

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г. И. Носова»
Многопрофильный колледж

**Методические указания
по подготовке к сдаче
демонстрационного экзамена
для обучающихся
специальности 09.02.01 Компьютерные системы и комплексы
квалификация: специалист по компьютерным системам**

Магнитогорск, 2024

Предметно-цикловой комиссией
«Информатика и вычислительная
техника»
Председатель Т.Б.Ремез
Протокол № 5 от «31» января 2024г.

Методической комиссией МпК
Протокол № 3 от «21» февраля
2024г.

Составители:

Разработчик (и):

преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный
колледж

Т.Б. Ремез

преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный
колледж

А.П. Иванченко

преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный
колледж

Н.А.Криворучко

Методические указания разработаны на основе ФГОС СПО по
специальности 09.02.01 Компьютерные системы и комплексы,
утвержденного приказом Министерства просвещения Российской
Федерации от 25 мая 2022 г. N 362.

СОДЕРЖАНИЕ

1 ОБЩИЕ ПОЛОЖЕНИЯ	5
2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ДЕМОНСТРАЦИОННОМУ ЭКЗАМЕНУ	9
3 ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	10

1 ОБЩИЕ ПОЛОЖЕНИЯ

Демонстрационный экзамен направлен на определение уровня освоения выпускником материала, предусмотренного образовательной программой, и степени сформированности профессиональных умений и навыков путем проведения независимой экспертной оценки выполненных выпускником практических заданий в условиях реальных или смоделированных производственных процессов.

Демонстрационный экзамен направлен на контроль освоения следующих основных видов деятельности и соответствующих им общих и профессиональных компетенций (согласно КОД 09.02.01-1-2024 Том 1 <https://bom.firpo.ru/file/9777/%D0%9A%D0%9E%D0%94%2009.02.01-1-2024%20%D0%A2%D0%BE%D0%BC%201.pdf>):

Вид деятельности (вид профессиональной деятельности)	Перечень оцениваемых ОК, ПК	Перечень оцениваемых умений, навыков (практического опыта)
<i>ИНВАРИАНТНАЯ ЧАСТЬ КОД</i>		
Проектирование цифровых систем	ПК: Анализировать требования технического задания на проектирование цифровых систем	Умение: применять методы анализа требований
		Умение: применять рекомендуемые и руководящие материалы на разрабатываемые цифровые системы
	ПК: Разрабатывать схемы электронных устройств на основе интегральных схем разной степени интеграции в соответствии с техническим	Практический опыт: моделирования цифровых устройств в специализированных программах
	с	Практический опыт: создания принципиальных схем в специализированных программах

	заданием	Практический опыт: монтажа печатных плат макетов устройств
	ПК: Оформлять техническую документацию на проектируемые устройства	Умение: разрабатывать рабочие чертежи в соответствии с требованиями стандартов организации, национальных стандартов и технических регламентов Умение: использовать прикладные программы для разработки конструкторской документации
Проектирование управляющих программ компьютерных систем и комплексов	ПК: Проектировать, разрабатывать и отлаживать программный код модулей управляющих программ	Умение: применять выбранные языки программирования для написания программного кода; Умение: использовать выбранную среду программирования и средства системы управления базами данных Умение: применять методы и приемы отладки программного код Умение: проводить оценку работоспособности программного продукта
	ПК: Выполнять интеграцию модулей в управляющую	Умение: выполнять процедуры сборки программных модулей и компонент в

	программу	программный продукт
		Практический опыт: подключения программного продукта к компонентам внешней среды
		Умение: писать программный код процедур интеграции программных модулей
Техническое обслуживание и ремонт компьютерных систем и комплексов	ПК: Проводить контроль параметров, диагностику и восстановление работоспособности цифровых устройств компьютерных систем и комплексов	Практический опыт: разработки и документирования программных интерфейсов
		Умение: применять контрольно-измерительную аппаратуру и специализированные средства для контроля и диагностики цифровых устройств компьютерных систем и комплексов
		Умение: выполнять поиск дефектов и неисправностей цифровых устройств компьютерных систем и комплексов
	ПК: Проверять	Практический опыт: устранения дефектов и замены устройств компьютерных систем и комплексов
		Умение: выполнять

	<p>работоспособность, выполнять обнаружение и устранять дефекты программного кода управляющих программ компьютерных систем и комплексов</p>	<p>инсталляцию, конфигурирование и настройку операционной системы, драйверов, резидентных программ</p>
		<p>Умение: выявлять дефекты и отклонения в функционировании программного обеспечения компьютерных систем и комплексов</p>
		<p>Практический опыт: отладки аппаратно-программных компьютерных систем и комплексов</p>

Для проведения демонстрационного экзамена составляется расписание экзамена и консультаций.

Демонстрационный экзамен по специальности 09.02.01 Компьютерные системы и комплексы может проводиться на базовом или профильном уровнях (по выбору).

Демонстрационный экзамен базового или профильного уровня проводится по решению образовательной организации на основании заявлений выпускников на основе требований к результатам освоения образовательных программ среднего профессионального образования, установленных в соответствии с ФГОС СПО.

Комплект оценочной документации включает комплекс требований для проведения демонстрационного экзамена, перечень оборудования и оснащения, расходных материалов, средств обучения и воспитания, план застройки площадки демонстрационного экзамена, требования к составу экспертных групп, инструкции по технике безопасности, а также образцы заданий.

Задание демонстрационного экзамена включает комплексную практическую задачу, моделирующую профессиональную деятельность и выполняемую в режиме реального времени.

5.2 Типовое задание для демонстрационного экзамена

профильного уровня

5.2.1 Структура и содержание типового задания

Демонстрационный экзамен профильного уровня проводится с использованием единых оценочных материалов, включающих в себя конкретные комплекты оценочной документации (КОД), варианты заданий и критерии оценивания, разрабатываемых оператором. Комплект оценочной документации приведен в <https://bom.firpo.ru/file/9777/%D0%9A%D0%9E%D0%94%2009.02.01-1-2024%20%D0%A2%D0%BE%D0%BC%201.pdf>.

Задание базового уровня состоит из 2 модулей:

Модуль 1: Проектирование цифровых систем

Задание модуля 1:

Проектирование цифровой системы согласно требованиям технического задания.

Время на выполнение: 1 час 30 мин

Сценарий:

Вам необходимо спроектировать цифровую систему согласно требованиям технического задания. Полная реализация данной задачи включает выполнение следующих этапов:

1. Провести анализ требований технического задания на проектирование цифровой системы:

- необходимо проанализировать предложенную техническую задачу и подобрать необходимые комплектующие для ее выполнения, используя нормативные документы и справочные материалы;

- необходимо аргументировать применение используемых нормативных документов;

- необходимо назвать выбранные элементы, их номинал и необходимое количество для реализации задачи.

2. Разработать схему цифровой системы в соответствии с техническим заданием:

- необходимо смоделировать цифровое устройство в специализированной программе;

- создать принципиальную схему цифровой системы с применением специализированных программ;

- на основе разработанной принципиальной схемы необходимо разработать 3D модель будущей печатной платы.

3. Оформить техническую документацию на проектируемые устройства:

- необходимо составить чертеж 3D модели печатной платы;
- используя прикладные программы, оформить техническую документацию для указанного в задании электронного компонента проектируемого устройства

Модуль 2: Проектирование управляющих программ компьютерных систем и комплексов

Задание модуля 2:

Программирование модулей управляющих программ компьютерной системы

Время на выполнение: 1 час

Сценарий:

Вам необходимо осуществить программирование модулей управляющих программ и продемонстрировать способность автоматизированной робототехнической системы выполнять базовые алгоритмы на микропроцессорных архитектурах в автономном режиме.

Полная реализация данной задачи включает выполнение следующих этапов:

1. Разработать и осуществить отладку программного кода модулей управляющих программ:

- разработайте программу для управления двумя моторами так, чтобы автономное устройство могло двигаться по заданной траектории с использованием датчиков линии;
- разработайте программу для обработки данных с углового гироскопа. Отобразите угловые скорости и углы поворота в реальном времени;
- напишите программу для считывания данных с цифрового энкодера, подключенного к контроллеру. Отобразите количество произведенных оборотов и угловую скорость в реальном времени.
- напишите программу для считывания данных сразу с двух дальномерных датчиков.

2. Выполнить интеграцию модулей в управляющую программу:

- создайте программу, которая будет управлять мотором постоянного тока с использованием данных с датчиков расстояния.

В качестве датчиков расстояния можно использовать инфракрасный либо ультразвуковой датчик (например, при получении значений с датчика больше 15 см, мотор крутится по часовой стрелке).

- напишите программу, которая будет генерировать ШИМ- сигнал для управления скоростью мотора постоянного тока. Продемонстрируйте 3

режима скорости: медленное вращение, средняя скорость, быстрое вращение.

- разработайте алгоритм ПИД (Пропорционально-интегрально-дифференцирующий) регулятора для управления мотором постоянного тока с использованием датчика энкодер.

3. Заполнить разделы журнала технического специалиста по мобильной робототехнике. Журнал должен содержать следующие разделы:

- раздел, посвященный каркасу/конструктивному исполнению робота;

- раздел, посвященный электропроводке робота;

- раздел, посвященный управлению движением робота;

- раздел, посвященный управлению объектом с помощью робота;

- раздел, посвященный программированию робота.

Технический журнал, описывающий робота, должен быть представлен двумя документами в форматах PDF и DOCX (Word)

Задание профильного уровня состоит из 3 модулей:

Модуль 1: Проектирование цифровых систем

Задание модуля 1:

Проектирование цифровой системы согласно требованиям технического задания.

Время на выполнение: 1 час 30 мин

Сценарий:

Вам необходимо спроектировать цифровую систему согласно требованиям технического задания. Полная реализация данной задачи включает выполнение следующих этапов:

1. Провести анализ требований технического задания на проектирование цифровой системы:

- необходимо проанализировать предложенную техническую задачу и подобрать необходимые комплектующие для ее выполнения, используя нормативные документы и справочные материалы;

- необходимо аргументировать применение используемых нормативных документов;

- необходимо назвать выбранные элементы, их номинал и необходимое количество для реализации задачи.

2. Разработать схему цифровой системы в соответствии с техническим заданием:

- необходимо смоделировать цифровое устройство в

специализированной программе;

- создать принципиальную схему цифровой системы с применением специализированных программ;

- на основе разработанной принципиальной схемы необходимо разработать 3D модель будущей печатной платы.

3. Оформить техническую документацию на проектируемые устройства:

- необходимо составить чертеж 3D модели печатной платы;

- используя прикладные программы, оформить техническую документацию для указанного в задании электронного компонента проектируемого устройства

Модуль 2: Проектирование управляющих программ компьютерных систем и комплексов

Задание модуля 2:

Программирование модулей управляющих программ компьютерной системы

Время на выполнение: 1 час

Сценарий:

Вам необходимо осуществить программирование модулей управляющих программ и продемонстрировать способность автоматизированной робототехнической системы выполнять базовые алгоритмы на микропроцессорных архитектурах в автономном режиме.

Полная реализация данной задачи включает выполнение следующих этапов:

1. Разработать и осуществить отладку программного кода модулей управляющих программ:

- разработайте программу для управления двумя моторами так, чтобы автономное устройство могло двигаться по заданной траектории с использованием датчиков линии;

- разработайте программу для обработки данных с углового гироскопа. Отобразите угловые скорости и углы поворота в реальном времени;

- напишите программу для считывания данных с цифрового энкодера, подключенного к контроллеру. Отобразите количество произведенных оборотов и угловую скорость в реальном времени.

- напишите программу для считывания данных сразу с двух дальномерных датчиков.

2. Выполнить интеграцию модулей в управляющую программу:

- создайте программу, которая будет управлять мотором постоянного тока с использованием данных с датчиков расстояния.

В качестве датчиков расстояния можно использовать инфракрасный либо ультразвуковой датчик (например, при получении значений с датчика больше 15 см, мотор крутится по часовой стрелке).

- напишите программу, которая будет генерировать ШИМ- сигнал для управления скоростью мотора постоянного тока. Продемонстрируйте 3 режима скорости: медленное вращение, средняя скорость, быстрое вращение.

- разработайте алгоритм ПИД (Пропорционально-интегрально-дифференцирующий) регулятора для управления мотором постоянного тока с использованием датчика энкодер.

3. Заполнить разделы журнала технического специалиста по мобильной робототехнике. Журнал должен содержать следующие разделы:

- раздел, посвященный каркасу/конструктивному исполнению робота;

- раздел, посвященный электропроводке робота;

- раздел, посвященный управлению движением робота;

- раздел, посвященный управлению объектом с помощью робота;

- раздел, посвященный программированию робота.

Технический журнал, описывающий робота, должен быть представлен двумя документами в форматах PDF и DOCX (Word)

Модуль 3: Техническое обслуживание и ремонт компьютерных систем и комплексов

Задание модуля 3: Техническое обслуживание и ремонт компьютерных систем и комплексов

Время на выполнение: 1 час

Сценарий:

В робототехническую систему и программный код управляющих систем вносятся неисправности. Необходимо провести диагностику, выявить неисправности и привести робототехническую систему в рабочее состояние.

Полная реализация данной задачи включает выполнение следующих этапов:

1. Провести диагностику и выявить неисправность в работе цифровых устройств компьютерных систем и комплексов:

- необходимо обнаружить и продемонстрировать неисправность в работе робототехнической системы;

- по результатам диагностики необходимо устранить причину неисправности, при необходимости – произвести замену устройств робототехнической системы.

Результатом работы является демонстрация передвижения мобильного робота, например, необходимо продемонстрировать способность распознавания штрих-кода роботизированной мобильной платформой, посредством использования «камеры».

Необходимо продемонстрировать возможность робота распознавать штрих или QR кода и выводить на экран расшифровку в заданной кодировке. Необходимо продемонстрировать распознавание 3х разных кодов.

2. Проверка работоспособности, обнаружение и устранение дефектов программного кода управляющих программ компьютерных систем и комплексов:

- необходимо обнаружить и продемонстрировать дефект программного кода управляющей программы;

- произвести отладку программного кода и продемонстрировать передвижения мобильного робота согласно заданию.

Результатом работы является демонстрация передвижения мобильного робота, например, необходимо продемонстрировать способность мобильной роботизированной системы к следованию за предметом. Робот должен автоматически распознавать и следовать за движущимся объектом, используя визуальные датчики или датчики движения

5.2.2 Оснащение рабочего места для проведения демонстрационного экзамена по типовому заданию

Материально-техническая база соответствует инфраструктурному листу КОД 09.02.01-1-2024 Том 1.

5.3 Критерии оценки выполнения задания демонстрационного экзамена

Процедура оценивания результатов выполнения заданий демонстрационного экзамена осуществляется членами экспертной группы по 80-балльной системе в соответствии с требованиями комплекта оценочной документации.

Распределение баллов по критериям оценивания демонстрационного экзамена базового и профильного уровней представлена в таблицах.

№ п/п	Модуль задания (вид деятельности, вид профессиональной деятельности)	Критерий оценивания ⁴	Баллы
1	Проектирование цифровых систем	Анализ требований технического задания на проектирование цифровых систем	4,00
		Разработка схем электронных устройств на основе интегральных схем разной степени интеграции в соответствии с техническим заданием	13,00
		Оформление технической документации на проектируемые устройства	9,00
2	Проектирование управляющих программ компьютерных систем и комплексов	Проектирование, разработка и отладка программного кода модулей управляющих программ	14,00
		Выполнение интеграции модулей в управляющую программу	10,00
ИТОГО			50,00

№ п/п	Модуль задания (вид деятельности, вид профессиональной деятельности)	Критерий оценивания ⁵	Баллы
1	Проектирование цифровых систем	Анализ требований технического задания на проектирование цифровых систем	4,00
		Разработка схем электронных устройств на основе интегральных схем разной степени интеграции в соответствии с техническим заданием	13,00
		Оформление технической документации на проектируемые устройства	9,00
2	Проектирование управляющих программ компьютерных систем и комплексов	Проектирование, разработка и отладка программного кода модулей управляющих программ	14,00
		Выполнение интеграции модулей в управляющую программу	10,00
3	Техническое обслуживание и ремонт компьютерных систем и комплексов	Проведение контроля параметров, диагностики и восстановление работоспособности цифровых устройств компьютерных систем и комплексов	15,00
		Проверка работоспособности, обнаружение и устранение дефектов программного кода управляющих программ компьютерных систем и комплексов	15,00
ИТОГО			80,00

Необходимо осуществить перевод количества баллов в оценки «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Перевод полученного количества баллов в оценки осуществляется государственной экзаменационной комиссией с обязательным присутствием главного эксперта.

Перевод баллов в оценку может быть осуществлен на основе таблицы:

Оценка ГИА	«2»	«3»	«4»	«5»
Отношение полученного количества баллов к максимально возможному (в	0,00 - 19,99%	20,00 - 39,99%	40,00 - 69,99%	70,00 - 100,00%

процентах)				
------------	--	--	--	--

Баллы выставляются в протоколе проведения демонстрационного экзамена, который подписывается каждым членом экспертной группы и утверждается главным экспертом после завершения экзамена для экзаменационной группы.

При выставлении баллов присутствует член ГЭК, не входящий в экспертную группу, присутствие других лиц запрещено.

Подписанный членами экспертной группы и утвержденный главным экспертом протокол проведения демонстрационного экзамена далее передается в ГЭК для выставления оценок по итогам ГИА.

Оригинал протокола проведения демонстрационного экзамена передается на хранение в образовательную организацию в составе архивных документов.

Статус победителя, призера чемпионатов профессионального мастерства, проведенных Агентством (Союзом «Агентство развития профессиональных сообществ и рабочих кадров «Молодые профессионалы (Ворлдскиллс Россия)») либо международной организацией «WorldSkills International», в том числе «WorldSkills Europe» и «WorldSkills Asia», и участника национальной сборной России по профессиональному мастерству по стандартам «Ворлдскиллс» выпускника по профилю осваиваемой образовательной программы среднего профессионального образования засчитывается в качестве оценки «отлично» по демонстрационному экзамену в рамках проведения ГИА по данной образовательной программе среднего профессионального образования.

2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ДЕМОНСТРАЦИОННОМУ ЭКЗАМЕНУ

Проектирование цифровых систем

1. В среде САПР Multisim создайте схемный файл для испытания *аналого-цифрового преобразователя* с ЦАП (рис. 36.4) и установите в диалоговых окнах компонентов их параметры или режимы работы. Скопируйте схему (рис. 36.4) в отчет. В схему (рис. 36.4) включены собственно библиотечный 8-разрядный АЦП (ADC); источники опорного напряжения E1 и E2 (подключены к входам Vref+ и Vref- АЦП); генератор E4 для синхронизации работы (подключен к входу SOC) и разрешения (вход OE) на выдачу двоичной информации на выходы D0, ..., D7 АЦП, с которыми соединены входы логического анализатора XLA1 и пробники X0, ..., X7; функциональный генератор XFG1 в качестве источника входного сигнала $u_{вх}$ (подключен к входу Vin); ЦАП (DAC) и осциллограф XSC1. Выход EOC служит для передачи двоичной информации АЦП, например на ЭВМ.

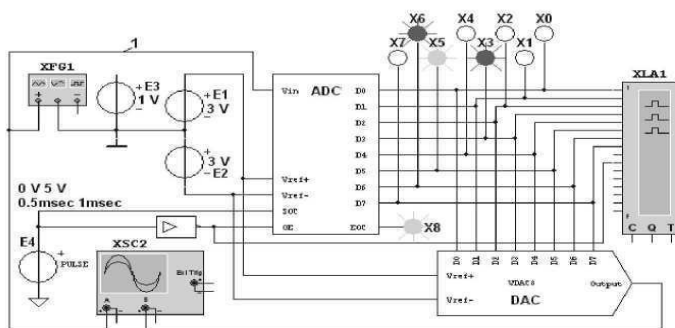


Рис. 36.4

2. Выполните моделирование работы схемы устройства и проверьте её работоспособность с помощью виртуальных инструментов и приборов.

2.1 **Исследовать** процесс преобразования входного напряжения треугольной формы в цифровые коды, а затем с помощью ЦАП — в ступенчатое напряжение, аппроксимирующее напряжение $u_{вх}$. Для этого:

- **удалить** провод, соединяющий выход генератора E3 с входом Vin АЦП, и **восстановить** провод 1, соединяющий выход «+» функционального генератора XFG1 с входом Vin АЦП (см. рис. 36.4);

- **установить** параметры генератора XFG1 (рис. 36.5я): напряжение треугольной формы со скажностью $N=99$ и амплитудой 1 В (диапазон от -1 В до 0,98 В) и его частоту/ $f_r = 50$ Гц;

- **запустить** программу моделирования АЦП;

- **получить и скопировать** в отчет осциллограмму входного

напряжения $u_{вх}$, осциллограмму ступенчатого напряжения $u_{вых(цап)}$ с выхода ЦАП (см. рис. 36.5б) и временные диаграммы сигналов с выходов D0,..., D7 АЦП, поступающих на входы логического анализатора XLA1 и являющихся двоичными эквивалентами дискретных отсчетов $u_{вх}(k\Delta t)$ входного напряжения (рис. 36.6);

– воспользовавшись визирными линиями, **провести анализ** формирования напряжения $u_{вых(цап)}$, аппроксимирующего входное напряжение $u_{вх}$, в частности измерить напряжение и высоту его ступеней в разные моменты преобразования (с интервалом в 1 мс в моменты положительного перепада тактового импульса синхронизации) и сравнить их с отсчетами $u_{вх}(k\Delta t)$ напряжения $u_{вх}$.

Так, при частоте синхронизации $f_c = 1$ кГц и частоте пилообразного напряжения $f_r = 50$ Гц образовалось на выходе ЦАП двадцать ступеней напряжения $u_{вых(цап)}$, средняя высота которых равна $U_{СТ} \approx 93,7$ мВ при расчетном значении $\Delta u = u_{вх,MAX} / (N + 1) = 1,98/21 = 94$ мВ. Первая ступень высотой 66 мВ сформировалась по истечении 0,5 мс с момента включения моделирования при уровне входного напряжения $u_{вх} = -93,4$ мВ, вторая — при $u_{вх} = -0,849$ В высотой 93,75 мкВ и т.д.

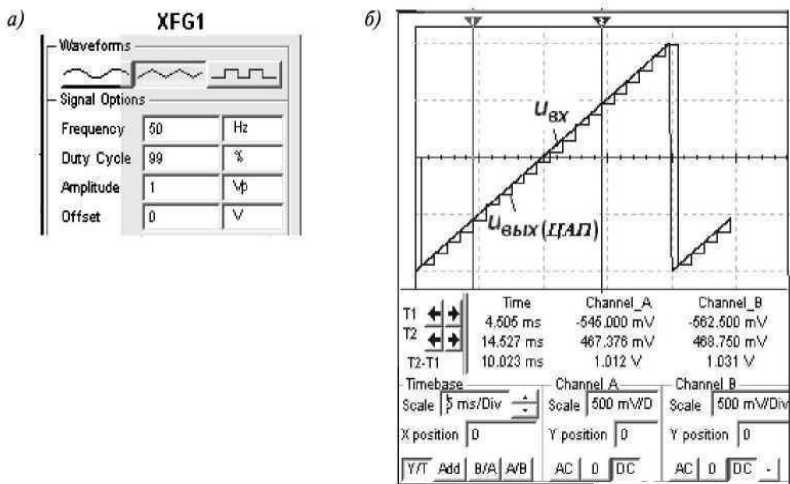


Рис. 36.5

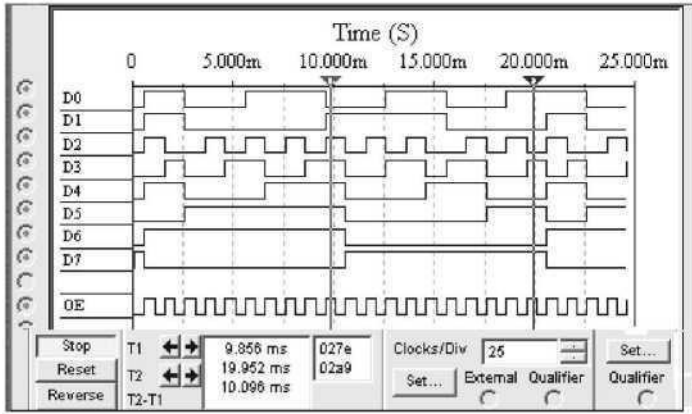


Рис. 36.6

2.2 **Исследовать** процесс преобразования АЦП входного синусоидального напряжения в цифровые коды, а затем с помощью ЦАП — в ступенчатое напряжение.

С этой целью:

- **щелкнуть мышью** на кнопке «Синусоидальное напряжение» генератора **XFG1** (см. рис. 36.5a) и **установить** частоту напряжения $f_r = 25$ Гц, а затем, при остановке моделирования, $f_r = 5$ Гц с изменением времени развертки лучей осциллографа с 10 мс/дел на 50 мс/дел. **Сместить** вверх на 0,6 деления осциллограмму входного напряжения и $u_{вх}$ (рис. 36.7);

- **измерить** напряжение $u_{вых(ЦАП)}$ и высоту его ступеней в разные моменты преобразования и **сравнить** их с отсчетами напряжения $u_{вх}(k\Delta t)$ входного напряжения $u_{вх}$ для моментов положительного перепада тактового импульса синхронизации.

Двоичные эквиваленты отсчетов напряжения $u_{вх}(k\Delta t)$ с выходов АЦП преобразуются с помощью ЦАП в аналоговый ступенчатый сигнал $u_{вых(ЦАП)}$ (см. рис. 36.7). При этом с уменьшением частоты сигнала увеличивается число ступеней и преобразованная кривая хорошо аппроксимирует входной сигнал. Высота ступеней переменная, от 46 мВ до 141 мВ, так как интервал дискретизации Δt при заданной частоте синхронизации постоянный. Особенно заметны верхняя и нижняя ступени с отклонением от амплитуды входного напряжения приблизительно на 15,5 мВ, так как на интервалах дискретизации около амплитуд скорость изменения напряжения минимальная.

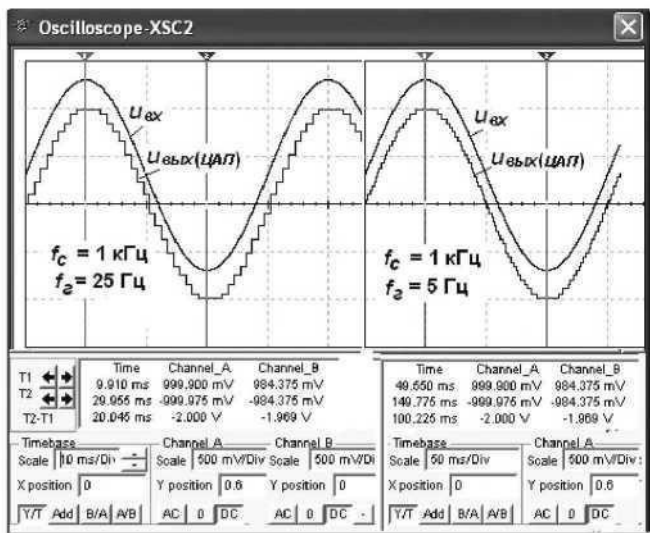


Рис. 36.7

3. Создайте конструкторские документы «схема электрическая принципиальная» и «перечень компонентов» средствами САПР.

4. Результат представьте в виде файла (.docx), содержащего:
- результаты тестирования (таблицы состояний, скриншоты показаний приборов и осциллографов);

- схему электрическую принципиальную;
- перечень элементов.

5. По электрической принципиальной схеме устройства подберите ИМС для реализации устройства, используя справочные источники. Оформите результаты поиска таблицей, в которой приведен перечень элементов, их количество, габаритные размеры и способ установки на печатную плату.

6. Выполните конструкторские расчеты: рассчитайте размеры печатной платы электронного узла (используя данные, полученные в п.5), параметры печатного монтажа, надежность узла, потребляемую мощность и ударопрочность. Оформите результаты расчетов.

7. Разработайте топологию печатной платы (одно- или двухстороннюю) и 3D модель прототипа печатной платы устройства с использованием САПР по рассчитанным размерам.

8. Оформите результаты в виде файла (.docx), содержащего:

- компоновку печатной платы в масштабе 1:1;
- топологию печатной платы с шелкографией в масштабе

1:1;

- 3D модель прототипа и/или фото прототипа.
- 9. Изучите требования техники безопасности при выполнении сборочных работ с применением пайки.
- 10. Выполните сборку и пайку прототипа проектируемого устройства.

Проектирование управляющих программ компьютерных систем и комплексов

Задание 1. Разработать, отладить и моделировать работу программы для микроконтроллера

для управления семисегментным индикатором в онлайн сервисе Tinkercad.

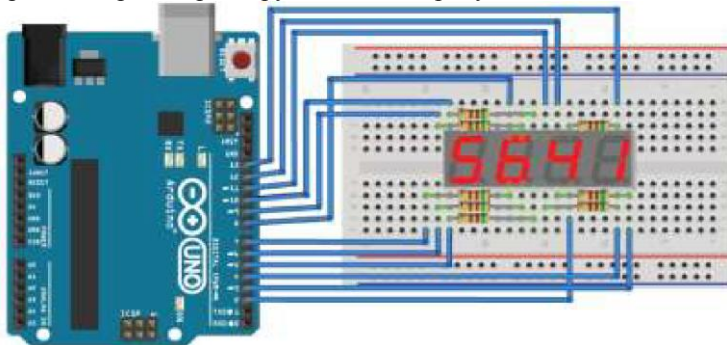
В этом задании необходимо подключить к МК Arduino 4-разрядную семисегментную

матрицу. Необходимые компоненты:

- контроллер Arduino UNO R3;
- плата для прототипирования;
- 4-разрядная семисегментная матрица;
- резистор 510 Ом – 8 штук;
- провода.

Матрица 4-разрядная из семисегментных индикаторов состоит из четырех семисегментных индикаторов и предназначена для одновременного вывода на матрицу 4 цифр, также есть возможность вывода десятичной точки.

Схема подключения 4-разрядной матрицы на 7-сегментных индикаторах к микроконтроллеру показана на рисунке:



На индикатор выводим дату рождения в формате ДДММ.

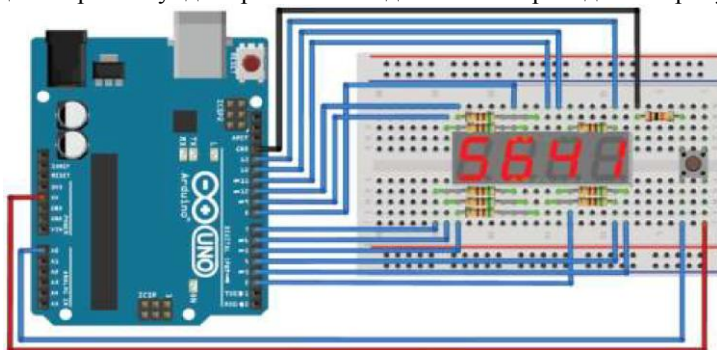
Задание 2. Собрать схему МПС на МК Ардуино с подключенным семисегментным индикатором, загрузить отлаженную программу и продемонстрировать работоспособность схемы и программы. На индикаторе отображается дата рождения.

Задание 3. Разработать, отладить и моделировать работу программы для микроконтроллера для вывода показаний таймера (секундомера) на семисегментный индикатор в онлайн сервисе Tinkercad.

Необходимые компоненты:

- контроллер Arduino UNO R3;
- плата для прототипирования;
- 4-разрядная семисегментная матрица;
- резистор 510 Ом – 8 штук;
- резистор 10 кОм – 1 штука;
- тактовая кнопка – 1 штука;
- провода.

Внести в предыдущую схему изменения, подключив тактовую кнопку для старта секундомера. Схема подключения приведена на рисунке:



Задание 4. Собрать схему МПС на МК Ардуино с подключенным семисегментным индикатором и кнопкой, загрузить отлаженную программу и продемонстрировать работоспособность схемы и программы. На индикаторе отображаются показания секундомера, старт и стоп осуществляется кнопкой.

Задание 5. Выполните модернизацию базового робота: добавьте способность за наименьшее время найти выход из лабиринта. Пример лабиринта представлен на рис. 12.1, где выход из лабиринта обозначен перечеркнутым кругом. Все проходы лабиринта проходимы по ширине и допускают беспрепятственную возможность разворота робота. В поисках выхода робот не видит лабиринт целиком и не имеет о нем дополнительной информации.

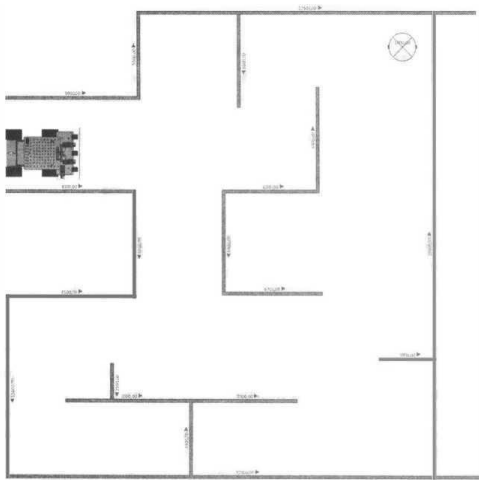


Рис. 12.1. Робот в лабиринте

Для достижения поставленной цели нам потребуется решить следующие задачи:

- Разобраться, как можно установить наличие препятствий (с помощью каких датчиков и решений).
- С учетом выбранных датчиков и принятых технических решений создать алгоритм движения, гарантирующий прохождение лабиринта.
- На основании созданного алгоритма составить программу, позволяющую роботу преодолеть лабиринт.
- Реализовать разработанного робота, провести его тестирование и добиться стабильной работы.

Способ обхода лабиринта

Известен принцип, согласно которому, если при движении в лабиринте придерживаться одной его стороны (стенки): левой или правой, то выход обязательно будет достигнут. Это верно для лабиринтов с выходом наружу. Графически возможный путь робота показан на рис. 12.2 (вдоль правой стенки лабиринта) и 12.3 (вдоль его левой стенки).

Следует обратить внимание, что по пути, показанному на рис. 12.3, робот достигнет выхода быстрее, но он не может знать заранее, где выход, и какой путь короче. Конечно, это может знать оператор и выбрать для робота выигрышный режим («направо») или («налево») перед входом в лабиринт. Впрочем, как правило, в соревнованиях роботов такая подсказка оператора недопустима.

Составим алгоритм обхода лабиринта для робота, оснащенного уже знакомым нам ультразвуковым датчиком, установленным на его

поворотной голове (рис. 12.4). Пусть робот в этот раз будет совершать обход лабиринта по правой его стороне. При этом он должен всегда держаться на небольшом расстоянии (3-4 см) от правой стенки и адекватно реагировать на изменение обстановки. Добавим к этому расстоянию половину ширины робота и получим нижнюю границу `Dist_Right`. Робот измеряет расстояния с двух сторон: справа и прямо, затем анализирует полученные результаты, выполняет соответствующие движения и снова останавливается для сбора данных. Если расстояние до препятствия справа находится в пределах от 5 до 7 см, значит, поворот не нужен, и он немного проезжает вперед (если при этом нет препятствия спереди).

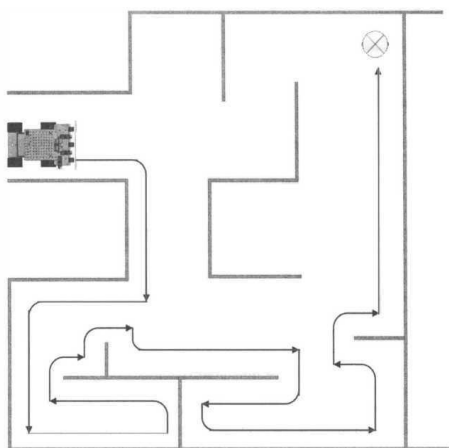


Рис. 12.2. Путь робота вдоль правой стенки

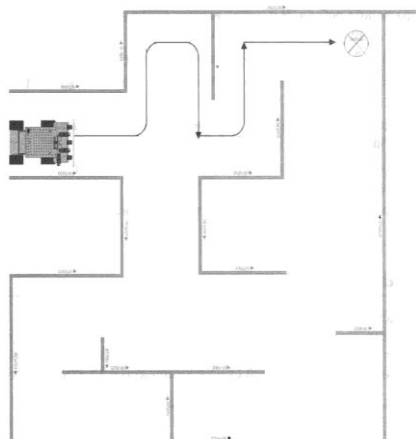


Рис. 12.3. Путь робота вдоль левой стенки

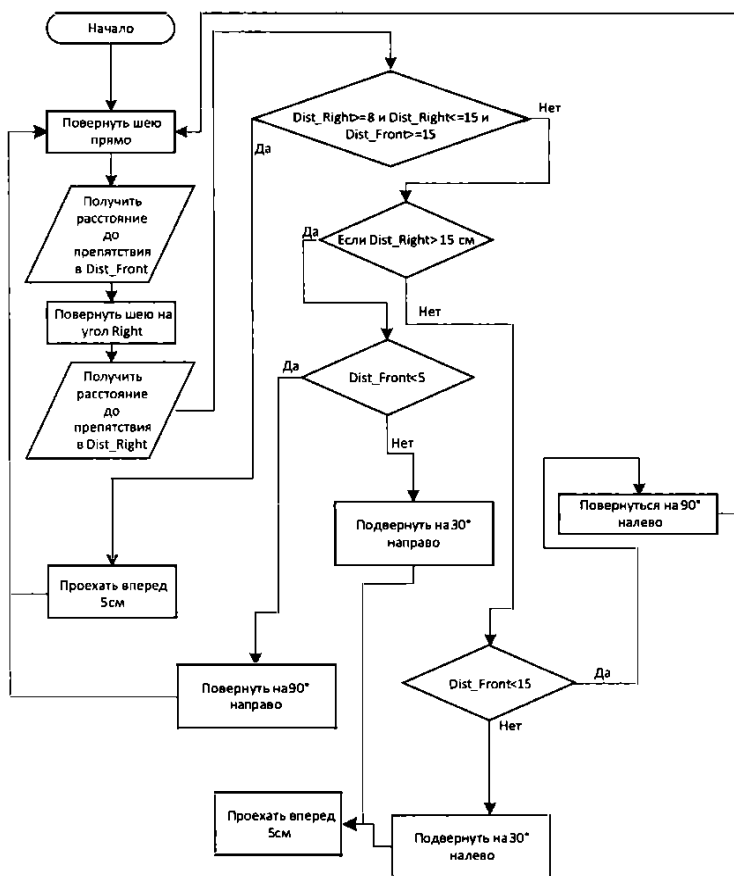


Рис. 12.4. Алгоритм движения вдоль правой стенки с помощью ультразвукового датчика

Если дистанция справа стала больше 7 см, значит, робот стал отходить от правой стены, и нужно начать поворот вправо. Если робот уперся в стену, ему следует повернуть на месте влево и снова получить данные.

Программа

Соответствующая рассмотренному алгоритму программа представлена в листинге 12.1. В ней много настраиваемых значений, которые определяются экспериментально. ^о, например, углы поворота головы. Дело в том, что механически установить голову так, чтобы значение 90° соответствовало точно переднему положению головы, довольно трудно, поэтому мы определяем этот угол экспериментально, изменяя значения, передаваемые на сервомотор, и следя за положением головы робота. Временные константы, связанные с движением и поворотами, зависят от дорожного покрытия и заряда аккумуляторов. Например,

поворот на ковровом покрытии займет больше времени, чем на линолеуме или ламинате, а на гладком и скользком полу возможно инерционное скольжение робота. Следите за зарядом аккумуляторов — если они сильно разрядятся, робот станет передвигаться медленнее, а это приведет к неверным значениям поворотов. В любом случае, значения переменных `left_ang`, `front_ang` и `right_ang`, отвечающих за поворот головы, а также значения `time_90`, `time_180`, `time_10cm` и `time_7cm`, отвечающих за время, требуемое на производство роботом простейших движений, настраиваются самостоятельно.

Листинг 12.1. Программа движения вдоль правой стенки с помощью ультразвукового датчика

```

// Подключаем библиотеку управления сервомоторами.
#include <Servo.h>
// Подключаем библиотеку, управляющую моторами.
#include "motor.h"
// Подключаем библиотеку, управляющую дальномером.
#include "sonar.h"
// Создаем сервомотор поворота головы.
Servo neck;
// Константы - постоянные значения для уточнения углов.
const int left_ang=168;
// Без поправки 160
const int front_ang=98;
// Без поправки 90
const int right_ang=28;
// Без поправки 20
// Временные константы служат для точного задания времени на поворот, разворот,
движение вперед.
// в миллисекундах.
const int time_90=390;
const int time_180=750;
const int time_10cm=220;
const int time_7cm=120;
void setup()
{
// Инициализируем дальномер Trig = 13, Echo = 12.
Sonar_init(13, 12);
// Инициализируем сервомотор, управление 9-м портом.
neck.attach(14);
// Переменные - номера контактов (пинов) Arduino.
// Для левых и правых моторов машинки.
setup_motor_system(2, 3, 4, 5);
_stop(); // Двигатели остановлены.
}
// Основная программа.
void loop()
{
// Создаем переменные для хранения двух
// дистанций до препятствий спереди, справа.
int Dist_left, Dist_front, Dist_right;
_stop();
// Ждем, т. к. поворот занимает небольшое время.

```

```

delay(250);
// Записываем расстояние до препятствия впереди.
Dist_front = Sonar(40);
// Поворачиваем голову направо. neck.write(right_ang);
// Ждем, т. к. поворот заснимает небольшое время.
delay(250);
// Записываем расстояние до препятствия впереди.
Dist_right = Sonar(40);
neck.write(front_ang);
// Если условия позволяют двигаться прямо
if((Dist_right>=8)&&(Dist_right<=15)&&(Dist_front>=15)) {
forward();
// едем вперед.
delay(time_10cm);
}
else
{
// если появился поворот направо
if(Dist_right>15)
{
// если абсолютно уперлись в стену!
// узкий правый поворот.
if(Dist_front<5)
{
right();
// поворачиваем направо.
delay(time_90);
}
}
else
// нормальный правый поворот.
{
// поворот направо в движении.
right();
delay(time_90/3);
forward();
// едем вперед 7 см.
delay(time_90/3);
}
}
else
// если уперлись в стену.
if(Dist_front<15)
{
left();
// поворачиваем налево.
delay(time_90);
}
}
else
{
// поворот налево в движении.
left ();
delay(time_90/3);
forward ();
// едем вперед 5 см.
delay(time_90/3);
}
}

```

}
}
}

Сравнение и выбор датчиков

Робот, определяя расстояния с помощью ультразвукового датчика расстояния HC-SR04, часто останавливается и находит выход из лабиринта относительно медленно. Можно попытаться ускорить его работу, задействовав дополнительные датчики.

В качестве таких датчиков могут быть использованы детекторы препятствия и инфракрасные датчики расстояния.

Ультразвуковой датчик HC-SR04

Ультразвуковой датчик HC-SR04 использует два порта Arduino. Датчик определяет расстояние от 2 до 400 сантиметров, а функция `sonar(unsigned long Limit)`, может настраиваться на максимально измеряемое расстояние, которое возвращается в сантиметрах.

Время, которое требуется для измерения, изменяется от 1 до 200 миллисекунд и увеличивается с увеличением измеряемого расстояния. Между последовательными измерениями для подавления эхо-эффекта требуется время в 50 миллисекунд. В итоге минимальное количество замеров в секунду — 4, максимальное — 19. Это не много, особенно с учетом того, что для каждого измерения роботу приходится останавливаться, а для измерения расстояния с разных сторон еще и вращать головой. Неудивительно, что прохождение лабиринта при этом происходит относительно медленно.

Инфракрасный детектор препятствия

Инфракрасный датчик (детектор) препятствия, , определяет наличие препятствия по интенсивности отраженного света: если она выше пороговой, на выходе появляется низкий уровень напряжения, а если препятствия нет, то уровень напряжения на выходе высокий. Это важно не перепутать! Датчик не информирует о расстоянии до препятствия, он только сбрасывается в ноль при его наличии.

Расстояние до препятствия, на котором срабатывает датчик, можно регулировать в диапазоне от 3 до 80 см, вращая бегунок переменного резистора (в центре датчика). Плата снабжена двумя информационными светодиодами: красный — наличие питания, зеленый — наличие препятствия. В передней части платы датчика расположены инфракрасный излучатель и приемник (рис. 12.5).

Недостатком этого детектора является зависимость расстояния срабатывания от цвета объекта. Фактически это говорит нам о том, что датчики нужно подстраивать под цвет проходимого лабиринта. Датчик также может срабатывать на несуществующее препятствие при наличии посторонних источников инфракрасного освещения — к таковым относится прямой или отраженный от близлежащих объектов солнечный свет.



Рис. 12.5. Инфракрасный детектор препятствия

Инфракрасный датчик Sharp GP2Y0A21YK

Инфракрасный датчик расстояния Sharp GP2Y0A21 YK, отличается от прочих тем, что выдает данные о расстоянии в виде аналогового сигнала, — его выходное напряжение изменяется в зависимости от расстояния. При этом измеряемое расстояние лежит в диапазоне от 10 до 80 см. Для измерений требуется подключить датчик к питанию 5 вольт, а его выход— к аналоговому входу Arduino, например, A0. Следует учесть, что зависимость показаний датчика от расстояния нелинейная, поэтому для расчета расстояния в сантиметрах нужно использовать специальную функцию обработки. Пример такой функции приведен в листинге 12.2. Время, которое требуется для измерения, составляет около 0,1 миллисекунды, паузы между замерами не нужны. Максимальное количество замеров в секунду — около 10000. Основные недостатки таких датчиков заключаются в их относительно высокой стоимости и малом диапазоне измеряемых расстояний.

Листинг 12.2. Функция обработки показаний датчика расстояния Sharp GP2Y0A21YK

```
int IR_Ranging(int nk)
{
  if((nk>13)&&(nk<22))
  {
    int vole=analogRead(nk) ;
    if(vole<18) vole=18;
    return 5461/(vole - 17) - 2;
  }
  else return 1000;
}
```

Обоснование выбора датчиков препятствия

В связи с тем, что ультразвуковой датчик уже используется в проекте, а инфракрасные детекторы препятствия примерно в 10 раз дешевле и чаще встречаются в продаже, чем инфракрасный датчик расстояния Sharp GP2Y0A21YK, то мы воспользуемся ими.

Впрочем, датчик Sharp GP2Y0A21 YK со счетов мы тоже

сбрасывать не станем, поскольку он практически не дает сбоев, и, если стоимость робота не критична, рекомендуем использовать именно его, немного модифицировав приведенный далее код.

Модуль 3: Техническое обслуживание и ремонт компьютерных систем и комплексов

Задание 1. Модернизация робота

Так как робот часто останавливается и находит выход из лабиринта относительно медленно, можно попытаться ускорить его работу, используя дополнительные датчики — детекторы препятствия.

Монтаж детекторов препятствия

Установим на передний бампер робота три детектора (датчика) препятствия, чего для наших целей вполне достаточно (рис. 12.6-12.9). У нашего робота два детектора будут смотреть вперед и один — вправо. Наличие двух детекторов, направленных вперед, улучшает качество определения препятствий спереди, поскольку сектор определения одного детектора не захватывает всю переднюю зону.

Крепятся датчики на шасси робота при помощи винтов. Перед установкой датчиков на шасси следует подсоединить или припаять к ним провода. Датчики линии, используемые ранее, следует демонтировать, их контакты будут использованы под детекторы препятствия и потребуется провести еще три провода и задействовать один дополнительный порт/пин Arduino.

Желательно также защитить сенсоры от ударов, иначе от столкновений с препятствиями они быстро погнутся и выйдут из строя.

Полную схему робота здесь приводить не имеет смысла — она уже была приведена в предшествующих главах. Изменения коснулись только порядка подключения детекторов. Электропитание +5V они в общем случае получают от модуля драйвера двигателей L298N (рис. 12.10), а для нашего робота — через плату Arduino Sensor Shield v5.0. «Земля» (контакт GND) берется с любого доступного места — например, от того же L298N. Сигнальные выводы подключаются к 7, 8 и 9-му портам Arduino (рис. 12.11). Получать питание 5 В датчики могут также от Arduino, но на некоторых платах Arduino стоят слабые стабилизаторы напряжения, которые не следует перегружать, поэтому 5 В и подается на датчики от L298N.

Фронтальные датчики мы подключаем к портам 8 (датчик S2) и 9 (датчик SB), а датчик, расположенный с правой стороны, к 7-му порту (датчик S1).

Итоговая принципиальная схема подключения датчиков препятствия приведена на рис. 12.12. Расположение контактов на датчиках у разных изготовителей может быть различным, поэтому внимательно следите за названиями контактов. Так, питающий контакт может иметь обозначение VCC или 5V.

После подключения датчиков следует снять с робота колеса, включить его электропитание и настроить расстояние срабатывания датчиков, используя переменные резисторы и зеленые сигнальные светодиоды, которые загораются при обнаружении препятствия.



Рис. 12.6. Робот, три датчика препятствия и крепежные винты



Правый датчик закреплен снизу на одном винте с передним датчиком

Рис. 12.7. Три датчика препятствия размещены на верхней плате шасси

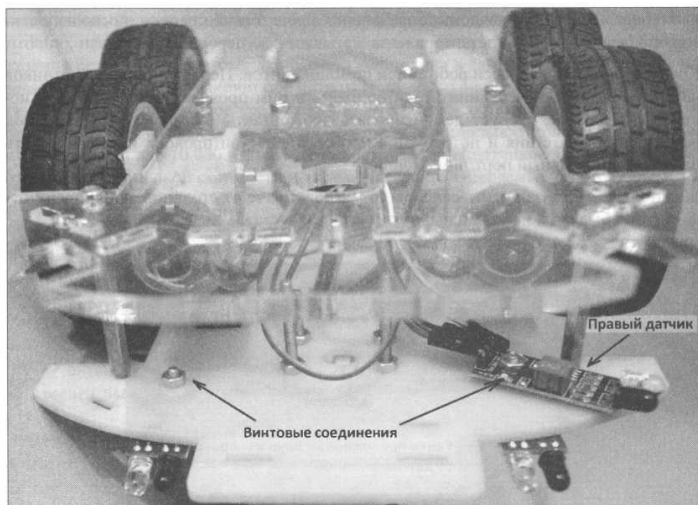


Рис. 12.8. Крепление датчиков препятствия винтами



Рис. 12.9. Пример альтернативной установки датчиков при помощи термоклея

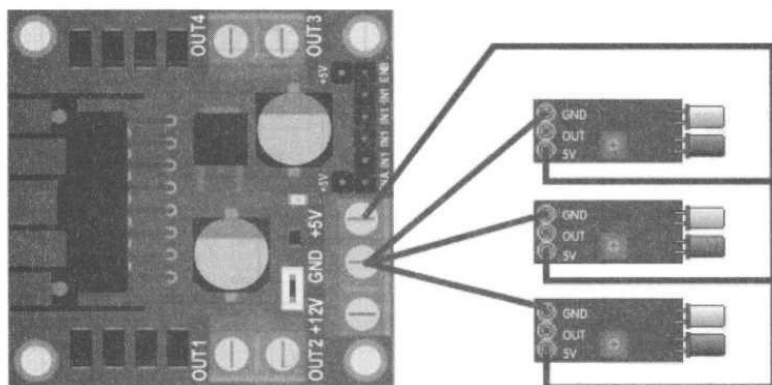


Рис. 12.10. Электропитание датчиков

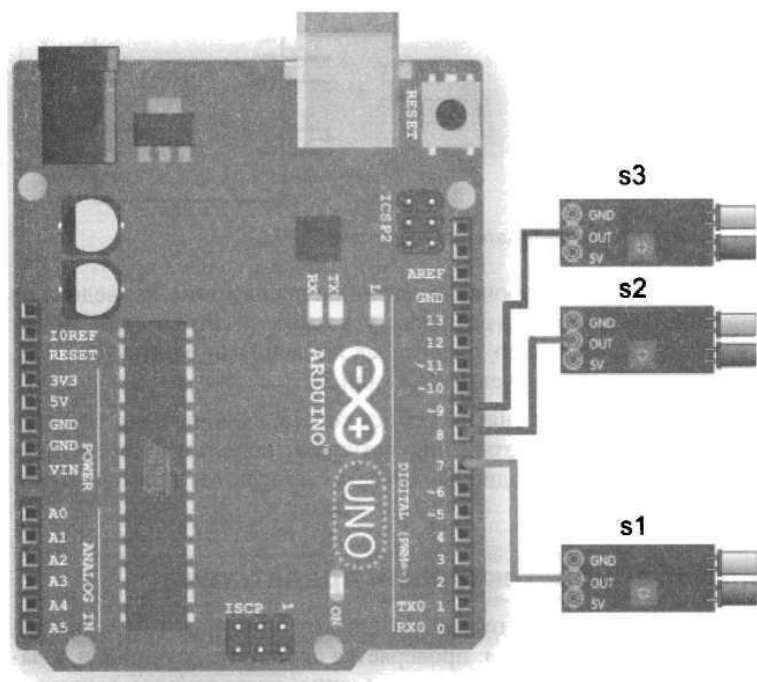


Рис. 12.11. Подключение датчиков к Arduino UNO

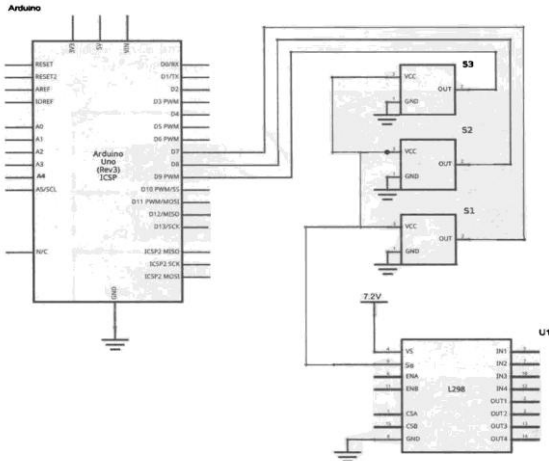


Рис. 12.12. Принципиальная схема подключения датчиков препятствия

Программа для робота с детекторами препятствия

Алгоритм движения робота с тремя детекторами препятствия (рис. 12.13) проще, чем алгоритм с ультразвуковым датчиком: проверяется состояние датчиков и, в зависимости от полученных данных, принимается решение, позволяющее двигаться вдоль правой стены.

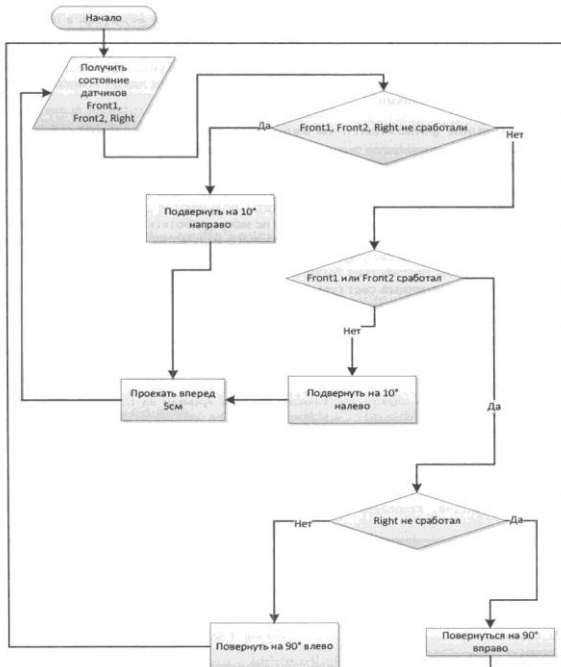


Рис. 12.13. Алгоритм движения робота с тремя детекторами препятствия

Загрузите программу (листинг 12.3). Поэкспериментируйте со значениями переменных времени для подбора точных поворотов. Собственно везде, где в программе стоит функция delay (), создается пауза, в течение которой и выполняется предыдущая команда. Если вам кажется, что можно создать программу лучше, чем та, что приведена в листинге, смело программируйте и пробуйте. Для тестирования программы не обязательно наличие лабиринта — подойдет и любая комната с расставленными препятствиями.

Защитите фотоэлементы датчиков от столкновений!

Настоятельно рекомендую защитить фотоэлементы датчиков от повреждений в результате столкновения робота с препятствиями.

Используемые в проекте детекторы препятствия по-разному реагируют на препятствия различного цвета— лучшие результаты получаются для белых и светлых препятствий. Детекторы препятствия также не могут работать при наличии солнечного света и источников сильного инфракрасного излучения — это связано с особенностью используемых светочувствительных элементов (фототранзисторов). Поэтому важно, чтобы препятствия были однородными по цвету, и в комнату не попадал бы прямой солнечный свет (увы, приходится считаться с этими недостатками детекторов).

Листинг 12.3. *Программа движения робота с тремя детекторами препятствия*

```
// Подключаем библиотеку, управляющего моторами.
#include "motor.h"
// Временные константы служат для точного задания времени на поворот,
разворот,
движение вперед
// в миллисекундах.
const int time_90=390;
const int time_180=750;
const int time_10cm=220;
// Номера портов, к которым подключены датчики препятствия.
const int Front1=8, Front2=9, Right=7;
void setup()
{
// Заносим в переменные номера контактов (пинов) Arduino.
// Для левых и правых моторов машинки.
setup_motor_system(2, 3, 4, 5);
// Остановка.
_stop();
// Инициализируем порты датчиков препятствия.
pinMode (Front1, INPUT);
pinMode (Front2, INPUT);
pinMode(Right, INPUT);
// Устанавливаем скорость передачи данных по кабелю.
// Порт компьютера //Serial.begin(9600);
```

```

}
// Основная програма.
void loop()
{
boolean d_Frontl, d_Front2, d_Right;
d_Frontl = digitalRead(Front1);
d_Front2 = digitalRead(Front2);
d_Right = digitalRead(Right);
// Если ни один датчик не сработал.
if (d_Frontl && d_Front2 && d_Right)
{
forward_right();
// подворот вправо.
delay(time_90 / 9);
forward(); /
/ едем вперед.
delay(time_10cm / 2);
}
else
{
// Если сработал один из передних датчиков и не сработал правый.
if ((!d_Frontl)
//(!d_Front2))
{
// Если не сработал правый датчик.
if (d_Right)
{
// поворачиваем направо на 90 градусов.
right();
delay(time_cm);
}
else
{
// поворачиваем налево на 90 градусов.
left ();
delay(time_10cm);
}
}
else
{
// Если сработал правый датчик.
forward_left();
// подворот влево.
delay(time_90 / 9); f
orward();

```

```
// едем вперед. delay(time_10cm /2);  
}  
}
```

Задание 2. Дополните Журнал технического специалиста по мобильной робототехнике, внося в него изменения, которые появились в результате выполненной в задании 5 модернизации. Журнал должен содержать следующие разделы:

- раздел технического журнала, посвященный каркасу / конструктивному исполнению;
- раздел технического журнала, посвященный электропроводке;
- раздел технического журнала, посвященный управлению движением;
- раздел технического журнала, посвященный управлению объектом;
- раздел технического журнала, посвященный программированию.

Технический журнал, описывающий робота, должен быть представлен двумя документами в форматах PDF и DOCX (Word).

3 ИНФОРМАЦИОННО-МЕДИЦИСКОЕ ОБЕСПЕЧЕНИЕ

Основные источники:

1. Проектирование цифровых устройств : Учебник / А.В. Кистрин [и др.] ; Рязанский государственный радиотехнический университет имени В.Ф. Уткина. - 1. - Москва : ООО "КУРС", 2022. - 352 с. - (Среднее профессиональное образование). - Среднее профессиональное образование. - URL: <https://znanium.com/catalog/document?id=397139>. - URL: <https://znanium.com/cover/1495/1495622.jpg>. - ISBN 978-5-906818-59-1. - ISBN 978-5-16-104714-9. - ISBN 978-5-16-011833-8.
2. Марченко Алексей Лукич (Московский авиационный институт (национальный исследовательский университет)). Электротехника и электроника : В 2 томах Том 2: Электроника; Учебник / А.Л. Марченко, Ю.Ф. Опадчий ; Московский авиационный институт (национальный исследовательский университет). - 1. - Москва : ООО "Научно-издательский центр ИНФРА-М", 2023. - 391 с. - (Высшее образование). - ВО - Бакалавриат. - URL: <https://znanium.com/catalog/document?id=428651>. URL: <https://znanium.com/cover/2006/2006854.jpg>. - ISBN 978-5-16-014295-1. - ISBN 978-5-16-106791-8.
3. Юрков Н. К. Технология производства электронных средств [Электронный ресурс] / Н. К. Юрков ; Юрков Н. К. - 2-е изд., испр., доп. - Санкт-Петербург : Лань, 2022. - 480 с. - Рекомендовано УМО вузов РФ по образованию в области радиотехники, электроники, биомедицинской техники и автоматизации в качестве учебника для студентов вузов, обучающихся по направлению 211000 — «Конструирование и технология электронных средств». - Книга из коллекции Лань - Инженерно-технические науки. - URL: <https://e.lanbook.com/book/211457>. - URL: <https://e.lanbook.com/img/cover/book/211457.jpg>. - ISBN 978-5-8114-1552-6.
4. Конструирование блоков радиоэлектронных средств [Электронный ресурс] / Д. Ю. Муромцев [и др.] ; Муромцев Д. Ю., Белоусов О. А., Тюрин И. В., Курносов Р. Ю. - 2-е изд., стер. - Санкт-Петербург : Лань, 2022. - 288 с. - Книга из коллекции Лань - Инженерно-технические науки. - URL: <https://e.lanbook.com/book/226472>. - URL: <https://e.lanbook.com/img/cover/book/226472.jpg>. - ISBN 978-5-507-44388-8.
5. Шишов, О. В. Программируемые контроллеры в системах промышленной автоматизации : учебник / О.В. Шишов. — Москва : ИНФРА-М, 2022. — 365 с. + Доп. материалы [Электронный ресурс]. —

(Среднее профессиональное образование). - ISBN 978-5-16-015321-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1851436>

6. Зараменских, Е. П. Интернет вещей. Исследования и область применения: монография / Е.П. Зараменских, И.Е. Артемьев. — Москва: ИНФРА-М, 2023. - 188 с. — (Научная мысль). — DOI 10.12737/13342. - ISBN 978-5-16-011476-7. - Текст: электронный. - URL: <https://znanium.com/read?id=416080> – Режим доступа: по подписке.

7. Давыдкин, М. Н. Программирование микроконтроллеров : методические указания / М. Н. Давыдкин. — Москва : МИСИС, 2022. — 176 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/305492>— Режим доступа: для авториз. пользователей.

8. Белугина, С. В. Разработка программных модулей программного обеспечения для компьютерных систем. Прикладное программирование : учебное пособие для спо / С. В. Белугина. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 312 с. — ISBN 978-5-8114-9817-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://reader.lanbook.com/book/296975> — Режим доступа: для авториз. пользователей.

9. Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие для спо / Т. М. Зубкова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 252 с. — ISBN 978-5-8114-9556-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://reader.lanbook.com/book/276419> — Режим доступа: для авториз. пользователей.

10. Соколова, В. В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2023. — 175 с. — (Профессиональное образование). — ISBN 978-5-534-10680-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518008>

11. Белугина, С. В. Архитектура компьютерных систем. Курс лекций / С. В. Белугина. — 2-е изд., стер. — Санкт-Петербург : Лань, 2024. — 160 с. — ISBN 978-5-507-48577-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/356147>— Режим доступа: для авториз. пользователей.

12. Гагарина, Л. Г. Технические средства информатизации : учебное пособие / Л.Г. Гагарина, Ф.С. Золотухин. — 2-е изд., перераб. и доп. — Москва : ИНФРА-М, 2021. — 260 с. — (Среднее профессиональное

образование). — DOI 10.12737/1083293. - ISBN 978-5-16- 016140-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1083293> (дата обращения: 12.05.2023). – Режим доступа: по подписке.

13. Программный ремонт сотовых телефонов Siemens, Fly, Voxtel : практическое пособие / под ред. А. В. Родина и Н. А. Тюнина. - Москва : СОЛОН-Пресс, 2020. - 96 с. - (Серия «Ремонт», выпуск 109). - ISBN 978-5-91359-035-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1858794> (дата обращения: 12.05.2023). – Режим доступа: по подписке.

Дополнительные источники:

1. Пуховский Валерий Николаевич (Южный федеральный университет). Электротехника, электроника и схемотехника. Модуль «цифровая схемотехника» : Учебное пособие / В.Н. Пуховский, М.Ю. Поленов ; Южный федеральный университет. - Ростов-на-Дону : Издательство Южного федерального университета (ЮФУ), 2018. - 163 с. - ВО - Бакалавриат. - URL: <https://znanium.com/catalog/document?id=343877>. - URL: <https://znanium.com/cover/1039/1039797.jpg>. - ISBN 978-5-927-53079-3.

2. Гололобов, В. Н. Схемотехника с программой multisim для любознательных / В. Н. Гололобов. — Санкт-Петербург : Наука и Техника, 2019. — 272 с. — ISBN 978-5-94387-880-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/139132>

3. Скарпино, М. Разработка печатных плат в EAGLE : учебное пособие / М. Скарпино ; перевод с английского А. Э. Бряндинского. — Москва : ДМК Пресс, 2018. — 370 с. — ISBN 978-5-97060-479-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/105829>

4. Макуха, В. К. Микропроцессорные системы и персональные компьютеры : учебное пособие для среднего профессионального образования / В. К. Макуха, В. А. Микерин. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2022. — 156 с. — (Профессиональное образование). — ISBN 978-5-534-12091-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496183>

5. Сажнев, А. М. Микропроцессорные системы: цифровые устройства и микропроцессоры : учебное пособие для среднего профессионального образования / А. М. Сажнев. — 2-е изд., перераб. и доп.

— Москва : Издательство Юрайт, 2022. — 139 с. — (Профессиональное образование). — ISBN 978-5-534-12092-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/496182>

6. Дубков, И. С. Решение практических задач на базе технологии интернета вещей: учебное пособие / И. С. Дубков, П. С. Сташевский, И. Н. Яковина. — Новосибирск: НГТУ, 2017. — 80 с. — ISBN 978-5-7782-3161-0. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118206> — Режим доступа: для авториз. пользователей.

7. Макаров, С. Л. Arduino Uno и Raspberry Pi 3. От схемотехники к интернету вещей / С. Л. Макаров. - Москва : ДМК Пресс, 2019. - 204 с. - ISBN 978-5-97060-730-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2045974> (дата обращения: 30.04.2023). – Режим доступа: по подписке.

8. Микушин, А. В. Программирование микропроцессорных систем на языке C-51 / А. В. Микушин. — Санкт-Петербург : Лань, 2023. — 124 с. — ISBN 978-5-507-45539-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/311828> — Режим доступа: для авториз. пользователей.

9. Вязовик, Н.А.. Программирование на Java : Курс лекций / Н.А. Вязовик — Москва : Интуит НОУ, 2016. — 603 с. — ISBN 978-5-9556-0006-2. — URL: <https://book.ru/book/918118> — Текст: электронный.

11. Чащина Е.А. Обслуживание аппаратного обеспечения персональных компьютеров, серверов, периферийных устройств, оборудования компьютерной оргтехники. – М. ИЦ «Академия», 2018.- 112с

12. Зверева, В. П. Технические средства информатизации : Учебник / В. П. Зверева, А.В. Назаров. - Москва: ООО "КУРС", 2024. - 242 с. - Режим доступа: <https://znanium.ru/read?id=436552> (дата обращения: 19.04.2024) - ISBN 978-5-906818-54-6. - ISBN 978-5-16-105402-4.

13. Осокин, А. Н. Теория информации: учебное пособие для спо / А. Н. Осокин, А.Н. А. Н. Мальчуков. - Москва : Юрайт, 2023. - 208 с. - Режим доступа: <https://urait.ru/viewer/teoriya-informacii-542695#page/1> (дата обращения: 19.04.2024).- ISBN 978-5-534-17296-6.

14. Чистов, Д. В. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. -

Москва:Юрайт,2024. -293 с. – Режим доступа:
<https://urait.ru/viewer/proektirovanie-informacionnyh-sistem-538370#page/1>
(дата обращения: 19.04.2024).- ISBN 978-5-534-16217-2.

Интернет-ресурсы:

1. Интуит – национальный открытый университет. Основы цифровой техники [Электронный ресурс]. – Режим доступа:<https://www.intuit.ru/studies/courses/685/541/info>

Интуит – национальный открытый университет. Введение в цифровую электронику [Электронный ресурс]. – Режим доступа:
<https://intuit.ru/studies/courses/588/444/info>

2. Интуит – национальный открытый университет. Основы САПР [Электронный ресурс]. – Режим доступа:<https://www.intuit.ru/studies/courses/2264/227/info>

3. Интуит – национальный открытый университет. Моделирование, тестирование и диагностика цифровых устройств [Электронный ресурс]. – Режим доступа:
<https://intuit.ru/studies/courses/3440/682/info>

4. Сайт Паяльник. Справочные материалы.[Электронный ресурс]. – Режим доступа: <https://cxem.net/sprav/sprav.php>, свободный. – Загл. с экрана. Яз.рус.

5. Моделирование, тестирование и диагностика цифровых устройств [Электронный ресурс] - <https://www.intuit.ru/studies/courses/3440/682/info> , свободный. – Загл. с экрана. Яз. рус.

6. Радиолюбительские программы, схемы, документация. Справочные материалы. [Электронный ресурс] – Режим доступа:<http://www.texnic.ru/data/index.htm>, свободный. – Загл.с экрана. Яз. Рус.

7. EasyEDA — веб-среда для автоматизированного проектирования электронных устройств
<https://easyeda.com/>

8. Основы микропроцессорной техники - <https://www.intuit.ru/studies/courses/3/3/info>, свободный. – Загл. с экрана. Яз. рус.

9. Организация вычислительных систем [Электронный ресурс] - <https://www.intuit.ru/studies/courses/92/92/info>, свободный. – Загл. с экрана. Яз. рус.

10. Архитектура ЭВМ и язык ассемблера [Электронный ресурс] <https://intuit.ru/studies/courses/535/391/info>

11. TinkerCAD - онлайн-сервис по разработке электронных схем и программирования [Электронный ресурс] <https://www.tinkercad.com/>
12. Официальный сайт компании Arduino на русском языке [Электронный ресурс] <https://arduino.ru/>
13. Git – Book [Электронный ресурс]. – Режим доступа: <https://git-scm.com/book/ru/v2>, свободный. – Загл. с экрана. Яз. рус.
14. Введение в концепцию "интернета вещей" (IoT) [Электронный ресурс]. – Режим доступа: <https://nag.ru/articles/article/107810/vvedenie-v-kontseptsiyu-interneta-veschey-iot-.html>, свободный. – Загл. с экрана. Яз. рус.
15. Официальный сайт компании Studica на русском языке [Электронный ресурс] <https://www.studica.com/>