) между текущей и предыдущей итерацией.

Необходимые элементы:

Shift Register

Wait (ms)

While Loop

Tick Count (ms)

Задание 7: Рассчитать текущую скорость колеса по формуле $(S1 - S0) / (T1 - T0)$

Необходимые элементы:

П.Р. 10,11 While Loop Wait (ms)

Задание 8: Изучить структуру Master&Slave. Сделать следующие циклы:

- Цикл, где опрашивается ТОЛЬКО энкодер.
- Цикл, где подаются сигналы на вращение мотора.
- Цикл, где рассчитывается скорость колеса.
- Цикл, где будет прописана основная логика (конечный автомат)

Передача данных между циклами осуществляется с помощью Local Variable.

Необходимые элементы:

Express VI – PWM

Express VI – Digital

Output

Express VI – Encoder

Shift Register

Tick Count (ms)

Практическое занятие 9. Использование сенсорного датчика и ультразвукового датчика программная реализация обратной связи в управлении роботом

Задание:

1. Изучить специфику работы сенсорного датчика
2. Написать программу, запускающую движение робота по прямой при нажатии датчика касания.
3. Оснастить робота бампером, ход которого нажимает на датчик касания
4. Написать программу, останавливающую робота, если он столкнулся с препятствием.

Проектное задание: Написать программу, позволяющую роботу в случае столкновения с препятствием отъехать от него и повернуть на 90 градусов.

Задание:

1. Изучить специфику работы ультразвукового датчика
2. Написать программу, останавливающую прямолинейно движущегося робота, на расстоянии 20 см до стены или препятствия
3. Написать программу для «робота-прилипалы», держащего дистанцию в 15 см от препятствия. Алгоритм решения:
 - a. при значении показания ультразвукового датчика больше 15 см робот будет двигаться вперед, стараясь приблизиться к препятствию;
 - b. при значении показания ультразвукового датчика меньше 15 см робот будет двигаться назад, стараясь удалиться от препятствия.

Проектное задание: «Движение в лабиринте».

Практическое занятие 10. Использование датчика освещенности и программная реализация обратной связи в управлении роботом при движении по направляющей

Задание:

1. Изучить специфику работы датчика освещенности
2. Написать программу, изменяющую скорость движения робота в зависимости от интенсивности внешнего освещения.
3. Написать программу, останавливающую робота, если он наехал на черную линию

Проектное задание: Написать программу, движения робота на релейном регуляторе по направляющей перекрестками

Задание 1:

1. Изучить специфику и принцип работы P-регулятора
2. Изучить особенности использования блоков математики в написании программ.
3. Написать программу, движения робота по направляющей на P, PD -регуляторе
4. Подобрать скорость и значение коэффициента пропорциональности для плавного движения робота по линии

Проектное задание: заезды роботов по траектории

Задание 2:

1. Изучить специфику и принцип работы PD-регулятора
2. Изучить особенности использования переменных и блоков математики в написании программ.
3. Написать программу, движения робота по направляющей на PD-регуляторе
4. Подобрать скорость, значение коэффициента пропорциональной и дифференциальной составляющей для плавного движения робота по линии

Проектное задание: В среде LM EV3 создать Урок 9 с элементами электронного учебника для изучения темы «Движение по направляющей на PD -регуляторе»

Теоретический материал

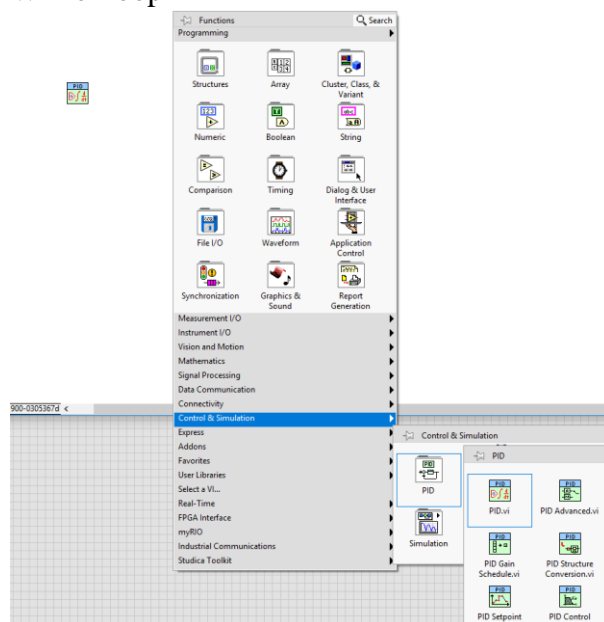
1. Изучить принцип работы PID – регулятора.

Необходимые элементы:

PID

Wait (ms)

While Loop



Упражнение 1: Используя основу П-регулятора, добавить в нее новый цикл с PID – регулятором, где вход в PID – регулятор «Setpoint» будет скоростью, к которой необходимо стремиться колесу, «Process Variable» – текущая скорость колеса, а значение с выхода «Output» с помощью локальной переменной подать на ШИМ (PWM) сигнал мотора. Убедиться в корректной работе VI.

Необходимые элементы:

Local Variable

Wait (ms)

Практическое занятие 13. Основы машинного зрения: распознавание образов

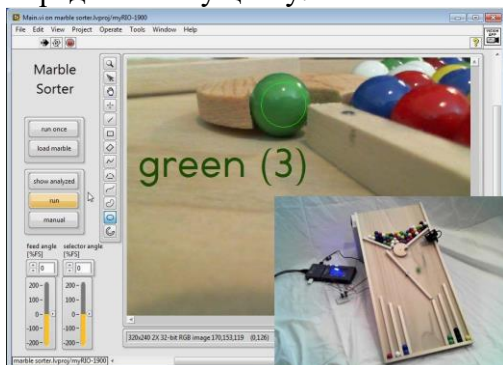
Вопросы для проработки:

Техническое зрение. Конфигурация вебкамеры. Основные параметры: яркость, насыщенность, время экспозиции, фокус, баланс белого. Чтение штрихкодов. Распознавание заданного эталонного шаблона при помощи сопоставления с образцом (Pattern Matching). Распознавание заданного шаблона при помощи выделения границ (Geometric Matching).

Проектное задание: «Сортировщик шариков»

Теоретический материал

Видеосистемы преуспели в сортировке объектов по категориям в зависимости от размера, формы и цвета. Стандартная система сортировки состоит из бункера, загрузчика, камеры, алгоритма обработки видеоизображения и отборщика. Разработайте сортировщик шариков с использованием цветной камеры и двух серводвигателей, который автоматически рассортировывает полный бункер шариков по контейнерам, каждый из которых соответствует определенному цвету.



Лицевая панель приложения *Сортировщик шариков*

Функциональные требования: Приложение *Сортировщик шариков* должно выполнять следующие функции:

1. Сортировать перемешанные в бункере цветные шарiki по шести различным цветам в контейнеры аналогичного цвета, используя последовательность из четырех действия:
2. Поворот диск загрузчика, расположенного под бункером, для загрузки одного шарика в подающее отверстие диска,
3. Поворот диска таким образом, чтобы шарик оказался в области видимости вебкамеры для последующего анализа цвета шарика,
4. Поворот рычага отборщика к соответствующему контейнеру
5. Поворот диск загрузчика, чтобы подать шарик к рычагу отборщика контейнера, который, в свою очередь, направит шарик в нужный контейнер.
6. Загружать один шарик в ячейку диска загрузчика и поворачивать его в область видимости вебкамеры для анализа цвета, при щелчке по кнопке load marble.
7. При нажатии кнопки run once однократно выполнять четыре действия сортировки.
8. Непрерывно выполнять процедуру сортировки при нажатии кнопки run.
9. Визуализировать изображение с вебкамеры с максимально возможной частотой кадров во время простоя сортировщика шариков,
10. Непрерывно отображать обнаруженный цветной шарик и ассоциированный с ним числовой индекс в неискажающем слоя наложения, даже во время простоя сортировщика шариков,
11. Разрешать ручное управление загрузчиком и углами поворота серводвигателя с рычагом отборщика при нажатии кнопки manual.
12. Применять определяемую пользователем исследуемую область (ROI) изображения для анализа цвета; в отсутствии определяемой пользователем ROI, использовать область по умолчанию,
13. Останавливать сортировщик, если диск загрузчика пуст.
14. Визуализировать обработанное изображение, если нажата кнопка show analyzed, иначе отображать изображение с вебкамеры. На рисунке показана лицевая панель приложения *Marble Sorter* во время просмотра и анализа цвета зеленого шарика; вставка снимка сортировщика

шариков в нижнем левом углу иллюстрирует процесс подачи шарика в соответствующий контейнер.

Требуемые ресурсы

- NI LabVIEW и NI Vision Assistant
- NI myRIO с источником питания и USB-кабелем
- USB вебкамера
- USB концентратор (Stratom X-HUB или аналог)
- USB флэш-накопитель

Технические компоненты сортировщика шариков:

Шарики стандартного диаметра 16 мм с однотонным цветом, например, *Chinese Checker Marbles1* Два серводвигателя (*GWS S03N STD2*); для получения полной информации о серводвигателях и примеров кода LabVIEW обратитесь к главе 17 руководства *NI myRIO Project Essentials Guide3*. Дорожки для шариков; подробное описание конструкции можно найти в приложении E

Результаты работы:

Отчет по практической работе в бумажном или электронном виде, оформленный в соответствии с требованиями преподавателя; включите скриншоты лицевой панели, чтобы продемонстрировать, что ваше приложение удовлетворяет функциональным требованиям. .zip архив, включающий все файлы проекта:

Образцы изображений

Скрипт Vision Assistant

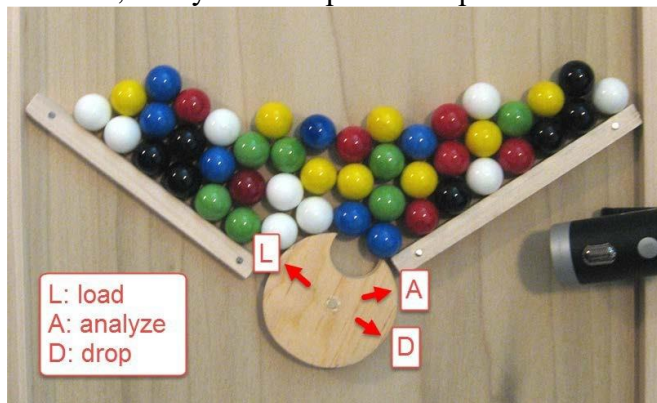
Набор файлов проекта LabVIEW

Способ реализации

Технические компоненты сортировщика шариков

Технические компоненты сортировщика включают два серводвигателя для того, чтобы привести в действие диск загрузчика и рычаг отборщика. Диск загрузчика имеет круглый вырез («карман») для одного шарика и рычаг отборщика, который поворачивается в позицию выбранного контейнера, расположенного в нижней части сортировщика. Поднимите верхнюю часть сортировщика примерно на 6 дюймов, чтобы обеспечить достаточную «силу тяжести» для скатывания шариков по направлению к приемным контейнерам, но недостаточную для того, чтобы шарики «выскочили» из конструкции.

Серводвигатель обеспечивает ход немного более, чем на 180°, поэтому диск загрузчика, изображенный на рисунке 16.2, может быть установлен в любое положение между позицией загрузки и выгрузки шарика. Диск загрузчика всегда должен быть повернут полностью в позицию загрузки, даже если карман уже занят. Это гарантирует, что шарик, оставшиеся в левой стороне бункера, попадут в карман загрузчика. Кроме того, вращающийся в разных направлениях диск загрузчика перемешивает в очередь шариков в бункере, что уменьшает риск заклинивания. В положении «analyze» шарик находится в области прямой видимости вебкамеры, а в положении «drop» шарик подается рычагом отборщика. Рычаг отборщика, изображенный, поворачивается к одному из шести цветных контейнеров в зависимости от результатов анализа цвета шарика видеосистемой, и опускает шарик в выбранный контейнер



Бункер с шариками, диск загрузчика и вебкамера (справа). Диск загрузчика поворачивается в положение "load", чтобы забрать один шарик, затем поворачивается обратно в положение "analyze" для анализа цвета шарика видеосистемой, и, наконец, выполняет поворот в положение "drop", чтобы отправить шарик в нужный контейнер.

Скрипт NI Vision Assistant

Шаг NI Vision Assistant *Color / Color Classification* (классификация по цвету) рассматривает исследуемую область изображения (ROI) и классифицирует ее, как соответствующую одному из N цветов, определенных файлом классификатора (.clf), созданного в интерактивном режиме с использованием NI Color Classification Training Interface (Интерфейс обучения классификации по цвету). Обучающий интерфейс может работать с несколькими цветами на одном или нескольких изображениях; последний режим подходит для использования нескольких изображений с вебкамеры, каждое из которых получено в положении "analyze" диска загрузчика с шариком разного цвета, в том числе изображение с пустым карманом диска загрузчика.

Шаг *Color Classification* предоставляет входы для параметров исследуемой области, определяемой пользователем, и пути к файлу классификатора для встраивания в NI Vision Assistant Express VI. Эти два входа обеспечивают возможность переопределять ROI с шариками при изменении позиции вебкамеры и для простой замены классификатора при изменении условия освещения.

Изучите видеоурок *Classify Colors (Классификация по цвету)*⁴, чтобы узнать, как создавать файл классификатора.

Управление серводвигателем

В главе 17 *NI myRIO Project Essentials Guide 5* описывается, как задать угол поворота серводвигателя с использованием выхода ШИМ (широтно-импульсной модуляции) NI myRIO PWM; более подробную информацию вы найдете в видеоуроке *Servo Interfacing Theory (Способ сопряжения с серводвигателем)*⁶ Используйте программный код, приведенный в этой главе, для преобразования необходимого значения угла поворота, выраженного в процентах от полной шкалы (от -100 до +100), в соответствующее значение скважности импульсов на выходе ШИМ. Более подробную информацию вы найдете в видеоуроке *Servo Demo Walk-Through (Разбор демонстрационной программы управления серводвигателем)*⁷, а код может быть загружен из архива *NI myRIO Project Essentials Guide .zip Files*⁸.

В демо-коде используются низкоуровневые PWM VI, а именно: Open и Set Duty Cycle and Frequency. Set Duty Cycle VI и Close VI расположены в субпалитре низкоуровневых ШИМ функции NI myRIO. Вы можете изменить этот демонстрационный код, чтобы добавить второй ШИМ-канал для удобства проведения экспериментов с двумя серводвигателями сортировщика шариков и получить необходимые углы поворота диска загрузчика и рычага отборщика контейнеров. Уделите некоторое время для управления сортировщиком шариков вручную, чтобы лучше понять как проектировать полноценное приложение автоматической сортировки шариков.

Совет: Обратитесь к схеме разъемов NI myRIO в приложении B.

ВАЖНО: Серводвигатели генерируют значительные электрические помехи, которые могут быть причиной перекрестных помех для двух каналов ШИМ, управляющих углами поворота серводвигателей. Чтобы решить эту проблему, используйте электролитический конденсатор 100 мкФ в качестве развязки по цепи питания 5 Вольт NI myRIO. Будьте осторожны! Соблюдайте полярность электролитического конденсатора.

Создайте set servo angle VI для задания угла поворота серводвигателя, чтобы упростить сопряжение двух ваших серводвигателей и ускорить разработку приложения *Marble Sorter*. Этот VI. принимает данные с пользовательского элемента управления servo info (состоит из ссылки на канал ШИМ и массива кластеров, которые содержат угол поворота серводвигателя, время задержки в миллисекундах и текстовое обозначение для каждой позиции серводвигателя), а также числовой индекс для массива кластеров. Например, в магистрали данных полоса servo info для диска загрузчика должна быть проинициализирована в состоянии (*qsm*) *initialize* ссылкой на ШИМ канал серводвигателя диска загрузчика и кластером значений для каждой из трех позиций "load", "analyze" и "drop"; задержка определяет необходимое время ожидания завершения

поворота серводвигателя в заданную позицию (см. встроенный VI Wait (ms) LabVIEW), а текстовая строка сообщает о наименовании каждого положения. Запуск на исполнение set servo angle с этой информацией для управления и индексом "1" инициирует поворот серводвигателя диска загрузчика в положение "analyze", а затем ожидание назначенной в миллисекундах задержки, чтобы позволить серводвигателю завершить движение.

СОВЕТ: *Используйте горячие клавиши (Ctrl + space bar) для поиска по имени и размещения необходимых элементов LabVIEW. Ознакомьтесь с приложением C для получения дополнительных советов по работе в LabVIEW. Аналогичным образом, в магистрали данных полоса servo info servo info для отборщика контейнеров инициализируется ссылкой на канал ШИМ серводвигателя и шести углов, задержек времени и текстовых обозначений цветowych контейнеров. Запуск на исполнение set servo angle с этой информацией для управления и индексом "2" инициирует поворот серводвигателя диска загрузчика в направлении к третьему контейнеру, а затем ожидание назначенной в миллисекундах задержки.*

Обратитесь к видео Create "Coin Info" Cluster Array (Создание массива кластеров "Coin Info")⁹, в котором изложена общая процедура, необходимая для создания пользовательского элемента управления servo info и инициализации его значений. Так как ссылка на ШИМ низкоуровневых PWM VI является определителем типа, следуйте процедуре, проиллюстрированной в видеоуроке, начиная с временной отметки 2:10, Step 5: Connect Script (Шаг 5: Подключение скрипта)¹⁰ при добавлении настраиваемого элемента управления.

Изучите видеоурок Create "Flag Change" SubVI (Создание SubVI "Изменение флага")¹¹, с примером, иллюстрирующим действия, необходимых для создания собственного set servo angle VI. Обязательно установите режим выполнения "Preallocated clone reentrant execution", как показано на отметке времени 6:40.

Операции в LabVIEW

Для этого проекта понадобится реализовать в LabVIEW следующие операции: Get ROI from Image Display or Default¹² – Получение исследуемой области ROI из основного изображения с помощью узла свойств "Property Node". Использование ROI по умолчанию, определенной скриптом Vision Assistant, если ROI не указана на дисплее.

Overlay Text¹³ – Формирование текста в неискажающем слое наложения, с помощью "IMAQ

Overlay Text" VI.

Select Overlay Text Font¹⁴ – Выбор пользователем шрифта текста "NI Vision" и задание его параметров, таких, как размер шрифта и выравнивание__

Советы по разработке

Образцы изображений

Сначала в режиме ручного управления задавайте произвольные углы поворота диска загрузчика серводвигателем. Определите значение угла для положения "analyze".

Установите низкое разрешение вебкамеры для обеспечения высокой частоты кадров. В этом приложении вебкамера это не более, чем датчика цвета, поэтому высокое разрешение не требуется.

Для надежного цветового восприятия установите все параметры вебкамеры вручную, в частности, "Focus" (фокусировку) и "White Balance" (баланс белого).

Щелкните правой кнопкой мыши на дисплее изображения и выберите "Save Image" (Сохранить изображение). Повторите эту операцию, получая изображение для шариков каждого цвета, а также для изображения с пустым карманом диска загрузчика.

Скрипт Vision Assistant

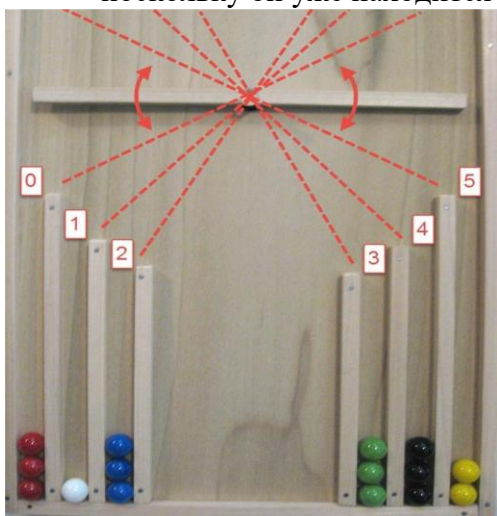
Шаг *Color Classification* формирует строку из выбранных вами обозначений. Используйте значимые обозначения, такие, как "красный", "белый" и "синий" для удобства чтения. Однако, вы можете также использовать числа в качестве обозначений, например, "0", "1", "2", и так далее, используя функцию LabVIEW Decimal String To Number, которая позволяет преобразовывать обозначения в числовые индексы цвета, по которым можно определять угол поворота серводвигателя из массива углов и, соответственно, приемный контейнер. Будет лучше использовать оба компонента в обозначении в следующем формате: <<0: красный", "1: белый" и

"2: синий", т.к. VI может выбрать десятичное число из текста. После внедрения в Vision Assistant Express VI классификатора цвета формирует в качестве выходных данных массив строк, поскольку классификатор может работать и с составными ROI. Используйте функцию Index Array для выбора одного обозначения.

Дополнительные функции:

Возможно вы захотите добавить одну или несколько дополнительных функций:

- Отобразить двоичный код индекса считанного цвета в качестве шаблона для включения светодиодов на плате pi myrio; см. функцию number to boolean array.
- добавить жидкокристаллический (lcd) дисплей для отображения кода считанного цвета; более подробную информацию о lcd дисплее можно найти в главе 8.
- удерживать текущее положение серводвигателя рычага при совпадении цвета текущего и предыдущего шариков, устранив таким образом время ожидания. другими словами, когда в карман диска загрузчика проходят несколько шариков одного цвета, исключается необходимость ожидания перемещения рычага, поскольку он уже находится в нужном положении



Рычаг отборщика расположен под диском загрузчика. Рычаг поворачивается к одному из шести контейнеров в зависимости от результата анализа цвета скриптом

Практическое занятие 15. Разработка дидактических и методических рекомендаций для организации демонстрационного экзамена по мобильной робототехнике

Вопросы для проработки

1. Документы регламентирующие проведение ДЭ
2. Вопросы организации ДЭ
3. Оборудование полигона для ДЭ
4. Оценка квалификации студента (выпускника) в ходе демонстрационного экзамена

Проектное задание: подготовить всю необходимую документацию для организации и проведения ДЭ в учебном формате

Практическое занятие 16. Организация демонстрационного экзамена по мобильной робототехнике

Контрольная работа: проектное задание: «Демонстрационный экзамен»

РЕГЛАМЕНТ ДЕМОНСТРАЦИОННОГО ЭКЗАМЕНА ПО НАПРАВЛЕНИЮ «МОБИЛЬНАЯ РОБОТОТЕХНИКА»

Регламент включает в себя следующие разделы:

1. введение
2. требования к участникам (квалификация и объем работ)
3. этапы проведения
4. экзаменационное задание
5. оценка

6. отраслевые требования техники безопасности

7. материалы и оборудование

1. ВВЕДЕНИЕ

1.1. Название и описание направления

1.1.1 Мобильная робототехника

1.1.2 Описание предметной области

Под направлением «Мобильная робототехника» понимается использование технологий проектирования, конструирования, изготовления, сборки, программирования, управления и обслуживания механических, электрических систем и систем управления мобильным роботом для выполнения задач автоматизации производственных и сервисных процессов. В условиях инженерного образования подростков в образовательных организациях направление «Мобильная робототехника» является неотъемлемой частью обучения принципам инженерной разработки и конструирования, а также изготовления конечного материального продукта с использованием широкого круга разнообразных инструментов, станков и технологий. Мобильная робототехника включает в себя элементы механики и компьютерных технологий (программирование автоматизированных систем управления).

1.2. Область применения

1.2.1. Каждый член экзаменационной комиссии (Эксперт) и Участник демонстрационного экзамена (ДЭ) обязан ознакомиться с данным Регламентом.

1.2.2. Данный регламент используется только совместно с документами:

- Положение о демонстрационном экзамене для обучающихся по дополнительным общеразвивающим программам углублённого уровня;
- Требования к Базовой площадке (материалы и оборудование, которые предоставляются Базовой площадкой участникам демонстрационного экзамена);
- Отраслевые требования техники безопасности;
- Примерное экзаменационное задание.

1.3. Время проведения демонстрационного экзамена

Подготовка: 12 часов

Время проведения экзамена: 2 часа.

1.4. Форма и вид участия

Форма участия в демонстрационном экзамене - очная. Вид участия: командный (не более 2-х человек). В состав команды могут входить специалист по электромеханическим системам и специалист по электронике и программированию, либо включать участников, обладающих знаниями в обеих областях.

1.5. Возраст участников

Возраст участников демонстрационного экзамена –18 - 20 лет.

2. ТРЕБОВАНИЯ К УЧАСТНИКАМ (КВАЛИФИКАЦИЯ И ОБЪЁМ РАБОТ)

Участник демонстрационного экзамена обязан продемонстрировать свои знания и умения в конструировании, изготовлении, сборке, программировании, управлении и обслуживании механических, электрических узлов и систем управления мобильных роботов, предназначенных для выполнения широкого круга задач, связанных с автоматизацией процессов, облегчающих деятельность человека в различных видах деятельности.

2.1 Участник демонстрационного экзамена должен знать и понимать:

- правила техники безопасности при работе с инструментом и оборудованием;
- правила технического обслуживания используемого оборудования, назначение используемых материалов, правила их безопасного применения;
- правила работы в составе команды.

Участник демонстрационного экзамена должен проявить следующие знания и понимание:

- Знания по проектированию, изготовлению, сборке, наладке и сдаче в эксплуатацию мобильного робота.
- Понимание систем программирования и управления.

Участник демонстрационного экзамена должен проявить следующие навыки и умения:

- Анализ реального применения мобильного робота для определения конкретных рабочих возможностей робота и их соответствия выполняемой работе.
- Определение конкретных блоков аппаратного обеспечения (различные датчики и т.п.), необходимых для обеспечения функционирования робота.
- Интегрирование разработанной системы управления в базовый блок управления мобильным роботом.
- Выбор соответствующих материалов и процессов для изготовления структурных и механических элементов, необходимых для дополнительной конструкции (прототипа).
- Выбор соответствующего аппаратного обеспечения (моторы, датчики), необходимого для соблюдения требований к функционированию дополнительной конструкции (прототипа).
- Умение читать технические описания и несложные чертежи, понимать постановку задачи, планировать свои действия.
- Умение обращаться с измерительными инструментами (линейка, штангенциркуль, микрометр, транспортир).
- Проектирование робота и дополнительной конструкции (прототипа) с учётом требований техники безопасности.
- Умение работать с ручным инструментом, собрать конструкцию.
- Владение основными приёмами инженерного моделирования.

2.2 Содержание экзамена

Экзамен является демонстрацией и оценкой предпрофессиональных навыков по направлению «Мобильная робототехника».

Экзаменационные испытания состоят из практических и теоретических заданий. Практическая часть экзамена заключается в разработке конструкции мобильного робота, изготовлении отдельных элементов конструкции.

Теоретические знания подвергаются проверке в ходе решения комплекта задач.

Содержанием Демонстрационного Экзамена является моделирование, разработка, сборка, регулировка и испытание, несложной робототехнической конструкции.

3. ЭТАПЫ ПРОВЕДЕНИЯ

3.1. Обучающий курс

3.2. Консультации в процессе самостоятельной работы по вопросам моделирования, разработки, сборки, регулировки и испытаний проводятся в течение учебного курса.

3.3. Ознакомление с техникой безопасности;

- ознакомление с рабочим пространством;
- решение возникающих у участников экзамена организационных вопросов.

3.4. Предварительная самостоятельная работа

Время на предварительную подготовку составляет 12 часов

Вид предполагаемого участия – командный.

3.5. Примерное экзаменационное задание.

Часть 1. Разработка и сборка конструкции мобильного робота из предложенного набора комплектующих элементов.

Часть 2. Сборка исполнительного механизма, монтаж исполнительного механизма на робота.

Часть 3. Разработка управляющей программы, тестирование, регулировка,

Часть 4. Испытания, демонстрация работы робота на поле.

3.6. За неделю до даты проведения контрольной работы в формате демонстрационного экзамена рабочая группа проверяет и тестирует оборудование.

3.7. В случае форс-мажорной ситуации (поломка оборудования, испорченные детали и т.п.), принимается решение о возможности предоставления участнику экзамена дополнительного времени, назначения испытания в другой день,

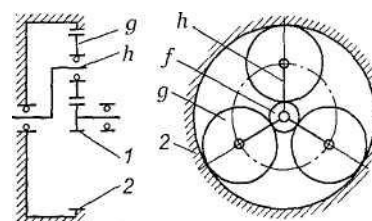
3.8. Участники должны знать и строго выполнять требования по охране труда и правила внутреннего распорядка во время проведения мероприятия. Участники, не прошедшие инструктаж по технике безопасности, к контрольной работе не допускаются.

4. ПРОВЕДЕНИЕ ДЕМОЭКЗАМЕНА

- 4.1. Формат и структура экзаменационного задания
 - 4.1.1. Задание состоит из 4 частей, распределённых на самостоятельную и аудиторную работу
 - 4.1.2. Каждая часть экзаменационного задания содержит задачи по компетенциям «Разработка и моделирование мобильных роботов» и «Программирование систем управления мобильных роботов».
 - 4.1.3. Последовательность выполнения задач каждой части задания участники команды определяют сами.
 - 4.1.4. По окончании времени, отведённого на выполнение контрольной работы в формате демозамена, все материалы, оборудование, комплектующие сдаются экзаменационной комиссии.
 - 4.1.5. Участники, завершившие выполнение задания ранее отведённого времени, сдают все экзаменационной комиссии и покидают рабочую зону.

3.2. Примеры тестовых заданий по вопросам общей механике и механическим передачам

1. Какое основное отличие зубчатой передачи от фрикционной?
 1. Постоянство передаточного числа
 2. Непостоянство передаточного числа
2. Движение в зубчатых передачах передается за счет...
 - 1) зацепления зубьев
 - 2) сил трения между зубьями
 - 3) прижатия колес друг к другу
 - 4) скольжения зубьев друг по другу
3. В цилиндрических зубчатых передачах передаточное отношение ...
 - 1) постоянное
 - 2) переменное
 - 3) $u = 0$
 - 4) $u = \infty$
5. У зубчатых колес находящихся в зацеплении должны быть одинаковыми ...
 - 1) делительные диаметры
 - 2) ширина колес
 - 3) числа зубьев
 - 4) модули зубьев
6. Как классифицируется по взаимному расположению осей колес передача на рисунке?
 1. Оси параллельны
 2. Оси пересекаются
 3. Оси скрещиваются
7. Как называют деталь h на рисунке?
 1. Водило
 2. Сателлиты
 3. Эпицикл
8. Определите, передаточное число червячной передачи, если число зубьев колеса равно $Z_2 = 30$, число витков червяка $z_1 = 2$
 - 1) 60
 - 2) 15
 - 3) 1/15
 - 4) Определить нельзя
9. Механическая передача является повышающей при ...
 - 1) $u < 1$, $n_1 < n_2$



- 2) $u > 1, n_1 > n_2$
 3) $u > 1, n_1 < n_2$
 4) $u < 1, n_1 > n_2$
10. Механическая передача является понижающей при ...
 1) $u < 1, n_1 < n_2$
 2) $u < 1, n_1 > n_2$
 3) $u > 1, n_1 < n_2$
 4) $u > 1, n_1 > n_2$
11. К механическим передачам с зацеплением относятся ...
 1) зубчатые, волновые, клиноременные
 2) зубчатые, фрикционные, червячные
 3) зубчатые, цепные, червячные, планетарные
 4) зубчатые, червячные, ременные, фрикционные
12. К механическим передачам трением относятся ...
 1) червячная
 2) клиноременная
 3) волновая зубчатая
 4) планетарная
 5) винтовая
13. Какое назначение механических передач
 1. Вырабатывать энергию
 2. Воспринимать энергию
 3. Затрачивать энергию на преодоление внешних сил, непосредственно связанных с процессом производства
 4. Преобразовывать скорость, вращающий момент, направление вращения
14. Как классифицируют зубчатую передачу по принципу передачи движения?
 1. Трением
 2. Зацеплением
 3. Непосредственно контактом деталей, сидящих на ведущем и ведомом валах
 4. Передача гибкой связью
15. Какая передача может использоваться для передачи вращения между валами, оси которых пересекаются?
 1. Коническая
 2. Червячная
 3. Цилиндрическая
 4. Кривошипно-шатунная
16. Какая передача может использоваться для передачи вращения между валами, оси которых параллельны?
 1. Цилиндрическая
 2. Червячная
 3. Кулисная
 4. Реечная
17. Какая передача может использоваться для передачи вращения между валами, оси которых перекрещиваются (но не пересекаются)?
 1. Червячная
 2. Гипоидная
 3. Коническая
 4. Винтовая

18. Макет какой передачи показан на фотографии?

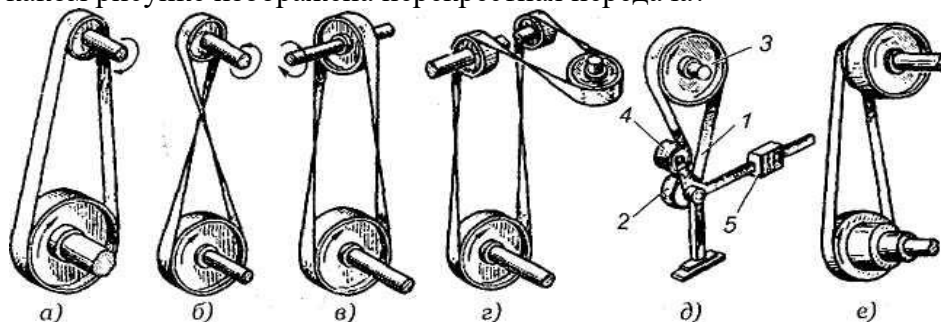
1. Червячной
2. Кулисный
3. Винтовой
4. Реечной



19. Какая передача как правило имеет меньший уровень шума при работе?

1. Цилиндрическая прямозубая
2. Коническая
3. Червячная
4. Цилиндрическая косозубая

20. На каком рисунке изображена перекрестная передача?



3.3. Содержание контрольной работы

Задание для контрольной работы в формате демонстрационного экзамена состоит из четырех частей:

Часть 1: Разработка и сборка конструкции мобильного робота из предложенного набора комплектующих элементов, демонстрация робота (выполняется в формате самостоятельной работы):

- разработать и собрать электромеханическую схему (компетенция «Разработка и моделирование мехатронных систем (мобильных роботов)»);
- разработать и собрать электронную схему (компетенция «Программирование систем управления мобильных роботов»);

Часть 2: Сборка исполнительного механизма, монтаж исполнительного механизма на робота, демонстрация работы робота (выполняется в формате самостоятельной работы):

- решить практическую задачу на механику (компетенция «Разработка и моделирование мехатронных систем (мобильных роботов)»);
- смоделировать исполнительный механизм (компетенция «Разработка и моделирование мехатронных систем (мобильных роботов)»);
- установить разработанный механизм на робота;

Часть 3: Разработка управляющей программы для мобильного робота (выполняется в формате самостоятельной работы):

- разработать управляющую программу передвижения робота (компетенция «Программирование систем управления мобильных роботов»);
- продемонстрировать способность робота к передвижению;
- разработать управляющую программу для исполнительного механизма (компетенция «Программирование систем управления мобильных роботов»);
- продемонстрировать работоспособность робота;

Часть 4: Корректировка управляющей программы для мобильного робота под уточненную практическую задачу на доле (выполняется во время аудиторной работы):

- решить комплект задач на разработку управляющей программы;

- продемонстрировать работоспособность робота на поле и способность выполнять манипуляции с объектами;

Задача каждого участника выполнить все части экзаменационного задания от начала до конца. Каждая часть оценивается отдельно, в соответствии с разработанными критериями

3.4. Экзамен

Вопросы к экзамену

1. Предпосылки возникновения и основные исторические этапы развития робототехники.
2. Основные понятия мобильной робототехники, история ее развития в России и за рубежом
3. Типы мобильных роботов и их применение в различных областях человеческой деятельности.
4. Учебное оборудование для изучения мобильной робототехники.
5. Симуляторы для мобильной робототехники.
6. Система конкурсных соревновательных и творческих мероприятий в России и за рубежом.
7. Техническое описание компетенции «Мобильная робототехника» WSI/WSR
8. Основные подсистемы мобильного робота, их назначение и способы реализации.
9. Основные виды трансмиссий ходовой части мобильных роботов.
10. Основы потокового программирования МК,
11. Физические основы функционирования цифровых датчиков,
12. Физические основы функционирования аналоговых датчиков,
13. Теоретические основы реализации простого (релейного) регулятора.
14. Теоретические основы реализации пропорционального регулятора
15. Теоретические основы реализации ПД - регулятора
16. Физические основы конструирования манипуляционной системы
17. Современные технологии в профессиональной сфере деятельности по компетенции «Мобильная робототехника
18. Основы LabVIEW
19. Основы работы с myRIO
20. Управление двигателем в LabVIEW.
21. Движения по показаниям гироскопа.
22. Движение по линии с использованием датчика света
23. Управление мобильным роботом с дифференциальным приводом
24. Управление манипулятором для захвата объектов.
25. Техническое зрение
26. Организация трансляции web-камеры на пульт оператора
27. Использование бинарного кодирования для распознавания образов
28. Чтение штрих- и QR-кодов
29. Организация и проведение демонстрационного экзамена по стандартам Ворлдскиллс Россия
30. Оценка квалификации обучающегося (выпускника)
31. Демонстрационный чемпионат, выполнение конкурсного задания.
32. Процедура оценивания конкурсных заданий.

Примеры практических задач

Практические задачи предполагают создание полнофункциональной конструкции и программного кода, позволяющего демонстрировать ее работоспособность

1. Сконструировать мобильную платформу с полноприводным механизмом
2. Сконструировать шагающую мобильную платформу
3. Сконструировать захватный механизм мобильного робота
4. Запрограммировать мобильную платформу на выполнение следующих действий:
 - a. Движение вперед
 - b. Движение назад
 - c. Движение на заданное расстояние
5. Запрограммировать мобильную платформу на выполнение следующих действий:

- d. Разворот на колесе
 - e. Разворот на места
 - f. Поворот по дуге
6. Написать программу, останавливающую робота, если он находится перед препятствием.
 7. Написать программу, позволяющую роботу двигаться по направляющей.
 8. Реализовать возможность робота перемещать объекты