



**Федеральное агентство морского и речного транспорта
ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»
Московская государственная академия водного транспорта - филиал
Федерального государственного бюджетного образовательного учреждения высшего
образования
«Государственный университет морского и речного флота имени адмирала С.О.
Макарова»
(МГАВТ - филиал ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)**

**Факультет Судомеханический
Кафедра Электрооборудования**



**УТВЕРЖДАЮ
Директор филиала**

**И.Н. Мищенко
«31» августа 2017 г.**

РАБОЧАЯ ПРОГРАММА

дисциплины СЗ.Б.14 «Микропроцессорные системы управления»

Специальность	<u>26.05.07 «Эксплуатация судового электрооборудования и средств автоматики»</u>
Уровень высшего образования	<u>специалитет</u>
Форма обучения	<u>очная / заочная</u>

Москва
2017

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы 26.05.07 Эксплуатация судового электрооборудования и средств автоматики.

В результате освоения дисциплины обучающийся должен овладеть следующими результатами:

Код компетенции	Результаты освоения ОПОП (содержание компетенций)	Планируемые результаты освоения дисциплины
ПК-1	способность генерировать новые идеи, выявлять проблемы, связанные с реализацией профессиональных функций, формулировать задачи и намечать пути исследования	Знать: основные области применения микроконтроллеров
		Уметь: различать типы микроконтроллеров, применять их в профдеятельности
		Владеть: методами применения микроконтроллерной техники
ПК-7	способность и готовность осуществлять безопасное техническое использование, техническое обслуживание судового электрооборудования и средств автоматики в соответствии с требованиями международных и национальных нормативно-технических документов	Знать: основные микропроцессорные и микроконтроллерные семейства, присутствующие на рынке (Ардуино)
		Уметь: читать схемы для подключения и эксплуатации МПС (на базе МК Ардуино)
		Владеть: методами сборки и агрегации типовых конструкторов МПС (Ардуино)
ПК-8	способность и готовность выполнять диагностирование, техническое обслуживание и ремонт судового электрооборудования и средств автоматики	Знать: особенности программной среды управления МПС (IDE Arduino)
		Уметь: найти и определить неисправность
		Владеть: методами устранения неисправностей МПС
ПК-9	способность и готовность осуществлять выбор электрооборудования и элементов систем автоматики для замены в процессе эксплуатации судового оборудования	Знать: виды и типы компонентов для МПС и МК
		Уметь: выбирать компоненты для различных схем и устройств
		Владеть: поиском компонентов на площадках сети Интернет
ПК-12	способность и готовность устанавливать причины отказов судового электрооборудования и средств автоматики, определять и осуществлять мероприятия по их предотвращению	Знать: методы диагностики неисправностей МПС (МК)
		Уметь: диагностировать неисправности МПС (МК)
		Владеть: прогнозированием возможных отказов МПС (МК)
ПК-15	способность применять базовые знания фундаментальных и профессиональных дисциплин, проводить технико-экономический анализ, обосновывать принимаемые решения по	Знать: основные законы электротехники
		Уметь: рассчитывать основные параметры электрических схем
		Владеть: электрическими измерительными инструментами

Код компетенции	Результаты освоения ОПОП (содержание компетенций)	Планируемые результаты освоения дисциплины
	использованию судового электрооборудования и средств автоматики, решать на их основе практические задачи профессиональной деятельности	
ПК-17	способность и готовность находить компромисс между различными требованиями (стоимости, качества, безопасности и сроками исполнения) при долгосрочном и краткосрочном планировании эксплуатации судового электрооборудования и средств автоматики, выбрать рациональное (оптимальное) решение	Знать: маркетинг рынка Ардуино Уметь: выбирать компоненты МПС по соотношению цена/ качество Владеть: оперативным поиском компонентов и библиотек к ним на площадках сети Интернет

В результате освоения дисциплины обучающийся должен овладеть следующими компетентностями (в соответствии с *Международной конвенцией о ПОДГОТОВКЕ И ДИПЛОМИРОВАНИИ МОРЯКОВ И НЕСЕНИИ ВАХТЫ 2011 г.*)

Таблица А-III/6

Спецификация минимальных стандартов компетентности для электромехаников

Функция: Электрооборудование, электронная аппаратура и системы управления на уровне эксплуатации		
Сфера компетентности	Знание, понимание и профессиональные навыки	Методы демонстрации компетентности
К-5 Эксплуатация компьютеров и компьютерных сетей на судах	К 5.2 Понимание: .1 основных характеристик обработки данных .2 создания и использования компьютерных сетей на судах	Экзамен и одобренная подготовка с использованием лабораторного оборудования

2. Место дисциплины в структуре образовательной программы

Дисциплина «Микропроцессорные системы управления» относится к профессиональному циклу. Для освоения дисциплины обучающимся необходимо обладать знаниями по следующим дисциплинам:

1. Информатика
2. Информационные технологии
3. Общая электротехника и электроника
4. Теоретические основы электротехники
5. Судовая электроника и силовая преобразовательная техника
6. Судовые информационно-измерительные системы
7. Судовые электроприводы
8. Судовые электрические машины
9. Элементы и функциональные устройства судовой автоматики
10. Основы технической эксплуатации судового электрооборудования и средств автоматизации
11. Электрические машины
12. Электрические аппараты

3. Объем дисциплины в зачетных единицах и виды учебных занятий

Общая трудоемкость дисциплины составляет 3 з.е., 108 час.

Вид учебной работы	Форма обучения			
	Очная		Заочная	
	Всего часов	из них в семестре № 9	Всего часов	на курсе № 6
Общая трудоемкость дисциплины	108	108	108	108
Контактная работа обучающихся с преподавателем, всего	52	52	20	20
В том числе:				
Лекции	26	26	10	10
Практические занятия	26	26	10	10
Самостоятельная работа, всего	47	47	79	79
В том числе:				
Другие виды самостоятельной работы	47	47	79	79
Промежуточная аттестация:	9	9	9	9
Экзамен (Э)	Э	Э	Э	Э

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 – 4.2. Содержание лекционных разделов (тем) дисциплины, совмещенных с практическими работами

№ п / п	Наименование раздела (темы) дисциплины	Реализуемые компетенции и компетентности	Содержание раздела (темы) дисциплины	Трудоемкость в часах по формам обучения	
				очная	заочная
1.	Микропроцессоры и микроконтроллеры.	ПК-9 ПК-15	Основные понятия курса, отличия и общее между микропроцессором (МП) и микроконтроллером (МК). МК Ардуино, типы и виды.	2	2
2.	Платформа Arduino. Подключение Arduino Среда программирования Arduino IDE		Среда Ардуино, подключение и запуск МК, тестовые примеры. Широтно-импульсная модуляция. Структура программы, язык C и Wiring, синтаксис языка, тестовые примеры. Последовательный порт. Цифровой и аналоговый сигнал.	2	2
3.	Компоненты для Arduino		Фоторезистор Зуммер (модуль звука) Датчик температуры Светодиоды и светодиодные матрицы Светодиодная LED шкала Светодиодная матрица и сдвиговый регистр	2	1
			Ультразвуковой датчик расстояния	2	1

№ п / п	Наименование раздела (темы) дисциплины	Реализу емые компете нции и компете нтности	Содержание раздела (темы) дисциплины	Трудоемкость в часах по формам обучения	
				очная	заочная
			Инфракрасный датчик расстояния		
			Транзистор Электромотор Сервопривод Шаговый двигатель	2	1
			Ходовая часть робота Управление моторами	2	
			Символьные и графические дисплеи Принципы вывода графики на дисплей	2	
			Радиочастотная идентификация Карта памяти SD	2	
			Аудио плеер	2	
4.	Сетевой обмен	К-5 ПК-9	Подключение Ардуино к сети интернет	2	1
5.	Bluetooth. Радиоуправление		Подключение по радиоканалу Удаленное управление	2	
6.	Создание проектов	К-5 ПК-1 ПК-9	- светофор - поливочный автомат через сеть - управление серво через сеть - робот, движущийся по линии	4	2
	ВСЕГО:			26	10

5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

5.1. Самостоятельная работа

№ п/п	Вид самостоятельной работы	Наименование работы и содержание в часах
1.	Самостоятельное изучение разделов тем из http://arduino.ru/	Язык Wiring. Синтаксис языка, операторы (арифметические, логические, сравнения, унарные). (9 час очное / 13 час заочное)
2.	Самостоятельное изучение разделов тем из http://arduino.ru/	Язык Wiring. Типы данных, преобразование типов. (9 час очное / 13 час заочное)
3.	Самостоятельное изучение разделов тем из http://arduino.ru/	Язык Wiring. Функции. (9 час очное / 13 час заочное)
4.	Самостоятельное изучение разделов тем из http://arduino.ru/	Изучение других микроконтроллерных систем. (9 час очное / 13 час заочное)
5.	Самостоятельное изучение разделов тем из http://arduino.ru/	Язык Processing. Синтаксис и функции. (9 час очное / 13 час заочное)
6.	Самостоятельное изучение разделов тем из http://arduino.ru/	Самостоятельный проект для «умного» дома - включение света по датчику присутствия и освещенности и т.п. (11 час очное / 14 час заочное)

5.2. Учебно-методическое обеспечение самостоятельной работы

№ п/п	Наименование работы, ее вид	Выходные данные	Автор (ы)
1.	Язык Wiring. Синтаксис языка, операторы (арифметические, логические, сравнения, унарные)	Проекты с использованием контроллера Arduino, http://arduino.ru/	В.А. Петин
2.	Язык Wiring. Типы данных, преобразование типов.	Проекты с использованием контроллера Arduino, http://arduino.ru/	В.А. Петин
3.	Язык Wiring. Функции.	Проекты с использованием контроллера Arduino, http://arduino.ru/	В.А. Петин
4.	Изучение других микроконтроллерных систем.	Микроконтроллеры для систем автоматики	А.М. Водовозов
5.	Язык Processing. Синтаксис и функции	https://processing.org/	
6.	Самостоятельный проект для «умного» дома - включение света по датчику присутствия и освещенности и т.п.	Знакомство с Ардуино, Изд. О'Брайен, 2010 http://we.easyelectronics.ru/ http://robocraft.ru/ http://arduino-project.net/ http://robotday.ru/	М. Банци

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в обязательном приложении к рабочей программе.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Название	Автор	Вид издания (учебник, учебное пособие)	Место издания, издательство, год издания, кол-во страниц
Основная литература			
Микропроцессорные системы управления	В.В. Якунчиков	Конспект лекций и практических работ	–М.: МГАВТ, 2018 - 181 с.
Дополнительная литература			
Arduino и Raspberry Pi в проектах Internet of Things (http://znanium.com)	Петин В.А.	Учебное пособие	- СПб:БХВ-Петербург, 2016. - 320 с. ISBN 978-5-9775-3646-2
Методические рекомендации по самостоятельной работе обучающихся	Якунчиков В.В.	Методические рекомендации	ЭБС МГАВТ, 2017

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», рекомендуемых для освоения дисциплины

№ п/п	Наименование информационного ресурса	Ссылка на информационный ресурс
1.	Портал разработчиков Arduino	https://www.arduino.cc/
2.	Портал разработчиков Processing	https://processing.org/
3.	Российский информационный портал по Ардуино	http://arduino.ru/
4.	Информационный портал по Ардуино	http://wiki.amperka.ru/
5.	Информационный портал по электронным проектам	http://meandr.org/
6.	Крупнейший поставщик электрических и электронных компонентов	http://www.electronshtik.ru
7.	Агрегатор новостных IT-ресурсов крупнейших электронных порталов «Хабрахабр» и «Гиктаймс»	https://sohabr.net
8.	Информационный портал по Ардуино и интернет-магазин	http://ardunn.ru/
9.	Информационный портал по Ардуино и интернет-магазин	http://arduino-kit.ru
10.	Информационный портал по Ардуино и интернет-магазин	http://iarduino.ru/
11.	Оптимальный поставщик электроники по соотношению цена/качество	http://www.robototehnika.ru
12.	Крупный интернет-магазин электроники	http://carduino.ru
13.	Крупный интернет-магазин электроники и робототехники	http://robot-kit.ru
14.	Крупный интернет-магазин электроники	http://chipster.ru
15.	Крупный интернет-магазин электроники	http://makerplus.ru
16.	Крупный интернет-магазин и справочник электроники	http://www.arduino-ic.ru
17.	Московский поставщик электроники, предпочтительный по ценам	http://electromicro.ru
18.	Московский поставщик электроники, предпочтительный по ценам	http://amperkot.ru
19.	Московский интернет-магазин электроники	http://www.arduino-boss.ru
20.	Московский интернет-магазин электроники	http://onpad.ru
21.	Сибирский интернет-магазин электроники	http://devictor.ru
22.	Московский интернет-магазин электроники	http://mcustore.ru
23.	Московский интернет-магазин электроники	http://geegrow.ru
24.	Портал по проектам Ардуино и интернет-магазин	http://we.easyelectronics.ru/
25.	Портал по проектам Ардуино	http://robocraft.ru/
26.	Портал по проектам Ардуино	http://arduino-project.net/
27.	Портал по робототехнике	http://robotday.ru/

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационно-справочных систем

№ п/п	Наименование информационной технологии /программного продукта	Назначение (базы и банки данных, тестирующие программы, практикум, деловые игры и т.д.)	Тип продукта (полная лицензионная версия, учебная версия, демоверсия и т.п.)
1.	Среда разработки Arduino IDE (язык Wiring)	Среда разработки программ для МК Ардуино	ПО распространяется по лицензии GPL
2.	Среда разработки Processing (язык Lua)	Надстройка для визуализации данных из среды Arduino IDE	ПО распространяется по лицензии GPL
3.	Операционная система Microsoft Windows 7	Операционная система	Полная лицензионная версия
4.	MS Office 2010 (Word, Excel, PowerPoint)	Офисный пакет приложений	Полная лицензионная версия

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование специализированных аудиторий	Перечень основного оборудования
1.	Компьютерная аудитория с подключением к сети Интернет (А.113)	<p>8 рабочих мест (ноутбуки), 1 ПК преподавателя, проектор, экран</p> <p>12 (7) комплектов микроконтроллеров линейки Ардуино с соединительными кабелями</p> <p>Комплект различных датчиков для Ардуино (65 шт):</p> <ul style="list-style-type: none"> - амперметр (2) - вольтметр (4) - атмосферный (температура, давление, влажность) (3) - вибрации (2) - магнитного поля (1) - касания (1) - контакта (4) - присутствия (3) - газоанализа (метан, пропан, СО, дым, СО₂, спирт) (4) - дальности ультразвуковой (8) - дальности инфракрасный (1) - звука (2) - ориентации по 3 осям и ускорения (2) - освещенности (6) - препятствий (4) - скорости вращения (3) - температуры (4) - цвета (2) - инфракрасного сигнала (2) - линии (4) - влажности (3) <p>Комплект соединительных проводов (200шт) и скоб (100шт)</p>

№ п/п	Наименование специализированных аудиторий	Перечень основного оборудования
		<p>Шилды для Ардуино по тематикам занятий (6 шт)</p> <p>Комплект исполнительных устройств</p> <ul style="list-style-type: none"> - мотор (4) - редуктор на 2 мотора (1) - шаговый двигатель (2) - сервопривод (8) - помпа (2) - контроллер двигателей (4) - дисплей цифровой (1) и графический (5) - 7-сегментный индикатор одинарный (8) - 7-сегментный индикатор на матрице x 4 (5) - динамик (4) - звуковая плата-шилд (2) - анализатор речи (1) - микроплеер (2) - радиомодуль (1) - зуммер (2) - усилитель звука (2) - фото/видеокамера (2) - видео передатчик (1) и видеоприемник (1) с переходником - манипулятор – захват (2) - шасси робота в комплекте (3) - таймер (4) - силовой ключ (2) - RFID считыватель (1) - сенсорная панель (1) - реле (4) - радиомодуль (4) - XBee модуль (6) - пульт GamePad XBee (1) - преобразователи напряжения (3) - джойстик (2) - АЦП (1) - GSM модуль (2) - SIM 808 модуль (1) - Ethernet модуль (2) - Bluetooth модуль (3) - пульт инфракрасный (2) - SD модуль (2) <p>Аккумуляторы (3) и батареи (2)</p> <p>Зарядное устройство для аккумуляторов (1)</p> <p>Блок питания 220В / 12В (2)</p> <p>Компоненты:</p> <ul style="list-style-type: none"> - потенциометры (6) - кнопки (10) - выключатели (1) - DIP переключатели (3) - тумблеры (4) - тестер сервопривода (1)

№ п/п	Наименование специализированных аудиторий	Перечень основного оборудования
		<ul style="list-style-type: none"> - плата прототипирования (7) - светодиоды 3х цветов (50) - матрицы светодиодов, включая RGB (8) - шкалы светодиодные (4) - разъемы разные (10) - переходники и кабели (4) - диоды (10) - комплект сопротивлений 15 номиналов (по 30) - транзисторы разные (15) - конденсаторы разные (30) - стабилитроны разные (100) - крепеж разный, кронштейны

11. Методические указания для обучающихся по освоению дисциплины

Рекомендации по освоению лекционного материала, подготовке к лекциям

Лекции являются основным видом учебных занятий курса, в ходе которых производится изложение современных научных взглядов и освещение основных проблем изучаемой области знаний.

Значительную часть теоретических знаний студент должен получать самостоятельно из рекомендованных основных и дополнительных информационных источников (учебников, Интернет-ресурсов, электронной образовательной среды университета).

Материал лекций большей частью излагается путем электронных презентаций и видеодемонстраций, и предоставляется обучающемуся в электронном виде для последующего самостоятельного изучения.

Во время лекций предусмотрены интерактивные формы освоения материала - воспроизведение практических методов, излагаемых во время лекции, на своих ПК с подключенным МК Ардуино, для закрепления материала и непосредственного прояснения вопросов по тематике с преподавателем.

Материалы лекций также рекомендуется использовать при подготовке к практическим занятиям, экзамену, контрольным тестам, при выполнении самостоятельных заданий.

Рекомендации по подготовке к практическим занятиям

Для подготовки к практическим занятиям необходимо заранее ознакомиться по тексту настоящей рабочей программы с перечнем задач, которые будут рассмотрены на занятии, рекомендуемой основной и дополнительной литературой, рекомендованными Интернет-ресурсами. Необходимо выделить основные проблемные вопросы, понятия и процессы, их закономерности и взаимные связи.

Приветствуется проведение доступных экспериментов у себя дома. На практических занятиях нужно выяснять у преподавателя ответы на интересующие или затруднительные вопросы, высказывать и аргументировать свое мнение, добиваясь ясности в понимании материала.

Рекомендации по организации самостоятельной работы

Самостоятельная работа включает изучение учебной литературы, изучение информации в сети Интернет, подготовку к практическим занятиям, экзамену, выполнение практических заданий, изучение теоретического материала, вынесенного на самостоятельное изучение, изучение прикладного программного обеспечения IDE Arduino и Processing.

Рабочая программа дисциплины составлена с учетом требований Федерального Государственного образовательного стандарта высшего профессионального образования (ФГОС ВПО) третьего поколения по специальности 26.05.07 Эксплуатация судового электрооборудования и средств автоматики.

Составитель:

/ В.В. Якунчиков /

Рабочая программа рассмотрена на заседании кафедры
Электрооборудования и утверждена протоколом №11 от «31» августа 2017 г.

Зав. кафедрой:

/Л.Ф. Мокеров/

Декан СМФ



/Якунчиков В.В./



**Федеральное агентство морского и речного транспорта
ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»
Московская государственная академия водного транспорта - филиал
Федерального государственного бюджетного образовательного учреждения высшего
образования
«Государственный университет морского и речного флота имени адмирала
С.О. Макарова»
(МГАВТ - филиал ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)**

**Факультет Судомеханический
Кафедра Электрооборудования**

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

дисциплины «Микропроцессорные системы управления»

Специальность	<u>26.05.07 Эксплуатация судового электрооборудования и средств автоматики</u>
Уровень высшего образования	<u>специалитет</u>
Форма обучения	<u>очная / заочная</u>

Москва
2017

1. Перечень компетенций и этапы их формирования в процессе освоения дисциплины

Рабочей программой дисциплины предусмотрено формирование следующих компетенций:

Код компетенции	Результаты освоения ОПОП (содержание компетенций)	Планируемые результаты освоения дисциплины
ПК-1	способность генерировать новые идеи, выявлять проблемы, связанные с реализацией профессиональных функций, формулировать задачи и намечать пути исследования	Знать: основные области применения микроконтроллеров
		Уметь: различать типы микроконтроллеров, применять их в профдеятельности
		Владеть: методами применения микроконтроллерной техники
ПК-7	способность и готовность осуществлять безопасное техническое использование, техническое обслуживание судового электрооборудования и средств автоматики в соответствии с требованиями международных и национальных нормативно-технических документов	Знать: основные микропроцессорные и микроконтроллерные семейства, присутствующие на рынке (Ардуино)
		Уметь: читать схемы для подключения и эксплуатации МПС (на базе МК Ардуино)
		Владеть: методами сборки и агрегации типовых конструкторов МПС (Ардуино)
ПК-8	способность и готовность выполнять диагностирование, техническое обслуживание и ремонт судового электрооборудования и средств автоматики	Знать: особенности программной среды управления МПС (IDE Arduino)
		Уметь: найти и определить неисправность
		Владеть: методами устранения неисправностей МПС
ПК-9	способность и готовность осуществлять выбор электрооборудования и элементов систем автоматики для замены в процессе эксплуатации судового оборудования	Знать: виды и типы компонентов для МПС и МК
		Уметь: выбирать компоненты для различных схем и устройств
		Владеть: поиском компонентов на площадках сети Интернет
ПК-12	способность и готовность устанавливать причины отказов судового электрооборудования и средств автоматики, определять и осуществлять мероприятия по их предотвращению	Знать: методы диагностики неисправностей МПС (МК)
		Уметь: диагностировать неисправности МПС (МК)
		Владеть: прогнозированием возможных отказов МПС (МК)
ПК-15	способность применять базовые знания фундаментальных и профессиональных дисциплин, проводить технико-экономический анализ, обосновывать принимаемые решения по использованию судового электрооборудования и средств автоматики, решать	Знать: основные законы электротехники
		Уметь: рассчитывать основные параметры электрических схем
		Владеть: электрическими измерительными инструментами

Код компетенции	Результаты освоения ОПОП (содержание компетенций)	Планируемые результаты освоения дисциплины
	на их основе практические задачи профессиональной деятельности	
ПК-17	способность и готовность находить компромисс между различными требованиями (стоимости, качества, безопасности и сроками исполнения) при долгосрочном и краткосрочном планировании эксплуатации судового электрооборудования и средств автоматики, выбрать рациональное (оптимальное) решение	Знать: маркетинг рынка Ардуино
		Уметь: выбирать компоненты МПС по соотношению цена/ качество
		Владеть: оперативным поиском компонентов и библиотек к ним на площадках сети Интернет

Рабочей программой дисциплины предусмотрено формирование следующих компетенций ПДНВ:

Код компетенции	Результаты освоения ОПОП (содержание компетенций)	Планируемые результаты освоения дисциплины
К-5	К 5.2 Судовые компьютерные сети Понимание: .1 основных характеристик обработки данных .2 создания и использования компьютерных сетей на судах	Знать: основные принципы работы современной микропроцессорной (микроконтроллерной) техники, основные способы обработки данных на МК, способы создания и использования компьютерных сетей с датчиками на судах, способы использования МК на мостике, в машинном отделении и для решения иных инженерных задач.
		Уметь: подключать и эксплуатировать локальную сеть на ПК и МК Ардуино в судовых условиях, разбираться и модернизировать стандартные скетчи ПО Ардуино под свои нужды
		Владеть: методами проектирования, сборки и агрегации типовых конструкторов МК Ардуино. Методами организации компьютерных сетей на судах с применением МК, их проверки и использования.
		Критерии для оценки компетентности Компьютеры (ПК и МК) правильно проверяются и используются.
		Методы демонстрации компетентности Одобрённая подготовка с использованием лабораторного оборудования

2. Паспорт фонда оценочных средств для проведения текущей и промежуточной аттестации обучающихся

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Микропроцессоры и микроконтроллеры. Подключение Arduino	ПК-1, ПК-7, К-5	Индивидуальное задание на подключение МК в среде Ардуино
2.	Среда программирования Arduino IDE	ПК-1, ПК-8, К-5	Индивидуальное задание на демонстрацию тестового примера на МК
3.	Компоненты для Arduino	ПК-1, ПК-9, ПК-8, ПК-12, ПК-15, ПК-17, К-5	Индивидуальное задание на сборку схемы и разработку проекта
	Компоненты для Arduino		Индивидуальное задание на сборку схемы и разработку проекта
	Компоненты для Arduino		Индивидуальное задание на сборку схемы и разработку проекта
	Компоненты для Arduino		Индивидуальное задание на сборку схемы и разработку проекта
	Компоненты для Arduino		Индивидуальное задание на сборку схемы и разработку проекта
	Компоненты для Arduino		Индивидуальное задание на сборку схемы и разработку проекта
	Компоненты для Arduino		Индивидуальное задание на сборку схемы и разработку проекта
4.	Сетевой обмен		Индивидуальное задание на сборку схемы и разработку проекта
5.	Bluetooth Радиоуправление		Индивидуальное задание на сборку схемы и разработку проекта
6.	Создание проектов		Индивидуальное задание на сборку схемы и разработку проекта

3. Критерии оценивания результата обучения по дисциплине и шкала оценивания

Результат обучения по дисциплине	Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине				Процедура оценивания
	2	3	4	5	
К-5 Знания основные принципы работы современной микропроцессорной (микроконтроллерной) техники, основные способы обработки данных на МК, способы создания и использования компьютерных сетей с датчиками на судах, способы использования МК на мостике, в машинном отделении и для решения иных инженерных задач.	Отсутствие знаний или фрагментарные представления по знанию принципов компьютерных сетей и правильного использования компьютеров и МК на судах, мостике и в машинном отделении и для решения иных инженерных задач	Неполные представления о: компьютерных сетях и правильном использовании компьютеров с МК на судах, мостике и в машинном отделении и для решения иных инженерных задач	Сформированные, но содержащие отдельные пробелы представления о компьютерных сетях и правильном использовании компьютеров с МК на судах, мостике и в машинном отделении и для решения иных инженерных задач	Сформированные систематические представления о Компьютерных сетях и правильном использовании компьютеров с МК на судах, мостике и в машинном отделении и для решения иных инженерных задач	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
Умения/ Владения подключать и эксплуатировать локальную сеть на ПК и МК Ардуино в судовых условиях, разбираться и модернизировать стандартные скетчи ПО Ардуино под свои нужды; владение методами проектирования, сборки и агрегации типовых конструкторов МК Ардуино. Методами организации компьютерных сетей на судах с применением МК, их проверки и использования.	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Подготовка с использованием лабораторного оборудования: Кейс-задания: индивидуальные задания и проекты на подключение МК платформы Ардуино к ПК
ПК-1 Знания основные области	Отсутствие знаний или фрагментарн	Неполные представления о: безопасное	Сформированн	Сформированн	Экзамен в виде кейс-задания: Индивидуальное

Результат обучения по дисциплине	Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине				Процедура оценивания
	2	3	4	5	
применения микроконтроллеров	ые представления по системе управления микроконтроллером Ардуино	техническое использование, техническое обслуживание судового электрооборудования и средств автоматизации	отдельные пробелы представления о системе управления микроконтроллером Ардуино	е представления об системе управления микроконтроллером Ардуино	задание на демонстрацию тестового примера на МК
Умения/ Владения различать типы микроконтроллеров, применять их в профессиональной деятельности владение методами применения микроконтроллерной техники	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
ПК-7 Знания основные микропроцессорные и микроконтроллерные семейства, присутствующие на рынке (Ардуино)	Отсутствие знаний или фрагментарные представления о размещении библиотек и компонентов в сети Интернет (Ардуино)	Неполные представления о размещении библиотек и компонентов в сети Интернет	Сформированные, но содержащие отдельные пробелы представления размещении библиотек и компонентов в сети Интернет	Сформированные систематические представления о размещении библиотек и компонентов в сети Интернет	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
Умения/ Владения читать схемы для подключения и эксплуатации МПС (на базе МК Ардуино) владение методами сборки и агрегации типовых конструкторов МПС (Ардуино)	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
ПК-8 Знания особенности программной среды управления МПС (IDE Ардуино)	Отсутствие умений или фрагментарные умения в среде программирования Arduino IDE	В целом удовлетворительные, но не систематизированные умения составлять скетчи в среде программирования Arduino IDE	В целом удовлетворительные, но содержащие отдельные пробелы умения составлять скетчи в среде программирования Arduino IDE	Сформированные умения составлять скетчи в среде программирования Arduino IDE	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
Умения/ Владения	Собранная схема не	Собранная схема работает не так,	Собранная схема	Собранная схема работает	Экзамен в виде кейс-задания:

Результат обучения по дисциплине	Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине				Процедура оценивания
	2	3	4	5	
найти и определить неисправность; владение методами устранения неисправностей МПС	работает	как предписано заданием (с отступлением от задания)	работает согласно заданию	согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Индивидуальное задание на демонстрацию тестового примера на МК
ПК-9 Знания виды и типы компонентов для МПС и МК	Отсутствие знаний или фрагментарные представления о составе МПС и МК	Неполные представления о составе МПС и МК	Сформированные, но содержащие отдельные пробелы представления о составе МПС и МК	Сформированные систематически представления о составе МПС и МК	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
Умения/ Владения выбирать компоненты для различных схем и устройств владение поиском компонентов на площадках сети Интернет	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
ПК-12 Знания методы диагностики неисправностей МПС (МК)	Отсутствие владения или фрагментарные владения определения неисправностей в проекте Ардуино	В целом удовлетворительные, но не систематизированные владения/применения навыков Использования электрического и электронного измерительного оборудования для обнаружения неисправностей	В целом удовлетворительные, но содержащие отдельные пробелы владения/применения навыков определения неисправностей в проекте Ардуино	Сформированные умения Определения и устранения неисправностей в проекте Ардуино	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
Умения/ Владения диагностировать неисправности МПС (МК) владение прогнозированием возможных отказов МПС (МК)	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
ПК-15 Знания основные законы электротехники	Отсутствие владения или фрагментарные владения обоснованием	В целом удовлетворительные, но не систематизированные владения/применения навыков	В целом удовлетворительные, но содержащие отдельные пробелы владения/применения навыков	Сформированные умения обоснованием принятых решений для формирования проекта	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК

Результат обучения по дисциплине	Критерии оценивания результата обучения по дисциплине и шкала оценивания по дисциплине				Процедура оценивания
	2	3	4	5	
	принятых решений для формирования проекта	проводить технико-экономический анализ, обосновывать принимаемые решения по использованию судового электрооборудования и средств автоматики	применения навыков обоснованием принятых решений для формирования проекта		
Умения/ Владения рассчитывать основные параметры электрических схем; владение электрическими измерительными инструментами	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
ПК-17 Знания маркетинг рынка Ардуино	Отсутствие владения или фрагментарные владения выбрать рациональное (оптимальное) решение в сети Интернет	В целом удовлетворительные, но не систематизированные владения/применения навыков выбрать рациональное (оптимальное) решение в сети Интернет	В целом удовлетворительные, но содержащие отдельные пробелы владения/применения навыков выбрать рациональное (оптимальное) решение в сети Интернет	Сформированные умения выбрать рациональное (оптимальное) решение в сети Интернет	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК
Умения/ Владения выбирать компоненты МПС по соотношению цена/ качество; владение оперативным поиском компонентов и библиотек к ним на площадках сети Интернет	Собранная схема не работает	Собранная схема работает не так, как предписано заданием (с отступлением от задания)	Собранная схема работает согласно заданию	Собранная схема работает согласно заданию, учитывается малое время сборки (в зависимости от сложности)	Экзамен в виде кейс-задания: Индивидуальное задание на демонстрацию тестового примера на МК

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

Вид текущего контроля: кейс-задание (30 вариантов)

Название кейса: **Проект слаботочной схемы автоматизации на платформе МК Ардуино** (см. приложение 1)

Время выполнения кейс-задания: 90 минут

Критерии оценивания:

Критерии / оценка	5	4	3	2
Работоспособность схемы	+	+	+	
Эффективность (минимизация количества компонентов) с аргументацией решения	+	+		
Аргументация предлагаемого решения, Предложение иного варианта решения задачи с учетом дополнительной литературы и сетевых источников	+			

Тестовые задания (перечень схем для сборки):

1. Среда Ардуино, подключение и запуск МК, тестовые примеры из стандартной библиотеки без подключения модулей.
2. Структура программы, язык C и Wiring, синтаксис языка, тестовые примеры из стандартной библиотеки без подключения модулей.
3. Последовательный порт, тестовые примеры из стандартной библиотеки без подключения модулей.
4. Сборка схемы: Фоторезистор
5. Сборка схемы: Зуммер (модуль звука)
6. Сборка схемы: Датчик температуры
7. Сборка схемы: Светодиоды и светодиодные матрицы
8. Сборка схемы: Светодиодная LED шкала
9. Сборка схемы: Светодиодная матрица и сдвиговый регистр
10. Сборка схемы: Ультразвуковой датчик расстояния
11. Сборка схемы: Инфракрасный датчик расстояния
12. Сборка схемы: Транзистор
13. Сборка схемы: Электромотор
14. Сборка схемы: Сервопривод
15. Сборка схемы: Шаговый двигатель
16. Сборка схемы: Символьные и графические дисплеи
17. Сборка схемы: Радиочастотная идентификация
18. Сборка схемы: Карта памяти SD
19. Сборка схемы: Аудио плеер
20. Сборка схемы: Подключение Ардуино к сети интернет
21. Сборка схемы: Подключение исполнительного модуля по радио-каналу
22. Сборка проекта: Настройка видеоканала
23. Сборка проекта: светофор
24. Сборка проекта: поливочный автомат

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

Вид аттестации: экзамен в виде практического задания – сборки проекта.

Наименование проекта: Проект слаботочной схемы автоматизации на платформе МК Ардуино с управлением через глобальную / локальную сеть

Содержание задания: Подключение датчика или исполнительного устройства к сети интернет на платформе МК Ардуино, управление ими через ПК (или смартфон).

Время выполнения: 45 минут

Критерии оценивания:

- работоспособность проекта;
- степень осознанности, понимания изученного;
- языковое оформление ответа.

Показатели и шкала оценивания:

Шкала оценивания	Показатели
5	проект выполнен без ошибок, обучающийся представил оригинальное и грамотное решение конструкции, отчетливо понимает ход расчета и умеет обосновать выбор исходных параметров и их взаимосвязь, подробно и безошибочно отвечает на все заданные ему вопросы, проявляет при работе достаточную самостоятельность
4	проект выполнен неоптимально или с незначительными ошибками, но при опросе обучающийся проявляет понимание ошибок и способов их исправления, не допускает существенных погрешностей в ответах на вопросы
3	проект выполнен без грубых ошибок, но при опросе обучающийся проявляет недостаточное понимание всех подробностей проделанной работы; допускает при ответах на вопросы неточности и неправильные формулировки; проект не закончен в установленный срок
2	принципиальные ошибки в представленном проекте, обучающийся не может устранить ошибки, проявляет полное пренебрежение к срокам выполнения проекта

ПЕРЕЧЕНЬ ТЕСТОВЫХ КЕЙС-ЗАДАНИЙ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

ПРОЕКТ 1: МИГАЕМ СВЕТОДИОДОМ

В этом эксперименте мы научимся управлять светодиодом. Заставим его мигать.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [светодиод](#);
- [резистор 220 Ом](#);
- [провод папа-папа](#).

Светодиод – это полупроводниковый прибор, преобразующий электрический ток непосредственно в световое излучение. По-английски светодиод называется light emitting diode, или LED. Цветовые характеристики светодиодов зависят от химического состава использованного в нем полупроводника. Светодиод излучает в узкой части спектра, его цвет чист, что особенно ценят дизайнеры. Светодиод механически прочен и исключительно надежен, его срок службы может достигать 100 тысяч часов, что почти в 100 раз больше, чем у лампочки накаливания, и в 5–10 раз больше, чем у люминесцентной лампы. Наконец, светодиод – низковольтный электроприбор, а стало быть, безопасный. Светодиоды поляризованы, имеет значение, в каком направлении подключать их. Положительный вывод светодиода (более длинный) называется анодом, отрицательный – катодом. Как и все диоды, светодиоды позволяют току течь только в одном направлении – от анода к катоду. Поскольку ток протекает от положительного к отрицательному, анод светодиода должен быть подключен к цифровому сигналу 5 В, а катод должен быть подключен к земле. Мы будем подключать светодиод к цифровому контакту D10 Arduino последовательно с резистором. Светодиоды должны быть всегда соединены последовательно с резистором, который выступает в качестве ограничителя тока. Чем больше значение резистора, тем больше он ограничивает ток. В этом эксперименте мы используем резистор номиналом 220 Ом. Схема подключения приведена на рис. 1.1. Как подобрать ограничительный резистор и как будет влиять номинал резистора на яркость светодиода, мы рассмотрим в эксперименте 3.

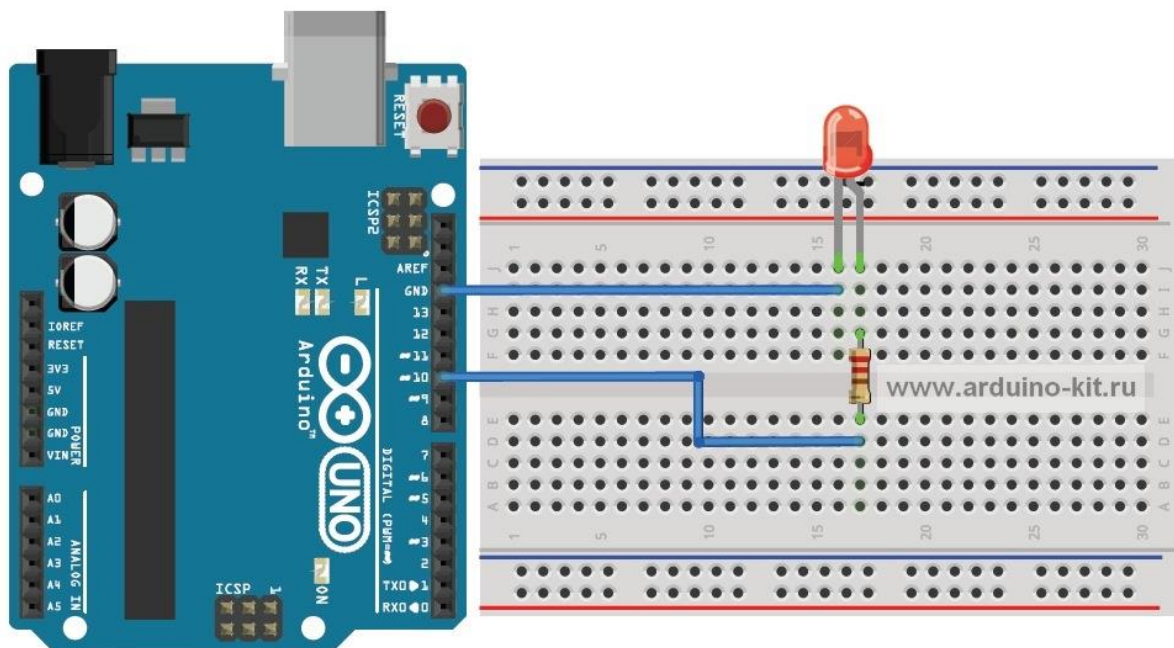


Рис. 1.1. Схема подключения светодиода

Светодиод последовательно с резистором подключаем к цифровому выводу Arduino D10. По умолчанию все выводы Arduino сконфигурированы как входы. Мы собираемся использовать вывод Arduino как выход, поэтому необходимо его переконфигурировать, выдав контроллеру соответствующую команду.

```
pinMode(10,OUTPUT);
```

Для мигания светодиода необходимо попеременно с определенным интервалом подавать на вывод Arduino сигналы HIGH (высокий уровень или 1) и LOW (низкий уровень или 0). Интервал изменения сигнала на выходе D10 Arduino будем устанавливать с помощью команды `delay()`, задерживающей выполнение скетча на заданное время в миллисекундах (мс).

Порядок подключения:

1. Длинную ножку светодиода (анод) подключаем к цифровому выводу D10 Arduino, другую (катод) – через резистор 220 Ом к выводу GND (см. рис. 1.1).
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Наблюдаем процесс мигания светодиода.

Теперь мы можем поэкспериментировать с периодом мигания светодиода, меняя в скетче значения задержки в функции `delay()`.

ПРОЕКТ 2: ОБРАБАТЫВАЕМ НАЖАТИЕ КНОПКИ НА ПРИМЕРЕ ЗАЖИГАНИЯ СВЕТОДИОДА

Это эксперимент по работе с кнопкой. Мы будем включать светодиод по нажатию кнопки и выключать по отпусканию кнопки. Рассмотрим понятие дребезга и программные методы его устранения.

Необходимые компоненты:

- контроллер Arduino UNO R3;
- плата для прототипирования;
- кнопка;
- светодиод;
- резистор 220 Ом;
- резистор 10 кОм;
- провода папа-папа.

В данном эксперименте мы будем использовать контакт D2 Arduino в качестве входа. Это позволяет подключить к нему кнопку для взаимодействия с проектом в режиме реального времени. При использовании Arduino в качестве входов используют pull-up- и pulldown-резисторы, чтобы вход Arduino не находился в «подвешенном» состоянии (в этом состоянии он будет собирать внешние наводки и принимать произвольные значения), а имел заранее известное состояние (0 или 1). Резисторы pull-up подтягивают вход к питанию +5 В, pull-down-резисторы подтягивают вход к GND. Кроме этого, pull-up- и pull-down-резисторы гарантируют, что кнопка не создаст короткого замыкания между +5 В и землей при нажатии. В нашем эксперименте для подключения кнопки мы будем использовать pulldown-резистор. Схема подключения представлена на рис. 2.1.

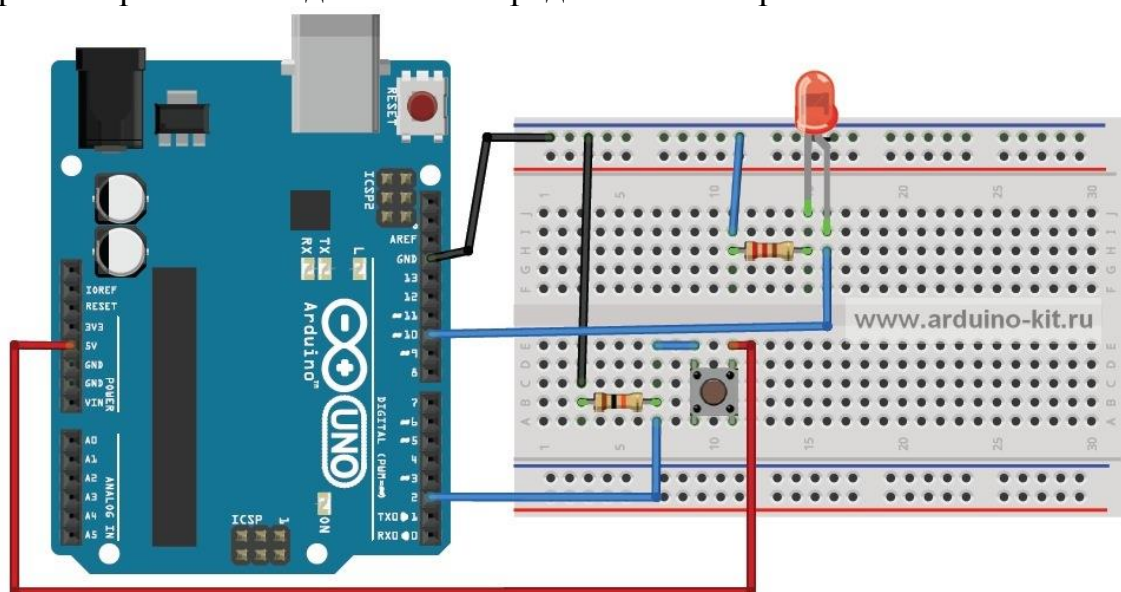


Рис. 2.1. Схема подключения кнопки и светодиода

Когда кнопка отключена, вход D2 будет подтянут к «земле» через резистор номиналом 10 кОм, который будет ограничивать поток тока, и на входном контакте будет установлено значение напряжения LOW. При нажатии на кнопку входной контакт напрямую связан с 5 В. Большая часть тока будет

протекать по пути наименьшего сопротивления через замкнутую кнопку, и на входе генерируется уровень HIGH. При нажатии на кнопку включаем светодиод, при отпускании – гасим.

Порядок подключения:

1. Длинную ножку светодиода (анод) подключаем к цифровому выводу D10 Arduino, другую (катод) – через резистор 220 Ом к выводу GND (см. рис. 2.1).
2. Один вход кнопки подключаем к +5 В, другой – через резистор 10 кОм к GND, выход кнопки подключаем к входу D2 Arduino (см. рис. 2.1).
3. Составляем самостоятельно и загружаем в плату Arduino скетч
4. При нажатии на кнопку светодиод должен гореть, при отпускании – затухнуть.

Усложним задачу – будем переключать состояние светодиода (включен/выключен) при каждом нажатии кнопки. Загрузим на плату Arduino скетч.

При нажатии кнопки светодиод должен изменять свое состояние. Но это будет происходить не всегда. Виной тому –дребезг кнопок. Кнопки представляют из себя механические устройства с системой пружинного контакта. Когда вы нажимаете на кнопку вниз, сигнал не просто меняется от низкого до высокого, он в течение нескольких миллисекунд меняет значение от одного до другого, прежде чем контакты плотно соприкоснутся и установится значение HIGH. Микроконтроллер фиксирует все эти нажатия, потому что дребезг неотличим от настоящего нажатия на кнопку. Устранить влияние дребезга можно программно. Алгоритм следующий:

1. Сохраняем предыдущее состояние кнопки и текущее состояние кнопки (при инициализации LOW).
2. Считываем текущее состояние кнопки.
3. Если текущее состояние кнопки отличается от предыдущего состояния кнопки, ждем 5 мс, потому что кнопка, возможно, изменила состояние.
4. После 5 мс считываем состояние кнопки и используем его в качестве текущего состояния кнопки.
5. Если предыдущее состояние кнопки было LOW, а текущее состояние кнопки HIGH, переключаем состояние светодиода.
6. Устанавливаем предыдущее состояние кнопки для текущего состояния кнопки.
7. Возврат к шагу 2. Добавляем к нашему скетчу подпрограмму устранения дребезга.

Загружаем скетч в плату Arduino и проверяем работу. Теперь все работает нормально, каждое нажатие кнопки приводит к изменению состояния светодиода.

ПРОЕКТ 3: ПОТЕНЦИОМЕТР. ПОКАЗЫВАЕМ ЗАКОН ОМА НА ПРИМЕРЕ ЯРКОСТИ СВЕТОДИОДА

В этом эксперименте мы познакомимся с потенциометром и будем управлять яркостью светодиода и изменением сопротивления потенциометра.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [потенциометр 2 кОм](#);
- [светодиод](#);
- [резистор 220 Ом](#);
- [провода папа-папа](#).

В [эксперименте 1](#) для подключения светодиода к цифровому выходу мы использовали ограничительный резистор номиналом 220 Ом. Сейчас мы рассмотрим, как подобрать ограничительный резистор и как будет влиять номинал резистора на яркость светодиода.

Самым главным уравнением для любого инженера-электрика является закон Ома. Закон Ома определяет отношения между напряжением, током и сопротивлением в цепи.

Закон Ома определяется следующим образом:

$$V = I \times R,$$

где V – напряжение в вольтах; I – ток в амперах; R – сопротивление в омах.

В электрической схеме каждый компонент имеет некоторое сопротивление, что снижает напряжение. Светодиоды имеют predetermined падение напряжения на них и предназначены для работы в определенном значении тока. Чем больше ток через светодиод, тем ярче светодиод светится, до предельного значения. Для наиболее распространенных светодиодов максимальный ток составляет 20 мА. Обычное значение падения напряжения для светодиода – около 2 В.

Напряжение питания 5 В должно упасть на светодиоде и резисторе, поскольку доля светодиода 2 В оставшиеся 3 В должны упасть на резисторе. Зная максимальное значение прямого тока через светодиод (20 мА), можете найти номинал резистора.

$$R = V/I = 3/0,02 = 150 \text{ Ом}.$$

Таким образом, со значением резистора 150 Ом ток 20 мА протекает через резистор и светодиод. По мере увеличения значения сопротивления ток будет уменьшаться. 220 Ом немного более, чем 150 Ом, но все же позволяет светиться светодиоду достаточно ярко, и резистор такого номинала очень распространен. Если мы будем увеличивать номинал резистора, то будем уменьшать ток, проходящий через светодиод, и, соответственно, яркость светодиода. Для изменения яркости светодиода мы будем использовать потенциометр. Потенциометры являются переменными делителями напряжения и выглядят как ручки. Они бывают разных размеров и форм, но все имеют три вывода. Номинал потенциометра определяет сопротивление

между крайними выводами, оно неизменно, поворотом ручки мы изменяем сопротивление между средним и крайним выводами от 0 до номинала потенциометра либо от номинала до нуля. В эксперименте потенциометр мы подключаем последовательно с резистором 220 Ом, чтобы не уменьшить значения ограничивающего резистора для светодиода до нуля и не сжечь светодиод. Схема подключения представлена на рис. 3.1.

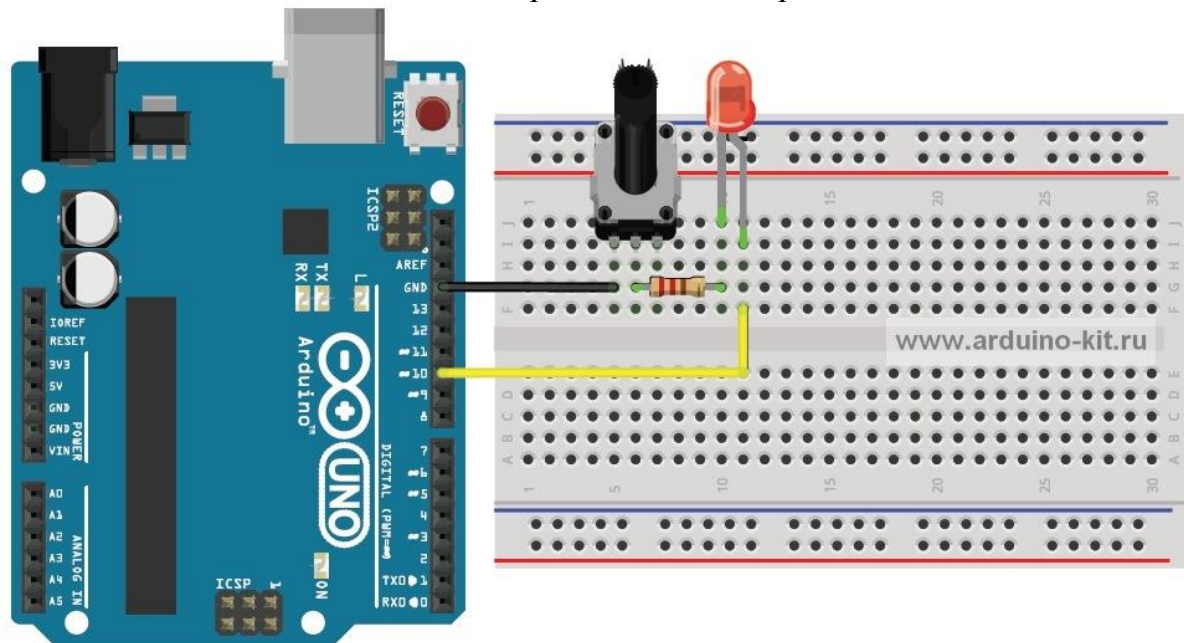


Рис. 3.1. Схема подключения потенциометра и светодиода

Он совсем простой — нам необходимо просто включить светодиод, подключенный к цифровому выводу D10 Arduino.

Порядок подключения:

1. Длинную ножку светодиода (анод) подключаем к цифровому выводу D10 Arduino, другую (катод) — к одной из ног резистора 220 Ом (см. рис. 3.1).
2. Свободную ногу резистора 220 Ом подключаем к средней ножке потенциометра, вторую (любую крайнюю) подсоединяем к GND (рис. 3.1).
3. Составляем самостоятельно и загружаем в плату Arduino скетч
4. Поворачиваем ручку потенциометра и наблюдаем изменение яркости светодиода от полного выключения до почти полной яркости.

ПРОЕКТ 4: СВЕТОДИОДНАЯ ШКАЛА 10 СЕГМЕНТОВ. ВРАЩЕНИЕМ ПОТЕНЦИОМЕТРА МЕНЯЕМ КОЛИЧЕСТВО СВЕТЯЩИХСЯ СВЕТОДИОДОВ

В этом эксперименте мы рассмотрим работу аналоговых входов Arduino, работу потенциометра в качестве аналогового датчика и будем демонстрировать показания аналогового датчика с помощью светодиодной шкалы.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [потенциометр 2 кОм](#);
- [светодиодная шкала 10 сегментов](#);
- [резистор 220 Ом](#);
- [провода папа-папа](#).

В предыдущих экспериментах мы рассматривали работу с цифровыми выводами Arduino, они имеют только два возможных состояния: включено или выключено, HIGH или LOW, 1 или 0. Но для получения информации об окружающем мире необходимо работать с аналоговыми данными, имеющими бесконечное число возможных значений в данном диапазоне. Для получения аналоговых данных Arduino имеет аналоговые входы, оснащенные 10-разрядным аналого-цифровым преобразователем для аналоговых преобразований. Точность АЦП определена разрешением. 10-разрядный означает, что АЦП может разделить аналоговый сигнал на 210 различных значений. Следовательно, Arduino может присвоить $2^{10} = 1024$ аналоговых значения, от 0 до 1023. опорное напряжение определяет максимальное напряжение, его значение соответствует значению 1023 АЦП. При напряжении 0 В на контакте АЦП возвращает значение 0, опорное напряжение возвращает значение 1023. Несмотря на то что можно изменить опорное напряжение, мы будем использовать опорное напряжение 5 В.

Рассмотрим, как использовать потенциометр в качестве аналогового датчика. Рисунок 4.1 показывает, как правильно подключить ваш

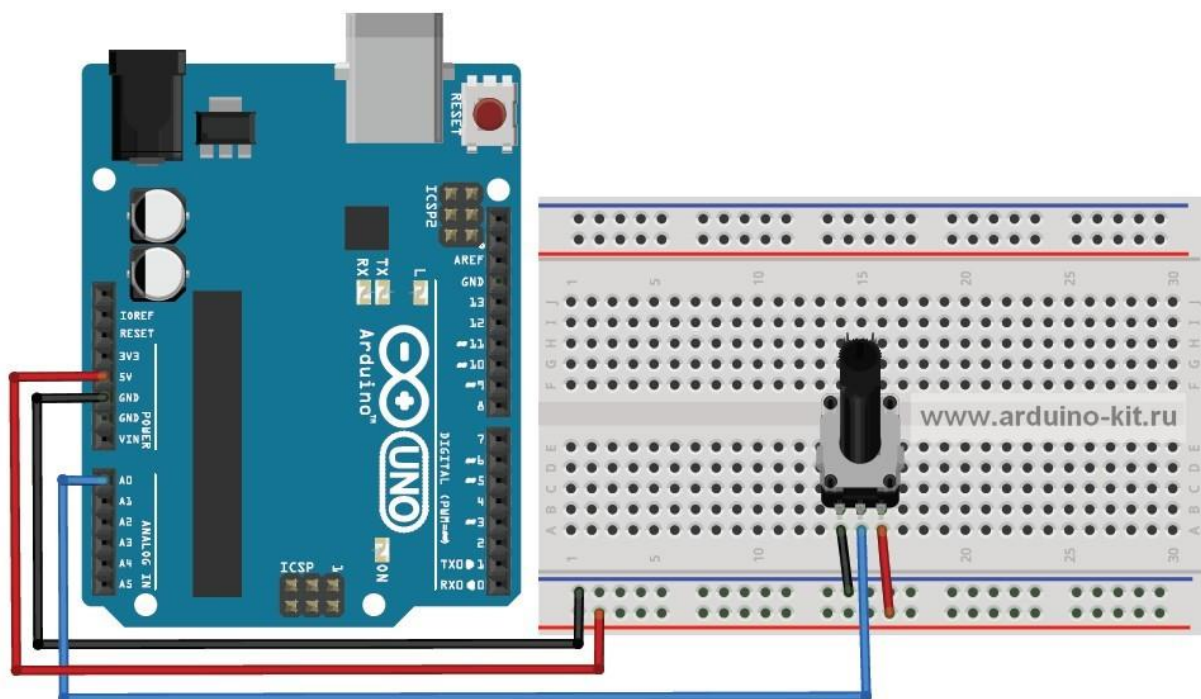


Рис. 4.1. Схема подключения потенциометра в качестве аналогового датчика

потенциометр к Arduino в качестве аналогового датчика. Мы подключаем один из крайних выводов на землю, другой крайний вывод – к +5 В. Средний вывод потенциометра подключаем к аналоговому входу A0 платы Arduino. Для считывания данных с аналогового порта в Arduino есть функция `analogRead()`.

Загружаем на плату Arduino скетч для считывания значений из аналогового порта и вывода их в монитор последовательного порта Arduino.

Порядок подключения:

1. Подключаем потенциометр по схеме на рис. 4.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Запускаем в Arduino IDE монитор последовательного порта.
4. Поворачиваем ручку потенциометра и наблюдаем вывод аналоговых значений потенциометра в монитор последовательного порта (см. рис. 4.2).

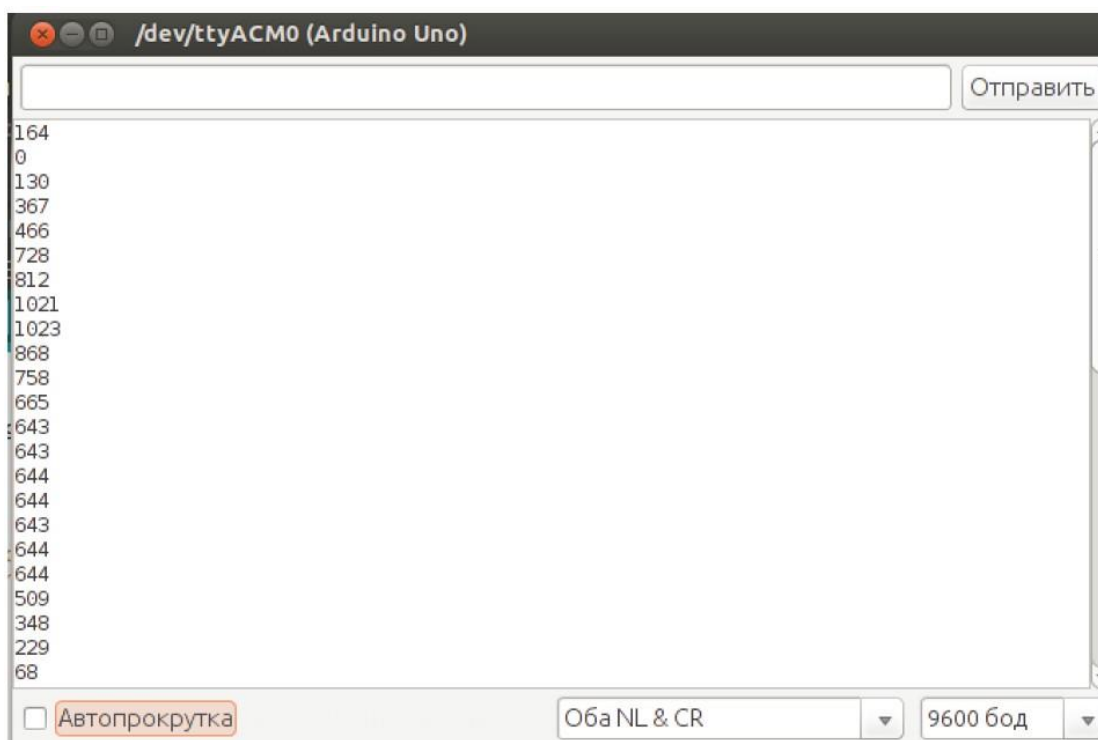


Рис. 4.2. Вывод аналоговых значений потенциометра в монитор последовательного порта

Теперь визуализируем аналоговые данные потенциометра с помощью 10-разрядной линейной светодиодной шкалы. Шкала представляет собой сборку из 10 независимых светодиодов с катодами со стороны надписи на корпусе. Для подключения шкалы к Arduino будем использовать 10 цифровых выводов D3–D12. Схема соединений показана на рис. 4.3. Каждый из светодиодов шкалы выводом анода соединен с цифровым выводом Arduino, а катодом на землю через последовательно соединенный ограничивающий резистор 220 Ом. Аналоговые данные потенциометра (0–1023) масштабируем в данные шкалы (0–10) с помощью функции `map()` и зажигаем соответствующее количество светодиодов.

Порядок подключения:

1. Подключаем потенциометр по схеме на рис. 4.1.
2. Подключаем выводы светодиодной шкалы контактами анодов через ограничительные резисторы номиналом 220 Ом к выводам Arduino D3–D12, контактами катодов – на землю (см. рис. 4.3).
3. Составляем самостоятельно и загружаем в плату Arduino скетч.
4. Поворачиваем ручку потенциометра и наблюдаем на светодиодной шкале

уровень значения потенциометра от максимального номинала.

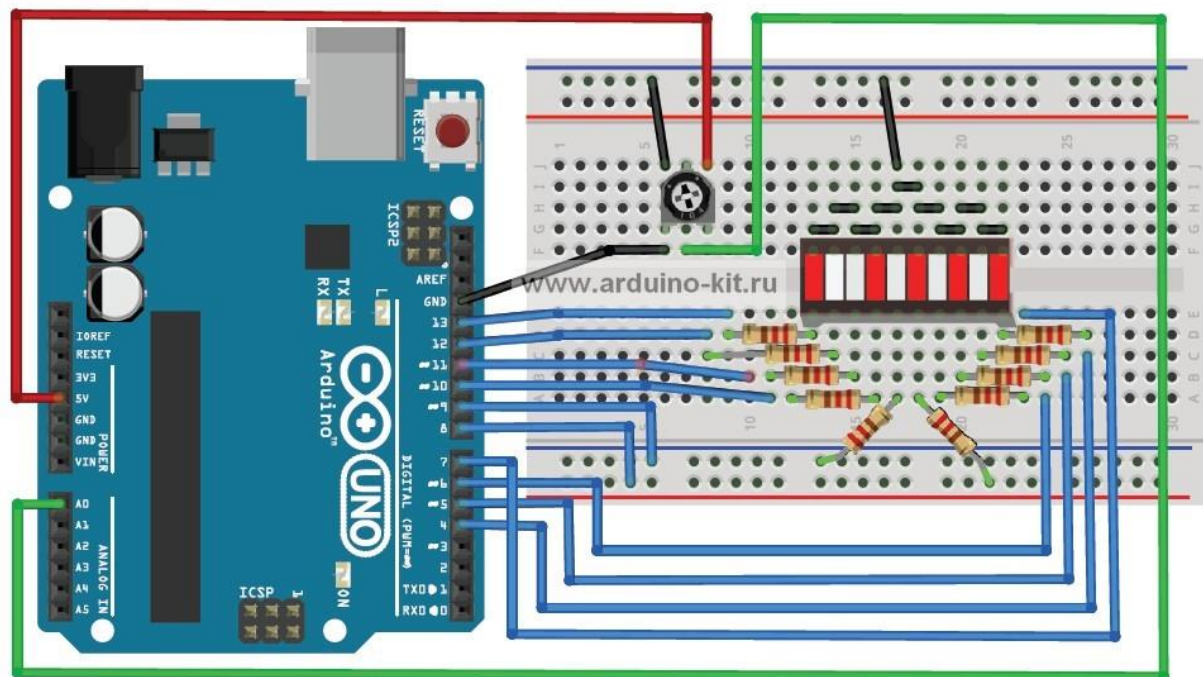


Рис. 4.3. Схема подключения линейной светодиодной шкалы

ПРОЕКТ 5: RGB-СВЕТОДИОД. ШИРОТНО-ИМПУЛЬСНАЯ МОДУЛЯЦИЯ. ПЕРЕЛИВАЕМСЯ ЦВЕТАМИ РАДУГИ

В этом эксперименте мы рассмотрим широтно-импульсную модуляцию, которая позволяет Arduino выводить аналоговые данные на цифровые выводы, и применим эти знания для создания произвольных цветов свечения с помощью RGB-светодиода.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [RGB-светодиод](#);
- [резистор 220 Ом](#);
- [провода папа-папа](#).

Arduino не может на цифровой вывод выдавать произвольное напряжение. Выдается либо +5 В (HIGH), либо 0 В (LOW). Но уровнем напряжения управляется многое: например, яркость светодиода или скорость вращения мотора. Для симуляции неполного напряжения используется ШИМ (широтно-импульсная модуляция, или PWM).

ШИМ – это операция получения изменяющегося аналогового значения посредством цифровых сигналов. Цифровой сигнал на выходе постоянно переключается между максимальным и минимальным значениями.

Переключение имеет частоту в тысячи герц. Глаз не замечает мерцания более 50 Гц, поэтому нам кажется, что светодиод не мерцает, а горит в неполную силу. Длительность включения максимального значения называется шириной импульса. Для получения различных аналоговых величин изменяется ширина импульса (см. рис. 5.1).

Arduino-функция `analogWrite()` выдает ШИМ-сигнал на цифровой вывод Arduino. После вызова `analogWrite()` на выходе будет генерироваться постоянная прямоугольная волна с заданной шириной импульса до следующего вызова `analogWrite()`, частота выдаваемого ШИМ-сигнала равна 490 Гц. На платах Arduino Nano и UNO ШИМ поддерживают выводы 3, 5, 6, 9, 10 и 11, на плате Mega – выводы 2–13. Данные выводы отмечены знаком тильды ~.

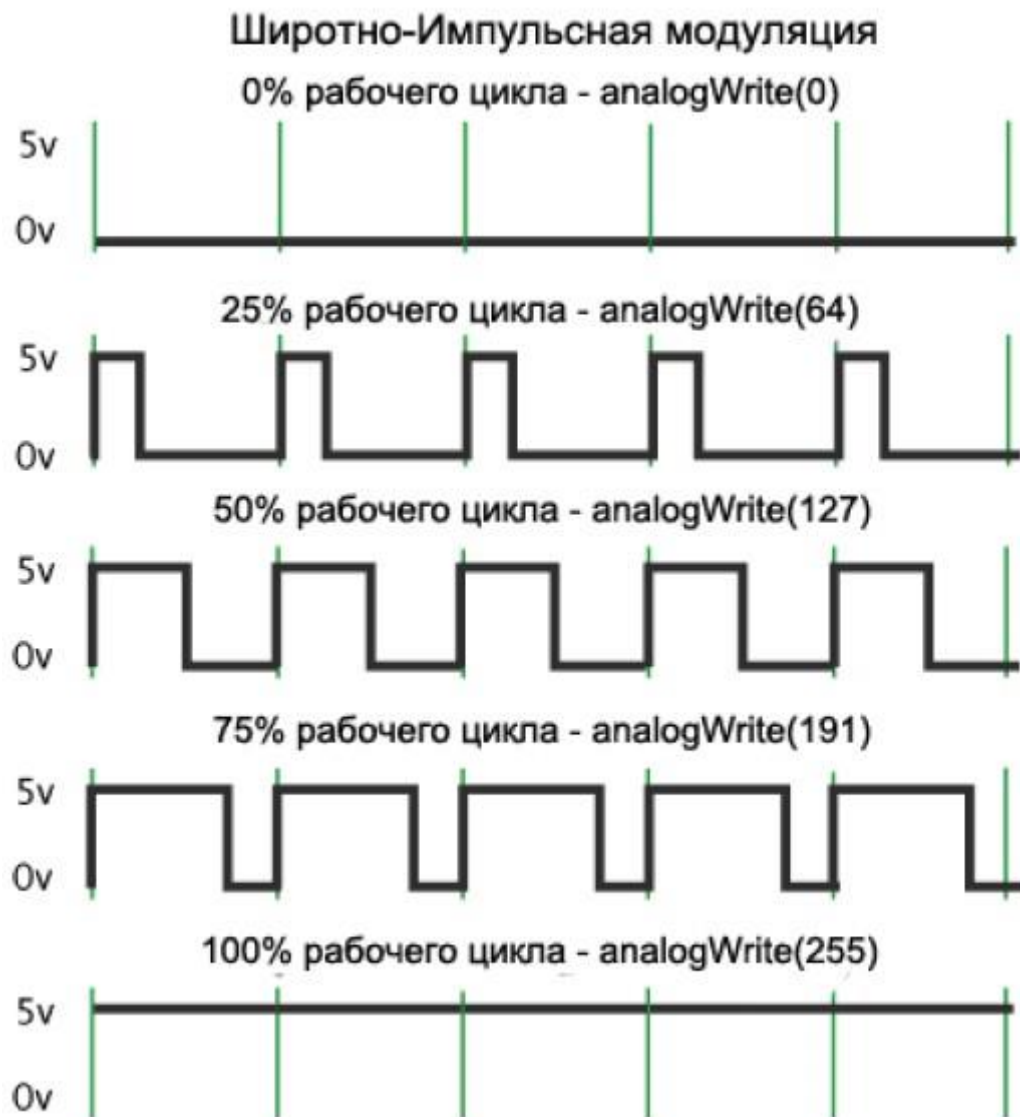


Рис. 5.1. Зависимость значений ШИМ от ширины импульса

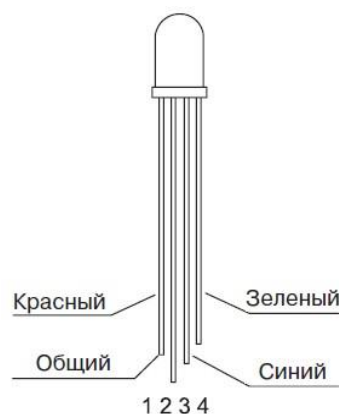


Рис. 5.2. Выводы RGB-светодиода

В данном эксперименте мы используем RGB-светодиод. RGB расшифровывается как аббревиатура Red, Green, Blue, при помощи этих цветов можно получить любой цвет путем смешения. Светодиод RGB отличается от обычного тем, что содержит 3 небольших кристалла R, G, B,

которые смогут синтезировать любой цвет или оттенок. RGB-светодиод имеет 4 вывода (см. рис. 5.2). Подключим RGB-светодиод к плате Arduino и заставим переливаться его цветами радуги. На рис. 5.3 показана схема подключения RGB-светодиода к плате Arduino.

Теперь перейдем к написанию скетча. На самом деле радуга имеет множество цветов, а 7 цветов были придуманы только потому, что эти цвета наиболее устойчиво воспринимаются и определяются глазом и мы их можем назвать и вспомнить поговоркой «Каждый Охотник Желает Знать, Где Сидит Фазан».

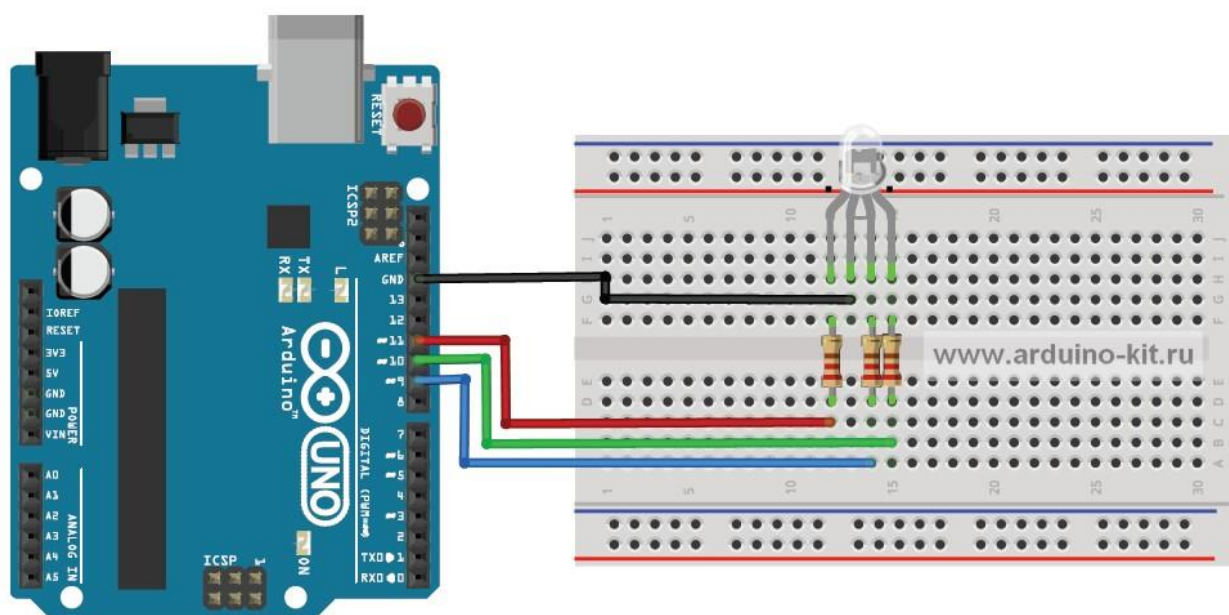


Рис. 5.3. Схема подключения RGB-светодиода

Список этих 7 основных цветов радуги с разложением по компонентам R, G и B представлен в табл. 5.1.

Цвет	R	G	B
Красный	255	0	0
Оранжевый	255	125	0
Желтый	255	255	0
Зеленый	0	255	0
Голубой	0	255	255
Синий	0	0	255
Фиолетовый	255	0	255

Наш светодиод должен переливаться от красного до фиолетового, проходя через все 7 основных цветов. Алгоритм вычисления любого промежуточного цвета радуги следующий:

1. Примем за начальную точку отсчета красный цвет (255, 0, 0).
2. Будем постепенно увеличивать значение зеленой составляющей G, пока не достигнем значения оранжевого (255, 125, 0), а затем и желтого цвета (255, 255, 0).
3. Постепенно уменьшим значение красной составляющей R до значения

зеленого цвета (0, 255, 0).

4. Постепенно увеличим значение синей составляющей В до значения голубого цвета (0, 255, 255).

5. Постепенно уменьшим количество зеленой составляющей G до значения синего цвета (0, 0, 255).

6. Постепенно увеличим количество красной составляющей R до значения фиолетового цвета (255, 0, 255).

7. Выдерживаем небольшую паузу и переходим к шагу 1.

Порядок подключения:

1. Чтобы видеть смешивание трех компонент R, G, B, а не отдельные составляющие, необходимо сделать поверхность светодиода шероховатой небольшой обработкой напильником или накрыть RGB-светодиод матовой пластиной.

2. Подключаем RGB-светодиод по схеме на рис. 5.3.

3. Составляем самостоятельно и загружаем в плату Arduino скетч

4. Наблюдаем свечение светодиода переливающимися цветами радуги.

ПРОЕКТ 6: СЕМИСЕГМЕНТНЫЙ ИНДИКАТОР ОДНОРАЗРЯДНЫЙ. ВЫВОДИМ ЦИФРЫ

В этом эксперименте мы рассмотрим работу с семисегментным светодиодным индикатором, которая позволяет Arduino визуализировать цифры.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [одноразрядный семисегментный индикатор](#);
- [резистор 510 Ом – 7 штук](#);
- [провода папа-папа](#).

Светодиодный семисегментный индикатор представляет собой группу светодиодов, расположенных в определенном порядке и объединенных конструктивно. Светодиодные контакты промаркированы метками от а до g (и дополнительно dp – для отображения десятичной точки), и один общий вывод, который определяет тип подключения индикатора (схема с общим анодом ОА, или общим катодом ОК). Зажигая одновременно несколько светодиодов, можно формировать на индикаторе символы цифр. Схема одноразрядного семисегментного индикатора показана на рис. 6.1.

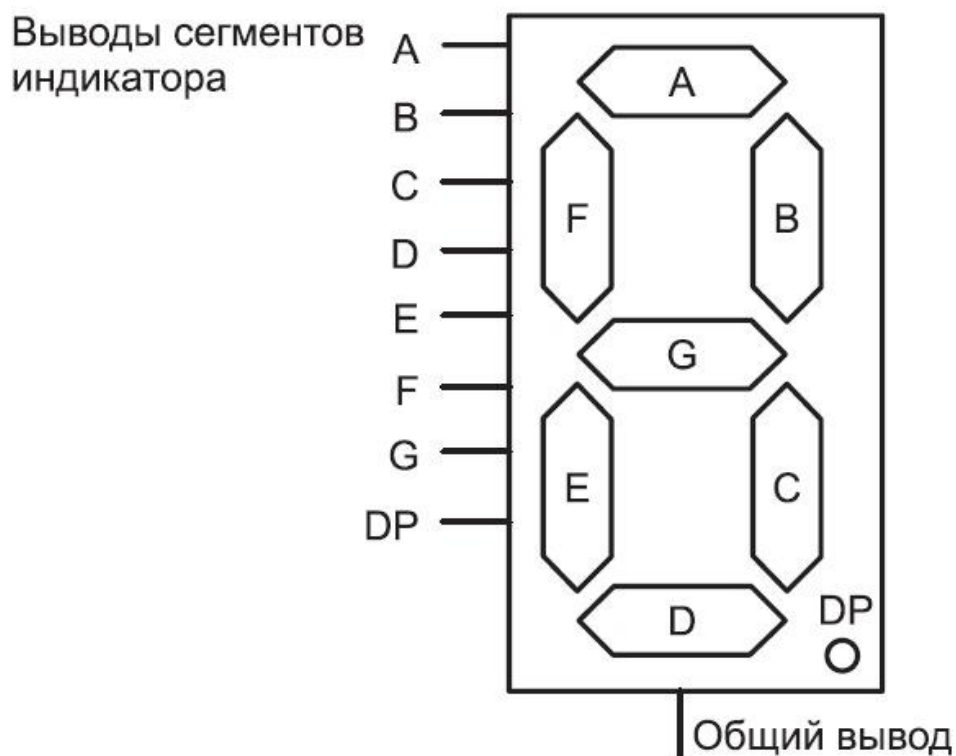


Рис. 6.1. Схема одноразрядного семисегментного индикатора

Для подключения одноразрядного светодиодного индикатора к Arduino будем задействовать 7 цифровых выводов, каждый из контактов a–g

The image shows an Arduino Uno R3 board connected to a breadboard. The breadboard contains a 7-segment display and several resistors. Blue wires connect the display's pins to the Arduino's digital pins. A black wire connects the display's common pin to ground. The display shows the number '8'.

Порядок подключения:

1. Подключаем семисегментный индикатор по схеме на рис. 6.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Наблюдаем вывод цифр на экран семисегментного индикатора.

ПРОЕКТ 7: МАТРИЦА 4-РАЗРЯДНАЯ ИЗ 7-СЕГМЕНТНЫХ ИНДИКАТОРОВ. ДЕЛАЕМ ДИНАМИЧЕСКУЮ ИНДИКАЦИЮ

В этом эксперименте мы рассмотрим работу Arduino с 4-разрядной семисегментной матрицей. Получим представление о динамической индикации, позволяющей использовать одни выводы Arduino при выводе информации на несколько семисегментных индикаторов.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [4-разрядная семисегментная матрица](#);
- [резистор 510 Ом – 8 штук](#);
- [кнопка](#);
- [резистор 10 кОм](#);
- [провода папа-папа](#).

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 4-разрядной матрицы на 7-сегментных индикаторах показана на рис. 7.1.

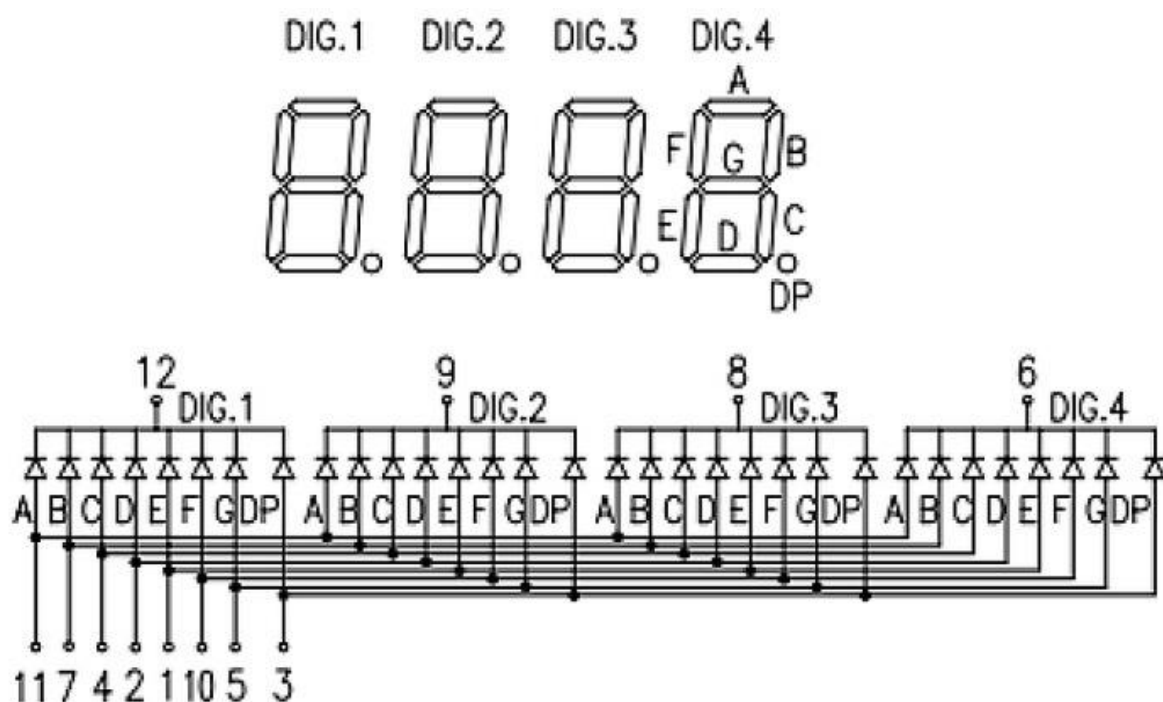


Рис. 7.1. Схема 4-разрядной матрицы на 7-сегментных индикаторах

Для вывода цифры необходимо зажечь нужные светодиоды на контактах A–G и DP и выбрать нужную матрицу подачей LOW на вывод 6, 8, 9 или 12. Подключим контакты матрицы к плате Arduino и будем выводить цифры на различные разряды матрицы. Для подключения нам понадобятся 12 выводов

Arduino. Схема соединений для подключения 4-разрядной матрицы к плате Arduino показана на рис. 7.2. При подключении контактов используются ограничительные резисторы 510 Ом.

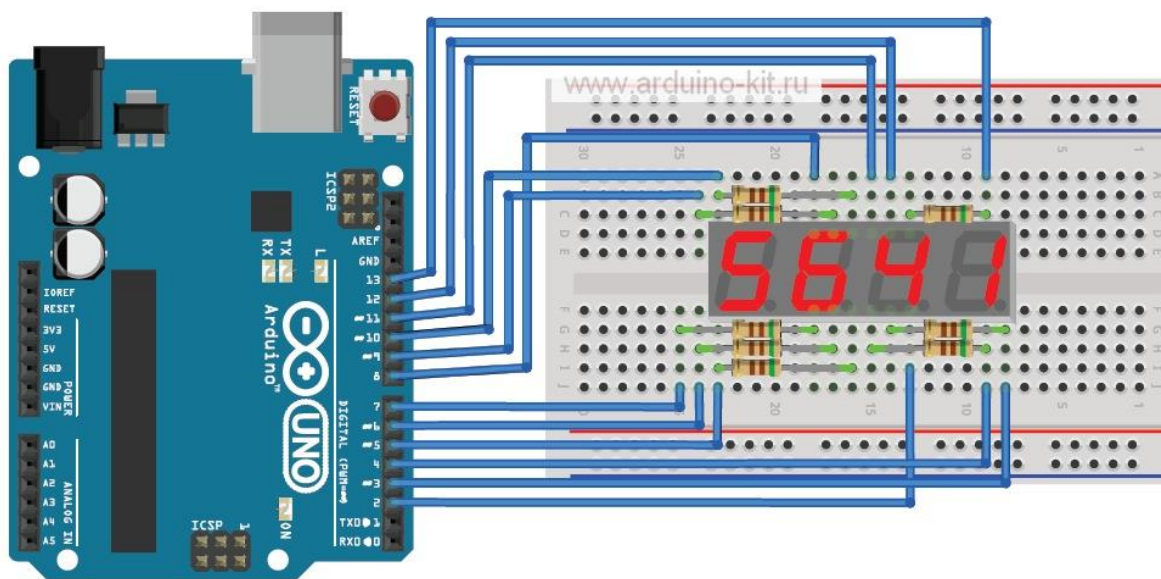


Рис. 7.2. Схема подключения 4-разрядной матрицы к Arduino

Напишем скетч последовательного вывода цифр (0–9) на произвольный регистр матрицы. Для выбора случайного значения из диапазона будем использовать функцию `random()`.

Порядок подключения:

1. Подключаем семисегментный индикатор по схеме на рис. 7.3.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Нажатием кнопки запускаем или останавливаем секундомер.

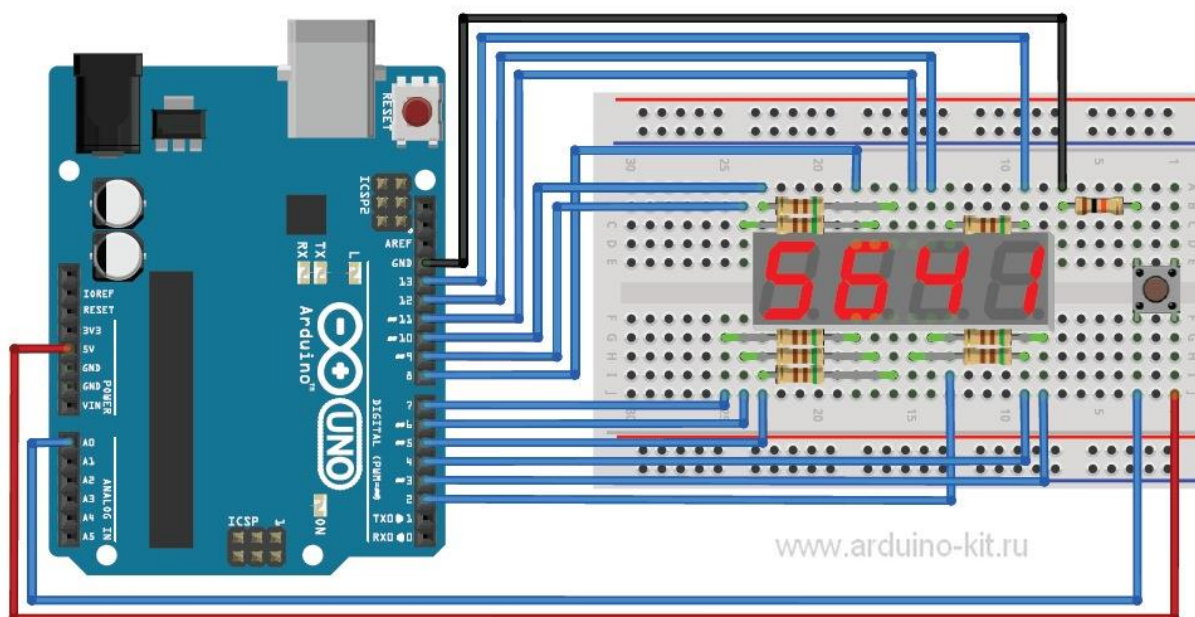


Рис. 7.3. Схема подключения для секундомера

ПРОЕКТ 8: МИКРОСХЕМА СДВИГОВОГО РЕГИСТРА 74НС595. УПРАВЛЯЕМ МАТРИЦЕЙ ИЗ 4 РАЗРЯДОВ, ЭКОНОМИМ ВЫХОДЫ ARDUINO

В этом эксперименте мы рассмотрим работу Arduino с микросхемой 74НС595 – расширителем выходов, позволяющей уменьшить количество выводов Arduino для управления 4-разрядной семисегментной матрицей.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [4-разрядная семисегментная матрица](#);
- микросхема 74НС595;
- [резистор 510 Ом – 7 штук](#);
- [провода папа-папа](#).

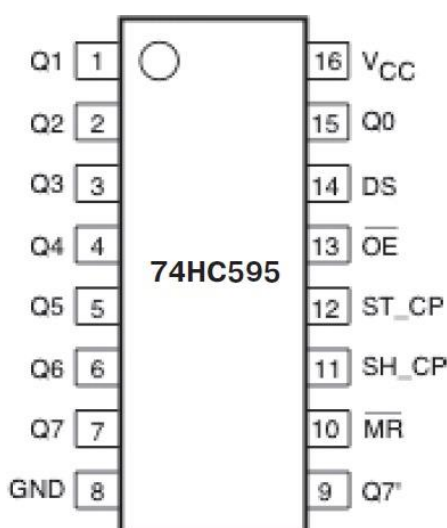


Рис. 8.1. Микросхема 74НС595

Цифровых выводов Arduino Nano и UNO, а иногда даже и Arduino Mega может не хватить, если требуется управлять большим количеством выводов. В этом случае можно использовать микросхему 74НС595. Микросхема 74НС595 – восьмиразрядный сдвиговый регистр с последовательным вводом, последовательным или параллельным выводом информации, с триггером-защелкой и тремя состояниями на выходе. Назначение контактов микросхемы 74НС595 показано на рис. 8.1.

Для управления нам вполне достаточно всего лишь трех выводов: SH_CP, ST_CP и DS. Когда на тактовом входе SH_CP появляется логическая единица, регистр считывает бит со входа данных DS и записывает его в самый младший разряд. При поступлении на тактовый вход следующего импульса все повторяется, только бит, записанный ранее, сдвигается на один разряд, а его место занимает вновь пришедший бит. Когда все восемь битов заполнились, и приходит девятый тактовый импульс, регистр снова начинает заполняться с младшего разряда, и все повторяется вновь.

Чтобы данные появились на выходах Q0...Q7, нужно их «защелкнуть». Для этого необходимо подать логическую единицу на вход ST_CP. Чтобы мы ни

Вывод Q7" предназначен для последовательного (каскадного) соединения сдвиговых регистров. При таком подключении биты из первого регистра будут проталкиваться в следующий в каскаде регистр, из него – в следующий и т. д. Таким образом, каскад из двух 8-битных регистров будет работать как один 16-битный. Можно соединить сколько угодно микросхем.

В следующем эксперименте со светодиодной матрицей мы рассмотрим каскадное подключение микросхем 74НС595, в данном эксперименте мы используем только одну микросхему – восемь выводов микросхемы 74НС595 подключены к восьми сегментам матрицы, четыре вывода матрицы для выбора разряда подключены к четырем выводам платы Arduino. Схема подключения представлена на рис. 8.2.

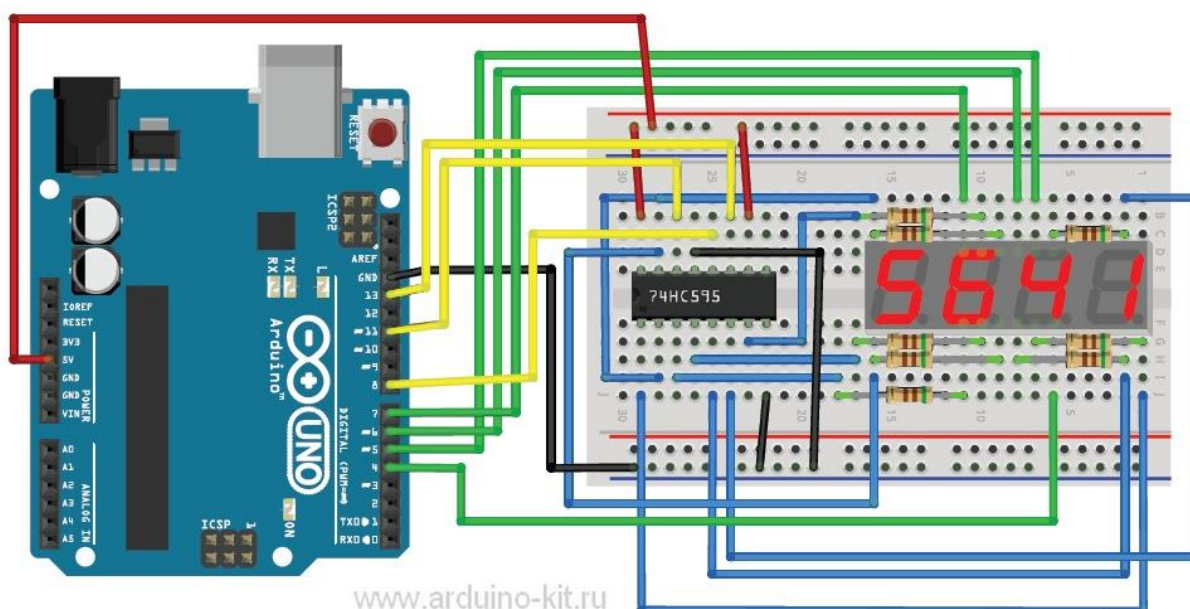


Рис. 8.2. Схема подключения 4-разрядного семисегментного индикатора с использованием сдвиговых регистров 74НС595

Приступаем к написанию скетча запуска и останова секундомера 0–999 сек с точностью 0.1 сек. Используем библиотеку Arduino SPI. Поскольку при использовании библиотеки SPI применяются Arduino выводы 11 и 13, для выбора регистров матрицы используем выводы Arduino 4, 5, 6, 7.

1. Подключаем семисегментный индикатор по схеме на рис. 8.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Нажатием кнопки запускаем или останавливаем секундомер.

ПРОЕКТ 9: МАТРИЦА СВЕТОДИОДНАЯ 8X8

В этом эксперименте мы рассмотрим каскадное подключение нескольких микросхем 74HC595, что позволит, используя 3 вывода Arduino, управлять множеством контактов, что будет продемонстрировано в примере вывода фигур на экран светодиодной матрицы 8×8.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [светодиодная RGB матрица 8×8](#);
- микросхема 74HC595 – 2 штуки;
- [провода папа-папа](#).
- [провода папа-мама](#).

В светодиодных матрицах светодиоды расположены в определенном порядке, а выводы расположены в удобном для монтажа порядке. Светодиодные матрицы бывают одноцветными, двухцветными и RGB. В эксперименте будем использовать двухцветную светодиодную матрицу FYM-23881BUG-11, которая представляет собой набор из 64 светодиодов зеленого цвета, собранных в матрицу 8×8. Расположение выводов матрицы показано на рис. 9.1.

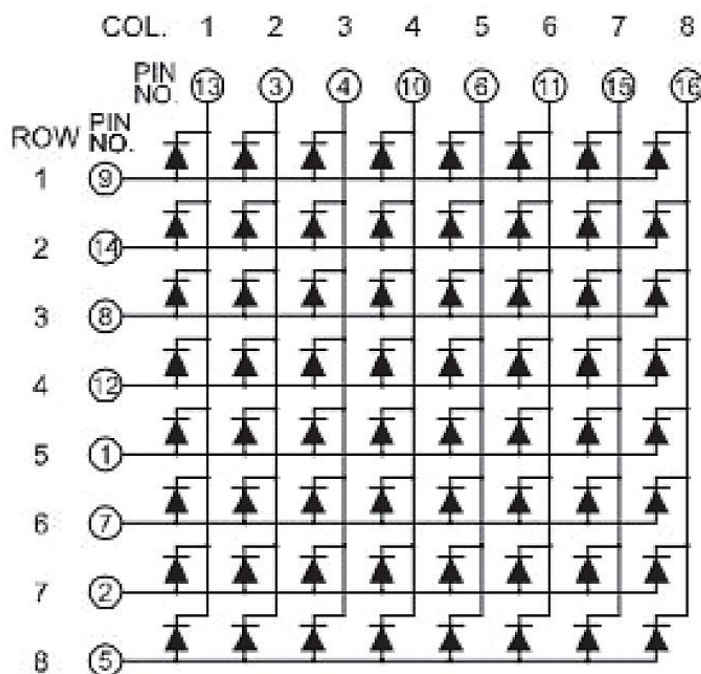


Рис. 9.1. Расположение выводов матрицы FYM-23881BUG-11

Для подключения светодиодной матрицы к Arduino будем использовать каскадное подключение 2 микросхем 74HC595. При таком подключении биты из первого регистра будут проталкиваться в следующий в каскаде

Для формирования изображения матрицы будем использовать динамическую индикацию для каждого столбца. Каждые 3 секунды будем менять фигуру для матрицы. Данные фигур хранятся в массиве `figure[]`.

1. Подключаем матрицу по схеме на рис. 9.2.
2. Загружаем в плату Arduino
3. Наблюдаем процесс попеременного вывода фигур на экране матрицы.

1. Подключаем матрицу по схеме на рис. 9.2.
2. Загружаем в плату Arduino
3. Наблюдаем процесс попеременного вывода фигур на экране матрицы.

ПРОЕКТ 10: УПРАВЛЯЕМ ПЬЕЗОИЗЛУЧАТЕЛЕМ: МЕНЯЕМ ТОН, ДЛИТЕЛЬНОСТЬ, ИГРАЕМ МУЗЫКУ

В этом эксперименте мы произведем генерацию звуков на Arduino с помощью пьезоизлучателя.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [пьезоизлучатель](#);
- [резистор 100 Ом](#);
- [провода папа-папа](#).

Самым простым вариантом генерации звука является использование пьезоизлучателя. Пьезокерамические излучатели (пьезоизлучатели) – электроакустические устройства воспроизведения звука, использующие обратный пьезоэлектрический эффект – возникновение механических деформаций под действием электрического поля. Пьезоизлучатели бывают двух типов – со встроенным генератором и без. Пьезоизлучатели со встроенным генератором излучают фиксированный тональный сигнал сразу после подачи на них номинального напряжения. Они не могут воспроизводить произвольного сигнала. Их обычно используют для простого звукового оповещения. Если требуется проиграть мелодию, то используют пьезоизлучатели без встроенного генератора и генерируют сигнал отдельно. В эксперименте мы используем пьезоизлучатель без встроенного генератора. Схема подключения пьезоизлучателя показана на рис. 10.1.

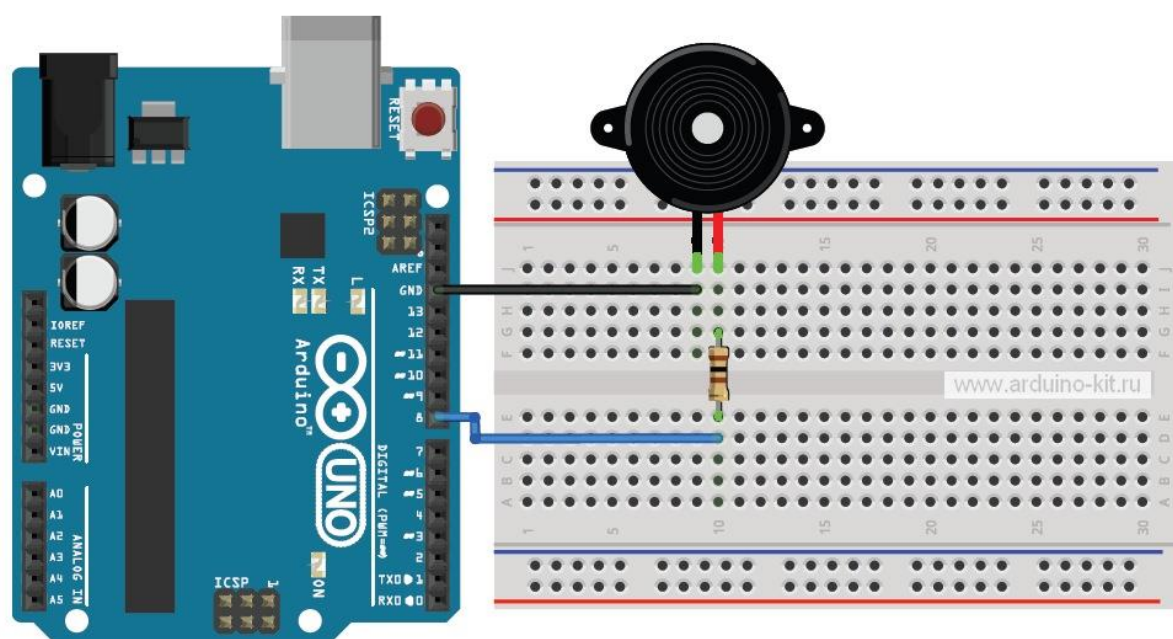


Рис. 10.1. Схема подключения пьезоизлучателя к Arduino

Приступим к написанию скетча. Для воспроизведения мелодии необходимо

подавать последовательно звуки определенной частоты и длительности. Для генерации звуков определенной частоты и длительности будем использовать Arduino-функцию `tone()`: `tone(pin,frequency,duration)`; Функция `tone()` генерирует на выводе прямоугольный сигнал заданной частоты (с коэффициентом заполнения 50%). Функция также позволяет задавать длительность сигнала. Если длительность сигнала не указана, он будет генерироваться до тех пор, пока не будет вызвана функция `noTone()`. Значения частот для нот первой и второй октав представлены в табл. 10.1.

1 октава	Обозначение	Частота, Гц	2 октава	Обозначение	Частота, Гц
до	C	261	до	c	523
до-диез	C#(R)	277	до-диез	c#(r)	554
ре	D	293	ре	d	587
ре-диез	D#(S)	311	ре-диез	d#(s)	622
ми	E	329	ми	e	659
фа	F	349	фа	f	698
фа-диез	F#(T)	370	фа-диез	f#(t)	740
соль	G	392	соль	g	784
соль-диез	G#(U)	415	соль-диез	g#(u)	830
ля	A	440	ля	a	880
си-бимоль	B	466	си-бимоль	b	932
си	H	494	си	h	988

Составим мелодию понотно, занесем в массив `melody[]`, список длительностей нот занесем в массив `duration[]`.



Рис. 10.2. Фрагмент Имперского марша

Данные с обозначением нот занесем в массив `notes[]`, а данные с частотами для соответствующих нот – в массив `frequency[]`.

Порядок подключения:

1. Подключаем пьезоэлемент к плате Arduino по схеме на рис. 10.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. После загрузки скетча слушаем мелодию на монтажной плате (рис. 10.3).

ПРОЕКТ 11: ТРАНЗИСТОР MOSFET. ПОКАЗЫВАЕМ УСИЛИТЕЛЬНЫЕ КАЧЕСТВА ТРАНЗИСТОРА. НА ПРИМЕРЕ ЭЛЕКТРОДВИГАТЕЛЯ ИЗМЕНЯЕМ ОБОРОТЫ

В этом эксперименте мы познакомимся с транзистором MOSFET и с помощью него будем управлять мощной нагрузкой – электродвигателем.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- транзистор MOSFET IRF540;
- диод 1N4007;
- двигатель DC;
- [блок питания 5 В](#);
- [провода папа-папа](#).

Выводы Arduino, сконфигурированные как OUTPUT, находятся в низкоимпедансном состоянии и могут отдавать 40 мА в нагрузку и не в состоянии обеспечить питание мощной нагрузки и большого напряжения. Одним из способов управления мощной нагрузкой является использование полевых MOSFET-транзисторов. MOSFET-транзистор – это ключ для управления большими токами при помощи небольшого напряжения (в отличие от биполярных транзисторов, управляемых током). В нашем эксперименте мы будем управлять скоростью вращения мотора изменением напряжения, подаваемого на MOSFET. Управлять напряжением, подаваемым на MOSFET, будем с помощью ШИМ (широтно-импульсной модуляции). В эксперименте 5 мы уже рассматривали использование ШИМ для получения изменяющегося аналогового значения посредством цифровых сигналов. Для регулирования скорости двигателя будем использовать потенциометр. Схема подключения элементов для данного эксперимента показана на рис. 11.1.

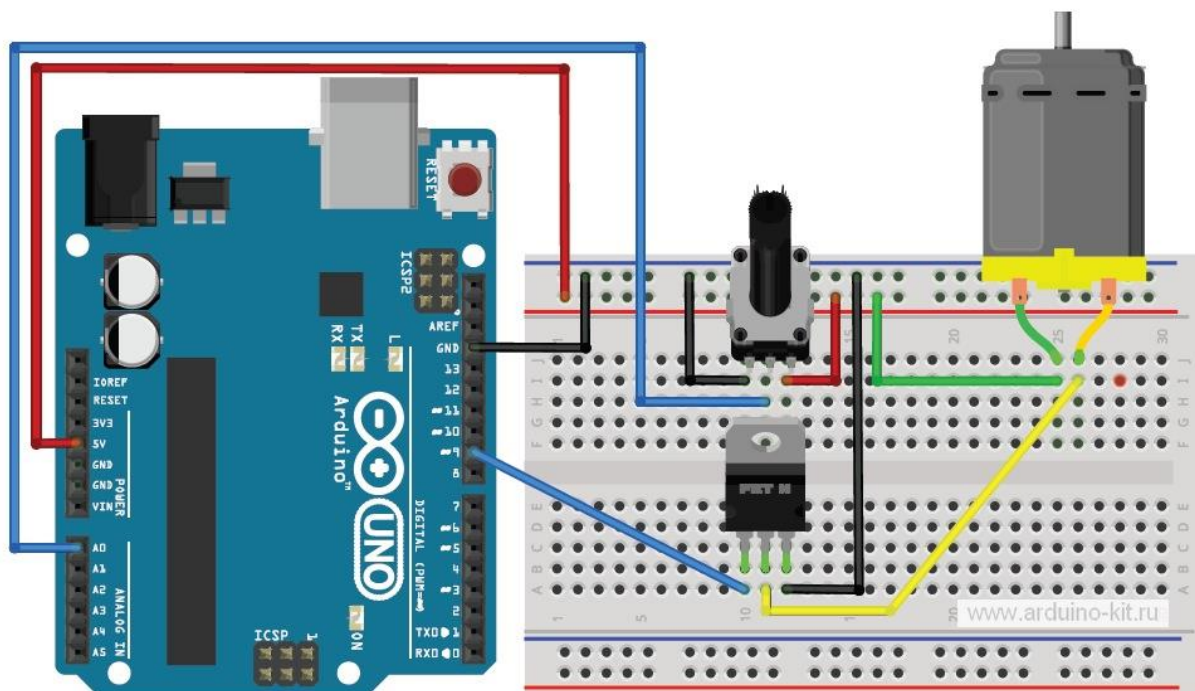


Рис. 11.1. Схема подключения мотора к Arduino

Порядок подключения:

1. Подключаем элементы к плате Arduino по схеме на рис. 11.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Крутим потенциометр – изменяем скорость вращения мотора.

ПРОЕКТ 12: УПРАВЛЯЕМ РЕЛЕ ЧЕРЕЗ ТРАНЗИСТОР

В этом эксперименте мы познакомимся с реле, с помощью которого с Arduino можно управлять мощной нагрузкой не только постоянного, но и переменного тока.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- биполярный транзистор С945;
- диод 1N4004;
- реле;
- [провода папа-папа](#).
- [провода папа-мама](#).

Реле – это электрически управляемый, механический переключатель, имеет две отдельные цепи: цепь управления, представленная контактами (А1, А2), и управляемая цепь, контакты 1, 2, 3 (см. рис. 12.1).

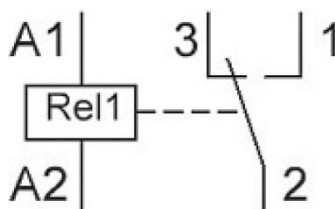


Рис. 12.1. Схема подключения реле к Arduino (n-канальное управление)

Цепи никак не связаны между собой. Между контактами А1 и А2 установлен металлический сердечник, при протекании тока по которому к нему притягивается подвижный якорь (2). Контакты же 1 и 3 неподвижны. Стоит отметить, что якорь подпружинен, и пока мы не пропустим ток через сердечник, якорь будет прижатым к контакту 3. При подаче тока, как уже говорилось, сердечник превращается в электромагнит и притягивается к контакту 1. При обесточивании пружина снова возвращает якорь к контакту 3.

При подключении реле к Arduino контакт микроконтроллера не может обеспечить мощность, необходимую для нормальной работы катушки. Поэтому следует усилить ток – поставить транзистор. Для усиления удобнее применять n-p-n-транзистор, включенный по схеме ОЭ (см. рис. 12.2). При таком способе можно подключать нагрузку с большим напряжением питания, чем питание микроконтроллера.

Резистор на базе – ограничительный. Может варьироваться в широких пределах (1–10 кОм), в любом случае, транзистор будет работать в режиме насыщения. В качестве транзистора может быть любой n-p-n-транзистор. Коэффициент усиления практически не имеет значения. Выбирается транзистор по току коллектора (нужный нам ток) и напряжению коллектор–эмиттер (напряжение, которым запитывается нагрузка).

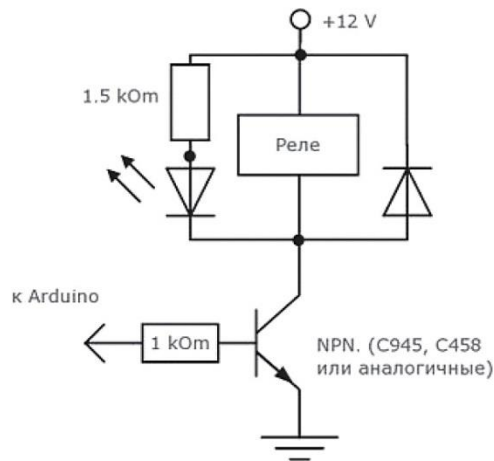


Рис. 12.2. Схема подключения реле к Arduino (n-канальное управление)

Для включения реле, подключенного по схеме с ОЭ, на вывод Arduino необходимо подать 1, для выключения – 0. Подключим реле к плате Arduino по схеме на рис. 12.3 и напишем скетч управления реле. Каждые 5 секунд реле будет переключаться (включаться/выключаться). При переключении реле раздается характерный щелчок.

Порядок подключения:

1. Подключаем элементы к плате Arduino по схеме на рис. 12.3.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Каждые 5 секунд происходит щелчок переключения реле если подключить контакты реле, например в разрыв подключенной к сети 220 В патрона с лампой накаливания, то увидим процесс включения/выключения лампы накаливания раз в 5 секунд (рис. 12.3).

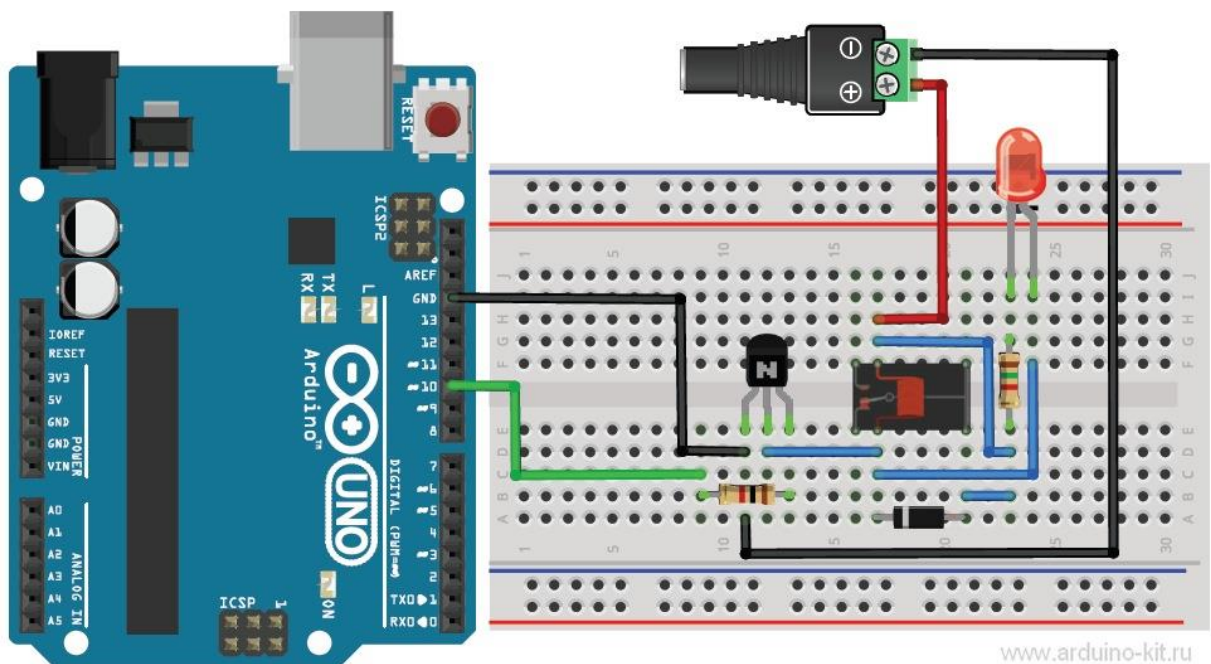


Рис. 12.3 Схема подключения

ПРОЕКТ 13: ФОТОРЕЗИСТОР. ОБРАБАТЫВАЕМ ОСВЕЩЁННОСТЬ, ЗАЖИГАЯ ИЛИ ГАСЯ СВЕТОДИОДЫ

В этом эксперименте мы познакомимся с аналоговым датчиком для измерения освещенности – фоторезистором (рис. 13.1).

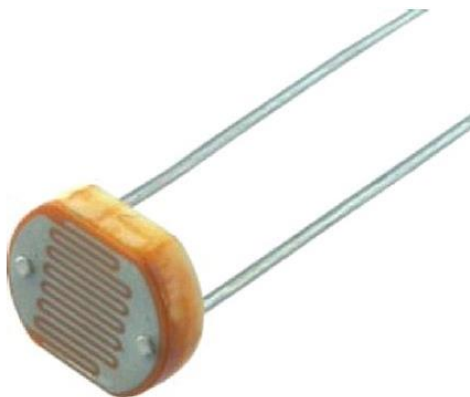


Рис. 13.1. Фоторезистор

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- фоторезистор;
- [резистор 10 кОм](#);
- [резистор 220 Ом – 8 штук](#);
- [светодиод](#);
- [провода папа-папа](#).

Распространённое использование фоторезистора – измерение освещённости. В темноте его сопротивление довольно велико. Когда на фоторезистор попадает свет, сопротивление падает пропорционально освещенности. Схема подключения фоторезистора к Arduino показана на рис. 13.2. Для измерения освещённости необходимо собрать делитель напряжения, в котором верхнее плечо будет представлено фоторезистором, нижнее – обычным резистором достаточно большого номинала. Будем использовать резистор 10 кОм. Среднее плечо делителя подключаем к аналоговому входу A0 Arduino.

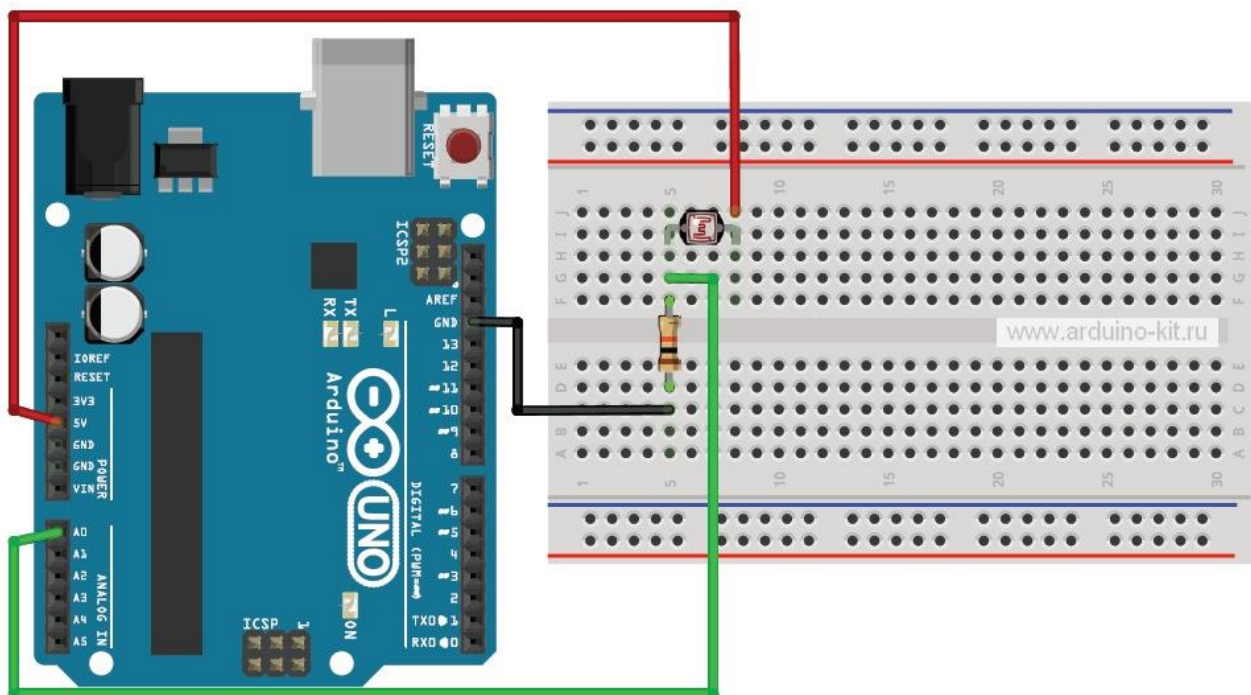


Рис. 13.2. Схема подключения фоторезистора к Arduino

Напишем скетч чтения аналоговых данных и отправки их в последовательный порт.

Порядок подключения:

1. Подключаем фоторезистор по схеме на рис. 13.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Регулируем рукой освещенность фоторезистора и наблюдаем вывод в последовательный порт изменяющихся значений, запоминаем показания при полной освещенности помещения и при полном перекрытии светового потока.

Теперь создадим индикатор освещенности с помощью светодиодного ряда из 8 светодиодов. Количество горящих светодиодов пропорционально текущей освещенности. Собираем светодиоды по схеме на рис. 13.3, используя ограничительные резисторы номиналом 220 Ом.

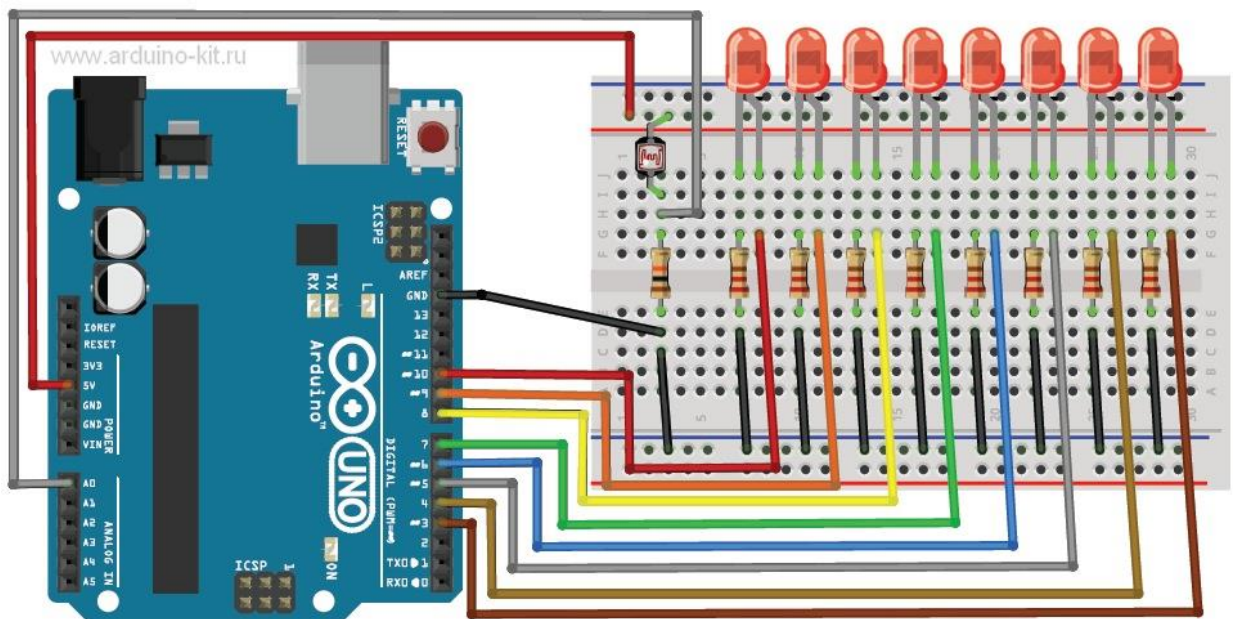


Рис. 13.3. Схема подключения фоторезистора и светодиодов к Arduino

Порядок подключения:

1. Подключаем фоторезистор и светодиоды по схеме на рис. 13.3.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Регулируем рукой освещенность фоторезистора и по количеству горящих светодиодов определяем текущий уровень освещенности (рис. 13.3).

Нижний и верхний пределы освещенности мы берем из запомненных значений при проведении эксперимента по предыдущему скетчу. Промежуточное значение освещенности мы масштабируем на 8 значений (8 светодиодов) и зажигаем количество светодиодов пропорциональное значению между нижней и верхней границами.

ПРОЕКТ 14: ДАТЧИК ТЕМПЕРАТУРЫ АНАЛОГОВЫЙ LM335. ПРИНЦИП РАБОТЫ, ПРИМЕР РАБОТЫ

В этом эксперименте мы познакомимся с аналоговым датчиком для измерения температуры LM335.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- датчик температуры LM335;
- [резистор 2,2 кОм](#);
- [RGB-светодиод](#);
- [резистор 220 Ом – 3 штуки](#);
- [провода папа-папа](#).

LM335 – это недорогой температурный чувствительный элемент с диапазоном от -40°C до $+100^{\circ}\text{C}$ и точностью в 1°C . По принципу действия датчик LM335 представляет собой стабилитрон, у которого напряжение стабилизации зависит от температуры. При повышении температуры на один градус Кельвина напряжение стабилизации увеличивается на 10 милливольт. Для измерения температуры используются 2 вывода, третий нужен для калибровки датчика. В качестве примера использования датчика LM335 создадим индикатор температуры окружающей среды на RGB-светодиоде. Схема подключения показана на рис. 14.1.

Приступим к написанию скетча. Нам необходимо получить значение с аналогового входа A0 подключения датчика LM335, перевести в значение в вольтах, исходя из значения опорного напряжения +5 В. Мы получим значение температуры в Кельвинах. Для получения значения в градусах Цельсия полученное значение необходимо уменьшить на величину 273.15. Определим комфортное значение температуры в интервале MIN_T–MAX_T ($20\text{--}27^{\circ}\text{C}$). При попадании значения в этот интервал RGB-светодиод горит желтым цветом, при пониженном значении – синим, при повышенном – красным.

Для проверки выводим значение температуры в монитор последовательного порта Arduino IDE.

ПРОЕКТ 15: ИНДИКАТОР LCD1602. ПРИНЦИП ПОДКЛЮЧЕНИЯ, ВЫВОД ИНФОРМАЦИИ НА НЕГО

В этом эксперименте мы познакомимся с жидкокристаллическими индикаторами Winstar для вывода символьной информации. Научимся в Arduino-проектах применять библиотеки и создадим проект вывода показаний датчика температуры LM335 на экран дисплея.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [LCD-экран WH1602](#);
- [резистор 2,2 кОм](#);
- [резистор 50 Ом](#);
- потенциометр 1 кОм;
- датчик температуры LM335;
- [провода папа-папа](#).
- [внешний блок питания +5 В](#).

Жидкокристаллические индикаторы (ЖКИ, англ. LCD) являются удобным и недорогим средством для отображения данных ваших проектов. Символьный индикатор WH1602 позволяет выводить на экран 2 строки по 16 символов (размером 5×7 или 5×10 и дополнительная строка под курсор). Управляет работой дисплея контроллер.

На рис. 15.1 показан ЖКИ Winstar с контроллером HD44780.

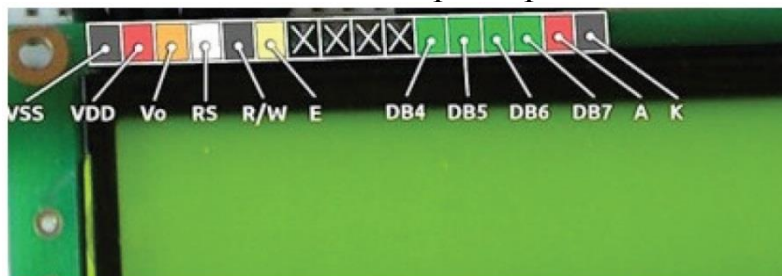


Рис. 15.1. ЖКИ WH1602 на контроллере HD44780

Назначение выводов контроллера:

- DB0–DB7 – отвечают за входящие/исходящие данные;
- RS – высокий уровень означает, что сигнал на выходах DB0–DB7 является данными, низкий – командой;
- R/W – определяет направление данных (чтение/запись). Так как операция чтения данных из индикатора обычно бывает неостребованной, то можно установить постоянно на этом входе низкий уровень;
- E – импульс длительностью не менее 500 мс на этом выводе определяет сигнал для чтения/записи данных с выводов DB0–DB7, RS и W/R;
- V0 – используется для задания контраста изображения;
- A, K – питание подсветки (анод и катод), если она имеется;
- VSS – земля;
- VDD – питание ЖК-индикатора.

Для управления ЖК-индикатором необходимо 6 или 10 выводов Arduino, в зависимости от того, выбран 4- или 8-битный режим обмена данными. Для сокращения требуемого числа выводов микроконтроллера можно работать в 4-битном режиме. В этом случае на выводах DB4–DB7 индикатора сначала будут передаваться старшие четыре бита данных/команды, затем – младшие четыре бита. Выводы DB0–DB3 останутся незадействованными.

В нашем эксперименте мы будем считывать данные с датчика температуры LM335, который мы рассмотрели в эксперименте 13, и выводить на экран ЖКИ значение температуры в Кельвинах и градусах Цельсия. Схема подключения датчика температуры и ЖКИ в 4-битном режиме к плате Arduino показана на рис. 15.2. Заметьте, что для питания ЖКИ нужен отдельный блок питания +5 В.

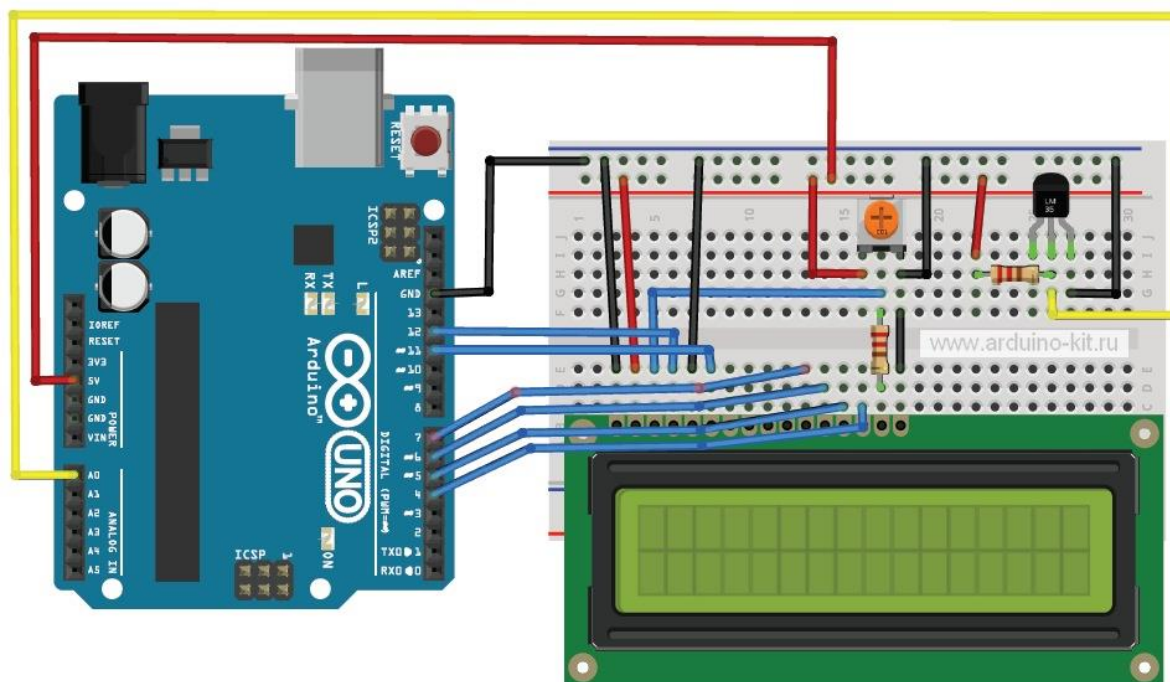


Рис. 15.2. Схема подключения датчика температуры и ЖКИ к Arduino

Приступим к написанию скетча. Функционал Arduino может быть расширен за счет использования библиотек. Библиотеки Arduino предоставляют дополнительную функциональность для использования в скетчах и сильно упрощают процесс написания программ. Ряд основных библиотек устанавливается вместе со средой Arduino IDE, а дополнительные, которых очень много, вы можете установить сами. При работе Arduino с ЖКИ-дисплеями на контроллере HD44780 будем использовать библиотеку LiquidCrystal.

Порядок подключения:

1. Подключаем датчик LM335 и ЖКИ по схеме на рис. 15.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Смотрим на экране дисплея показания датчика температуры в Кельвинах и градусах Цельсия (рис. 15.2).

ПРОЕКТ 16: ГРАФИЧЕСКИЙ ИНДИКАТОР. ПОДКЛЮЧЕНИЕ ДИСПЛЕЯ NOKIA 5110

В этом эксперименте мы рассмотрим графический дисплей Nokia 5110, который можно использовать в проектах Arduino для вывода графической информации.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [графический дисплей Nokia 5110](#);
- [провода папа-папа](#).
- фоторезистор;
- [резистор 2 кОм](#).

Жидкокристаллический дисплей Nokia 5110 – монохромный дисплей с разрешением 84×48 на контроллере PCD8544, предназначен для вывода графической и текстовой информации. Питание дисплея должно лежать в пределах 2.7–3.3 В (максимум 3.3 В, при подаче 5 В на вывод VCC дисплей может выйти из строя). Но выводы контроллера толерантны к +5 В, поэтому их можно напрямую подключать к входам Arduino. Немаловажный момент – низкое потребление, что позволяет питать дисплей от платы Arduino без внешнего источника питания.

Схема подключения Nokia 5110 к Arduino показана на рис. 16.1.

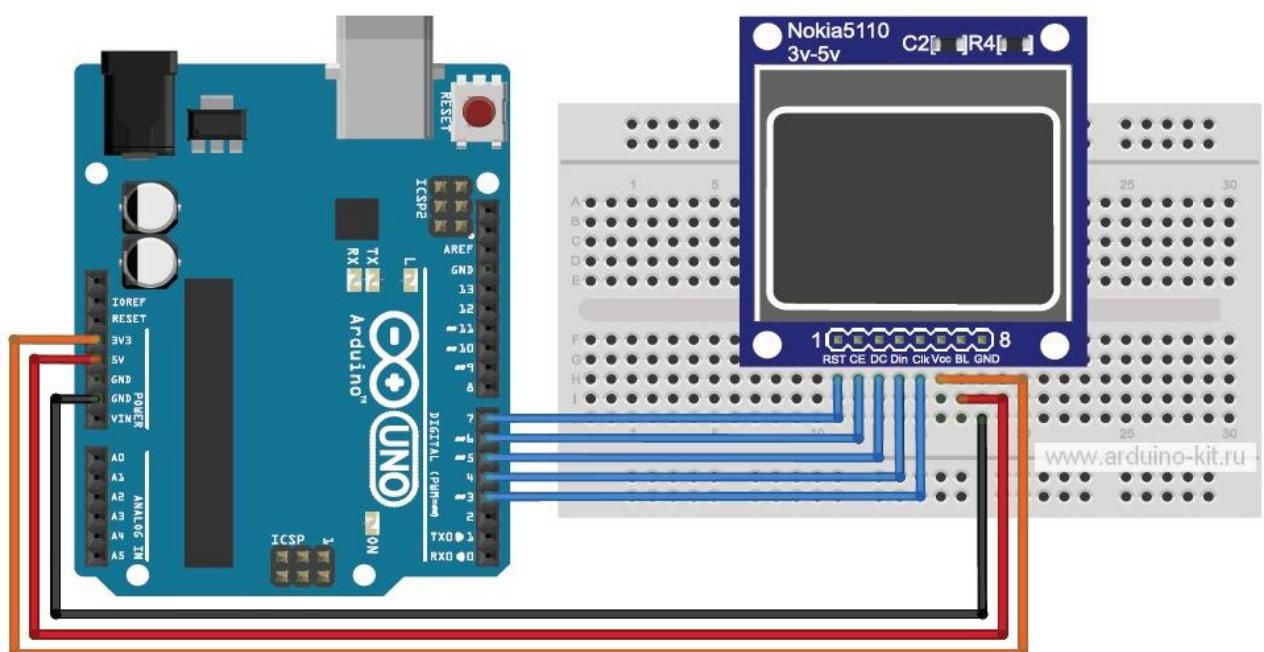


Рис. 16.1. Схема подключения Nokia 5110 к Arduino

Для работы с дисплеем Nokia 5110 будем использовать библиотеку Adafruit_GFX, которая имеет богатые возможности для вывода графики и текста. В нашем эксперименте мы будем получать данные освещенности с

фоторезистора, подключенного к аналоговому входу Arduino A0, и выводить данные освещенности в числовом и графическом представлениях. Схема подключения показана на рис. 16.2.

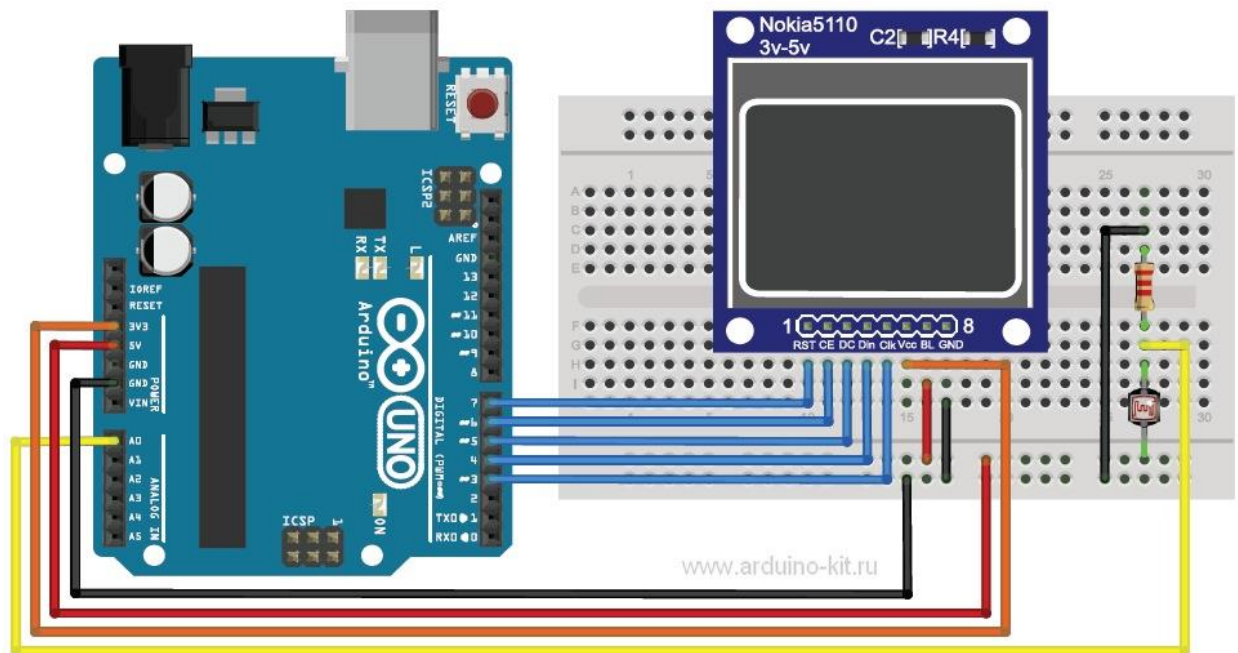


Рис. 16.2. Схема подключения Nokia 5110 и фоторезистора к Arduino

Мы считываем данные с фоторезистора и отображаем числовое значение, а также в графическом виде (прогресс-бар) значение освещенности в процентах от максимального значения. Значения минимальной и максимальной освещенности берем из эксперимента 13.

Порядок подключения:

1. Подключаем датчик дисплея Nokia 5110 и фоторезистор по схеме на рис. 16.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Перекрывая рукой поток света, смотрим на экране дисплея изменение показаний освещенности.

ПРОЕКТ 17: СЕРВОПРИВОД. КРУТИМ ПОТЕНЦИОМЕТР, МЕНЯЕМ ПОЛОЖЕНИЕ

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [сервопривод](#);
- [потенциометр 1 кОм](#);
- [провода папа-папа](#).
- [внешний блок питания +5 В](#).

Сервопривод (см. рис. 17.1) – устройство, обеспечивающее преобразование сигнала в строго соответствующее этому сигналу перемещение (как правило, поворот) исполнительного устройства. Представляет собой прямоугольную коробку с мотором, схемой и редуктором внутри и выходным валом, который может поворачиваться на строго фиксированный угол, определяемый входным сигналом. Как правило, этот угол имеет предел в 60 в 180. Кроме этого, еще бывают сервоприводы и постоянного вращения.



Рис. 17.1. Сервопривод

Сервопривод подключается с помощью трех проводов к управляющему устройству (драйверу или контроллеру) и источнику питания. Сервопривод управляется с помощью импульсов переменной длительности. Угол поворота определяется длительностью импульса, который подается по сигнальному проводу. Это называется широтно-импульсной модуляцией. Сервопривод ожидает импульса каждые 20 мс. Длительность импульса определяет, насколько далеко должен поворачиваться мотор. Например, импульс в 1,5 мс диктует мотору поворот в положение 90° (нейтральное положение). Когда сервопривод получает команду на перемещение, его управляющий орган перемещается в это положение и удерживает его. Если внешняя сила действует на сервопривод, когда он удерживает заданное положение, сервопривод будет сопротивляться перемещению из этого положения. Максимальная величина силы, которую может выдерживать сервопривод, характеризует вращающий момент сервопривода. Однако сервопривод не навсегда удерживает свое положение, импульсы позиционирования должны

повторяться, информируя сервопривод о сохранении положения. В нашем эксперименте мы будем управлять положением сервопривода с помощью потенциометра. Схема подключения сервопривода и потенциометра к плате Arduino показана на рис. 17.2.

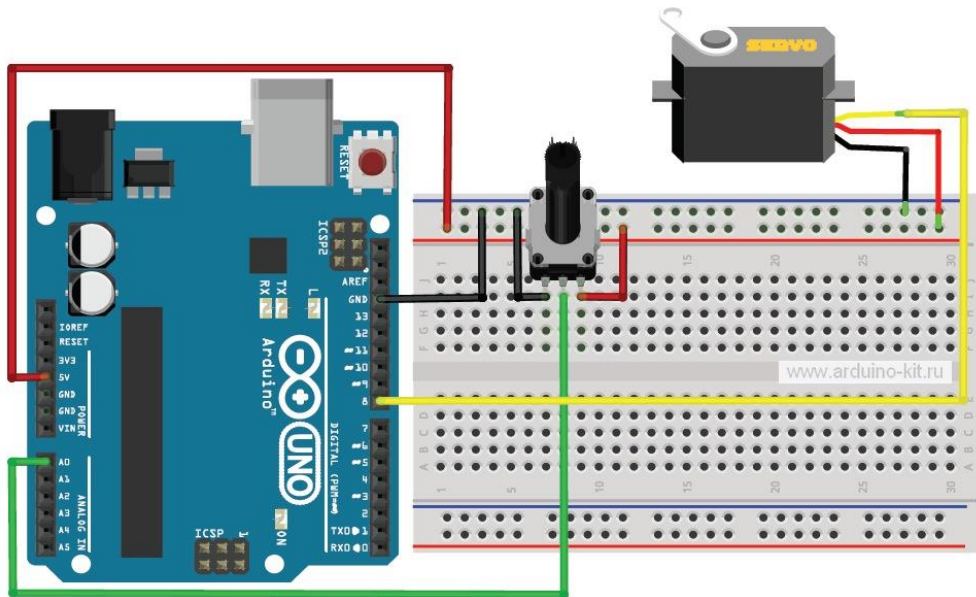


Рис. 17.2. Схема подключения сервопривода и потенциометра к Arduino

Сервопривод подключается тремя проводами: питание (Vcc), «земля» (Gnd) и сигнальный (C). Питание – красный провод, он может быть подключен к +5 В внешнего источника питания, черны (или коричневый) провод – «земля» – подключается к GND-выводу Arduino GND, сигнальный (оранжевый / желтый / белый) провод подключается к цифровому выводу контроллера Arduino. Для питания сервопривода используем отдельный блок питания +5 В. Для управления сервоприводом в Arduino имеется стандартная библиотека Servo. На платах, отличных от Mega, использование библиотеки отключает возможность применения analogWrite() (ШИМ) на пинах 9 и 10 (вне зависимости, подключены к этим пинам сервы или нет). На платах Mega до 12 сервоприводов могут использоваться без влияния на функциональность ШИМ, но использование от 12 до 23 сервомашинок отключит PWM ШИМ на пинах 11 и 12.

Аналоговые данные потенциометра (0–1023) масштабируем функцией map() в значения угла поворота сервопривода (0–180) и с помощью библиотечной функции servo.write(angle) даем сервоприводу команду для поворота.

Порядок подключения:

1. Подключаем датчик сервопривода и потенциометр по схеме на рис. 17.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Поворотом ручки потенциометра управляем положением сервопривода.

ПРОЕКТ 18: ОБРАБАТЫВАЕМ ДАННЫЕ ОТ ДЖОЙСТИКА. УПРАВЛЕНИЕ PAN/TILT BRACKET С ПОМОЩЬЮ ДЖОЙСТИКА

В этом эксперименте мы рассмотрим подключение к Arduino двухосевого аналогового джойстика.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль джойстика](#);
- [кронштейн Pan/Tilt Bracket](#);
- [сервопривод – 2 шт.](#);
- [провода папа-папа](#).

Для управления каким-либо устройством на основе Arduino, перемещающимся в двухмерной системе координат, отлично подойдет джойстик. Для плат Arduino существуют модули аналогового джойстика, имеющие ось X, Y (потенциометры 10 кОм) и дополнительную кнопку – ось Z. Джойстик позволяет плавно и точно отслеживать степень отклонения от нулевой точки. Сам джойстик подпружиненный, поэтому он будет возвращаться в центральное состояние после его отпускания из определенной позиции.

Контакты Vcc и GND между всеми тремя группами контактов соединены. Таким образом, для подключения нужно 5 проводов: ось X, ось Y, кнопка Z, питание Vcc и общий GND. Джойстики – пассивные модули и не потребляют какую-либо энергию от платы Arduino. Выводы VERT и HORZ подключаются к аналоговым входам A0 и A1 Arduino, SEL – к цифровому входу D2. Схема подключения показана на рис. 18.1.

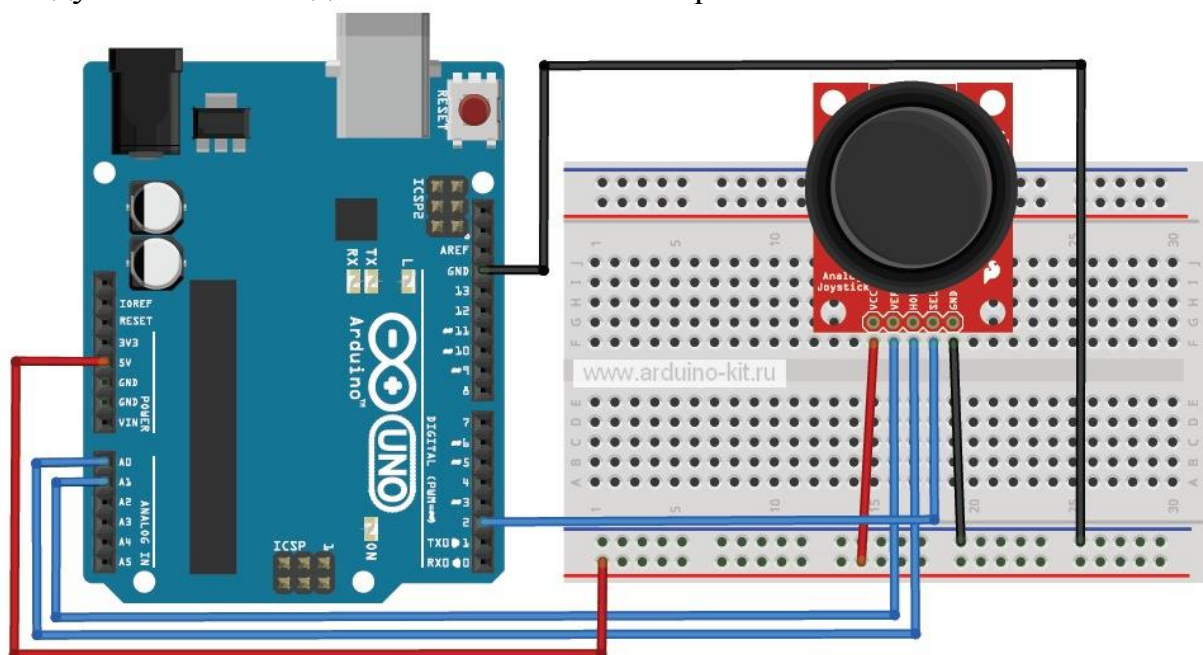


Рис. 18.1. Схема подключения платы джойстика к Arduino

Напишем скетч считывания данных джойстика и вывода значений в монитор последовательного порта Arduino IDE.

Порядок подключения:

1. Подключаем джойстик к плате Arduino по схеме на рис. 18.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Смотрим в мониторе последовательного порта Arduino IDE вывод значений смещения по осям X и Y и состояние кнопки (ось Z).

Создадим более понятный пример использования джойстика для управления положением кронштейна Pan/Tilt Bracket с двумя сервоприводами, на котором можно разместить, например, камеру и менять положение камеры влево/вправо и вниз/вверх с помощью джойстика. Схема соединений для данного эксперимента показана на рис. 18.2.

Перемещением джойстика по оси X мы будем управлять поворотом нижнего сервопривода (влево/вправо), перемещением джойстика по оси Y будем управлять поворотом верхнего сервопривода (вверх/вниз). Среднее нейтральное положение джойстика по каждой оси (при аналоговом значении 512) соответствует углу поворота сервопривода на угол 90°.

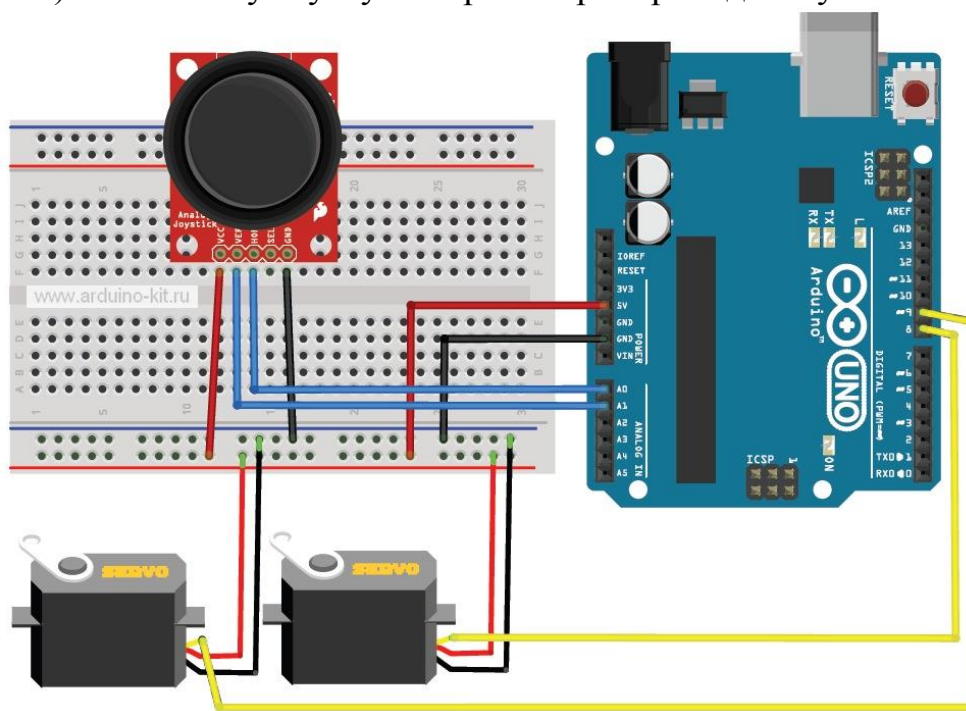


Рис. 18.2. Схема подключения платы джойстика и Pan/Title Bracket к Arduino

Порядок подключения:

1. Собираем Pan/Title Bracket и сервоприводы.
2. Подключаем джойстик и Pan/Title Bracket к плате Arduino по схеме на рис.
3. Составляем самостоятельно и загружаем в плату Arduino скетч
4. Управляем положением Pan/Title Bracket перемещением джойстика по осям X и Y.

ПРОЕКТ 19: ШАГОВЫЙ ДВИГАТЕЛЬ 4-ФАЗНЫЙ, С УПРАВЛЕНИЕМ НА ULN2003 (L293)

В этом эксперименте мы рассмотрим подключение к Arduino шагового двигателя.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- шаговый двигатель;
- микросхема L293;
- [провода папа-папа](#).
- [кнопка – 3 шт.](#);
- [резистор 1 кОм – 3 шт.](#);
- [внешний блок питания +5 В](#).

Шаговые двигатели представляют собой электромеханические устройства, задачей которых является преобразование электрических импульсов в перемещение вала двигателя на определенный угол. ШД нашли широкое применение в области, где требуется высокая точность перемещений или скорости. Наглядными примерами устройств с ШД могут служить принтеры, факсы и копировальные машины, а также более сложные устройства: станки с ЧПУ (числовым программным управлением), фрезерные, гравировальные машины и т. д.

Шаговый двигатель – синхронный бесщёточный электродвигатель с несколькими обмотками, в котором ток, подаваемый в одну из обмоток статора, вызывает фиксацию ротора. Последовательная активация обмоток двигателя вызывает дискретные угловые перемещения (шаги) ротора. Напрямую, к выводам Arduino подключать ШД нельзя, для подключения используют либо драйверы шаговых двигателей (например, A4988), либо драйверы двигателей постоянного тока (ULN2003, L293). В эксперименте будем использовать микросхему L293, которая содержит в себе четыре мощных усилителя (см. рис. 19.1). Если на вход усилителя подается 1, то выход сажается на 12 В, если на вход подается 0, то вывод сажается на землю. Таким образом, подавая комбинации 0 и 1 на разные входы, можно сажать выводы двигателя на шины разной полярности, вращая движок в разные стороны.

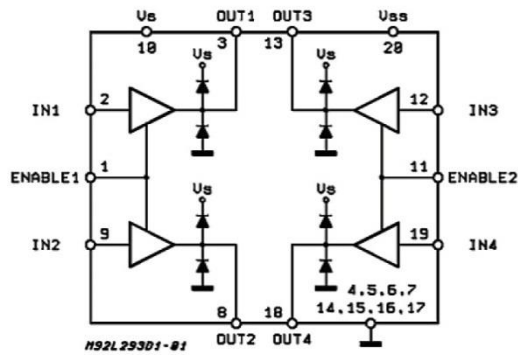


Рис. 19.1. Драйвер двигателей L293

Подключим к Arduino шаговый двигатель и с помощью кнопок будем задавать перемещение шагового двигателя в разные стороны. Схема соединений для данного эксперимента показана на рис. 19.2.

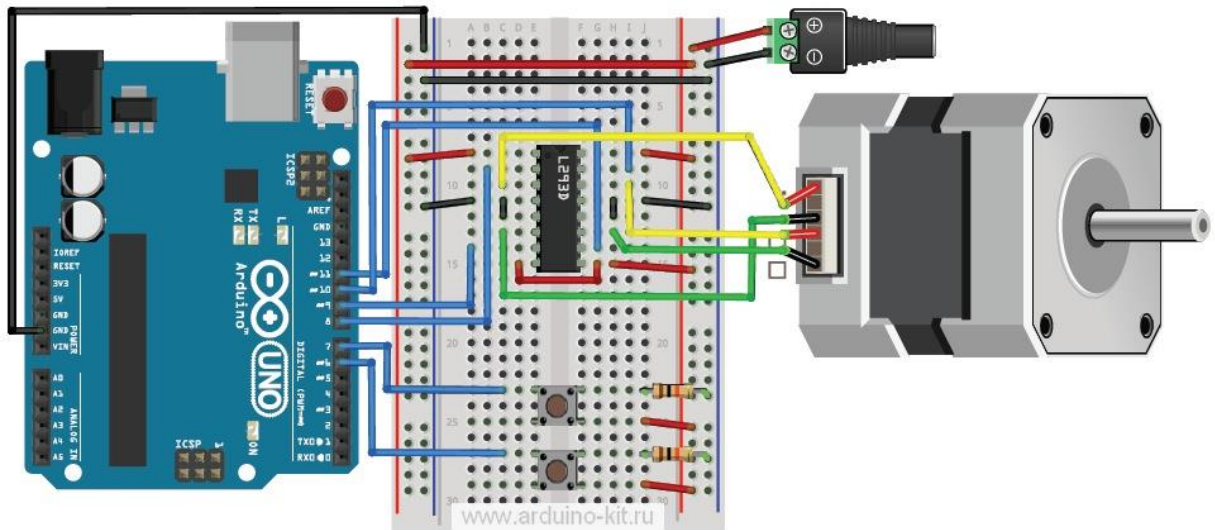


Рис. 19.2. Схема соединений шагового двигателя и Arduino

Напишем скетч управления поворотом шагового двигателя с помощью кнопок. При нажатии на первую кнопку шаговый двигатель перемещается на 200 шагов по часовой стрелке, при нажатии на другую кнопку шаговый двигатель перемещается на 200 шагов против часовой стрелки. При написании скетча будем использовать Arduino-библиотеку Stepper.

Порядок подключения:

1. Подключаем элементы к плате Arduino по схеме на рис. 19.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. При нажатии на одну из кнопок шаговый двигатель делает 200 шагов в одну сторону и останавливается, при нажатии на другую кнопку двигатель делает 200 шагов в обратную сторону.

ПРОЕКТ 20: ДАТЧИК ТЕМПЕРАТУРЫ DS18B20

В этом эксперименте мы рассмотрим популярный цифровой датчик температуры DS18B20, работающий по протоколу 1-Wire, и создадим проект вывода показаний датчика на экран ЖКИ WH1602.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [датчик DS18B20](#);
- [LCD-экран WH1602](#);
- [резистор 50 Ом](#);
- потенциометр 1 кОм;
- [провода папа-папа](#).
- [внешний блок питания +5 В](#).

DS18B20 – цифровой термометр с программируемым разрешением от 9 до 12 битов, которое может сохраняться в EEPROM-памяти прибора. DS18B20 обменивается данными по шине 1-Wire и при этом может быть как единственным устройством на линии, так и работать в группе. Все процессы на шине управляются центральным микропроцессором.

Диапазон измерений датчика: от $-55\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$ с точностью $0,5\text{ }^{\circ}\text{C}$ в диапазоне от $-10\text{ }^{\circ}\text{C}$ до $+85\text{ }^{\circ}\text{C}$. В дополнение DS18B20 может питаться напряжением линии данных (так называемое питание от паразитного источника) при отсутствии внешнего источника напряжения. Каждый датчик типа DS18B20 имеет уникальный 64-битный последовательный код, который позволяет общаться со множеством датчиков DS18B20, установленных на одной шине. Первые 8 битов – код серии (для DS18B20 – 28h), затем 48 битов уникального номера, и в конце 8 битов CRC-кода. Такой принцип позволяет использовать один микропроцессор, чтобы контролировать множество датчиков DS18B20, распределенных по большому участку.

В нашем эксперименте мы будем считывать данные с датчика температуры DS18B20 и выводить на экран ЖКИ WH1602, который мы рассматривали в эксперименте 16. Схема подключения датчика температуры DS18B20 и WH1602 к плате Arduino показана на рис. 20.1.

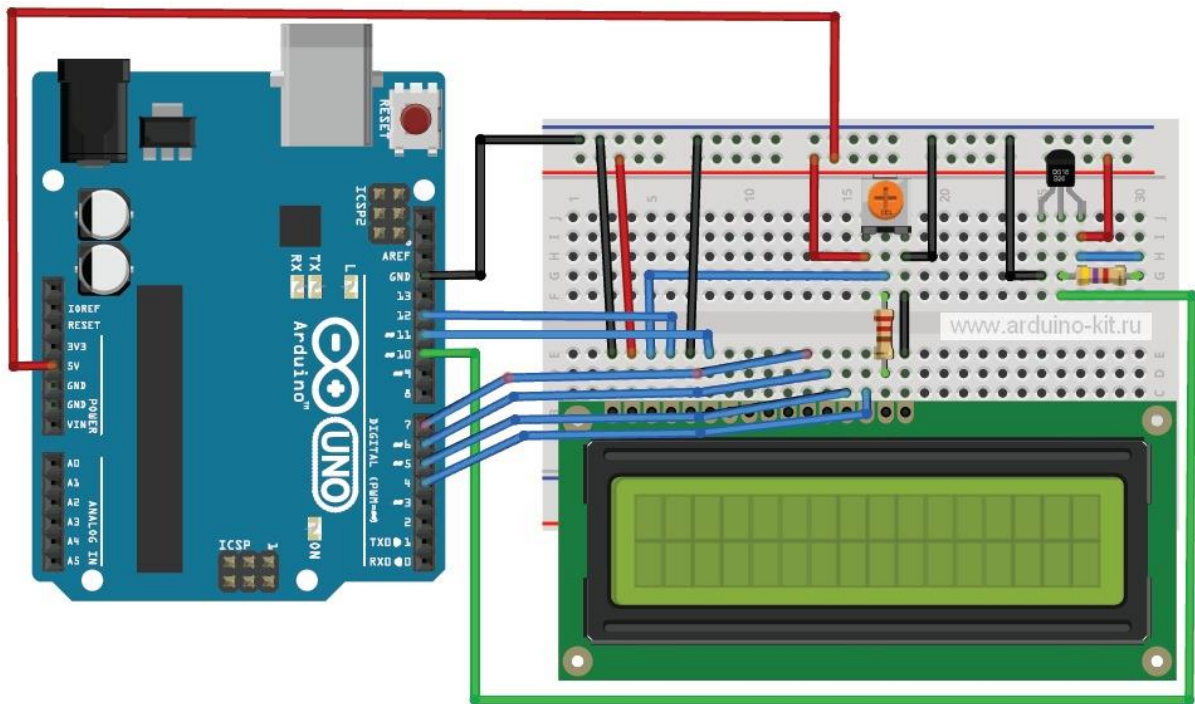


Рис. 20.1. Схема подключения датчика температуры DS18B20 и WH1602 к Arduino

Приступим к написанию скетча. Для работы с устройствами 1-Wire в Arduino есть стандартная библиотека OneWire.

Последовательность данных для чтения данных с устройств 1-Wire следующая:

1. Произвести RESET и поиск устройств на линии 1-Wire.
2. Выдать команду 0x44, чтобы запустить конвертацию температуры датчиком.
3. Подождать не менее 750 мс.
4. Выдать команду 0xBE, чтобы считать ОЗУ датчика (данные о температуре будут в первых двух байтах).

Порядок подключения:

1. Подключаем датчик DS18B20 и WH1602 по схеме на рис. 20.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Смотрим на экране дисплея показания датчика температуры.

ПРОЕКТ 21: ДАТЧИК ВЛАЖНОСТИ И ТЕМПЕРАТУРЫ DHT11

В этом эксперименте мы рассмотрим датчик для измерения относительной влажности воздуха и температуры DHT11 и создадим проект вывода показаний датчика на экран ЖКИ WH1602.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [датчик DHT11](#);
- [LCD-экран WH1602](#);
- [резистор 50 Ом](#);
- [потенциометр 1 кОм](#);
- [провода папа-папа](#).
- [внешний блок питания +5 В](#).

Датчик DHT11 (см. рис. 21.1) не отличается высоким быстродействием и точностью, однако может найти свое применение в радиолюбительских проектах из-за своей невысокой стоимости. Датчик DHT11 состоит из емкостного датчика влажности и термистора. Кроме того, датчик содержит в себе простенький АЦП для преобразования аналоговых значений влажности и температуры.

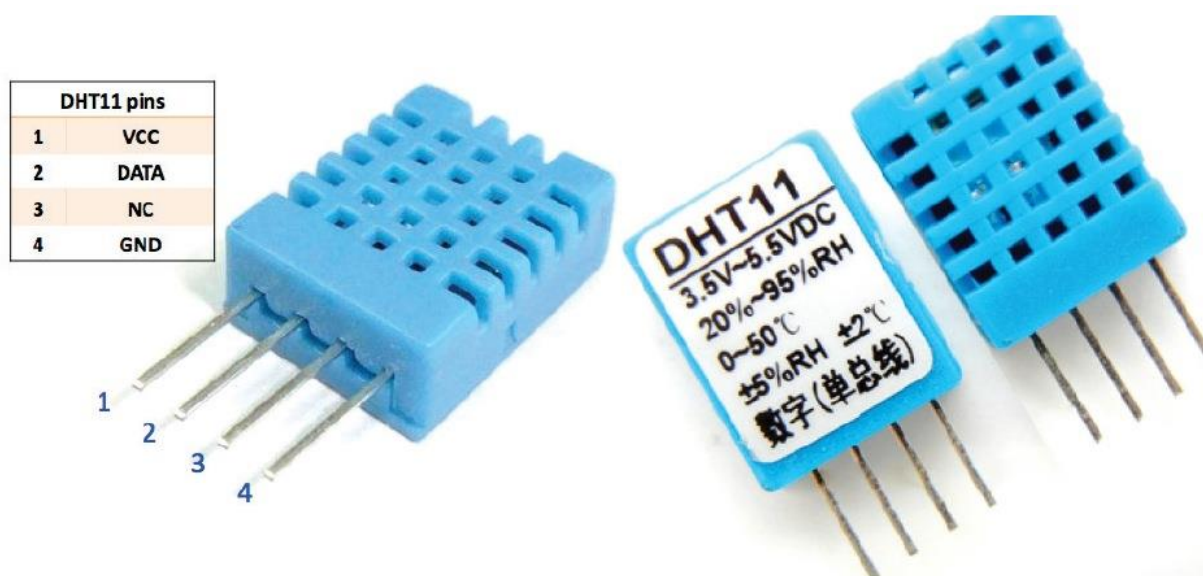


Рис. 21.1. Датчик DHT11

Датчик имеет 4 вывода в одну линию с шагом 2,54 мм:

- 1 – VCC (питание 3–5 В);
- 2 – DATA (вывод данных);
- 3 – не используется;
- 4 – GND (земля).

Протокол обмена – однопроводный, по структуре весьма похож на DS18B20, но с важными оговорками:

- DHT11 не умеет работать в «паразитном» режиме (питание по линии

данных);

- каждый DS18B20 имеет персональный идентификатор, что дает возможность подключения нескольких таких датчиков к одному пину Arduino. Однако у DHT11 подобной возможности нет – один датчик будет использовать строго один цифровой пин.

В нашем эксперименте мы будем считывать данные с датчика DHT11 и выводить на экран ЖКИ WH1602, который мы рассматривали в эксперименте 16. Рекомендуемая схема подключения к Arduino содержит обязательный для однопроводных линий резистор-подтяжку к VCC, в качестве опции рекомендуется конденсатор (фильтр по питанию между VCC и GND). У нас в наличии DHT11 в виде модуля, его можно подключать к Arduino напрямую – резистор и конденсатор там уже есть. Схема подключения датчика DHT11 и WH1602 к плате Arduino показана на рис. 21.2. Для питания ЖКИ нужен отдельный блок питания +5 В.

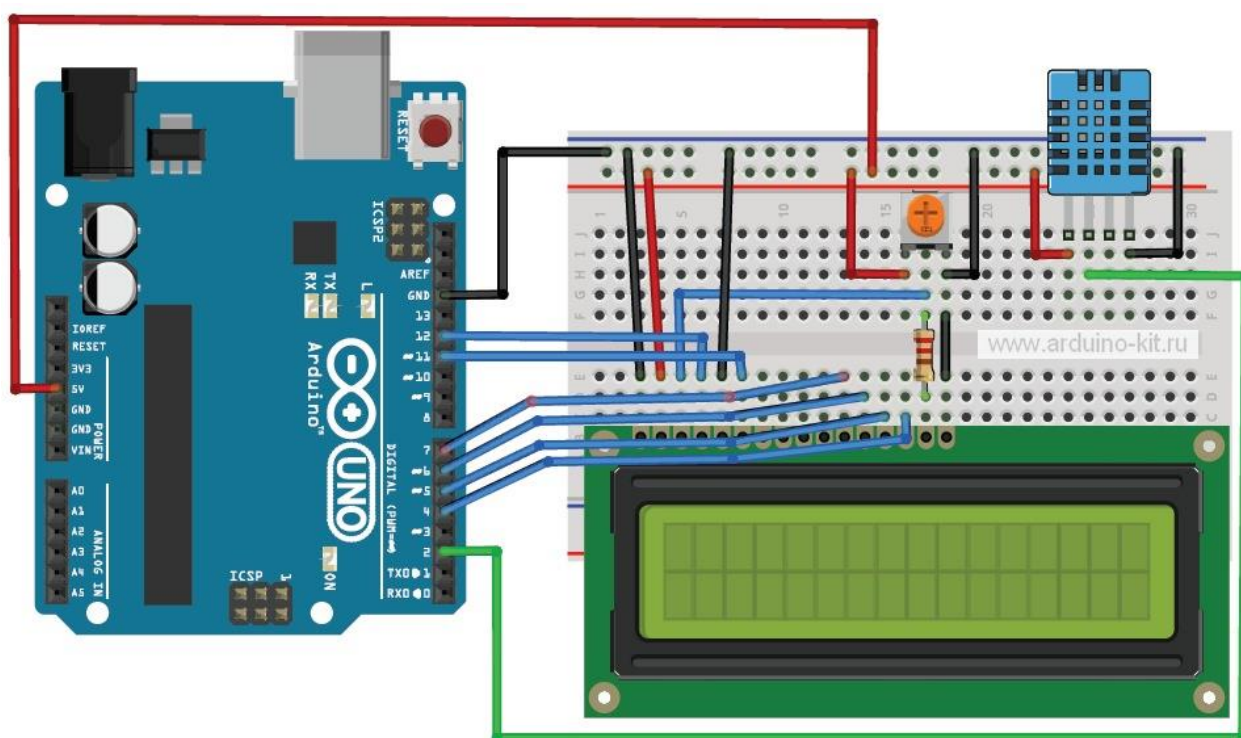


Рис. 21.2. Схема подключения датчика DHT11 и WH1602 к Arduino

Приступим к написанию скетча. Для работы с датчиками DHT11 (DHT21, DHT22) в Arduino есть библиотека OneWire.

Порядок подключения:

1. Подключаем датчик DHT11 и WH1602 по схеме на рис. 21.2.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Смотрим на экране дисплея показания относительной влажности и температуры.

ПРОЕКТ 22: ДАТЧИКИ ГАЗОВ. ПРИНЦИП РАБОТЫ, ПРИМЕР РАБОТЫ

В этом эксперименте мы рассмотрим ультразвуковой датчик для измерения расстояния и создадим проект вывода показаний датчика на экран ЖКИ WH1602.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [датчик газа MQ-4](#);
- газовая зажигалка;
- [светодиод](#);
- [резистор 220 Ом](#);
- [провода папа-папа](#).

Серия MQ-сенсоров для Ардуино, построены на базе мини-нагревателя внутри и используют электрохимический сенсор. Они чувствительны для определенных диапазонов газов и используются в помещениях при комнатной температуре. Вот некоторые из них:

- MQ-3 – сенсор паров алкоголя;
- [MQ-4](#) – сенсор для обнаружения метана, пропана;
- [MQ-5](#) и MQ-6 – предназначены для обнаружения пропана, бутана;
- [MQ-7](#) – чувствителен к угарному газу;
- MQ-8 – специализируется по водороду H₂.

Датчики содержат аналоговый и цифровой выходы для подключения к Arduino. Советуют подключать оба выхода для более точного результата, что вовсе не обязательно.

В нашем эксперименте подключим датчик MQ4 к плате Arduino и посмотрим, как он реагирует на наличие газов. Схема подключения датчика показана на рис. 22.1.

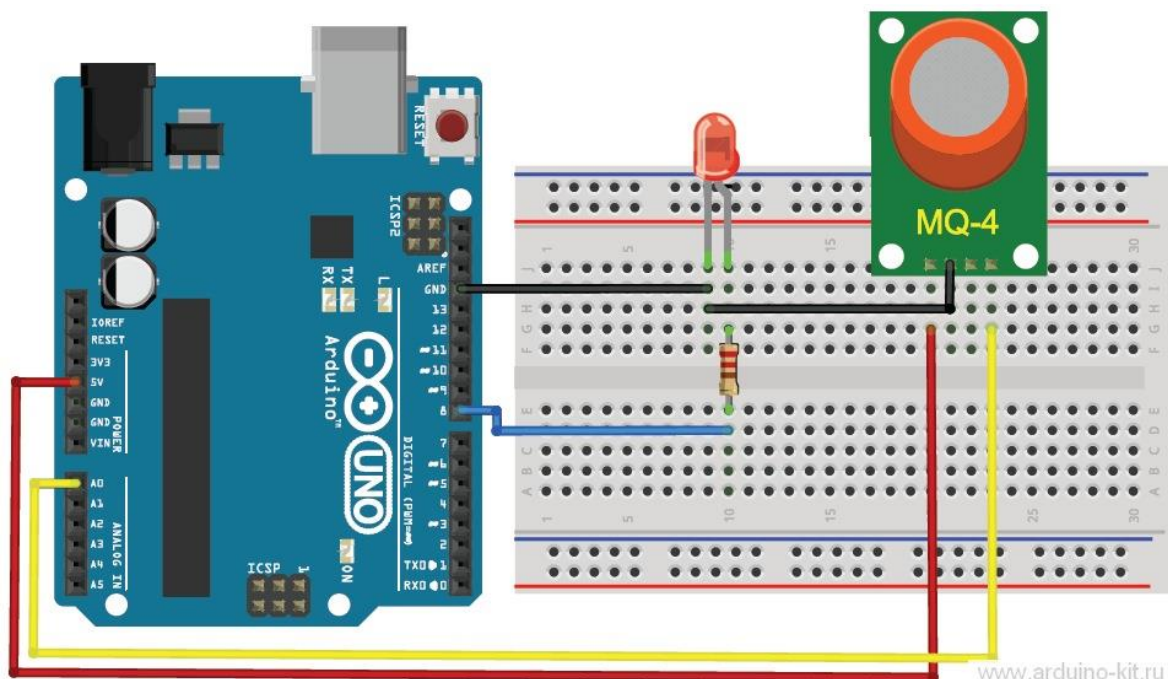


Рис. 22.1. Схема подключения датчика MQ4

Напишем скетч, считывающий показания с датчика MQ4 и выводящий показания в монитор последовательного порта. Если аналоговое значение с датчика превысит 750 (опасный уровень), будем зажигать светодиод, подключенный к цифровому выводу 8.

Порядок подключения:

1. Подключаем датчик MQ4 к плате Arduino по схеме на рис. 22.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Открываем монитор последовательного порта Arduino IDE.
4. После подачи питания датчику необходимо время, чтобы выйти на рабочий режим, примерно 10–15 секунд. Это время нужно, чтобы нагреватель внутри датчика поднял температуру до необходимого значения.
5. Подносим газовую зажигалку к датчику и открываем газ, наблюдаем изменение показаний от датчика MQ4 в мониторе последовательного порта Arduino IDE.

ПРОЕКТ 23: УЛЬТРАЗВУКОВОЙ ДАТЧИК РАССТОЯНИЯ HC-SR04. ПРИНЦИП РАБОТЫ, ПОДКЛЮЧЕНИЕ, ПРИМЕР

В этом эксперименте мы рассмотрим ультразвуковой датчик для измерения расстояния и создадим проект вывода показаний датчика на экран ЖКИ WH1602.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [ультразвуковой датчик расстояния HC-SR04](#);
- [пьезоизлучатель](#);
- [резистор 100 Ом](#);
- [сервопривод](#);
- [провода папа-папа](#).
- [внешний блок питания +5 В](#).

Ультразвуковой дальномер HC-SR04 (рис. 23.1) – это помещенные на одну плату приемник и передатчик ультразвукового сигнала. Излучатель генерирует сигнал, который, отразившись от препятствия, попадает на приемник. Измерив время, за которое сигнал проходит до объекта и обратно, можно оценить расстояние. Кроме самих приемника и передатчика, на плате находится еще и необходимая обвязка, чтобы сделать работу с этим датчиком простой и удобной.

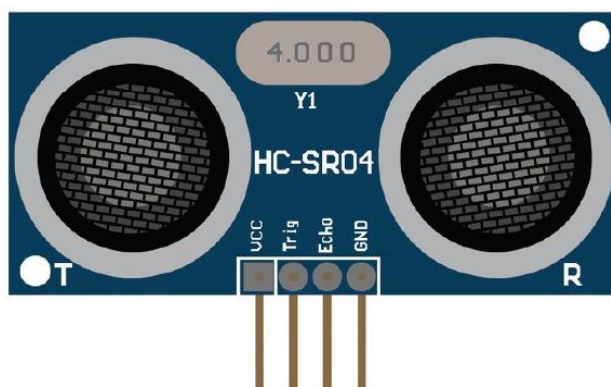


Рис. 23.1. Ультразвуковой дальномер HC-SR04

Характеристики ультразвукового дальномера HC-SR04:

- измеряемый диапазон – от 2 до 500 см;
- точность – 0,3 см;
- угол обзора – $< 15^\circ$;
- напряжение питания – 5 В.

Датчик имеет 4 вывода стандарта 2,54 мм:

- VCC – питание +5 В;

- Trig (T) – вывод входного сигнала;
- Echo (R) – вывод выходного сигнала;
- GND – земля.

Последовательность действий для получения данных такова:

- подаем импульс продолжительностью 10 мкс на вывод Trig;
- внутри датчика входной импульс преобразуется в 8 импульсов частотой 40 кГц и посылаются вперед через излучатель T;
- дойдя до препятствия, посланные импульсы отражаются и принимаются приемником R, в результате получаем выходной сигнал на выводе Echo;
- непосредственно на стороне контроллера переводим полученный сигнал в расстояние по формуле:

— ширина импульса (мкс) / 58 = дистанция (см);

— ширина импульса (мкс) / 148 = дистанция (дюйм).

В нашем эксперименте мы создадим звуковую сигнализацию, которая будет включаться при приближении к плате Arduino на расстояние меньше 1 м.

Датчик размещен на кронштейне вращающейся сервы и контролирует пространство с углом обзора 180°. Если датчик обнаруживает объект в радиусе 1 м, подается звуковой сигнал на пьезоизлучатель, вращение сервы прекращается. Схема соединения элементов представлена на рис. 23.2.

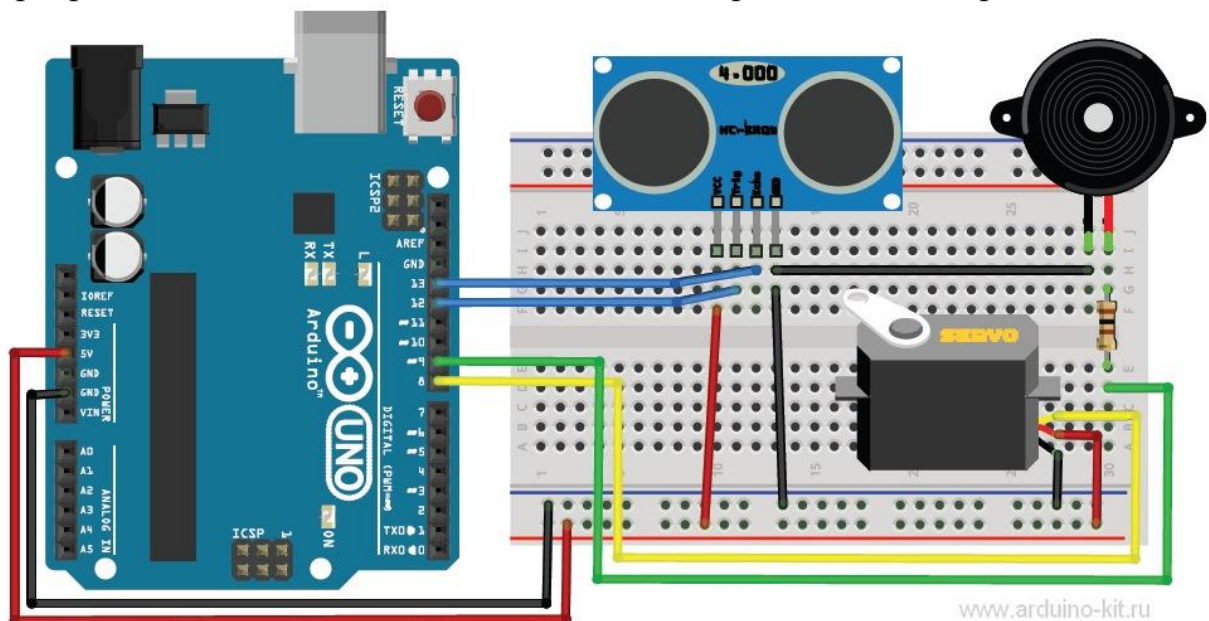


Рис. 23.2. Схема соединения элементов для звуковой сигнализации

При написании скетча будем использовать библиотеку Servo для работы с сервоприводом и библиотеку Ultrasonic.

Для работы Arduino с датчиком HC-SR04 имеется готовая библиотека – Ultrasonic.

Конструктор Ultrasonic принимает два параметра: номера пинов, к которым подключены выводы Trig и Echo, соответственно:

Ultrasonic ultrasonic(12,13);

Порядок подключения:

1. Закрепляем датчик расстояния HC-SR04 на сервоприводе.
2. Подключаем датчик HC-SR04, пьезозуммер и сервопривод к плате Arduino по схеме на рис. 23.2.
3. Составляем самостоятельно и загружаем в плату Arduino скетч
4. Наблюдаем за циклическим перемещением сервопривода, при попадании объекта в поле зрения датчика HC-SR04 пьезозуммер издает сигнал, сервопривод останавливается, при исчезновении объекта из поля зрения датчика сервопривод возобновляет движение.

ПРОЕКТ 24: 3-ОСЕВОЙ ГИРОСКОП + АКСЕЛЕРОМЕТР НА ПРИМЕРЕ GY-521

В этом эксперименте мы познакомимся с акселерометром и гироскопом и будем с помощью Arduino получать показания с этих датчиков.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль GY-521](#);
- [провода папа-папа](#).

Модуль GY-521 на микросхеме MPU6050 содержит гироскоп, акселерометр и температурный сенсор. На плате модуля GY-521 расположена необходимая обвязка MPU6050, в том числе подтягивающие резисторы, стабилизатор напряжения на 3,3 В с малым падением напряжения с фильтрующими конденсаторами. Обмен с микроконтроллером осуществляется по шине I2C.

Гироскоп представляет собой устройство, реагирующее на изменение углов ориентации контролируемого тела. Акселерометр – это устройство, которое измеряет проекцию кажущегося ускорения, то есть разницы между истинным ускорением объекта и гравитационным ускорением.

Схема соединений платы GY-521 к Arduino показана на рис. 24.1.

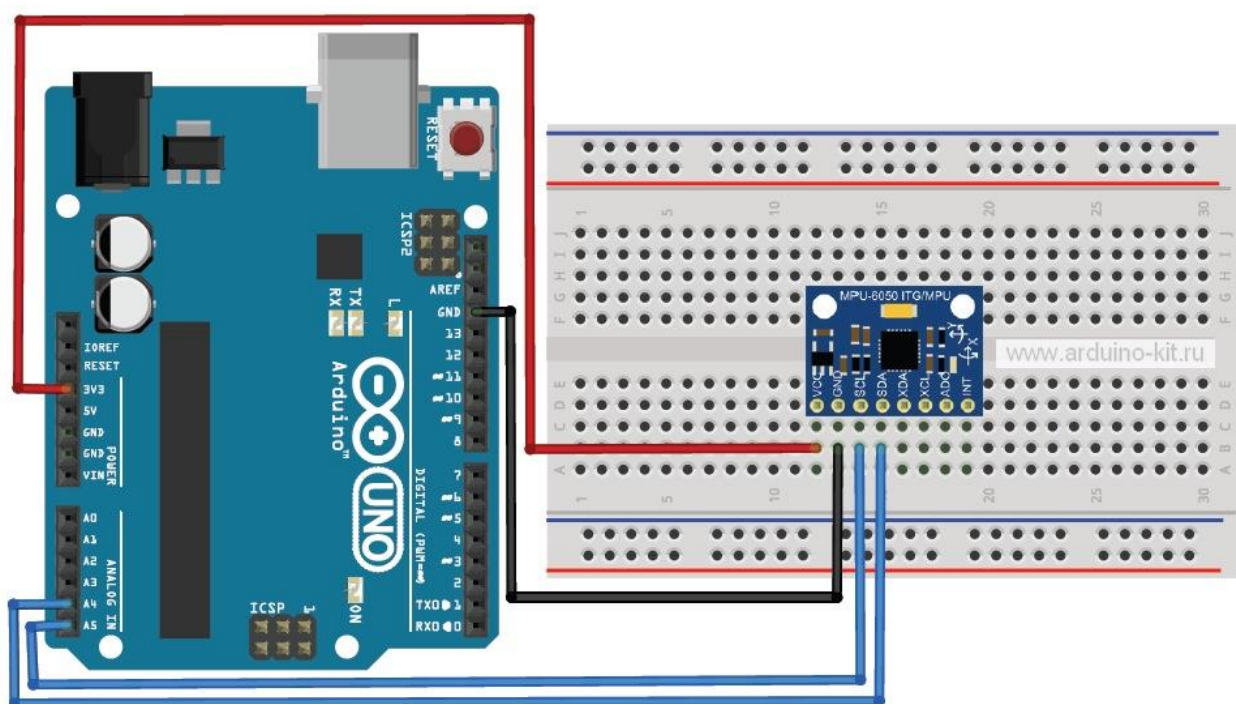


Рис. 24.1. Схема соединения GY-521 к Arduino

Порядок подключения:

1. Подключаем плату GY521 к плате Arduino по схеме на рис. 24.1.

2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Открываем монитор последовательного порта Arduino IDE и смотрим вывод данных гироскопа и акселерометра (см. рис. 24.2).
4. При поворотах датчика данные изменяются.

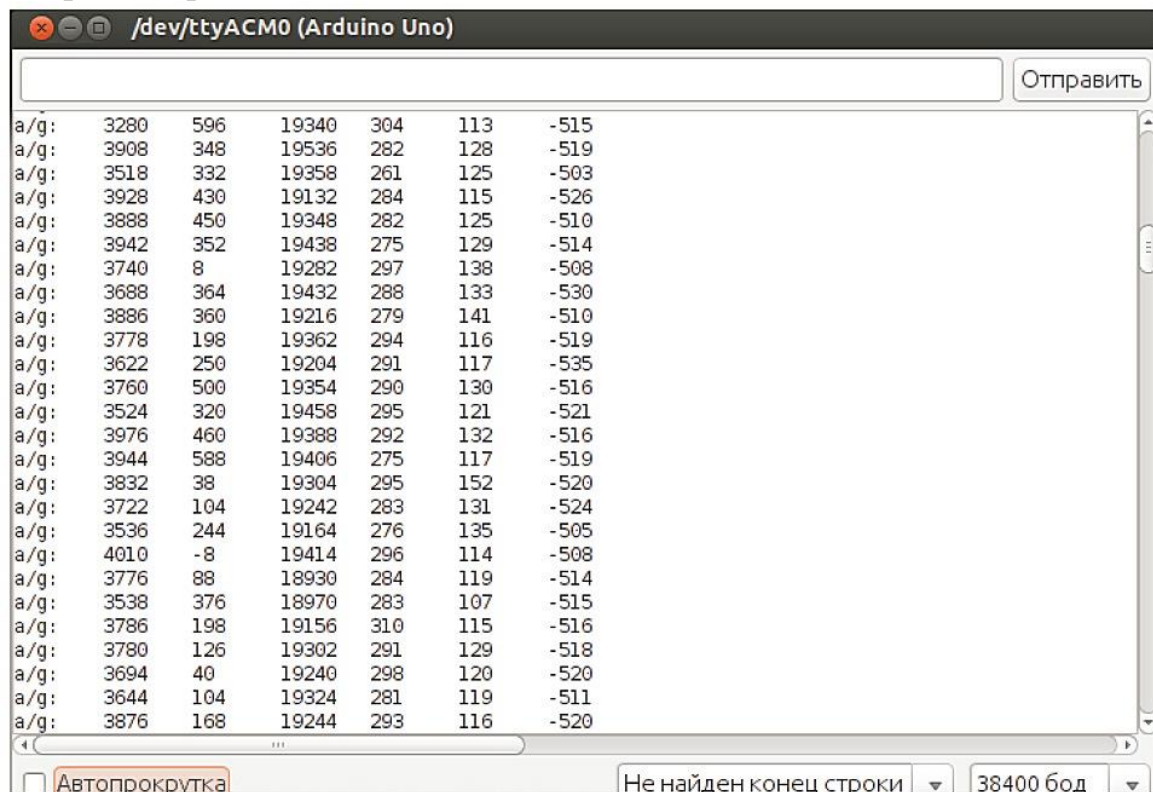


Рис. 24.2. Вывод данных гироскопа и акселерометра в монитор Arduino IDE

Область применения таких датчиков достаточно широка. Данный модуль часто применяют для стабилизации полета квадрокоптера по причине совместного использования гироскопа и акселерометра. Кроме этого, модуль можно использовать для координации различных устройств – от просто детектора движения до системы ориентации различных роботов или управления движениями какими-либо устройствами. Область подобных сенсорных устройств достаточно новая и интересная для изучения и применения в любительской технике.

ПРОЕКТ 25: ИК-ФОТОПРИЕМНИК И ИК-ПУЛЬТ. ОБРАБАТЫВАЕМ КОМАНДЫ ОТ ПУЛЬТА

В этом эксперименте мы организуем беспроводную ИК-связь, которая нам позволит отправлять на плату Arduino команды с помощью любого ИК-пульта.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- микросхема TSOP 31236;
- конденсатор 10 мкФ 25 В;
- [резистор 100 Ом](#);
- [светодиод – 8 шт.](#);
- [резистор 220 Ом – 8 шт.](#);
- [любой ИК-пульт](#);
- [провода папа-папа](#).

Устройства инфракрасного (ИК) диапазона волн часто применяются в робототехнике. Наличие дешевых приемников диапазона 36–40 кГц, а также наличие большого количества пультов от бытовых приборов позволяет организовать простое и понятное беспроводное управление. В качестве приемника будем использовать микросхему TSOP31236. В одном корпусе она объединяет фотодиод, предусилитель и формирователь. На выходе формируется обычный ТТЛ-сигнал без заполнения, пригодный для дальнейшей обработки микроконтроллером. Несущая частота 36 кГц. В качестве передатчика – любой пульт для управления бытовой техникой. Для обеспечения надежного приема и гарантированной защиты от помех при инфракрасной передаче используются модуляция сигнала и кодирование. К сожалению, нет единого и универсального протокола для ИК-пультов дистанционного управления, хотя среди всего многообразия есть наиболее распространенные. Наиболее распространенными протоколами для ИК-пультов дистанционного управления являются следующие:

- RC5;
- NEC;
- JVC;
- Sony.

Можно узнать протокол вашего пульта и написать скетч для получения кодов, отправляемых с пульта. К счастью, уже написана универсальная библиотека для приема и обработки кодов с любого пульта – Irremote. Ее мы и будем использовать при написании скетча. В нашем эксперименте мы будем с ИК-пульта зажигать светодиоды, подключенные к плате Arduino. Схема соединений показана на рис. 25.1.

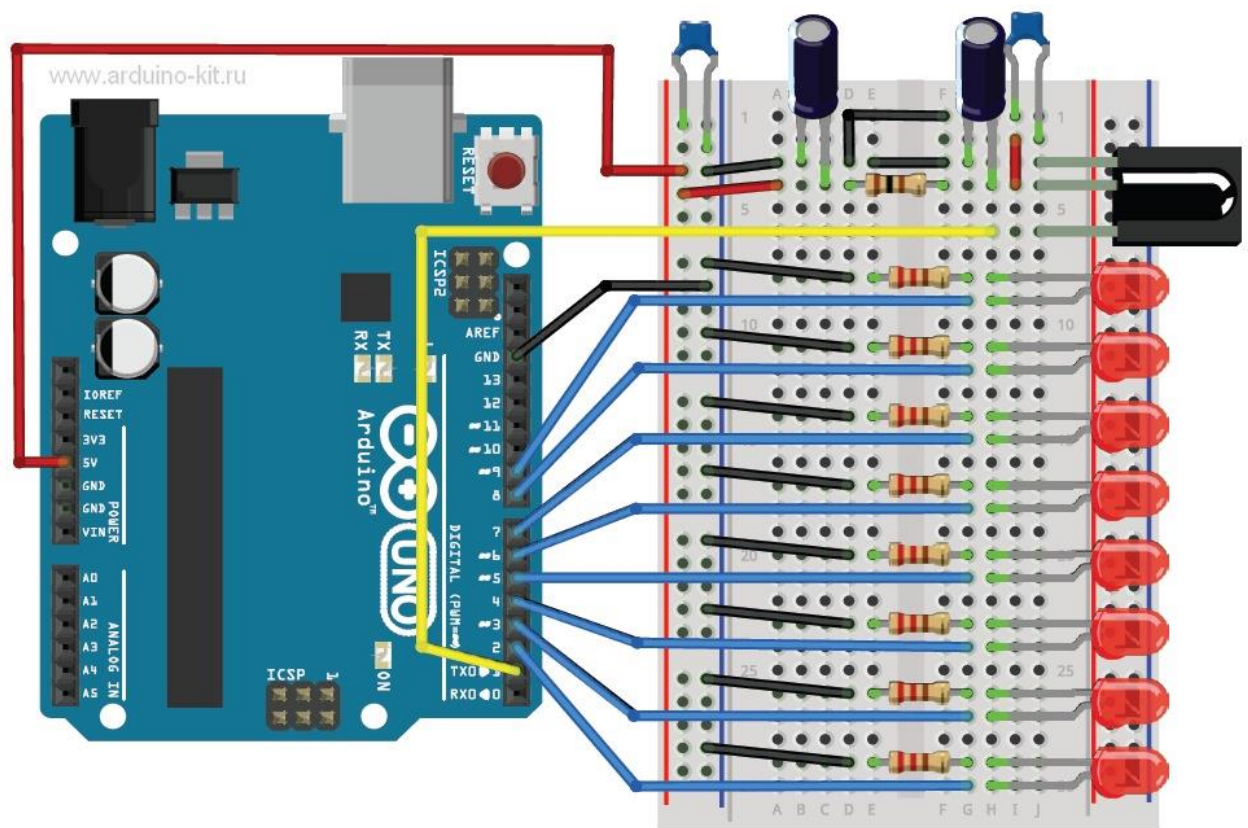


Рис. 25.1. Схема соединений ИК-управления

Выход ИК-приемника подсоединен к выводу 1 платы Arduino. Без фильтра питания будет работать нестабильно, с пропуском посылок, поэтому ставим RC-фильтр.

Сначала загрузим скетч, определяющий коды, приходящие с приемника, и выводящий их в монитор последовательного порта.

Порядок подключения:

1. Подключаем ИК-приемник и светодиоды к плате Arduino по схеме на рис. 25.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Открываем монитор последовательного порта Arduino IDE и смотрим коды, приходящие при нажатии кнопок на ИК-пульте. Запоминаем коды, приходящие при нажатии кнопок 2–9 на ИК-пульте. Пишем скетч, переключающий состояние светодиодов на контактах D2–D9, при получении определенного кода. Значение констант K2–K9 (кодов для клавиш 2–9) у вас будет другим.

Загружаем скетч 25.2 на плату Arduino и нажатием на пульте кнопок 2–9 переключаем состояние светодиодов, подключенных к выводам 2–9 платы Arduino (см. рис. 25.1).

ПРОЕКТ 26: ЧАСЫ РЕАЛЬНОГО ВРЕМЕНИ. ПРИНЦИП РАБОТЫ, ПОДКЛЮЧЕНИЕ, ПРИМЕРЫ

В этом эксперименте мы рассмотрим модуль часов реального времени на микросхеме DS1307.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль часов реального времени DS1307](#);
- [батарейка CR2032 3 В](#);
- [ЖКИ WH1602](#);
- потенциометр 10 кОм;
- [резистор 51 Ом](#);
- [провода папа-папа](#).
- [внешний блок питания +5 В](#).

Микросхема Dallas DS1307 представляет собой часы реального времени с календарем и дополнительной памятью NW SRAM (56 байт). Микросхема подключается к микроконтроллеру при помощи шины I2C. Количество дней в месяце рассчитывается с учетом високосных лет до 2100 г. В микросхеме DS1307 имеется встроенная схема, определяющая аварийное отключение питания и автоматически подключающая резервную батарейку. При этом отсчет времени продолжается, и после восстановления питания часы показывают правильное время. Также в этой микросхеме имеется программируемый генератор прямоугольных импульсов, позволяющий вырабатывать одну из четырех частот (1 Гц, 4096 Гц, 8192 Гц или 32768 Гц).

Часы подключаются по протоколу I2C всего двумя проводами. SCL и SDA – это выводы интерфейса I2C. Необходимо дополнительно подтянуть выводы, к которым подключаются часы к шине питания, с помощью резисторов 2 кОм. SCL и SDA на разных платах расположены на разных выводах:

- Uno, Nano – A4 (SDA), A5 (SCL);
- Mega2560 – 20 (SDA), 21 (SCL);
- Leonardo – 2 (SDA), 3 (SCL).

Вывод SDA часов подключается к выводу SDA контроллера. SCL часов – соответственно, к SCL контроллера. В нашем эксперименте мы будем выводить дату и время, получаемые с микросхемы DS1307, на экран LCD индикатора WH1602. Схема подключения показана на рис. 26.1.

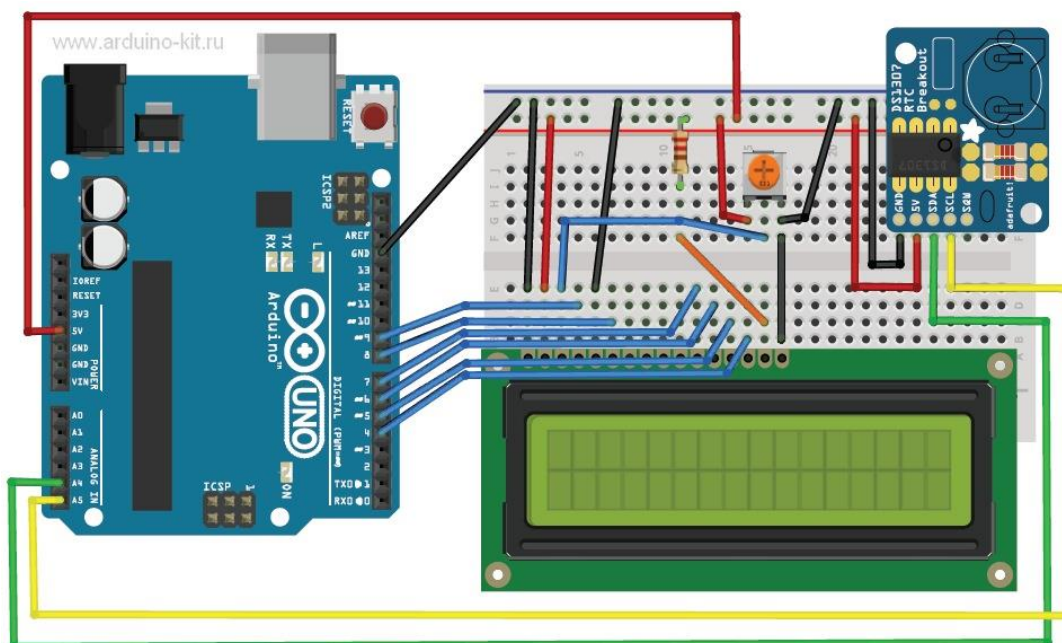


Рис. 26.1. Схема подключения модуля DS1307 и WH1602 к Arduino

При написании скетча используем библиотеку Time, которая является «оберткой» для библиотеки DS1307, и библиотеку Wire для работы с I2C-устройствами. Для работы с ЖКИ используем библиотеку LiquidCrystal.

Порядок подключения:

1. Подключаем модуль DS1307 и ЖКИ к плате Arduino по схеме на рис. 26.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Смотрим на экране ЖКИ меняющееся ежесекундно время и дату. Однако на экране дисплея мы видим неверное время и неверную дату. Дело в том, что при отсутствии питания значение времени в микросхеме DS1307 сбрасывается на 00:00:00 01/01/2000. Чтобы при отключении питания время не сбрасывалось, предусмотрено аварийное питание от батарейки 3 В. Для установки времени в библиотеке есть функция RTC.write(tmElements_ttm). Добавим в скетч возможность установки данных RTC по последовательному порту отправкой строки вида «dd/mm/YYYY hh:mm:ss». Теперь установим время из монитора последовательного порта отправкой строки «dd/mm/YYYY hh:mm:ss» и увидим на экране дисплея отображение верной даты и времени.

ПРОЕКТ 27: SD-КАРТА. ЧТЕНИЕ И ЗАПИСЬ ДАННЫХ

В этом эксперименте мы покажем, как к плате Arduino подключить SD-карту.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль SD-card](#);
- [модуль часов реального времени DS1307](#) с батареей;
- датчик температуры LM335;
- [резистор 2,2 кОм](#);
- [провода папа-папа](#).

Если вашим Arduino-проектам не хватает памяти, а объем энергонезависимой памяти EEPROM в платах Arduino совсем небольшой, можно использовать внешние носители. Один из самых простых по подключению к платам Arduino – это SD-карта. Можно подсоединиться к SD-карте напрямую, а можно использовать модули.

Подсоединим модуль SD-card к плате Arduino и напомним пример сохранения на SD-карте данных аналогового датчика температуры LM335. Для анализа данных температуры потребуется знать и время измерения данных, поэтому будем использовать и модуль часов реального времени RTC. Соберем схему, показанную на рис. 27.1.

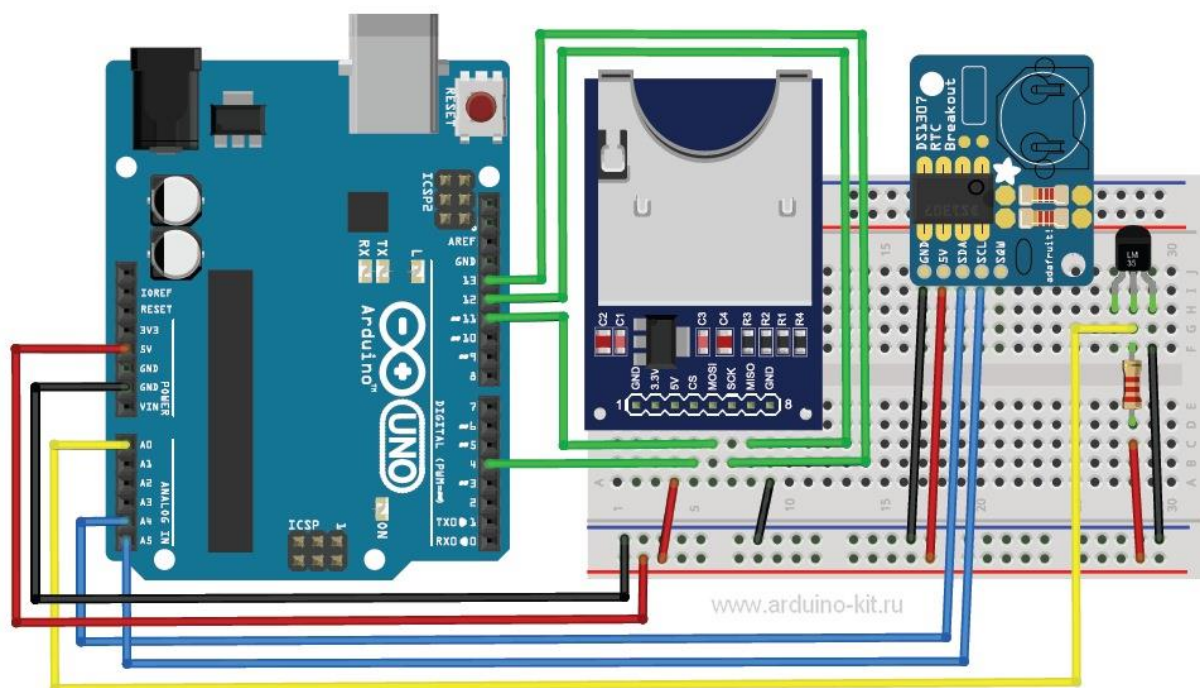


Рис. 27.1. Схема подключения модулей SD-card и DS1307 к Arduino

При написании скетча используем библиотеку SD для работы с SD-картой, а также библиотеки Time и DS1307 для работы с модулем RTC.

Каждые 5 минут мы считываем данные с датчика LM335, подключенного к аналоговому входу A0, и заносим время измерения и данные температуры в файл вида d-m-Y. В начале суток создаем новый файл на новый день.

Порядок подключения:

1. Подключаем модули SD card, DS1307 и датчик температуры LM335 к плате Arduino по схеме на рис. 27.1.

2. Составляем самостоятельно и загружаем в плату Arduino скетч
Ждем продолжительное время, затем вынимаем карту, вставляем в компьютер, открываем файл, соответствующий текущему дню, и смотрим его содержимое – по строкам времени замера и показания температуры.

ПРОЕКТ 28: СЧИТЫВАТЕЛЬ RFID НА ПРИМЕРЕ RC522. ПРИНЦИП РАБОТЫ, ПОДКЛЮЧЕНИЕ

В этом эксперименте мы покажем, как плата Arduino получает доступ к данным RFID-карт и брелоков Mifare с помощью RFID-считывателя RC522C.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [RFID-считыватель RC522](#);
- [брелок](#);
- [карта](#);
- [провода папа-папа](#).

Радиочастотная идентификация (RFID) – это технология автоматической бесконтактной идентификации объектов при помощи радиочастотного канала связи. Базовая система RFID состоит из:

- радиочастотной метки;
- считывателя информации (ридера);
- компьютера для обработки информации.

Идентификация объектов производится по уникальному цифровому коду, который считывается из памяти электронной метки, прикрепляемой к объекту идентификации. Считыватель содержит в своем составе передатчик и антенну, посредством которых излучается электромагнитное поле определенной частоты. Попавшие в зону действия считывающего поля радиочастотные метки «отвечают» собственным сигналом, содержащим информацию (идентификационный номер товара, пользовательские данные и т. д.). Сигнал улавливается антенной считывателя, информация расшифровывается и передается в компьютер для обработки. Подавляющее большинство современных систем контроля доступа (СКД) использует в качестве средств доступа идентификаторы, работающие на частоте 125 кГц. Это проксимити-карты доступа (только чтение), самыми распространенными являются карты EM-Marin, а также HID, Indala. Карты этого стандарта являются удобным средством открывания дверей и турникетов. Но не более. Эти карты не обладают никакой защищенностью, легко копируются и подделываются и, соответственно, ничего не дают для защиты объекта от несанкционированного проникновения.

Настоящую защиту от копирования и подделки обеспечивают такие идентификаторы, в чипах которых реализована криптографическая защита. Это бесконтактные смарт-карты, работающие на частоте 13,56 МГц, наиболее распространенными из них являются карты Mifare®. В картах этих стандартов криптозащита организована на высоком уровне, и подделка таких карт практически невозможна.

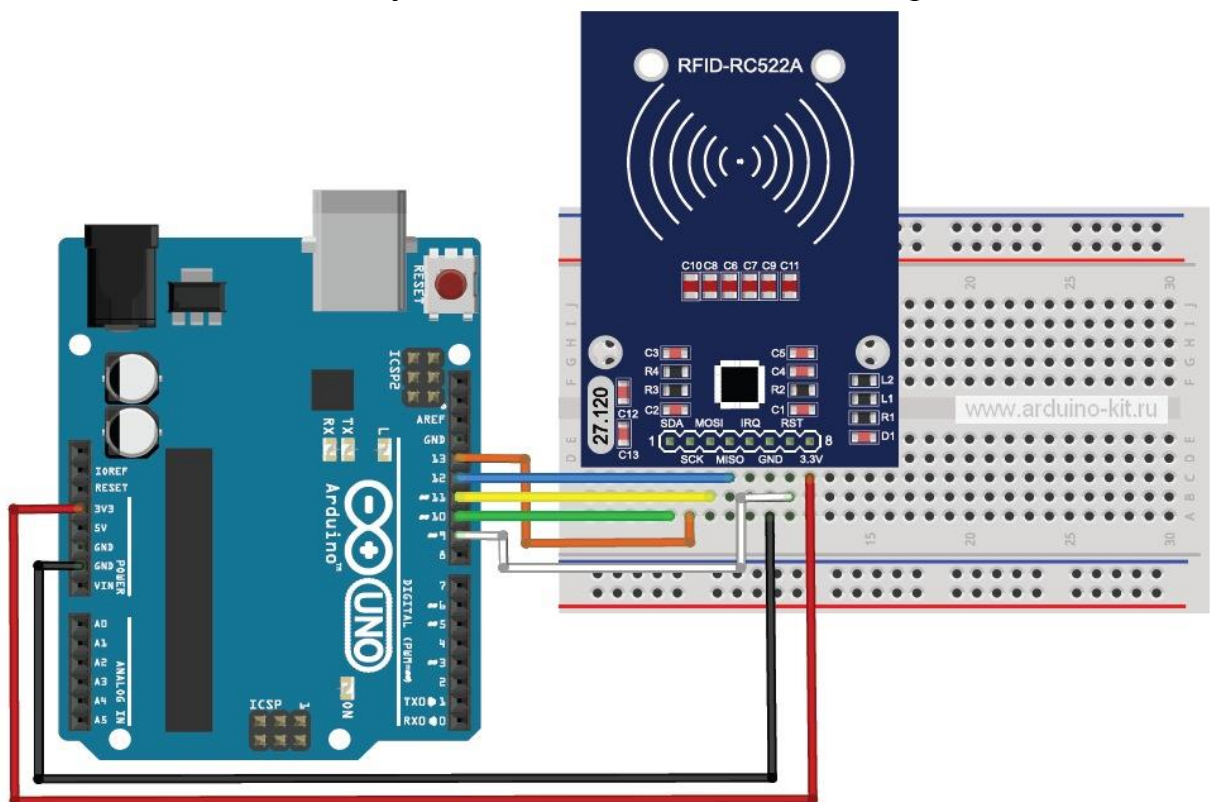
Модуль RC522 – RFID-модуль 13,56 МГц с SPI-интерфейсом. В комплекте к

модулю идут 2 RFID-метки – в виде карты и брелока.

Основные характеристики:

- основан на микросхеме MFRC522;
- напряжение питания: 3,3 В;
- потребляемый ток: 13–26 мА;
- рабочая частота: 13,56 МГц;
- дальность считывания: 0~60 мм;
- интерфейс: SPI, максимальная скорость передачи 10 МБит/с;
- размер: 40×60 мм;
- чтение и запись RFID-меток.

Схема подключения модуля к плате Arduino показана на рис. 28.1.



Открываем монитор последовательного порта.

3. Подносим метку (карту или брелок) к считывателю и видим вывод в последовательный порт данных метки UID и тип (рис. 28.2).



Рис. 28.2. Вывод в последовательный порт информации о метках

Метки Mirafe позволяют записывать на них информацию. В следующем скетче мы организуем на карте счетчик, который будет инкрементироваться при поднесении карты к считывателю. В последовательный порт будем выводить показания счетчика.

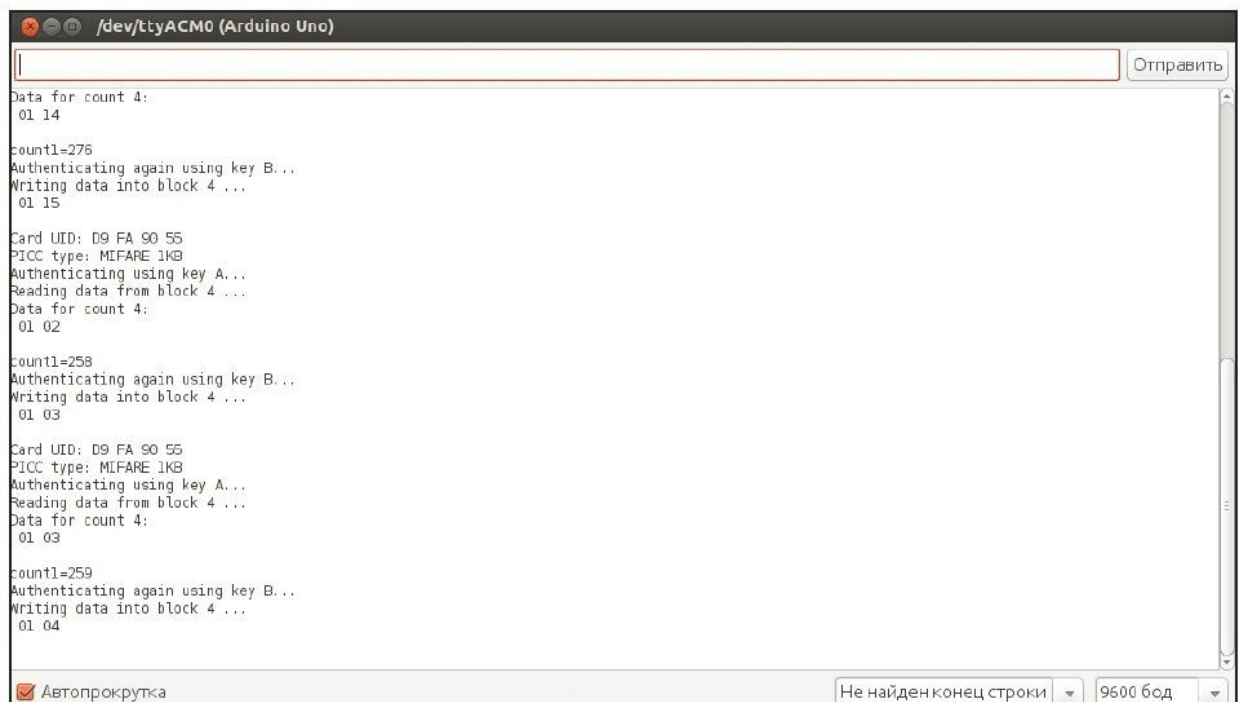


Рис. 28.3. Вывод в последовательный порт информации о счетчике на метках

ПРОЕКТ 29: РАБОТА С ИНТЕРНЕТОМ НА ПРИМЕРЕ ARDUINO ETHERNET SHIELD W5100

В этом эксперименте мы покажем, как нашей плате Arduino получить доступ к сети Интернет с помощью модуля Ethernet shield W5100.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль Ethernet shield W5100](#);
- [светодиод – 2 шт.](#);
- [резистор 220 Ом – 2 шт.](#);
- [провода папа-папа](#).

Ethernet Shield позволяет легко подключить вашу плату Arduino к локальной сети или сети Интернет. Он предоставляет возможность Arduino отправлять и принимать данные из любой точки мира с помощью интернет-соединения. Например, можно реализовать удаленное управление вашими исполнительными устройствами, подключенными к реле, через веб-сайт или создать устройство, которое с помощью звукового сигнала оповестит вас о новом электронном письме.

В первой части эксперимента рассмотрим использование платы Arduino с подключенным Ethernet Shield в качестве сервера, выдающего клиенту (браузеру) веб-страницу и позволяющего запросами с браузера изменять состояния подключенного к Arduino светодиода.

Подключаем к плате Arduino Ethernet shield, а к выводам D7, D8 – светодиоды через резистор 220 Ом (см. рис. 29.1).

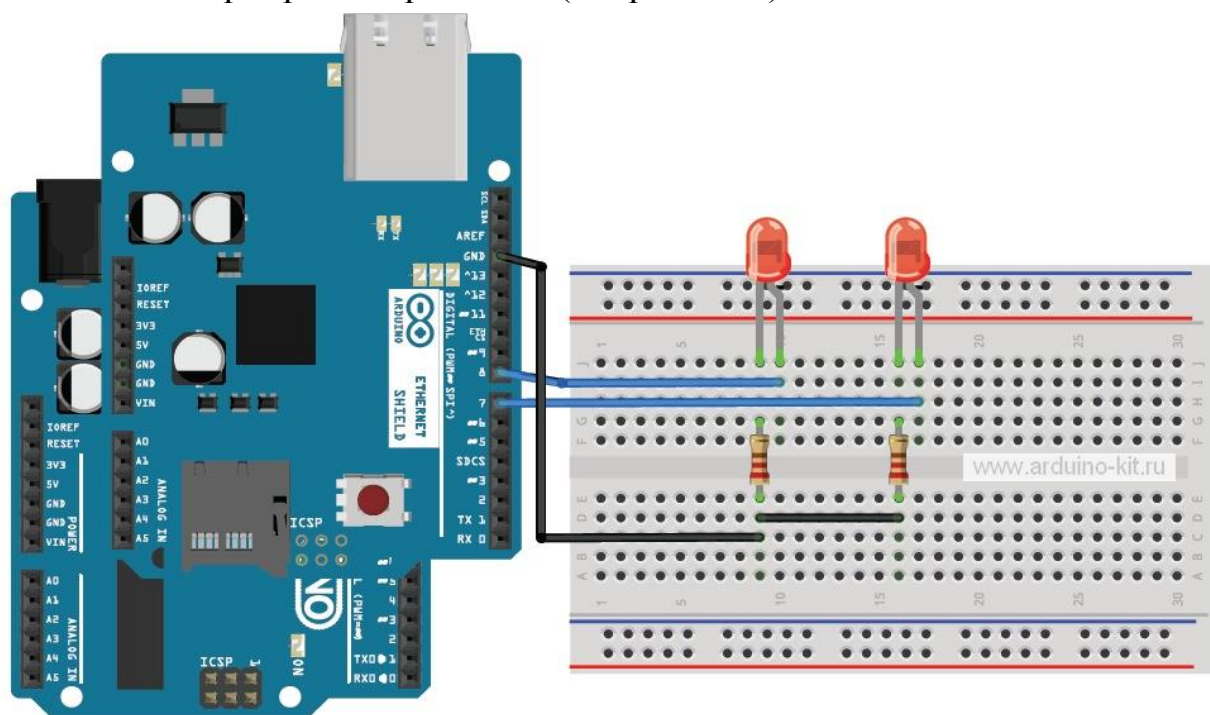


Рис. 29.1. Схема подключения модуля Ethernet shield и светодиодов

При написании скетча используем встроенную в Arduino IDE библиотеку Ethernet.

Порядок подключения:

1. Подключаем Ethernet shield к плате Arduino, с помощью кабеля RJ45 подключаем Ethernet shield к сети.
2. Подключаем светодиоды по схеме на рис. 29.1.
3. Составляем самостоятельно и загружаем в плату Arduino скетч
4. Открываем браузер на любом компьютере данной сети и в адресной строке набираем `http://192.168.0.214` (тот адрес, который вы присваиваете Arduino в скетче).
5. На странице (рис. 29.2), изменяя статус элементов input radio, видим изменение состояния светодиодов, подключенных к плате Arduino.



Рис. 29.2. Веб-страница, формируемая Arduino-сервером

ПРОЕКТ 30: БЕСПРОВОДНАЯ СВЯЗЬ. МОДУЛЬ WI-FI ESP8266

В этом эксперименте мы познакомимся с модулем ESP8266, с помощью которого можно подключить плату Arduino к сетям Wi-Fi, и напишем скетч для передачи данных датчика температуры на веб-сервис Народный мониторинг.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль ESP8266 ESP-01](#);
- датчик температуры LM335;
- [резистор 2,2 кОм](#);
- [провода папа-папа](#).
- [блок питания +5 в 1 А](#);
- преобразователь напряжения 3–30 В.

После своего появления платы на базе Wifi чипа ESP8266 стали по-настоящему народными. Огромные возможности и минимальная цена сделали свое дело. Платы на ESP8266 – это не просто модули для связи по Wi-Fi. Чип, по сути, является микроконтроллером со своими интерфейсами SPI, UART, а также портами GPIO, а это значит, что модуль можно использовать автономно без Arduino и других плат с микроконтроллерами. Существует около 11 официальных модификаций платы. В нашем распоряжении самая простая плата – ESP01. Распиновка платы показана на рис. 30.1. Покажем, как использовать ее в качестве Wi-Fi модуля для Arduino.

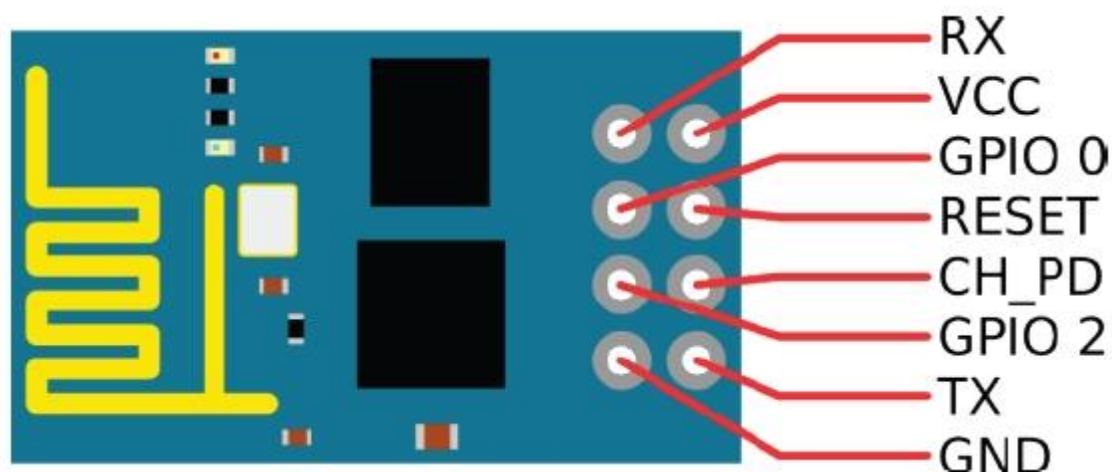


Рис. 30.1. Распиновка модуля ESP-01

Модуль ESP8266 рассчитан только на 3,3 В. Поэтому нам необходим источник питания 3,3 В. Схема подключения модуля ESP-01 к плате Arduino показана на рис. 30.2. Общение с модулем с помощью AT-команд. Список основных AT-команд показан в табл. 30.1. Загрузим на плату Arduino скетч, и

будем отправлять в модуль ESP-01 AT-команды. Результат выполнения команд показан на рис. 30.3.

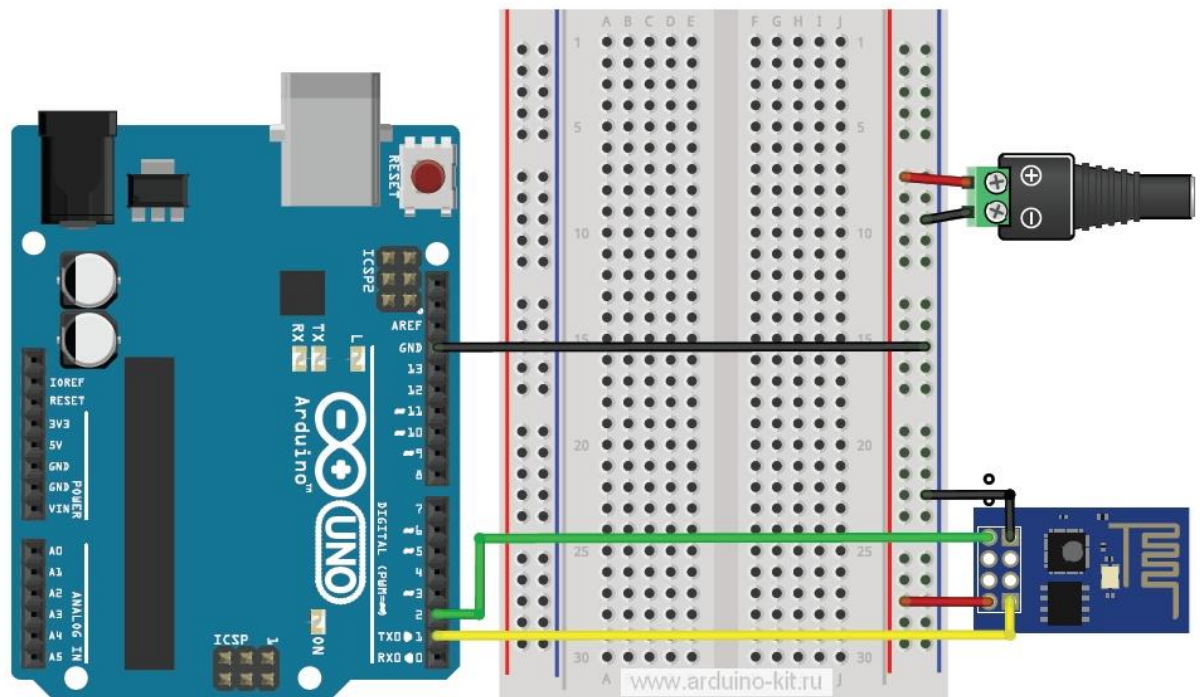


Рис. 30.2. Схема подключения модуля ESP-01 к Arduino

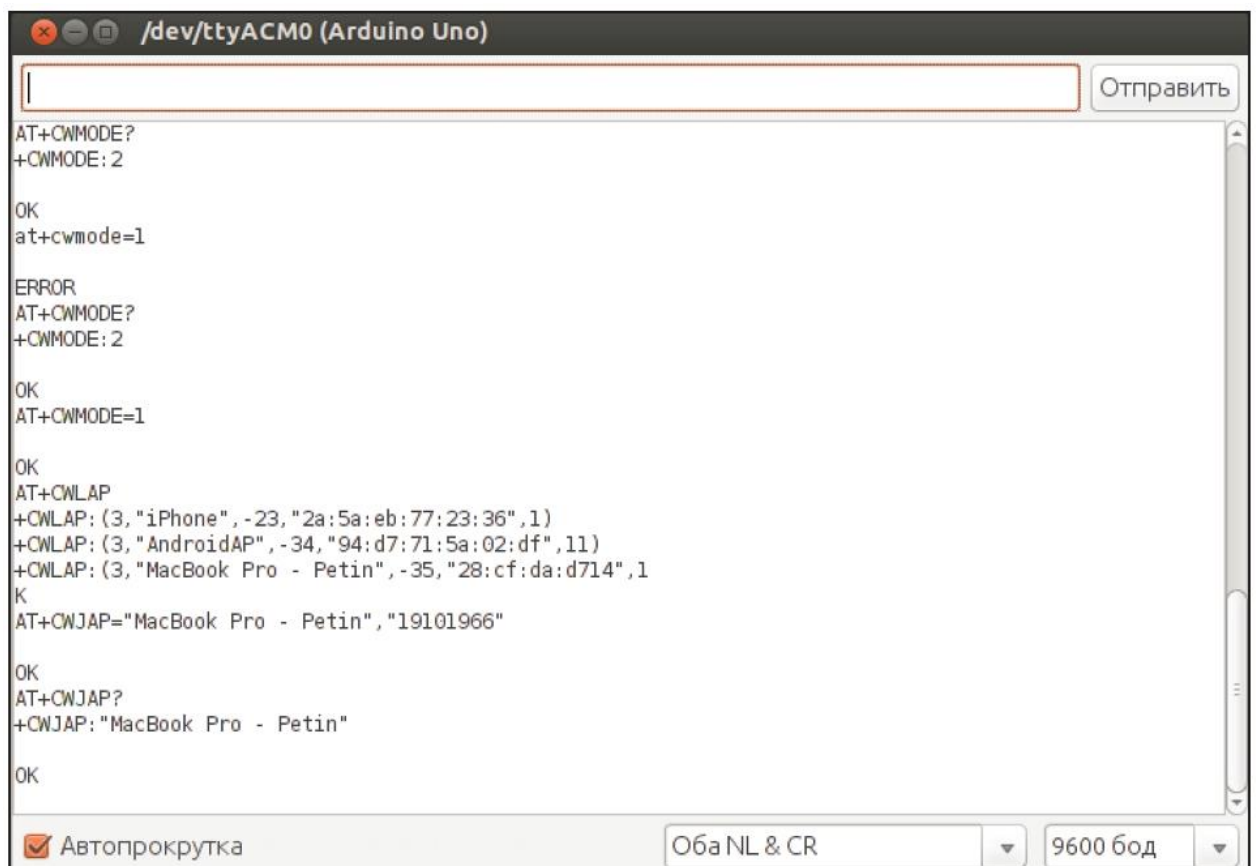


Рис. 30.3. Отправка AT-команд из Arduino IDE

Создадим скрипт отправки данных на сайт Народный мониторинг (<http://narodmon.ru>). Подключим к плате Arduino датчик температуры LM335

и каждые 10 минут будем отправлять данные. Схема соединений показана на рис. 30.4. Для отправки данных необходимо выполнить следующую последовательность действий:

1. Сброс ESP-01 и проверка готовности модуля (AT+RST).
2. Подключение к сети по Wi-Fi (AT+CWJAP="","").
3. Выбор режима одиночного соединения (AT+CWMUX=0).
4. Создание TCP-соединения (AT+CIPSTART="TCP","92.39.235.156", 8283).
5. Отправка данных (AT+CIPSEND= и сами данные #\n#\n#\n##).
6. Закрыть TCP-соединение (AT+CIPCLOSE).
7. Пауза 10 минут и переход к шагу 4.

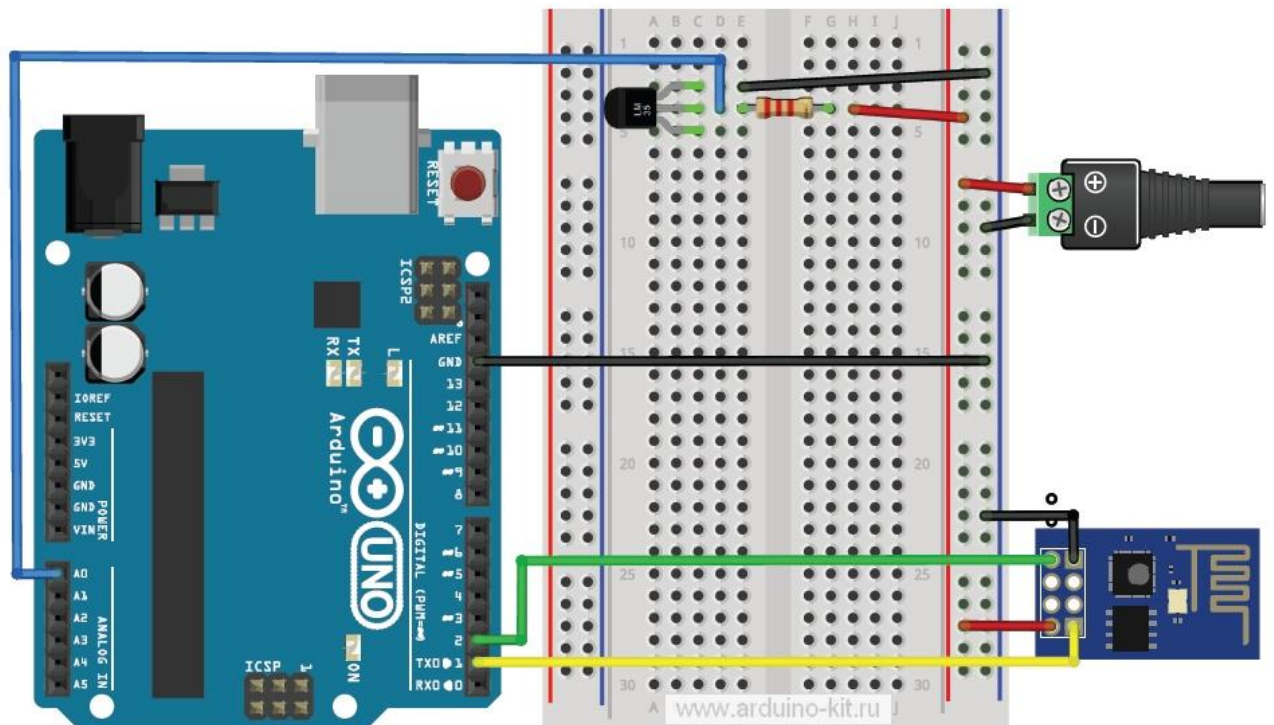


Рис. 30.4. Схема подключения модуля ESP-01 к Arduino

Таблица 30.1

Команда	Описание	Выполнение
AT	Проверка модуля. Если модуль успешно стартовал, то отвечает OK	AT
AT+RST	Перезапуск модуля	AT+RST
AT+RESTORE	Сбросить на заводские настройки	AT+RESTORE
AT+UART	Настройка последовательного интерфейса	AT+UART=baudrate,databits,stopbits,parity,flow control
AT+CWMODE	Переключение режима Wi-Fi. Для вступления в силу требуется перезапуск модуля командой AT+RST	AT+CWMODE? AT+CWMODE=1 (station) AT+CWMODE=2 (AP) AT+CWMODE=3 (station+AP)
AT+CWLJAP	Подключение к AP (точке доступа)	AT+CWLJAP=<идентификатор сети>,<пароль> AT+CWLJAP?
AT+CWLAP	Отобразить список доступных AP	AT+CWLAP
AT+CWQAP	Отключение от AP	AT+CWQAP
AT+CIPSTART	Установить подключение TCP или UDP	AT+CIPSTART="<TCP/UDP>","<IP>",<port>
AT+CIPSEND	Отправить данные	AT+CIPSEND=? AT+CIPSEND=<длина> AT+CIPSEND=<идентификатор><длина>
AT+CIPCLOSE	Закрыть подключение TCP или UDP	AT+CIPCLOSE

Составляем самостоятельно и загружаем в плату Arduino скетч.

ПРОЕКТ 31: БЕСПРОВОДНАЯ СВЯЗЬ. МОДУЛЬ BLUETOOTH HC-05

В этом эксперименте рассмотрим работу модуля Bluetooth HC-05, позволяющего плате Arduino установить беспроводную связь и обмениваться данными с другими устройствами по протоколу Bluetooth.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [модуль Bluetooth HC-05](#);
- [провода папа-папа](#).
- телефон или планшет с OS Android.

Bluetooth (с англ. – «голубой зуб») – одна из технологий беспроводной передачи данных. Спецификация была разработана в 1998 г. компанией Ericsson, а позднее оформлена группой Bluetooth Special Interest Group (SIG), официально зарегистрированной 20 мая 1999 г. Bluetooth позволяет объединять в локальные сети любую технику: от мобильного телефона и компьютера до холодильника. При этом одним из немаловажных параметров новой технологии являются низкая стоимость устройства связи (в пределах 20 долларов), его небольшие размеры (ведь речь идет о мобильных устройствах) и, что немаловажно, совместимость, простота встраивания в различные устройства. Мы будем использовать недорогой модуль HC-05. В нем используется чип BC417 плюс Flash-память и выводы GPIO.

Чип поддерживает спецификацию Bluetooth v2.0 + EDR, AT-команды, может работать в режиме Master или Slave, поддерживает скорость обмена от 2400 до 1 382 400. Напряжение питания модуля составляет 3,3 В, ток потребления ~50 мА, что позволяет питать его от вывода Arduino +3,3 В. Для программирования модуля с помощью AT-команд необходимо на вывод P1011 подать +3,3 В. Подключим модуль к плате Arduino и рассмотрим простейшие AT-команды. Схема подключения показана на рис. 31.1.

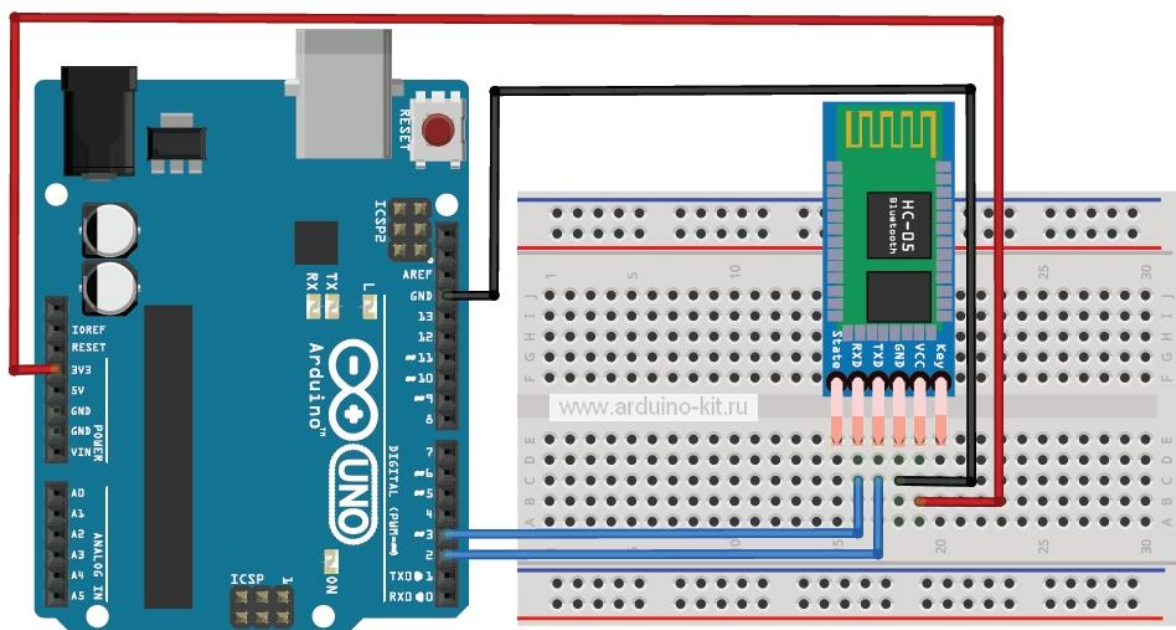


Рис. 31.1. Схема подключения модуля HC-05 к плате Arduino

Порядок подключения:

1. Подключаем модуль HC-05 к плате Arduino по схеме на рис. 31.1. Отключаем провод, ведущий к выводу 34 модуля (PIO11), от 3,3 В.
2. Загружаем на плату Arduino скетч
3. Светодиод на плате должен быстро мигать. Если не мигает или мигает иначе, отключаем питание модуля от 3,3 В, затем снова подключаем питание.
4. Подключаем провод, ведущий к выводу 34 модуля (PIO11), к 3,3 В.
5. Открываем монитор последовательного порта Arduino и набираем AT-команды из табл. 31.1. Смотрим результат выполнения команд (рис. 31.2).

Таблица 31.1

АТ-команда	Параметры, ответ	Описание
AT	Ответ модуля: OK	Тестовая команда
AT+NAME?	Ответ модуля: +NAME:<name>	Получить имя модуля
AT+NAME=<name>	<name> – имя Bluetooth-модуля. Ответ модуля: OK (или FAIL)	Установить новое имя модуля
AT+PSWD?	Ответ модуля: + PSWD:<password>	Получить код доступа к Bluetooth-модулю
AT+PSWD=<password>	<password> – код доступа к модулю. Ответ модуля: OK (или FAIL)	Установить код доступа к Bluetooth-модулю
AT+ROLE?	Ответ модуля: +ROLE:<role> – режим работы Bluetooth-модуля	Получить режим работы Bluetooth-модуля
AT+ROLE=<role>	<role> – режим работы: 0 – slave; 1 – master; 2 – slave-loop. Ответ модуля: OK (или FAIL)	Установить режим работы Bluetooth-модуля
AT+UART?	Ответ модуля: +UART:<p1>,<p2>,<p3> <p1> – скорость обмена; <p2> – стоп-бит; <p3> – бит паритета	Получить параметры обмена
AT+UART=<p1>,<p2>,<p3>	<p1> – скорость обмена (9600,19200,38400,57600,115200); <p2> – стоп-бит; <p3> – бит паритета. Ответ модуля: OK (или FAIL)	Установить параметры обмена
AT+ADDR?	Ответ модуля: +ADDR:<addr> <addr> – адрес Bluetooth-модуля NAP: UAP : LAP	Получить адрес модуля

Следующий шаг – двунаправленная передача данных между телефоном с OS Android и платой Arduino с модулем HC-05. Отсоединим контакт 34 Bluetooth-модуля от 3,3 В. Загрузим и установим на телефон из Play Market приложение Bluetooth Terminal (<https://play.google.com/store/apps/details?id=Qwerty.BluetoothTerminal&hl=ru>). Запустим программу и установим соединение с нашим модулем (см.рис. 31.2). Передаем Arduino и получаем (через монитор последовательного порта) из Arduino сообщения (см. рис. 31.3).

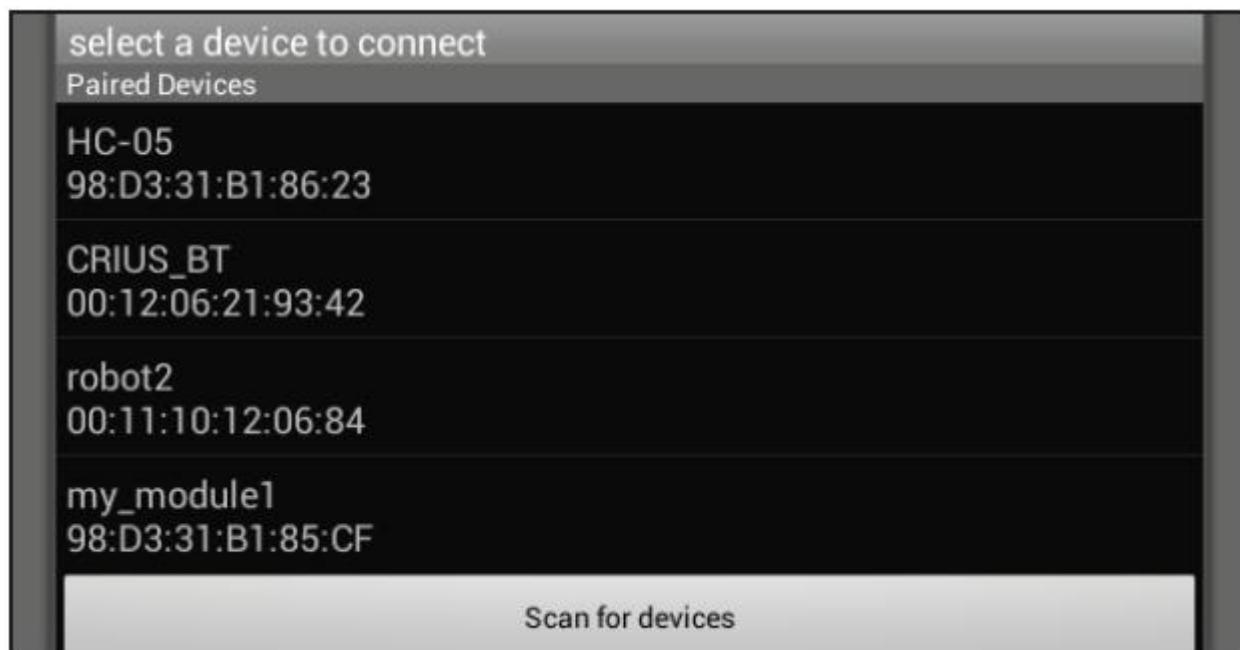


Рис. 31.2. Установка связи с модулем HC-05

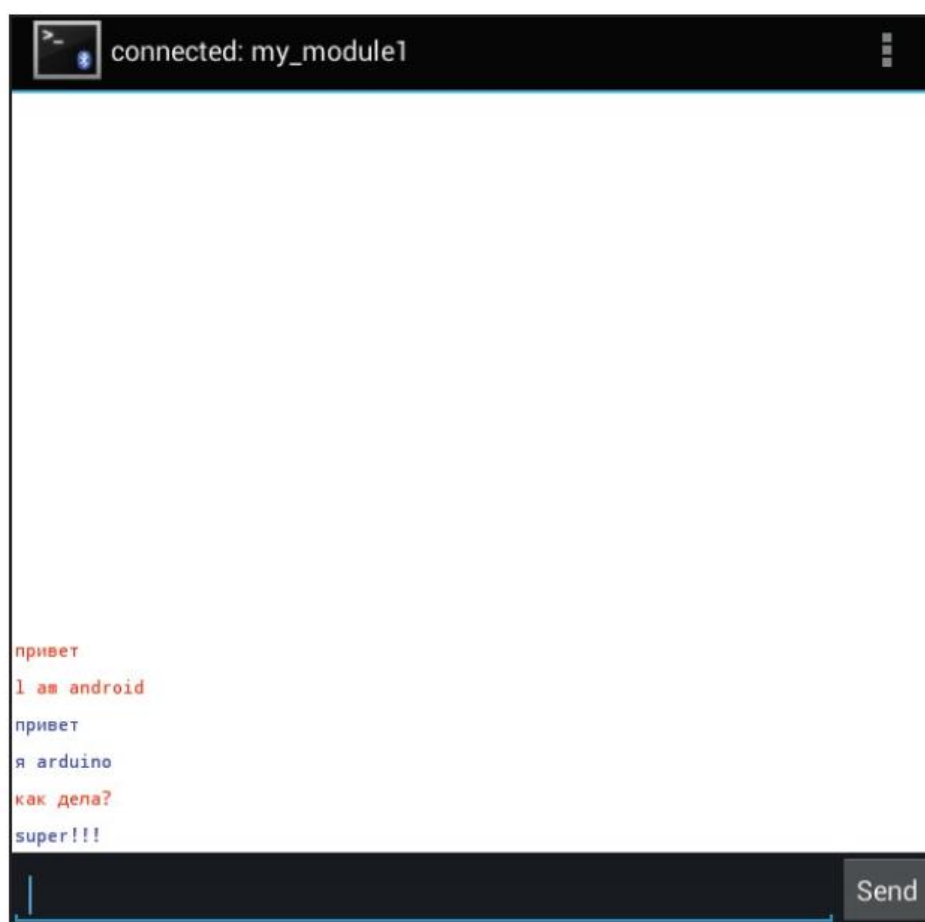


Рис. 31.3. Обмен сообщениями между телефоном и платой Arduino

ПРОЕКТ 32: БЕСПРОВОДНАЯ СВЯЗЬ. МОДУЛЬ GSM/GPRS SIM900

В этом эксперименте рассмотрим работу модуля GSM/GPRS shield – платы расширения, позволяющей Arduino работать в сетях сотовой связи по технологиям GSM/GPRS для приёма и передачи данных, SMS и голосовой связи.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [GSM/GPRS shield](#);
- работающая SIM-карта любого оператора;
- датчик температуры LM335;
- [резистор 2,2 кОм](#);
- [провода папа-папа](#).
- [блок питания +5 в 1 А](#);

GSM/GPRS shield на базе модуля SIMCom SIM900 выпускают несколько производителей, и платы имеют незначительные отличия. Также на некоторых платах расположены: слот для SIM-карты, стандартные 3,5 мм джек для аудиовхода и выхода и разъём для внешней антенны. На плате GSM/GPRS shield имеется несколько перемычек, позволяющих выбрать тип serial-соединения (hardware или software). GSM/GPRS shield имеет два способа включения – аппаратный (кратковременное нажатие кнопки PWRKEY) и программный (используется один из выходов Arduino). Рассмотрим пример отправки и получения SMS-сообщений с помощью GSM/GPRS shield. Каждые 30 минут будем отправлять на определенный номер показания аналогового датчика температуры LM335, подсоединенного к выводу A0. Схема соединений эксперимента показана на рис. 32.1.

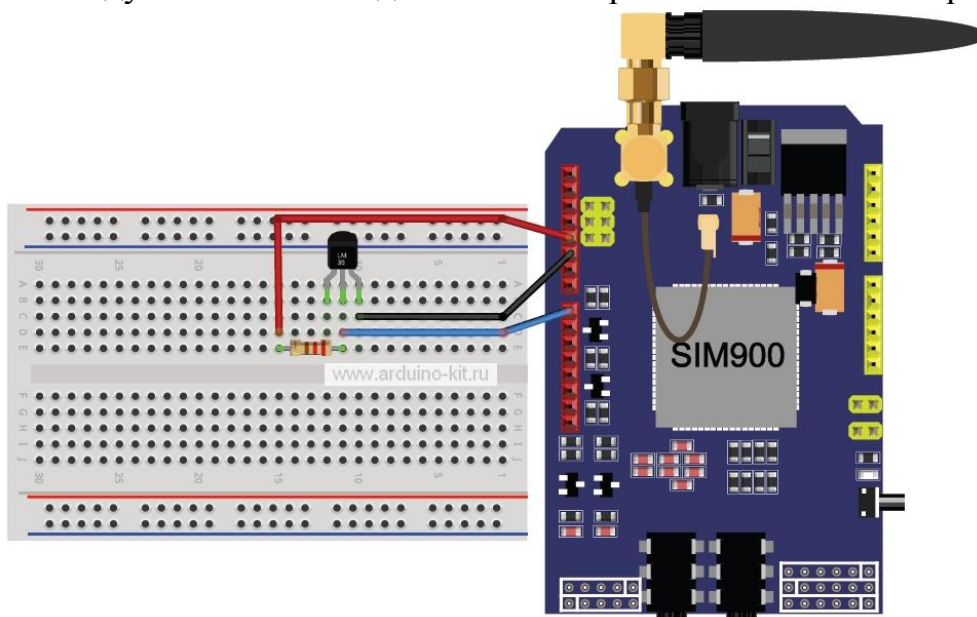


Рис. 32.1. Схема подключения модуля GSM/GPRS shield и датчика LM335.
Шилд SIM900 установлен на Arduino UNO

Порядок подключения:

1. Установим SIM-карту на GSM/GPRS Shield, а GSM/GPRS Shield – на Arduino. С помощью джамперов соединим контакты для работы через SoftwareSerial-эмуляцию.
2. Собираем схему, согласно рис. 32.1.
3. Составляем самостоятельно и загружаем в плату Arduino скетч
4. На телефон, указанный в скетче, раз в 30 минут должны приходить sms-сообщения с данными температуры. Теперь изменим скетч таким образом, чтобы Arduino отправляла sms-сообщение с данными температуры только при получении приходящего сообщения с текстом «temp».

ПРОЕКТ 33: МОДУЛЬ GPS. ПРИНЦИП РАБОТЫ, ПОДКЛЮЧЕНИЕ, ПРИМЕРЫ

В этом эксперименте рассмотрим работу модуля GPS-приемника, позволяющего определять наше местоположение с помощью глобальной системы GPS, и подключение данного приемника к плате Arduino.

Необходимые компоненты:

- [контроллер Arduino UNO R3](#);
- [плата для прототипирования](#);
- [GPS-приемник V.KEL VK16E](#);
- [провода папа-папа](#).

GPS (Global Positioning System) – это система, позволяющая с точностью не хуже 100 м определить местоположение объекта, то есть определить его широту, долготу и высоту над уровнем моря, а также направление и скорость его движения. Кроме того, с помощью GPS можно определить время с точностью до 1 наносекунды. GPS состоит из совокупности определенного количества искусственных спутников Земли (спутниковой системы NAVSTAR) и наземных станций слежения, объединенных в общую сеть. В качестве абонентского оборудования служат индивидуальные GPS-приемники, способные принимать сигналы со спутников и по принятой информации вычислять свое местоположение. Мы используем GPS-приемник V.KEL VK16E (см. рис. 33.1).



Рис. 33.1. GPS-приемник V.KEL VK16E

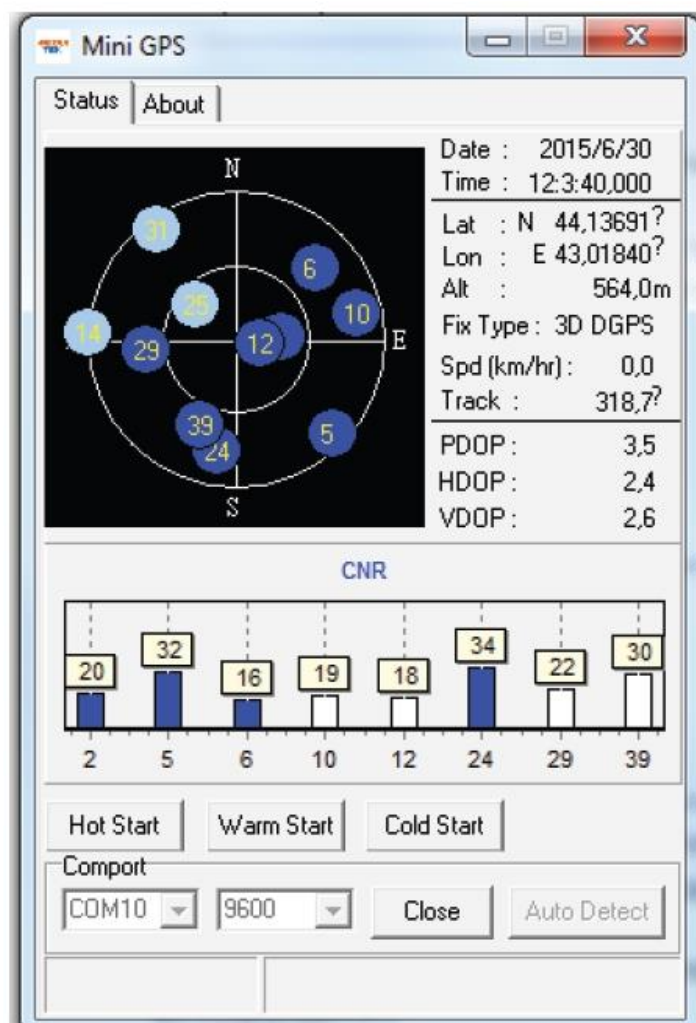


Рис. 33.2. Просмотр данных GPS-приемника в программе MiniGPS

Назначение выводов:

- BOOT – зарезервировано;
- GND – общий вывод;
- RX – вход для данных в последовательном формате UART;
- TX – выход для данных в последовательном формате UART;
- VCC – вход напряжения питания от 3,3 до 5 В;
- PPS – выход импульсов времени.

Холодный старт происходит, когда GPS-приемник был выключен длительное время, перемещался в выключенном состоянии на значительное расстояние или его часы не совпадают с данными спутника. При холодном старте со спутников скачивается альманах. Время **обновления альманаха – от 5 до 15 минут в зависимости от условий приема и количества видимых спутников. Особенность – приемник в это время должен быть неподвижен. Теплый старт происходит, когда приемник был выключен более 30 мин. При этом происходит прием уточняющих данных. Занимает 0,5–1,5 мин. Горячий старт

происходит, когда приемник был отключен непродолжительное время. Данные считаются свежими, и приемник просто находит спутники (опираясь на данные альманаха). Для проверки работоспособности подключим модуль по последовательному порту к компьютеру с ОС Windows, используя USBTTL-адаптер, и запустим программу MiniGPS_v1.7.1.exe. Программа показывает количество найденных приемником спутников и в случае их достаточного числа показывает наше местоположение – географические широту и долготу (см. рис. 33.2). Мигание зеленого светодиода сигнализирует о достаточном для определения положения количестве спутников. Теперь подключим GPS-приемник к плате Arduino и будем выводить данные о местоположении – географические широту и долготу в монитор последовательного порта Arduino. Схема подключения показана на рис. 33.3.

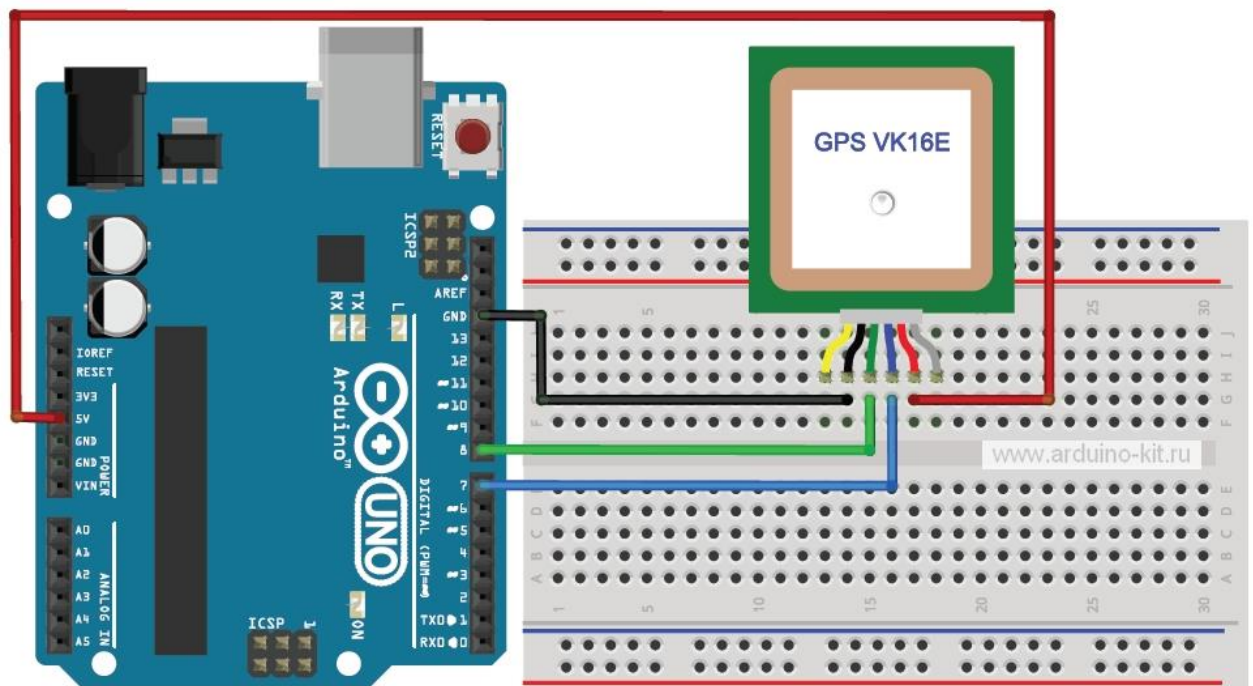


Рис. 33.3. Схема подключения GPS-приемника к Arduino

Приступим к написанию скетча. Будем использовать библиотеки SoftwareSerial и TinyGPS, позволяющие выделять нужные данные из всего потока, передаваемого приемником.

Порядок подключения:

1. Собираем схему, согласно рис. 33.1.
2. Составляем самостоятельно и загружаем в плату Arduino скетч
3. Ждем мигания зеленого светодиода на приемнике GPS, сигнализирующего о наличии данных о местоположении.
4. Смотрим в мониторе последовательного порта Arduino вывод данных широты и долготы. Также получаем текущую дату и время по Гринвичу (рис. 33.4).

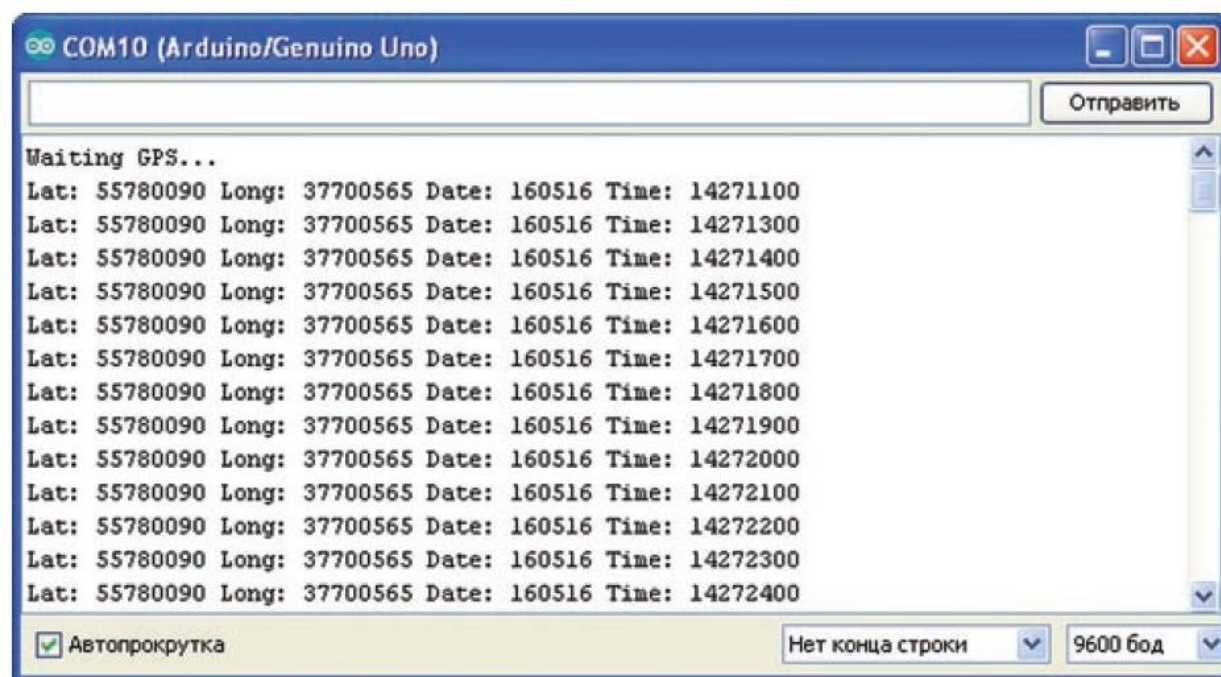


Рис. 33.4. Вывод данных GPS в монитор последовательного порта Arduino

ПРОЕКТ 34: МОДУЛЬ МАТРИЦЫ LED 8x8

LED матрица имеет 8 светодиодов в ширину и 8 в высоту, всего 64 светодиода, управление каждым светодиодом осуществляется при помощи микросхемы MAX7219. В матрице используется [динамическая индикация](#), это означает что каждый столбец загорается поочередно.

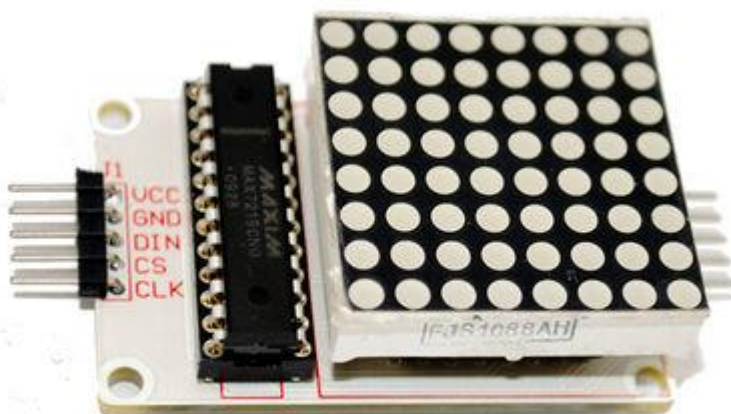
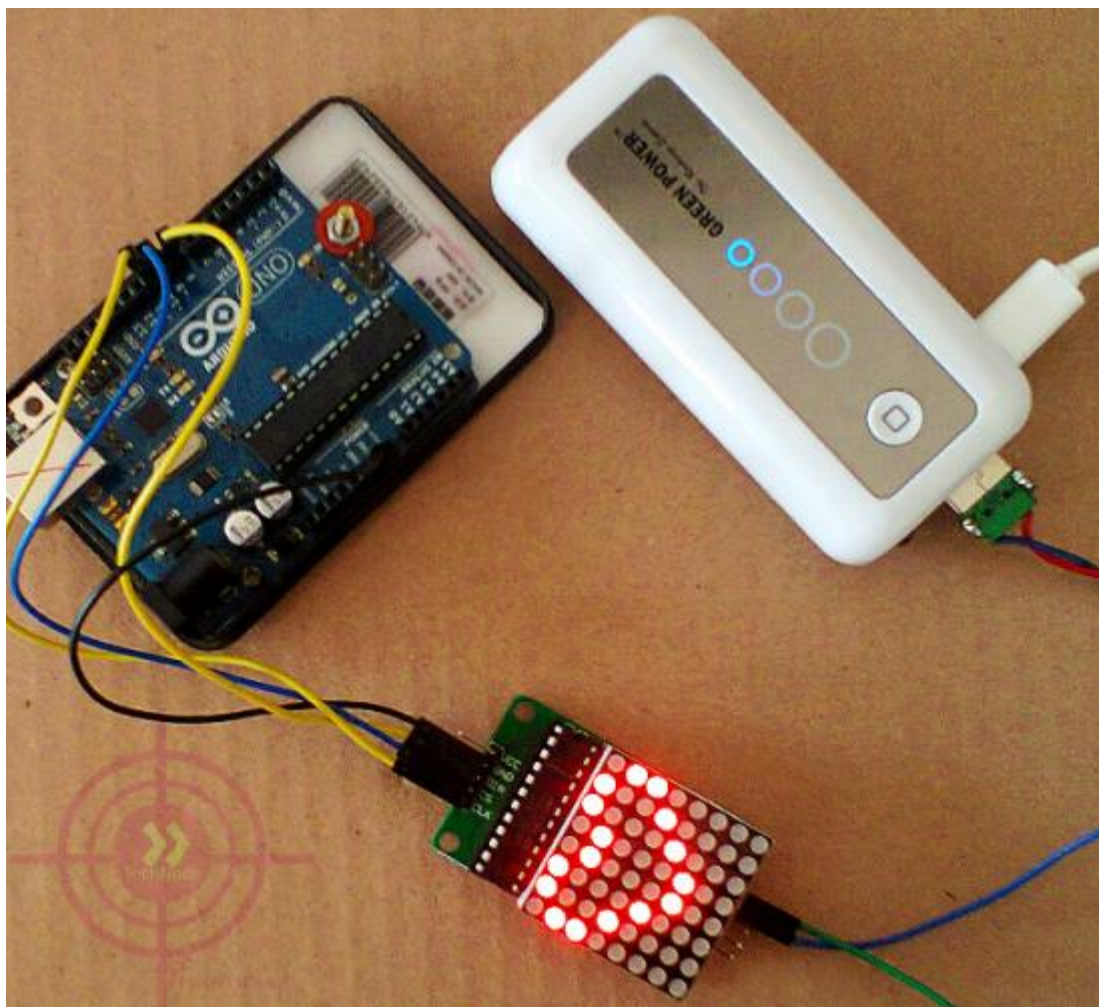
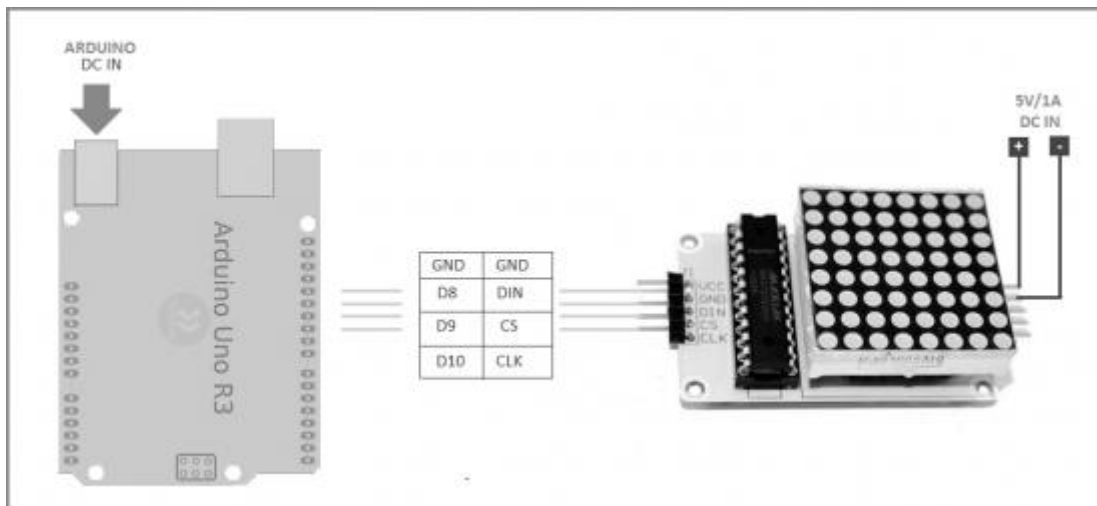


Рис. Подключение светодиодной матрицы 8x8 к Arduino



Скетч вывод сердечка

```

1 unsigned char i;
2 unsigned char j;
3 int Max7219_pinCLK = 10;
4 int Max7219_pinCS = 9;
5 int Max7219_pinDIN = 8;
6
7 unsigned char disp1[19][8]={
8  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
9  0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
10 0x00, 0x00, 0x00, 0x40, 0x40, 0x00, 0x00, 0x00,
11 0x00, 0x00, 0x80, 0x40, 0x40, 0x00, 0x00, 0x00,
12 0x00, 0x80, 0x80, 0x40, 0x40, 0x00, 0x00, 0x00,
13 0x40, 0x80, 0x80, 0x40, 0x40, 0x00, 0x00, 0x00,
14 0x60, 0x80, 0x80, 0x40, 0x40, 0x00, 0x00, 0x00,
15 0x60, 0x90, 0x80, 0x40, 0x40, 0x00, 0x00, 0x00,
16 0x60, 0x90, 0x88, 0x40, 0x40, 0x00, 0x00, 0x00,
17 0x60, 0x90, 0x88, 0x44, 0x40, 0x00, 0x00, 0x00,
18 0x60, 0x90, 0x88, 0x44, 0x44, 0x00, 0x00, 0x00,
19 0x60, 0x90, 0x88, 0x44, 0x44, 0x08, 0x00, 0x00,
20 0x60, 0x90, 0x88, 0x44, 0x44, 0x08, 0x10, 0x00,
21 0x60, 0x90, 0x88, 0x44, 0x44, 0x08, 0x10, 0x20,
22 0x60, 0x90, 0x88, 0x44, 0x44, 0x08, 0x10, 0x60,
23 0x60, 0x90, 0x88, 0x44, 0x44, 0x08, 0x90, 0x60,
24 0x60, 0x90, 0x88, 0x44, 0x44, 0x88, 0x90, 0x60
25};
26
27void Write_Max7219_byte(unsigned char DATA)
28{
29 unsigned char i;
30 digitalWrite(Max7219_pinCS,LOW);
31 for(i=8;i>=1;i--)
```

```

32 {
33 digitalWrite(Max7219_pinCLK, LOW);
34 digitalWrite(Max7219_pinDIN, DATA&0x80);
35 DATA = DATA<<1;
36 digitalWrite(Max7219_pinCLK, HIGH);
37 }
38}
39
40void Write_Max7219(unsigned char address, unsigned char dat)
41{
42 digitalWrite(Max7219_pinCS,LOW);
43 Write_Max7219_byte(address);
44 Write_Max7219_byte(dat);
45 digitalWrite(Max7219_pinCS,HIGH);
46}
47
48void Init_MAX7219(void)
49{
50 Write_Max7219(0x09, 0x00);
51 Write_Max7219(0x0a, 0x03);
52 Write_Max7219(0x0b, 0x07);
53 Write_Max7219(0x0c, 0x01);
54 Write_Max7219(0x0f, 0x00);
55}
56
57void setup()
58{
59 pinMode(Max7219_pinCLK,OUTPUT);
60 pinMode(Max7219_pinCS,OUTPUT);
61 pinMode(Max7219_pinDIN,OUTPUT);
62 delay(50);
63 Init_MAX7219();
64}
65void loop()
66{
67 for(j=0;j<19;j++)
68 {
69 for(i=1;i<9;i++)
70 Write_Max7219(i,disp1[j][i-1]);
71 delay(500);
72 }
73}

```


Скетч **ВЫВОД СИМВОЛОВ**

```
#define Din 8
#define Load 9
#define Clk 10

void FlashNow (int);
// матрица состояний индикатора 8x8 -одна строка - одно состояние
// 0-10 - цифры
// 11-20 ГЛАЗА
char data[][8]
{ {0x00,0x00,0xFF,0x81,0x81,0xFF,0x00,0x00}, //0- 0
  {0x00,0x00,0x00,0x20,0x40,0xFF,0x00,0x00}, //1-1
  {0x00,0x00,0xCF,0x89,0x89,0xF9,0x00,0x00}, //2- 2
  {0x00,0x00,0x91,0x91,0x91,0xFF,0x00,0x00}, //3- 3
  {0x00,0x00,0xF8,0x08,0x08,0xFF,0x00,0x00}, //4- 4
  {0x00,0x00,0xF1,0x91,0x91,0x9F,0x00,0x00}, //5- 5
  {0x00,0x00,0xFF,0x91,0x91,0x9F,0x00,0x00}, //6- 6
  {0x00,0x00,0x80,0x80,0x80,0xFF,0x00,0x00}, //7- 7
  {0x00,0x00,0xFF,0x91,0x91,0xFF,0x00,0x00}, //8- 8
  {0x00,0x00,0xF1,0x91,0x91,0xFF,0x00,0x00}, //9-9
  {0x20,0x40,0xFF,0x00,0xFF,0x81,0x81,0xFF}, //10-10
  {0x20,0x40,0xFF,0x00,0xFF,0x81,0x81,0xFF}, //10-10
  {0x3C,0x42,0x81,0x99,0x99,0x81,0x42,0x3C}, //11 - ГЛАЗ центр
  {0x18,0x18,0x18,0x18,0x18,0x18,0x18,0x18}, //12 - ГЛАЗ закрыт
  {0x3C,0x42,0xB1,0xB1,0x81,0x81,0x42,0x3C}, //13 - ВЕРХ левый угол
  {0x3C,0x42,0x81,0x8D,0x8D,0x81,0x42,0x3C}, //14 - ВНИЗ средне
  {0x3C,0x42,0x81,0x81,0x8D,0x8D,0x42,0x3C}, //15 - ВНИЗ направо
  {0x3C,0x42,0x81,0x81,0xB1,0xB1,0x42,0x3C}, //16 - ВЕРХ направо угол
  {0x3C,0x42,0x81,0xB1,0xB1,0x81,0x42,0x3C}, //17 - ВЕРХ средне
  {0x3C,0x42,0x81,0xE1,0xE1,0x81,0x42,0x3C}, //18 - ВЕРХ высоко
  {0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C,0x0C}, //19 - НИЖНЕЕ моргание
  {0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30}, //20 - ВЕРХНЕЕ моргание

  {0xFF,0x81,0x81,0x81,0x81,0x81,0x81,0xFF}, //21-КВАДРАТ
  {0xFF,0x81,0x81,0x99,0x99,0x81,0x81,0xFF}, //22-КВАДРАТНЫЙ ГЛАЗ
  {0xAA,0x80,0x01,0x80,0x01,0x80,0x01,0x55}, //23 БЕГУЩИЙ ОГОНЬ
ПОЛОЖЕНИЕ-1
  {0x55,0x01,0x80,0x01,0x80,0x01,0x80,0xAA}, //24 БЕГУЩИЙ ОГОНЬ
ПОЛОЖЕНИЕ-2 {0x18,0x3C,0x3C,0x24,0x24,0x24,0x24,0x18}, //25 -УЗКИЙ
ГЛАЗ ЛЕВО
  {0x18,0x24,0x24,0x3C,0x3C,0x24,0x24,0x18}, //26 -УЗКИЙ ГЛАЗ
  {0x18,0x24,0x24,0x24,0x24,0x3C,0x3C,0x18}, //27 -УЗКИЙ ГЛАЗ ПРАВО
```



```
{0x18,0x3C,0x24,0x24,0x24,0x24,0x24,0x18}, //28 -УЗКИЙ ГЛАЗ ЛЕВО  
ЛЕВЕЕ  
{0x18,0x24,0x24,0x24,0x24,0x24,0x3C,0x18}, //29 -УЗКИЙ ГЛАЗ ПРАВО  
ПРАВЕЕ  
{0x0A,0x05,0x05,0x07,0x07,0x05,0x05,0x0A}, //30 -УЗКИЙ ГЛАЗ ВНИЗУ  
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00} //31- пустая строка  
};
```

```
void setup()  
{  
  pinMode (Din, OUTPUT);  
  pinMode (Load, OUTPUT);  
  pinMode (Clk, OUTPUT);  
  // Устанавливаем Decode Mode, Intensity, Scan Limit, Shutdown и Display Test,  
  но сначала  
  digitalWrite (Clk, 0); // первый такт устанавливаем в 0  
  Data16sent(0x09,0x00); //Decode Mode - без декодирования.  
  Data16sent(0x0A,0x08); //Intensity - среднее свечение 17 из 32 см.таблицу  
  Data16sent(0x0B,0x07); //Scan Limit - используем все 8 катодов.  
  Data16sent(0x0C,0x01); //используем режим Shutdown в Normal Mode.  
  Data16sent(0x0F,0x00); //Display Test - Normal Mode.  
}
```

```
void loop()  
{  
  FlashNow(11);  
  delay(1000);  
  FlashNow(12);  
  delay(400);  
  FlashNow(13);  
  delay(900);  
  FlashNow(14);  
  delay(90);  
  FlashNow(15);  
  delay(800);  
  FlashNow(16);  
  delay(200);  
  FlashNow(17);  
  delay(1000);  
  FlashNow(18);  
  delay(120);  
  FlashNow(19);  
  delay(800);  
  FlashNow(20);  
  delay(200);
```

```

}
void FlashNow (int sost)
{ // Зажигает номер комбинации - эскиз из двумерного массива data[][]
for (int i=0;i<8;i++)
{ //заполняет 8 столбцов матрицы
//готовим столбец - катод №1 меняется только последняя цифра, начинаем с
1, поэтому i+1
Data16sent(i+1,data[sost][i]); //зажигаем один i-й столбец
}
}
void Data16sent (char address, char data)
{ //Принимает 8 бит адреса и 8 бит данных, отправляет 16 битную посылку в
MAX7219
digitalWrite (Load, 0);
shiftOut (Din, Clk, MSBFIRST, address); //первым уходит старший байт
shiftOut (Din, Clk, MSBFIRST, data); //вторым младший
digitalWrite (Load, 1);
}
Загружаем в плату Arduino указанный скетч.

```