

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ</b> .....	3
ГЛАВА 1. Теоретические основы электроники в дистанционном курсе.....	7
1.1 Электроника в основе занятий по технологии .....	7
1.2 Программируемая микроэлектроника в основе платформ для программирования .....	9
1.3 Плата разработки Arduino.....	11
1.4. Обзор платформы для дистанционного курса .....	17
<b>Выводы по первой главе</b> .....	22
ГЛАВА 2. ДИСТАНЦИОННЫЙ ФОРМАТ ПРОВЕДЕНИЯ ЗАНЯТИЙ ПО ПРЕДМЕТУ ТЕХНОЛОГИЙ .....	23
2.1 Методические рекомендации по планированию и проведению занятий в дистанционном формате. ....	23
2.3. Проектирование практических заданий .....	33
2.4. Педагогический эксперимент .....	59
<b>Выводы по второй главе</b> .....	61
<b>ЗАКЛЮЧЕНИЕ</b> .....	62
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	63
<b>Приложение 1</b> .....	66
<b>Приложение 2</b> .....	69

## Реферат

Магистерская диссертация: 71 страница, 25 источников.

**Ключевые слова:** Схемотехника, контроллер, микроконтроллер, Arduino, дистанционных формат, дистанционный курс, Google класс, тест.

**Цель работы:** разработка педагогических условий для формирования базовых инженерных компетенций у бакалавров педагогического образования для дальнейшего преподавания в дополнительном образовании.

**Объект исследования:** процесс обучения в дополнительном образовании.

**Предмет исследования:** педагогические условия, направленные на формирование базовых инженерных компетенций.

### **Задачи исследования:**

1. Провести анализ существующих программ по электроники в основном и дополнительном образовании.
2. Сформировать комплекс организационных, дидактических условий (методик, технологий и т.п.) для формирования базовых инженерных компетенций.
3. Апробировать в опытно-экспериментальной работе комплекс организационных, дидактических условий для формирования базовых инженерных компетенций.

По теме исследования опубликована одна статья: Выходные данные статьи: Михайлов .. ДИСТАНЦИОННОЕ ДОПОЛНИТЕЛЬНО ОБРАЗОВАНИЕ ДЛЯ ШКОЛЬНИКОВ // Студенческий вестник: электрон. научн. журн. 2021. № 46(191). URL: <https://studvestnik.ru/journal/stud/herald/191> (дата обращения: 19.12.2021).

## Abstract

Master's thesis: 71page, 25 sources.

Keywords: Schematic, controller, microcontroller, Arduino, remote format, distance course, Google class, test.

The purpose of the work: the development of pedagogical conditions for the formation of basic engineering competencies in bachelors of pedagogical education for further teaching in additional education.

Subject of the study: the process of learning in additional education.

Subject of the study: pedagogical conditions aimed at the formation of basic engineering competencies.

Research Objectives:

1. Analyze existing electronics programs mainly and additional education.
2. To form a set of organizational, didactic conditions (techniques, technologies, etc.) for the formation of basic engineering competencies.
3. Test in experimental work a set of organizational, didactic conditions for the formation of basic engineering competencies.

One remains on the topic of the study: Output of the article: Mikhailov.. DISTANCE ADDITIONAL EDUCATION FOR SCHOOLCHILDREN//Student Bulletin: electron. scientifically. magazine. 2021. № 46(191). URL: <https://studvestnik.ru/journal/stud/herald/191> (case date: 19.12.2021).

## ВВЕДЕНИЕ

**Актуальность и практическая значимость** данной тематики обусловлена тем, что полученные на занятиях знания становятся для ребят необходимой теоретической и практической основой их дальнейшего участия в техническом творчестве, выборе будущей профессии, в определении жизненного пути. Овладев навыками технического творчества сегодня, они, в дальнейшем, сумеют применить их с нужным эффектом в своих трудовых делах.

В связи с этим, исследование, направленное на формирование базовых инженерных компетенций у обучающихся, является актуальным.

В современных реалиях, реализовать эту программу можно с помощью дистанционных курсов. Огромное количество сервисов для работы через интернет, помогают не только преподавателю выдавать задания и развиваться самому, но и ученикам, осуществлять выполнение заданий, искать информацию в интернете и обрабатывать ее.

Схемотехника и управляемая электроника очень широкое понятие, состоящее из множества предметов и направлений.

**Объект исследования:** процесс обучения в дополнительном образовании.

**Предмет исследования:** педагогические условия, направленные на формирование базовых инженерных компетенций.

**Цель исследования заключается в разработке педагогических условий для формирования базовых инженерных компетенций у бакалавров педагогического образования для дальнейшего преподавания в дополнительном образовании.**

Объект, предмет и цель исследования определили следующие задачи:

1. Провести анализ существующих программ по электроники в основном и дополнительном образовании.
2. Сформировать комплекс организационных, дидактических условий (методик, технологий и т.п.) для формирования базовых инженерных компетенций.
3. Апробировать в опытно-экспериментальной работе комплекс организационных, дидактических условий для формирования базовых инженерных компетенций.

Работа состоит из введения, двух глав, заключения и списка литературы, состоящего из 25 наименований. Текст иллюстрируют 3 таблиц, 30 рисунков. Объем работы составляет 71 страница.

Апробация работы проводила на базе детского технопарка “Кванториум” г.Красноярска.

## ГЛАВА 1. Теоретические основы электроники в дистанционном курсе

### 1.1 Электроника в основе занятий по технологии

Основа дистанционного курса, является электроника, которая является элементной базой для современных устройств.

Электроника – это область науки и техники, которая занимается разработкой и практическим применением различных устройств и приборов.

В школьной программе рассматриваются понятия электричества, принципиальных схем. Рассмотрим более подробно появление и развитие электроники.

После открытия и дальнейшего изучения электричества, стали появляться первые электронные устройства. Первым изобретенным устройством было радио, которое сразу же нашло применение в военном направлении. Все первые устройства были изготовлены на электронных лампах. В связи с появлением ламп, активно начало развиваться направление вакуумной электроники. В процессе изучения вакуумной электроники появились телевизоры и радары, которые так же активно использовались и развивались в военном направлении, т.к. происходило все во времена второй мировой войны.

У электронных ламп есть ряд существенных недостатков. Главные недостатки связаны с большими габаритами и высоким потреблением электроэнергии, эти сильно осложняло использование портативных устройств. Ряд недостатков подтолкнул к дальнейшему развитию электронной базы и дальнейший переход с ламповой техники, на твердотельную электронику (диоды и транзисторы).

Дальнейший скачком для развития электроники послужило создание первого компьютера. Именно это достижение позволило проводить первые вычисления на технике. Первый ЭВМ был полностью собран из транзисторов, недостатками были огромные размеры, большое потребление электроэнергии и низкая надежность. Для решения возникших проблем, стали использовать

микросборки, а после и микросхемы. Количество элементов в сборках увеличивалось, появились первые микропроцессоры. В наше время толчком для развития служит развитие сотовой связи, а также беспроводные устройства.

Вся современная элементная база построена на основе твердотельной электронике. Впервые этот термин появился в середине XX века и обозначал устройства на полупроводниковой элементной базе: транзисторы и диоды, заменившие громоздкие и низкоэффективные электровакуумные лампы. С появлением твердотельных элементов начался процесс миниатюризации устройств. За несколько десятков лет элементы уменьшились в несколько миллиардов раз, если ЭВМ в момент своего появления занимал целые комнаты, то сейчас смело умещается в ладони и вычислительная мощность при этом в разы больше. Для примера компьютер, который был установлен в первой космической ракете имел вычислительную мощность современного калькулятора.

Основные элементы в электронных устройствах:

- Диод
- Биполярный транзистор
- Полевой транзистор
- Динисторы и тиристоры
- Интегральные микросхемы

После появления твердотельной элементной базы и уменьшения габаритов, появилась необходимость в автономном управлении.

## 1.2 Программируемая микроэлектроника в основе платформ для программирования

Платформы для программирования и написания программ состоят из ряда элементов, необходимых для работы, основным элементом является микроконтроллер.

**Микроконтроллер** – микросхема, предназначенная для управления электронными устройствами.

Обычно микроконтроллер сочетает в себе функции процессора и периферийных устройств, содержит ОЗУ и ПЗУ. Является однокристалльным компьютером, позволяющий выполнять простые задачи. Отличается от микропроцессора встроенными в микросхему устройствами ввода-вывода, таймерами и другими устройствами.

С появлением микроконтроллеров начинается массовое применение автоматизации в области управления. Так появился термин “микроконтроллер”. Controller – регулятор, управляющее устройство.

С увеличением импорта техники и спадом отечественного производства, произошла смена терминов. Термин “микроконтроллер” вытеснил ранее использовавшийся “однокристалльный микроЭВМ”.

В 1984 году появилась компания полупроводниковых электронных компонентов Atmel Corporation. Вклад этой компании колоссальный, сложно представить современное устройство без их контроллеров.

Первый микроконтроллер компании Atmel появился в 1993 году и был основан на основе прорывного микроконтроллера компании intel i8051. Компания включает множество различных микроконтроллеров на основе собственных разработок. Atmel представляет свои устройства как стандартные и полностью укомплектованные.

Именно компания Atmel стала началом развития популярной в наше время платы для разработчиков Arduino. Все платы основаны на контроллерах



Atmel, имеют полную и законченную версию для производства. Платы несут на себе все необходимое для успешного применения в различных задачах и проектах.

### 1.3 Плата разработки Arduino

На смену простых платформ, таких как Lego, приходят более сложные устройства. Именно одна из таких платформ и послужила началом изучения программирования и схемотехники. Именно эта плата позволяет производить прототипирование и изучение в старших классах и студентов.

Arduino – торговая марка аппаратно-программных средств, созданная для прототипирования и построения систем или моделей. Позволяет быстро оценить работоспособность спроектированных схем, провести эксперименты в области электроники, автоматики и робототехники.

Аппаратная часть, построена на микроконтроллерах компании Atmel, имеет встроенную память и весь необходимый набор периферийных устройств для связи и ввода-вывода информации. Открытая архитектура позволяет свободно копировать и изменять самостоятельно под конкретные задачи платформу.

Бесплатная программное обеспечение является собственной разработкой фирмы Arduino и позволяет пользоваться любому желающему. Простой и русскоязычный интерфейс делает программирование интуитивно понятным. Оболочка состоит из:

- Текстовый редактор
- Менеджер проектов
- Процессор
- Компилятор
- Инструменты для загрузки программы в микроконтроллер

В основе программы находится Java, которая позволяет работать на любых устройствах под управлением Windows, Mac OS и Linux. Необходимые для работы с устройствами библиотеки являются тоже бесплатными и находятся в открытом доступе в интернете.

Язык программирования Arduino называется Arduino C и является видоизмененным C++, имеет небольшие отличия от классического языка программирования, собирается и компилируется при помощи avr-gcc, с особенностями, облегчающими написание работающей программы. При компиляции программа автоматически сохраняется и создает временный файл формата \*.cpp.

Загрузка программы в плату Arduino происходит через предварительно запрограммированный загрузчик, который уже содержится на плате и во многом упрощает процесс загрузки кода в контроллер. Загрузчик создан на основе программатора Atmel AVR Application Note AN109. В зависимости от модели платы и внешних подключенных устройств, может загружать код через RS-232, USB или Ethernet. В некоторых платах из-за маленьких габаритов, такие варианты загрузки не подходят и необходимо использовать программатор или другую Arduino в роли программатора для загрузки кода.

Пользователь имеет возможность самостоятельно запрограммировать загрузчик к чистый контроллер. Для этого в программное обеспечение встроена поддержка популярных программаторов и сделано на проекте AVRDUDE.

Популярность, простота программирования и устройства вызвала большой спрос и как следствие стало появляться огромное количество сторонних программных решений. В основе этих решений лежит интеграция компилятора и загрузчика Arduino в имеющиеся оболочки для программистов. Среди популярных инструментов можно встретить такие как:

- Processing
- Eclipse
- Microsoft Visual Studio
- Atmel Studio
- Scratch for Arduino

Существуют как оригинальные платы, так и совместимые аналоги. Также существуют стандартные версии плат и уменьшенные. Среди оригинальных плат можно выделить наиболее популярные:

- Uno
- Pro
- Leonardo
- Mega 2560
- Due

Выпускаются платы уменьшенного размера в габарите DIP-корпусов. Они предназначены для установки в макетные платы и уменьшения провод для подключения.

К уменьшенным версиям относятся:

- Nano
- Nano Every
- Micro

Ко всем платам существуют дополнительные платы для увеличения функционала, чаще называются платы расширения или шилды расширения. Уменьшенные версии лишены плат расширения из-за малых габаритов.

Сторонние разработчики создают огромное количество аналогов, они имеют отличия во внешнем виде и расположение выводов на плате, оставляют только в основе архитектурную и программную совместимость. Самый маленький клон был выпущен компанией Fentoduino. Размер получившейся платы очень мал и составляет 15\*20 мм, включая разъем microUSB, стабилизатор напряжения и полный комплект ввода-вывода Arduino Uno. Эта же компания выпустила и самую богатую плату по набору и функционалу, является клоном Arduino Leonardo с поддержкой USB Host (клавиатура и мышь), Bluetooth 4 Low Energy, шестиосный гироскоп/акселерометр,

трехосный магнитометр (компас) и барометр. Размер получившегося устройства составляет 16\*40 мм.

Arduino используется не только для проверки мини-проектов или выполнения каких-то простых задач. Платы имеют достаточную мощность для выполнения более сложных задач. В интернете огромное количество устройств собранные на базе Arduino такие как: домашние метеостанции, будильник, наручные часы, радио и пр. По мимо простых устройств, можно встретить и более сложные начиная от 3D принтеров, фрезеров и лазеров, до полноценных станков на производстве. Для облегчения создания управляющих плат под производственные задачи, существуют модули позволяющие подключить клеммные колодки или установить в шкаф на Din-рейку. Подобные решения различные защиты от высокого напряжения и изолированными.

Официально компания не выпускает модули расширения для использования на производстве. На официальном сайте компании Arduino можно встретить адаптацию под производство. Такая продукция выпускается компанией Archiduino и Industrial Shields. Компании предлагают адаптацию под установку на Din-рейку, позволяющие установить дополнительные периферийные устройства. Торговая марка Controllino выпускает клон Arduino 2560 в адаптированной версии под производство с проводным Ethernet. Компания Norvi производит промышленные версии на процессорах AVR и ESP32.

К производству промышленных версии подключаются и крупные компании, специализирующиеся на промышленной автоматике. Компания AutomationDirect выпускает промышленные контроллеры и модули ввода-вывода, совместимые с программным обеспечением Arduino IDE и модулями расширения Arduino. Так же компания выпустила дополнения для стандартной программы Arduino IDE с графическим языком программирования и набором библиотек автоматике.

В классической линейке устройств Arduino в основном применяются микроконтроллеры Atmel AVR. Следующие МК можно встретить на указанных распространённых платах:

- **ATmega2560** (16 МГц, 256 Kb Flash, 8 Kb RAM, 54 порта, из них до 15 с ШИМ и 16 АЦП). Платы Mega.
- **ATmega32U4** (16 МГц, 32 Kb Flash, 2,5 Kb RAM, 20 портов, из них до 7 с ШИМ и 12 АЦП). Платы Leonardo, Micro, Yun.
- **ATmega328** (16 МГц, 32 Kb Flash, 2 Kb RAM, 14 портов, из них до 6 с ШИМ и 8 АЦП). Платы UnoR3, Mini, NanoR2, Pro, Pro mini, различные варианты плат uno и nano, такие как Wifi Uno и nano + nrf42101
- **ATtiny85** (20 МГц, 8 Kb Flash, 512 b RAM, 6 портов, из них 4 ШИМ и 4 аналоговых). Платы Digispark, также часто применяются вне плат.
- **ATmega168** (16 МГц, 16 Kb Flash, 1 Kb RAM, порты и распиновка аналогично ATmega328) Платы Uno R1, Uno R2, Pro mini, NanoR1.

В некоторых платах состав доступных портов и частота тактирования могут отличаться.

Со временем в разнообразие контроллеров стали появляться процессоры ARM. С 2020 года появились модули Portenta с ARM Cortex-M7 (STM32H747 @ 480 МГц).

Напряжение питания процессоров ARM на платах Arduino — 3,3 вольта. На такое же напряжение должны быть рассчитаны датчики для этих плат.

Сторонние разработчики внедрили в Arduino модуль популярного Wi-Fi микроконтроллера ESP8266 или клона ESP12. Удобство использования возрастает из-за отсутствия провода для загрузки кода и появлением возможности загрузки через Wi-Fi.

Все производимые аналоги Arduino с использованием модуля ESP8266 продаются под маркой Wemos. Имеется два размера, один в размере Arduino

Uno, полностью повторяющим расположение разъемов и поддержки шилдов расширения, второй размер является собственной разработкой уменьшенной версии и имеет название Wemos R1 и Wemos R2. Эти платы также возможно запрограммировать через программное обеспечение Arduino IDE, которое содержит встроенные библиотеки под эти платы.

В рамках сотрудничества со сторонними производителями в Arduino IDE была включена поддержка некоторых аппаратных средств Intel x86. Intel Galileo (процессор Intel Quark X1000 400 МГц), Intel Edison и Arduino 101 — Arduino-совместимые платы на Intel x86 архитектуре. Платы механически и электрически совместимы с периферийными платами Ардуино. Платы функционируют под собственной ОС Linux, поверх которой работает приложение, позволяющее загружать и исполнять скетчи Arduino.

#### 1.4. Обзор платформы для дистанционного курса

Существует огромное множество платформ для дистанционных занятий, но у всех есть ряд достоинств и недостатков. Проанализировав существующие решения выбор был остановлен на бесплатной платформе Google Класс. Платформа имеет версию для компьютеров и мобильных устройств, что является удобным для отслеживания присланных заданий и контролирования курса, вне зависимости от местоположения, нет привязки к компьютеру и рабочему месту. Наличие мобильной версии позволяет ответить на возникшие вопросы практически мгновенно.

Особенностью сервиса является привязка к Google аккаунту и отдельная папка на Google диске для каждого курса, можно зайти в папку конкретного ученика и посмотреть все сданные работы. Также система позволяет хранить все загруженные материалы для изучения в одном месте.

Система поддерживает различные системы оценки заданий. Есть возможность отслеживать успеваемость. Выполненные задания оцениваются и в случае ошибок, могут быть отправлены на изменения с комментариями от преподавателя. После оценивания, задания могут редактироваться только преподавателями, имеющими доступ к курсу.

Преподаватель может публиковать объявления в ленте, учащиеся могут их комментировать, задавать возникшие вопросы. Ученики могут отправлять сообщения на прямую преподавателю, в свою очередь преподавателю будет приходиться уведомление о сообщении и дополнительно сообщение об этом на почту, пропустить сообщение сложно. Из-за обширного функционала сервиса Google, есть возможность прикреплять видео с YouTube. Есть интеграция с почтовым сервисом Gmail. Доступ к курсу можно получить через интернет или присоединиться с помощью приложения на мобильном устройстве под управлением Android и iOS.



Система так же имеет ряд недостатков, но для начала обучения в дистанционном формате полностью подходит и закрывает все потребности преподавателя.

Каждый урок содержит в себе видеоматериал, задание и дополнительная литература для выполнения задания. Для входа на курс необходимо иметь Google аккаунт.

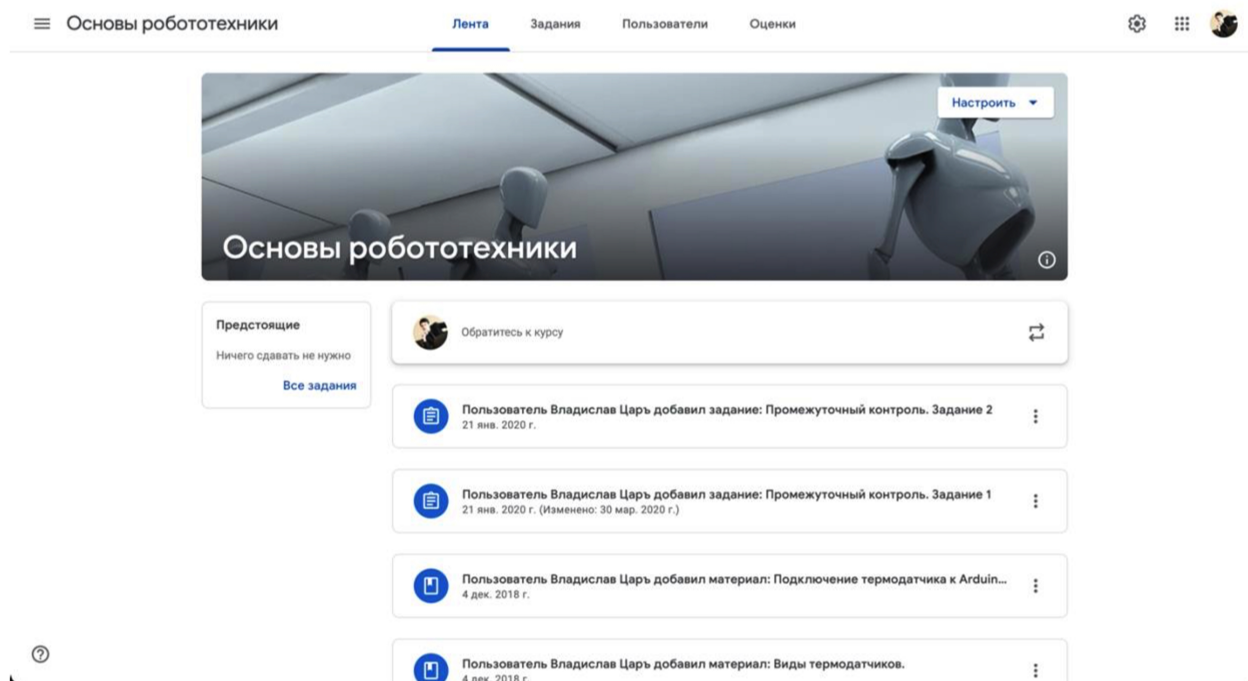


Рис.1

Основная страница, на которую попадает слушатель курса после присоединения. Главная страница состоит из нескольких вкладок.

Вкладка “лента” содержит последние добавленные материалы, а также комментарии и вопросы слушателей. В левой части отражается срок сдачи того или иного задания.

Вкладка “задание” содержит основную информацию по всем заданиям курса, а также весь материал для изучения и задания для промежуточного контроля. На этой вкладке можно посмотреть сроки сдачи того или иного задания.

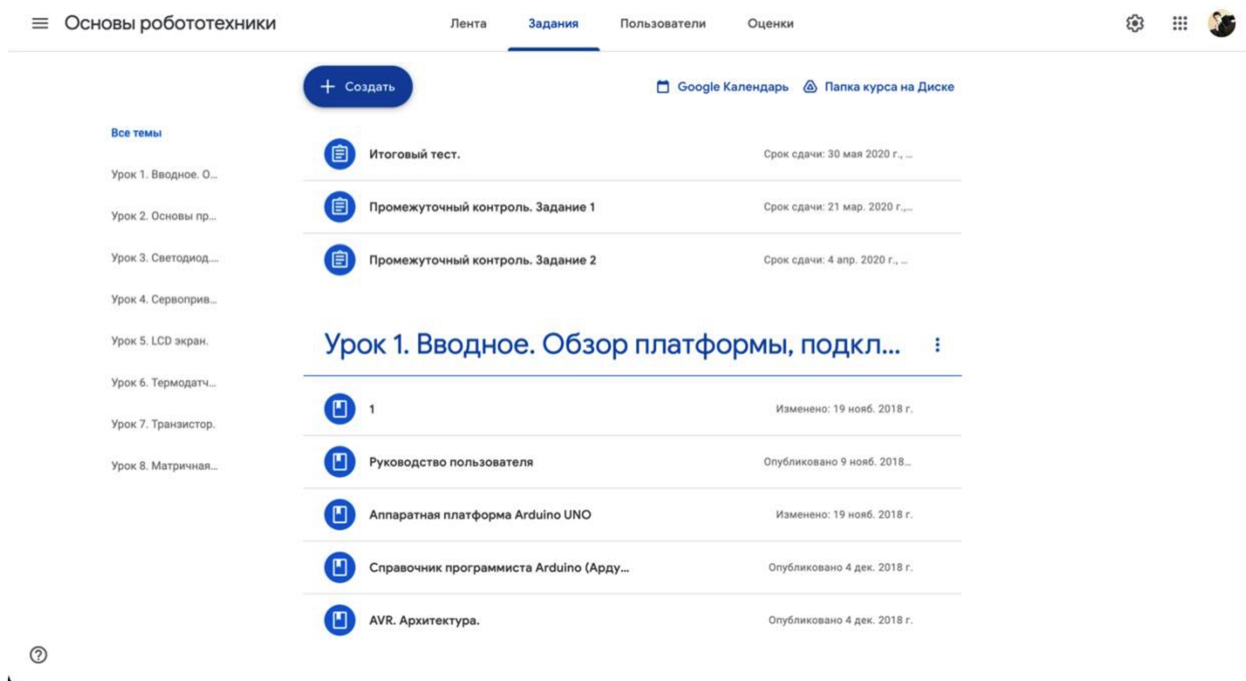


Рис.2

Вкладка “Пользователи” отображает преподавателей, у которых есть доступ к этому курсу и всех учащихся.

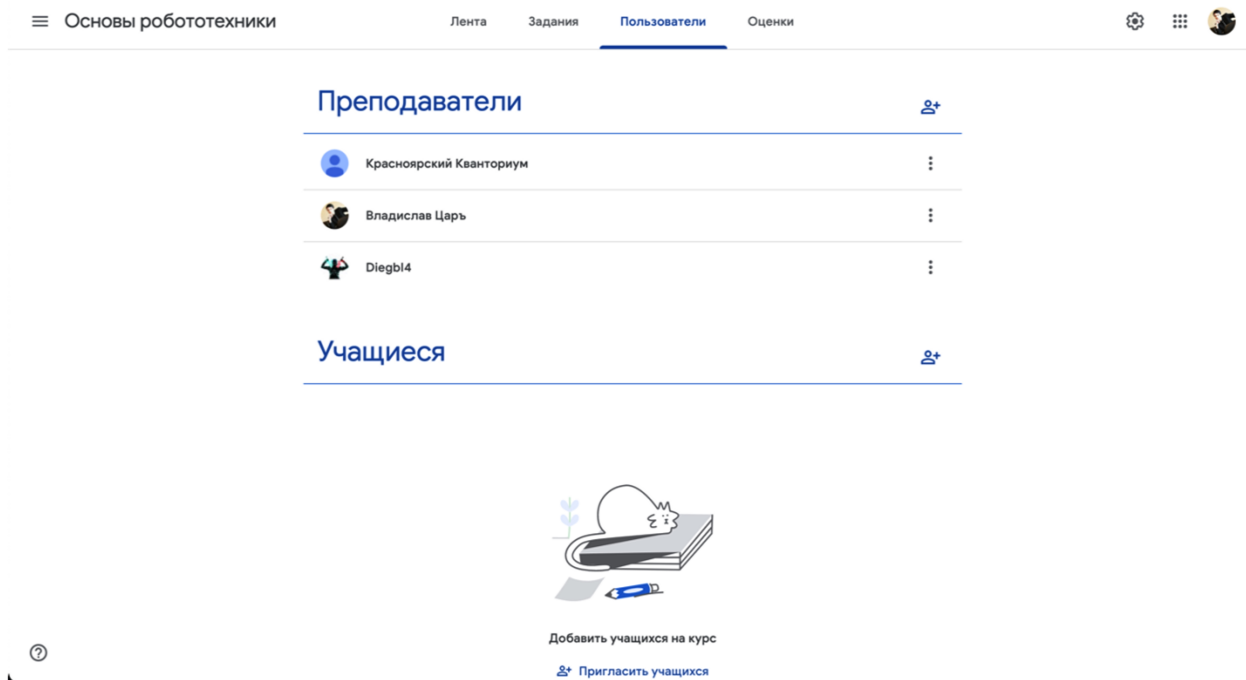


Рис.3

Вкладка “Оценки” выполняет роль журнала. Оценки или баллы за выполнения задания могут выставляться двумя способами, в зависимости от

выставленных настроек и типа заданий. Первый способ, ручное выставление оценки за урок или тестирование. Второй способ более интересный и позволяет выставлять оценки автоматически. Второй способ не всегда является эффективным и возникает ряд неполадок в процессе в проверки работ, так как оценивается только полученный результат и не проверяется способ получения этого ответа, что является не маловажным фактором в создании схем и программного кода.

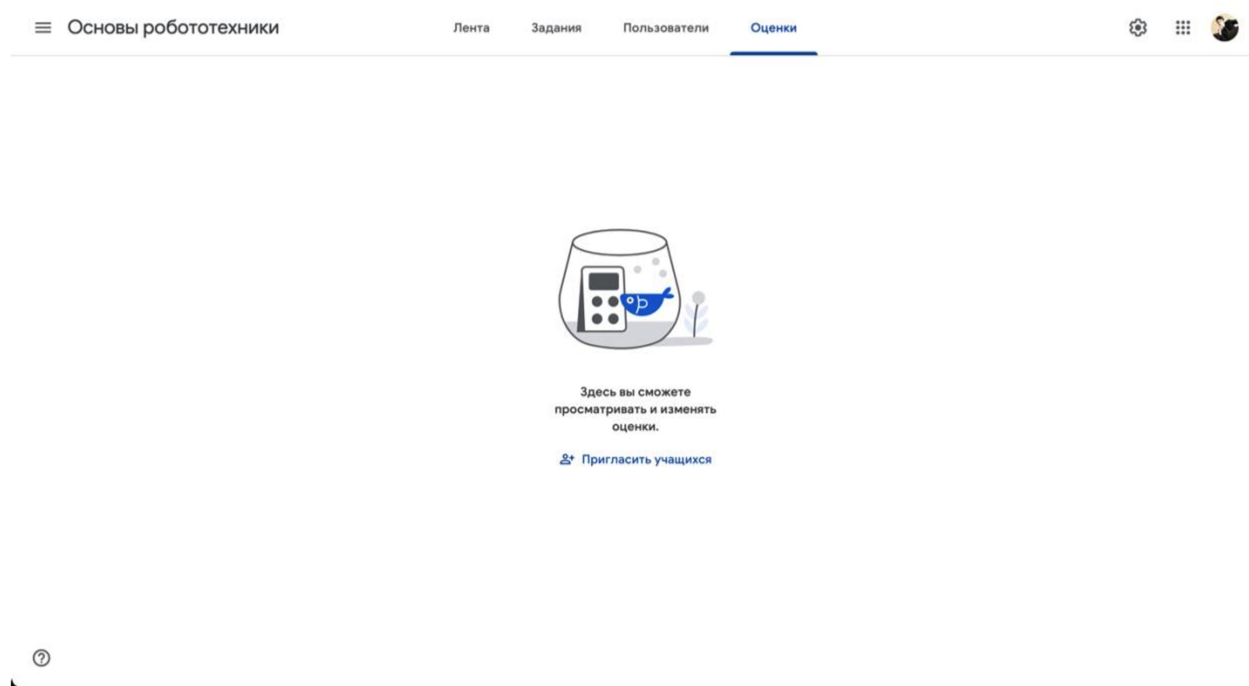


Рис.4

Перейдем к более детальному рассмотрению представленного материала на примере первого урока.

## Урок 1. Вводное. Обзор платформы, подкл... ⋮

---






 1	Изменено: 19 нояб. 2018 г.
 Руководство пользователя	Опубликовано 9 нояб. 2018...
 Аппаратная платформа Arduino UNO	Изменено: 19 нояб. 2018 г.
 Справочник программиста Arduino (Арду...	Опубликовано 4 дек. 2018 г.
 AVR. Архитектура.	Опубликовано 4 дек. 2018 г.

Рис.5

В данном уроке содержится несколько типов представленного материала. Первым и основным материалом, это является видео урок, в котором рассказывается о плате, с которой предстоит работать и немного истории создания и массового распространения. Так же в уроке содержится материал для самостоятельного изучения в виде ссылок на ресурсы сети интернет.

## Выводы по первой главе

Анализ истории появления электорники, микроэлектроники и литературы по данным темам, можно сказать, что развитие происходит колоссальными темпами. На базе простой элементной базы можно создать сложные и многозадачные устройства. Все современные сложные устройства можно разделить на простые задачи и проследить как они работают. Твердотельная элементная база продолжает активно развиваться и уже сейчас можно с легкостью выполнять задачи, которые еще десять или 15 лет назад были невыполнимые.

На примере одной платы разработчика можно создать сложные устройства и механизмы, позволяющие автоматизировать повторяющиеся действия и максимально упростить жизнь человека.

Самой популярной платой для разработчика в данный момент является Arduino, в большинстве своем из-за простоты программирования и низкой стоимости, любой желающий может присоединиться к программированию и создавать свои проекты.

## ГЛАВА 2. ДИСТАНЦИОННЫЙ ФОРМАТ ПРОВЕДЕНИЯ ЗАНЯТИЙ ПО ПРЕДМЕТУ ТЕХНОЛОГИЙ

### 2.1 Методические рекомендации по планированию и проведению занятий в дистанционном формате.

Практические задания дистанционного курса для школьников и студентов является дополнительной общеобразовательной программой «Основы робототехники», реализуемой в детском технопарке «Кванториум». Основа для программы была использована из общеобразовательной программы для первого года обучения учеников технопарка, которая была адаптирована под более старший возраст. Программа реализуется в рамках уроков технологии в районах Красноярского края. Образовательная программа разработана авторами Михайловым В.С. и Сапичевым В.В.

Дополнительная общеобразовательная общеразвивающая программа «Основы робототехники» (далее - программа) имеет техническую направленность и ориентирована на обучающихся 12-22 лет. Учебные занятия проходят по заочной форме обучения с применением дистанционных образовательных технологий на сайте Google Class. Учебные темы доступны для обучающихся в соответствии с учебным графиком.

Все задания программы выполняются с помощью персонального компьютера и необходимых программных средств (Arduino IDE).

Целью реализации программы является формирование начальных основ робототехники и программирования на примере датчиков и вывода информации.

Программа позволяет решить следующие задачи:

- Сформировать принципы работы робототехнических элементов, представление о состояниях и перспективах робототехники в настоящее время.

- Сформировать представление о правилах безопасного поведения при работе с электротехникой, инструментами, необходимыми при конструировании робототехнических моделей.
- Сформировать интерес к техническим знаниям, развить у обучающихся техническое и тактическое мышление, изобретательность.
- Развивать способности осознанно ставить перед собой конкретные задачи, разбивать их на отдельные этапы и добиваться их выполнения.

Отличительной особенностью программы является основанность на реальных примерах использования автоматизированных устройствах.

В рамках программы, обучающиеся произведут сборку и программирования простейших электронных компонентов, а также разработают собственные решения. Кроме того, участники получают ценные навыки поиска и анализа информации.

Учебные занятия проходят по заочной форме обучения с применением дистанционных образовательных технологий на сайте Google Class. Учебные темы доступны для обучающихся в соответствии с учебным графиком. Объем программы 30 учебных часов в год, режим занятий - 1 раза в неделю по 2 учебных часа.

Обучение по программе определяется как самостоятельная учебная деятельность с использованием дистанционных образовательных технологий, при этом взаимодействие с обучающимися производится в образовательном процессе в следующих режимах:

online, используя средства коммуникации и одновременно взаимодействуя друг с другом;

offline, когда обучающийся выполняет какую-либо самостоятельную работу, а педагог оценивает правильность ее выполнения и дает рекомендации.

Занятия проводятся в форме электронных лекций, семинаров, консультаций (индивидуальные и групповые), контроля (тестирование).

Программа предусматривает проведение итоговой и промежуточной аттестации обучающихся. Форма, порядок и периодичность аттестации обучающихся определяются учебно-тематическим планом. Способы отслеживания результатов, параметры и критерии оценивания результатов освоения содержания определены в описании программы.

Контроль качества усвоения программы осуществляется в форме выполнения тестов, контрольных работ. Педагогом реализуется рейтинговая система оценки образовательных результатов в баллах (получение зачета о прохождении программы).

### **Образовательные результаты**

- написание простейших кодов, объединение в сложные алгоритмы.
- способность обучающегося принимать и сохранять учебную цель и задачи;
- умение планировать собственную деятельность в соответствии с поставленной задачей и условиями её реализации;
- умение корректировать свои действия на основе полученной оценки;
- анализировать характер ошибок и исправлять их;
- способность к самообразованию;
- умение осуществлять информационный поиск.

Итоговый контроль освоения образовательной программы осуществляется через написание итогового теста по всей программе.



Учебно-тематический план

Таблица 1

№ п/п	Наименование тем	Общее количество о часов	В том числе:	
			Теорети- ческих	Практи- ческих
1.	Презентация программы. Ответы на вопросы. Набор группы	1	1	0
2.	Вводное занятие. Обзор платформы Arduino. Подключение к компьютеру.	3	2	1
3.	Основы программирования. Основные команды для начала программирования.	4	2	2
4.	Светодиод. RGB светодиод. Подключение и управление.	3	2	1
5.	Сервопривод. Виды, отличия. Управление.	3	2	1
6.	Промежуточный контроль.	1		1
7.	LCD экран. Подключение, вывод значений.	3	2	1
8.	Термодатчик. Виды, управление. Вывода значений в монитор порта.	3	2	1
9.	Промежуточный контроль.	1		1
10.	Транзистор. Виды, подключение.	3	2	1
11.	Матричная клавиатура. Подключение, управление.	4	2	2
12.	Итоговый тест.	1	0	1
<b>Итого часов:</b>		<b>30</b>	<b>17</b>	<b>13</b>

## Содержание образовательной программы

Таблица 2

<p>1. Презентация программы. Ответы на вопросы. Набор группы.</p>	<p>Ознакомление с программой Ознакомление с платформой Google class</p>
<p>2. Вводное занятие. Обзор платформы Arduino. Подключение к компьютеру.</p>	<p>История создания и использование платформы Arduino Основные моменты при работе с платформой Arduino</p>
<p>3. Основы программирования. Основные команды для начала программирования.</p>	<p>Вводное представление об языке программирования Представление среды программирования, программы ArduinoIDE</p>
<p>4. Светодиод. RGB светодиод. Подключение и управление.</p>	<p>Схема сборки светодиода Используемые элементы Особенности подключения светодиода Написание и объяснение кода для зажигания светодиода Проверка работоспособности</p>
<p>5. Сервопривод. Виды, отличия. Управление.</p>	<p>Особенности сервоприводов, их отличия от других двигателей Особенности подключения и управления Принцип и способы управления Написание кода Проверка работоспособности</p>

<p>6. Промежуточный контроль. Написание программы для сервопривода и светодиода.</p>	<p>Проверка усвоенного материала Задача объединить два пройденных урока и выполнить поставленную задачу Написание кода Проверка работоспособности</p>
<p>7. LCD экран. Подключение, вывод значений.</p>	<p>Виды и типы устройств вывода информации Особенности подключения Написание кода Проверка работоспособности</p>
<p>8. Термодатчик. Виды, управление. Подключение вывода значений в монитор порта.</p>	<p>Устройства ввода информации Виды устройств, позволяющих получить информацию из окружающего мира Написание кода Проверка работоспособности</p>
<p>9. Промежуточный контроль. Написание кода вывод температуры на экран.</p>	<p>Проверка усвоенного материала Задача объединить два пройденных урока и выполнить поставленную задачу Написание кода Проверка работоспособности</p>
<p>10. Транзистор. Виды, подключение.</p>	<p>Виды и назначения транзисторов Применение в электротехнике Особенности подключения и управления Написание кода Проверка работоспособности</p>

11. Матричная клавиатура. Подключение, управление.	Устройство ввода информации Какие существуют виды клавиатур и их особенности Подключение Написание кода Проверка работоспособности
12. Итоговый тест.	Проверка усвоенного материала Задача объединить четыре пройденных элемента и создать собственное устройство Описать его назначение Написание кода Проверка работоспособности

**Планируемые результаты освоения практических заданий**

Таблица 3

Задание	Содержание	Планируемые результаты
1. Презентация программы. Ответы на вопросы. Набор группы.	Ознакомление с программой Ознакомление с платформой Google class	- присоединение к курсу - получение заданий
2. Вводное занятие. Обзор платформы Arduino. Подключение к компьютеру.	История создания и использование платформы Arduino Основные моменты при работе с платформой Arduino	- установка программного обеспечения - научиться различать лицевую панель

<p>3. Основы программирования. Основные команды для начала программирования.</p>	<p>Вводное представление об языке программирования Представление среды программирования, программы ArduinoIDE</p>	<p>- выучить структуру программы - освоить и выучить основные команды</p>
<p>4. Светодиод. RGB светодиод. Подключение и управление.</p>	<p>Схема сборки светодиода Используемые элементы Особенности подключения светодиода Написание и объяснение кода для зажигания светодиода Проверка работоспособности</p>	<p>- сборка схемы - написание кода - проверка результатов - выполнение и отправка задания</p>
<p>5. Сервопривод. Виды, отличия. Управление.</p>	<p>Особенности сервоприводов, их отличия от других двигателей Особенности подключения и управления Принцип и способы управления Написание кода Проверка работоспособности</p>	<p>- сборка схемы - написание кода - проверка результатов - выполнение и отправка задания</p>
<p>6. Промежуточный контроль. Написание программы для сервопривода и светодиода.</p>	<p>Проверка усвоенного материала Задача объединить два пройденных урока и выполнить поставленную задачу Написание кода Проверка работоспособности</p>	<p>- выполнение и отправка задания</p>

<p>7. LCD экран. Подключение, вывод значений.</p>	<p>Виды и типы устройств вывода информации Особенности подключения Написание кода Проверка работоспособности</p>	<p>- сборка схемы - написание кода - загрузка и проверка работы - выполнение задания</p>
<p>8. Термодатчик. Виды, управление. Подключение вывода значений в монитор порта.</p>	<p>Устройства ввода информации Виды устройств, позволяющих получить информацию из окружающего мира Написание кода Проверка работоспособности</p>	<p>- сборка схемы - написание кода - загрузка и проверка работы - выполнение задания</p>
<p>9. Промежуточный контроль. Написание кода вывод температуры на экран.</p>	<p>Проверка усвоенного материала Задача объединить два пройденных урока и выполнить поставленную задачу Написание кода Проверка работоспособности</p>	<p>- выполнение и отправка задания</p>
<p>10. Транзистор. Виды, подключение.</p>	<p>Виды и назначения транзисторов Применение в электротехнике Особенности подключения и управления Написание кода Проверка работоспособности</p>	<p>- сборка схемы - написание кода - загрузка и проверка работы - выполнение задания</p>

<p>11. Матричная клавиатура. Подключение, управление.</p>	<p>Устройство ввода информации Какие существуют виды клавиатур и их особенности Подключение Написание кода Проверка работоспособности</p>	<p>- сборка схемы - написание кода - загрузка и проверка работы - выполнение задания</p>
<p>12. Итоговый тест.</p>	<p>Проверка усвоенного материала Задача объединить четыре пройденных элемента и создать собственное устройство Описать его назначение Написание кода Проверка работоспособности</p>	<p>- выполнение и отправка задания</p>

### 2.3. Проектирование практических заданий

#### 1. Презентация программы. Ответы на вопросы. Набор группы.

Цель: познакомиться с учащимися, рассказать о программе и проведении обучения.

Теоретическая часть: данный урок имеет формат онлайн и при необходимости проходит в любой программе для видеоконференций. Необходимо рассказать о заданиях, как происходит прием и оценивание работ. Установка дат и времени для консультаций. Ответ на возникшие вопросы.

Практическая часть: присоединение к куру. При необходимости установка и настройка видеоконференции.

#### 2. Вводное занятие. Обзор платформы Arduino. Подключение к компьютеру.

Цель: показать платформу для программирования.

Теоретическая часть:

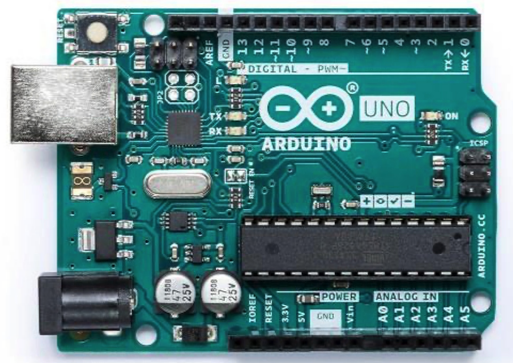


Рис.6

Платформа имеет 14 цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку



перезагрузки. Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи.

Обеспечить питанием плату можно путем подключения USB к компьютеру и внешнему питанию 5 Вольт (в зависимости от производителя платы, разъем может отличаться, чаще это microUSB или USB B). Так же внешнее питание может подаваться через специальный разъем, посредством разъема 2.1 мм с центральным положительным полюсом.

При подключении внешнего питания, плата может работать в диапазоне напряжения от 6 до 20 Вольт. Рекомендуемое внешнее напряжение от 7 до 12 Вольт.

Габариты платы Arduino Uno составляют 6,9 на 5,3 сантиметра. Разъем USB и разъем питания выходят за пределы платы. Отверстия в плате позволяют закрепить на поверхности.

### 3. Основы программирования. Основные команды для начала программирования.

Цель: показать платформу для программирования, инструменты для предварительной настройки и программирования. Основные команды программирования, структура программы.

Теоретическая часть:

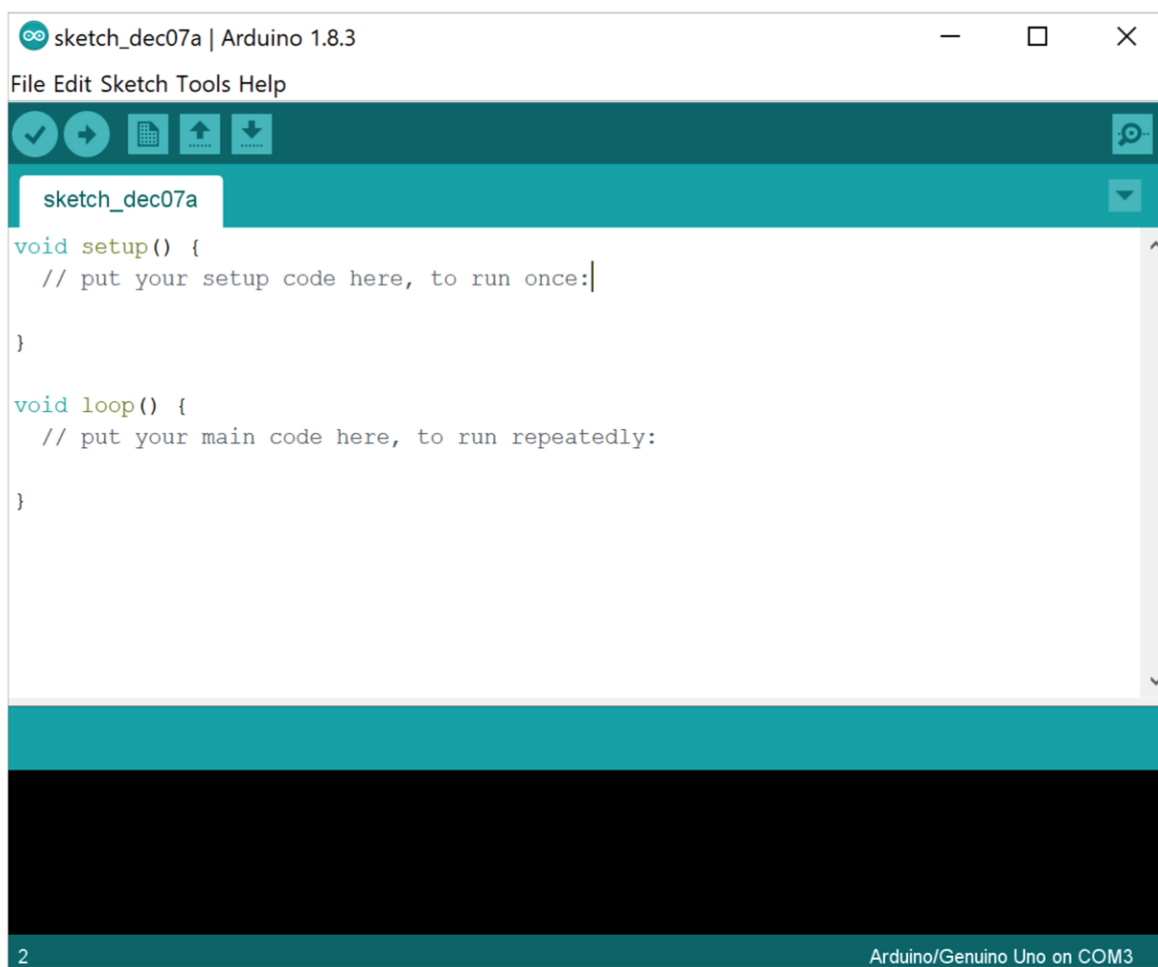


Рис.7

Программа Arduino IDE состоит из:

- Текстовый редактор программного кода
- Области сообщений
- Панель инструментов с часто используемыми командами

Код, написанный в программе, называется скетч. Код пишется как в обычном блокноте. Во время компиляции кода или сохранения в области сообщений появляются сообщения об успешном сохранении и ошибках в коде (программа проверяет синтаксис кода, но не правильность написания самого кода). Панель инструментов позволяет проверить или загрузить программу, открыть новый код или ранее сохраненный, сохранить текущую программу.



*Verify/Compile*

Проверка программного кода на ошибки, компиляция.



*Stop*

Остановка мониторинга последовательной шины (Serial monitor) или отображение других окон.



*New*

Создание нового скетча.



*Open*

Открытие меню доступа ко всем скетчам в библиотеке. Открывается нажатием в текущем окне.



*Save*

Сохранение скетча.



*Upload to I/O Board*

Компилирует программный код и загружает его в устройство Arduino. Описание загрузки приведено ниже.



*Serial Monitor*

Открытие мониторинга последовательной шины (Serial monitor).

Меню программы: File, Edit, Sketch, Tools, Help. Доступ к некоторым функциям может быть ограничен, в зависимости от выполняемой программы.

## Edit

- Copy for Discourse  
Копирует в буфер обмена подходящий для размещения на форуме код скетча с выделением синтаксиса.
- Copy as HTML  
Копирует код скетча в буфер обмена как HTML код, для размещения на веб-страницах.

## Sketch

- Verify/Compile  
Проверка скетча на ошибки.
- Import Library  
Добавляет библиотеку в текущий скетч, вставляя директиву #include в код скетча. Подробная информация в описании библиотек ниже (Libraries).
- Show Sketch Folder  
Открывает папку, содержащую файл скетча, на рабочем столе.
- Add File...  
Добавляет файл в скетч (файл будет скопирован из текущего места расположения). Новый файл появляется в новой закладке в окне скетча. Файл может быть удален из скетча при помощи меню закладок.

## Tools

- Auto Format  
 Данная опция оптимизирует код, например, выстраивает в одну линию по вертикали открывающую и закрывающую скобки и помещает между ними утверждение.
- Board  
 Выбор используемой платформы. Список с описанием платформ приводится ниже.
- Serial Port  
 Меню содержит список последовательных устройств передачи данных (реальных и виртуальных) на компьютере. Список обновляется автоматически каждый раз при открытии меню Tools.
- Burn Bootloader  
 Пункты данного меню позволяют записать Загрузчик (Bootloader) в микроконтроллер на платформе Arduino. Данное действие не требуется в текущей работе с Arduino, но пригодится, если имеется новый ATmega (без загрузчика). Перед записью рекомендуется проверить правильность выбора платформы из меню. При использовании AVR ISP необходимо выбрать соответствующий программатору порт из меню Serial Port.

Основная программа состоит из следующих функций:

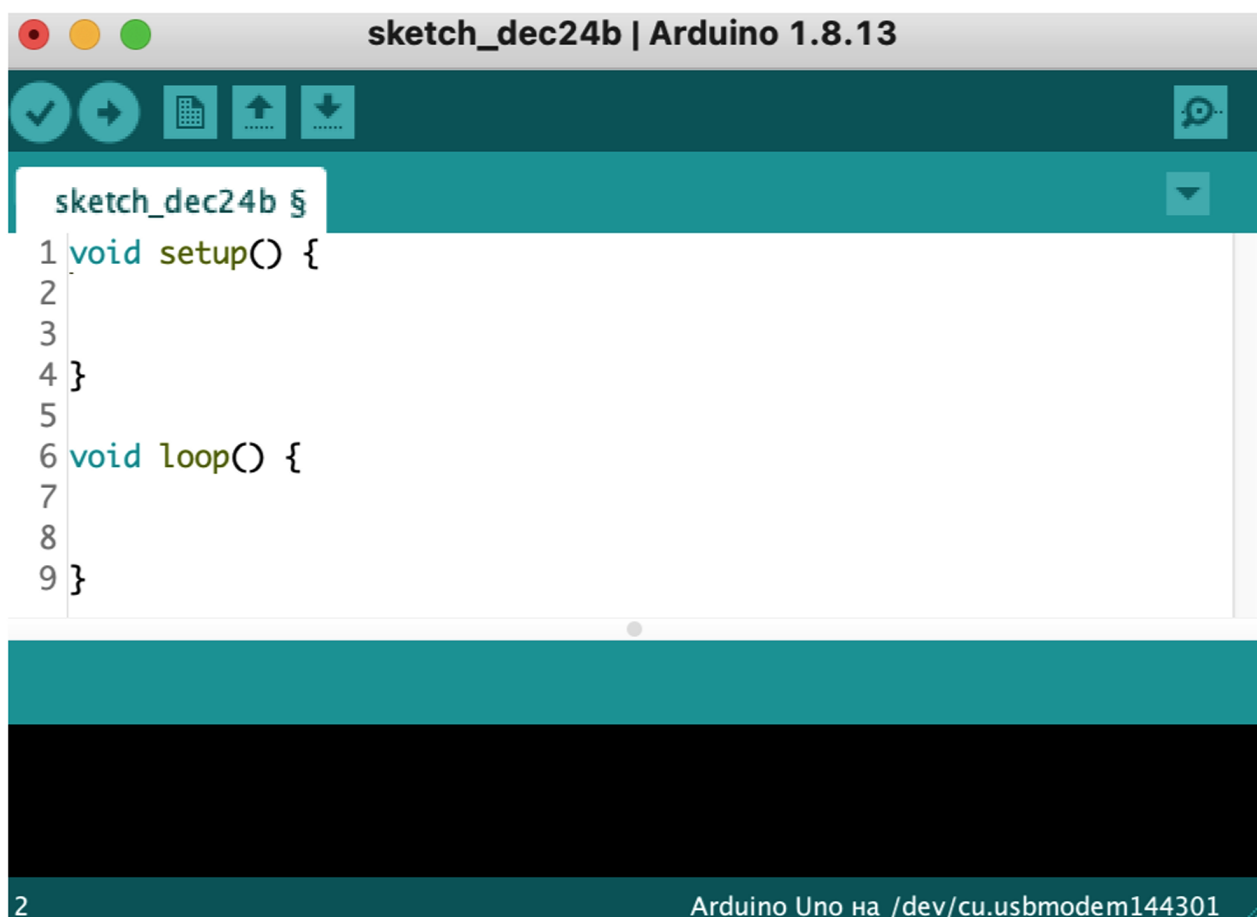


Рис.8

“void setup ()” - первоначальной настройки, здесь прописываются все переменные, задается режим работы, подключение библиотек и т.д. Функция выполняется один раз во время подачи напряжения или сброса платы физической кнопкой на плате “Reset”.

“void loop ()” – основная часть программы, по-другому называется “цикл”. В ней происходят первоначальные установки значений и инициализация, после выполняется бесконечное количество раз.

Список основных команд в Приложении 1.

Практическая часть: Установка и настройка программного обеспечения.

#### 4. Светодиод. Подключение и управление.

Цель: собрать схему со светодиодом, написать управляющий код, загрузка и проверка работоспособности кода.

Теоретическая часть: для успешного выполнения урока необходимы следующие элементы:

- Плата Arduino UNO
- Одноцветные светодиоды
- Макетная плата
- Макетные провода
- Резисторы (номиналом 240 Ом и стандартного ряда резисторов)

Практическая часть:

Необходимо собрать схему на макетной плате следующим образом:

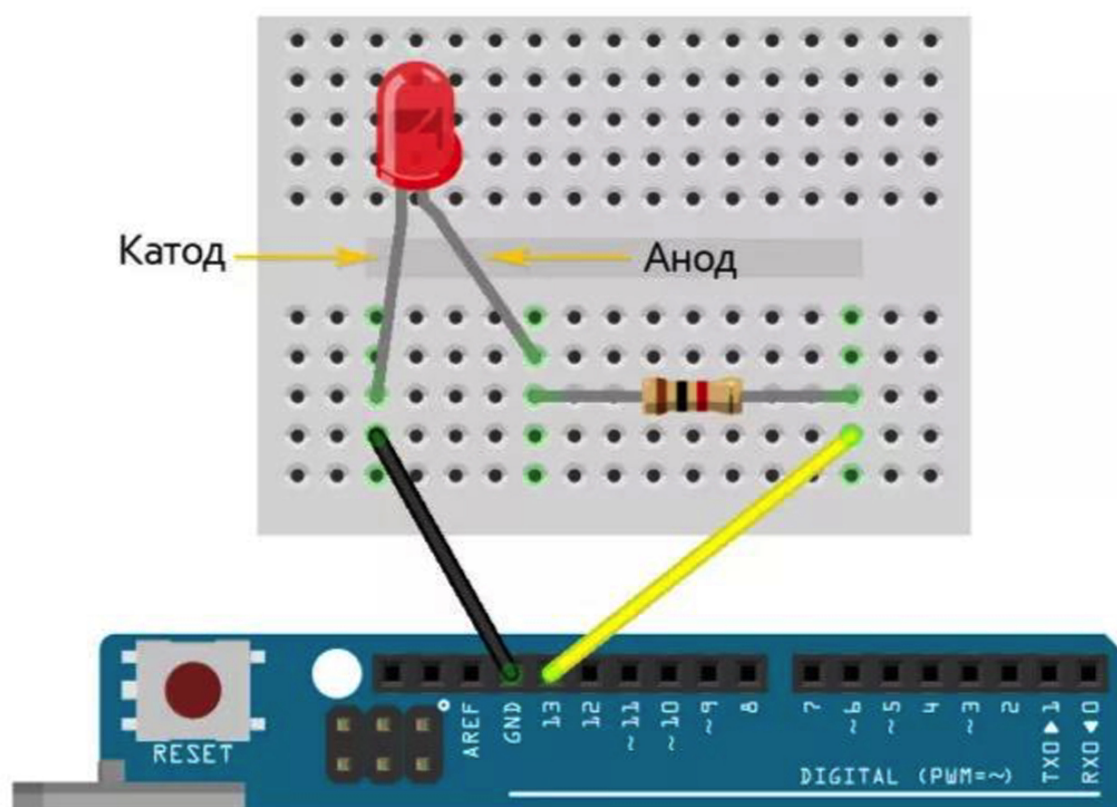


Рис.9

Светодиод имеет полярность. Длинная ножка является “+” и называется анод, ее необходимо подключить к резистору, вторая ножка является “-” и ее следует подключить с помощью макетных проводов к плате Arduino (GND). При несоблюдении полярности светодиод гореть не будет. У резистора

полярности нет и его можно подключать любой стороной, одна ножка светодиода подключается к светодиоду, свободная ножка через резистор подключается к цифровой ножке на плате Arduino (Pin13).

После успешной сборки схемы можно перейти к написанию управляющего кода. После символа “//” идут комментарии, которые служат для пояснения кода и не участвуют в основной программе.

```
int led = 13; // создаем переменную led присваиваем ей значение 13, эта
переменная хранит номер порта к которому подключен светодиод
void setup() {
  pinMode(led, OUTPUT);    // Устанавливаем порт 13 (Led) как исходящий
}
void loop() { // Этот цикл будет крутиться бесконечно
  digitalWrite(led, HIGH); // Присваиваем с помощью оператора digitalWrite
порту led (13) значение HIGH, это значит на пин 13 ардуина подаст 5В.
  delay(1000);           // пауза 1 секунда
  digitalWrite(led, LOW); // Присваиваем с помощью оператора digitalWrite
порту led (13) значение LOW, это значит на пин 13 ардуина подаст 0В.
  delay(1000);           // Еще одна секунда и цикл loop запустится с самого
начала
}
```

Задание: Написать программу, в которой будем 3 мигания светодиода с разной частотой мигания.

## 5. Сервопривод. Виды, отличия. Управление.

Цель: Подключение сервопривода, написать управляющий код, загрузка и проверка работоспособности кода.

Теоретическая часть:

Сервоприводы Arduino — это умные устройства. Используя только один входной пин, они получают значения для позиционирования от



микроконтроллера и переходят в это положение. Как можно увидеть на рисунке в самом начале статьи внутри они имеют двигатель и цепь обратной связи, которая гарантирует, что вал/рычаг сервопривода достигнет желаемого положения.

Но какой сигнал сервомоторы получают на входе? Это прямоугольная волна, подобная PWM (англ. - pulse-width modulation, широтно-импульсная модуляция). Каждый цикл в сигнале длится 20 миллисекунд, и большая часть времени в значении LOW. В начале каждого цикла значение сигнала становится HIGH на время от 1 до 2 миллисекунд.

При 1 миллисекунде она составляет 0 градусов, а при 2 миллисекундах - 180 градусов, а в промежутке значение от 0 до 180. Это очень хороший и надежный метод. График вьше упрощает понимание.



Рис.10

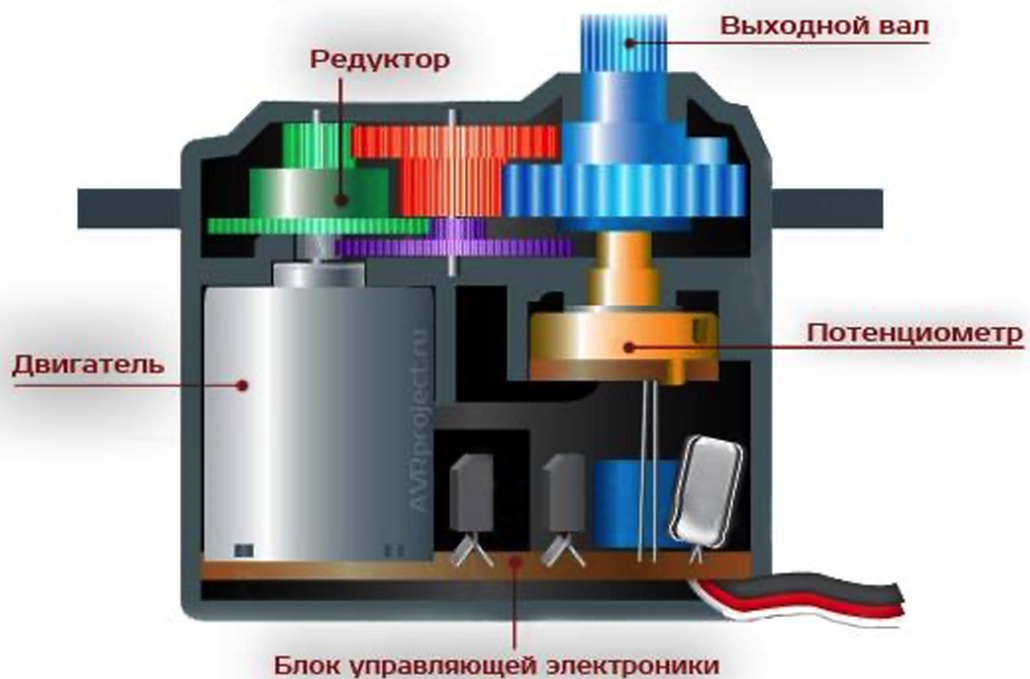


Рис.11

Для успешного выполнения урока необходимы следующие элементы:

- Плата Arduino UNO
- Сервопривод
- Макетные провода

Практическая часть:

Необходимо собрать схему следующим образом:

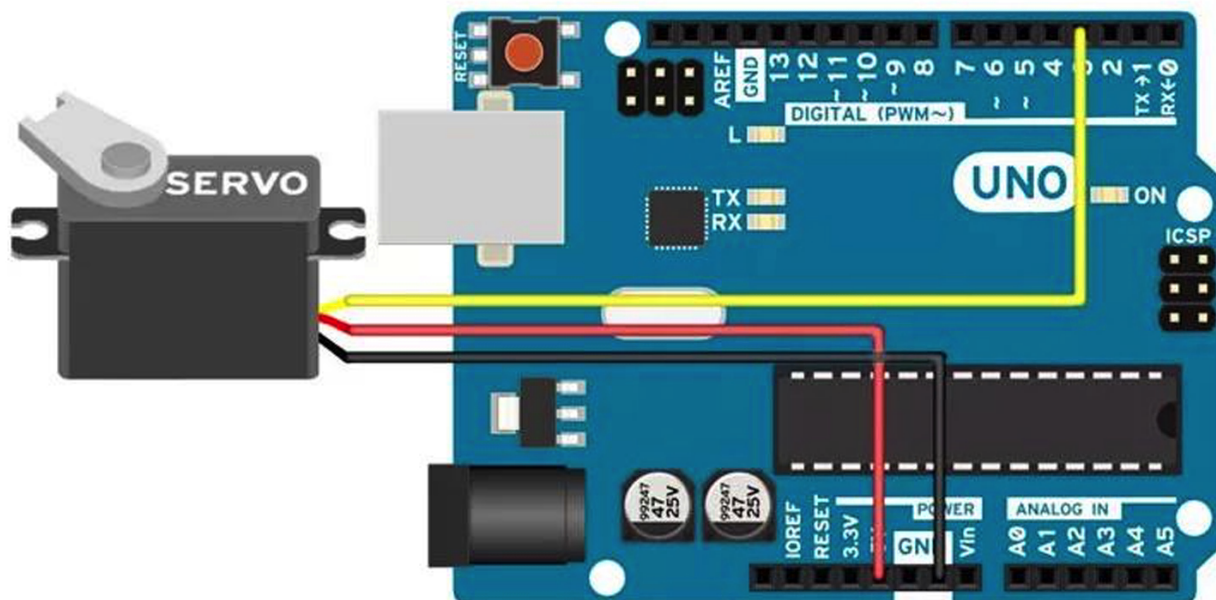


Рис.12

Сервопривод имеет 3 провода подключения, черный провод – это минус и подключается на плате к разъему GND, красный провод – это питающий провод и необходимо подключить к разъему 5Вольт, желтый провод – управляющий провод, необходимо подключить к цифровому разъему на плате, в нашем случае это Pin 3.

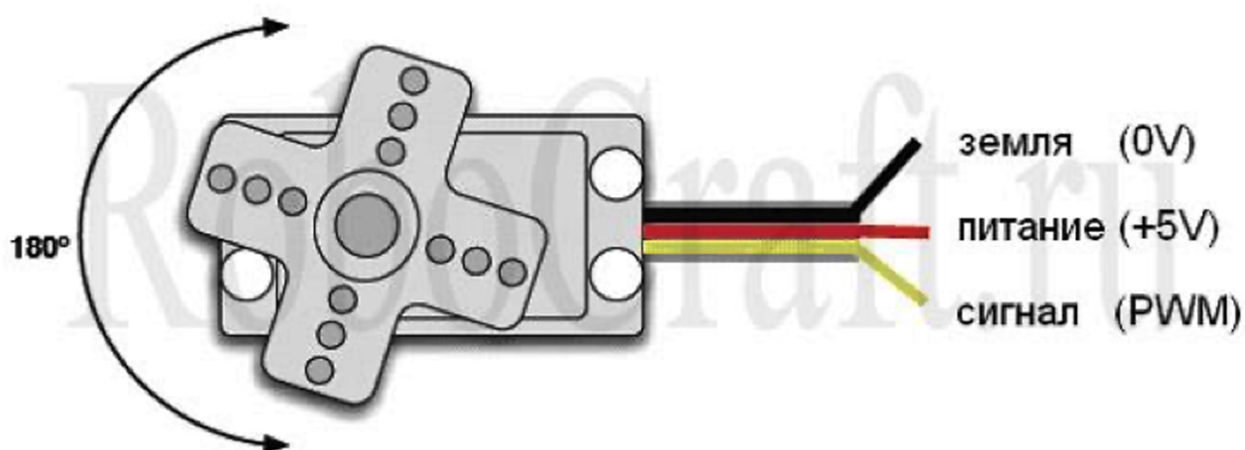


Рис.13

Код для управления сервоприводом:

```
#include <Servo.h> // подключаем библиотеку для работы с сервоприводом
```

```

Servo servo1; // объявляем переменную servo типа "servo1"

void setup() {
    servo1.attach(3); // привязываем сервопривод к цифровому входу 3
}

void loop() {
    servo1.write(0); // ставим угол поворота под 0
    delay(2000); // ждем 2 секунды
    servo1.write(90); // ставим угол поворота под 90
    delay(2000); // ждем 2 секунды
    servo1.write(180); // ставим угол поворота под 180
    delay(2000); // ждем 2 секунды
}

```

Задание: написать программу, в которой сервопривод будет поворачиваться на 8 углов.

6. Промежуточный контроль. Написание программы для сервопривода и светодиода.

Цель: Самостоятельная работа, проверка усвоенного материала.

Теоретическая часть: Объяснение задачи.

Практическая часть:

Объединение двух предыдущих урока, необходимо написать код, который будет выполнять следующее:

Сервопривод должен поворачиваться на 4 угла сервопривода, при этом, должен загораться один из четырех светодиодов.

## 7. LCD экран. Подключение, вывод значений.

Цель: Подключение источника вывода информации.

Теоретическая часть: Взаимодействие символьного Ардуино дисплея добавляет приятный элемент читаемости в проект. Многие из лучших проектов по всему миру демонстрируют спортивные дисплеи.

Эти ЖК-дисплеи используются для отображения информации от микроконтроллера или любого подключенного к нему датчика. Например, можно создать систему **контроля температуры**, которая отображает данный показатель на Arduino. Можно сконструировать собственный спидометр, который отображает скорость на экране.

Display для Arduino – это одно из самых простых устройств, которое можно использовать для отображения результатов проектов. Однако есть два различных типа данного устройства: графические и символьные. В этой статье используется персональный экран, так как он легче всего работает. Кроме того, в зависимости от размера экрана, существуют разные типы:

16x02 - 16 колонок и 2 строки

20x04 – 20 колонок и 4 строки

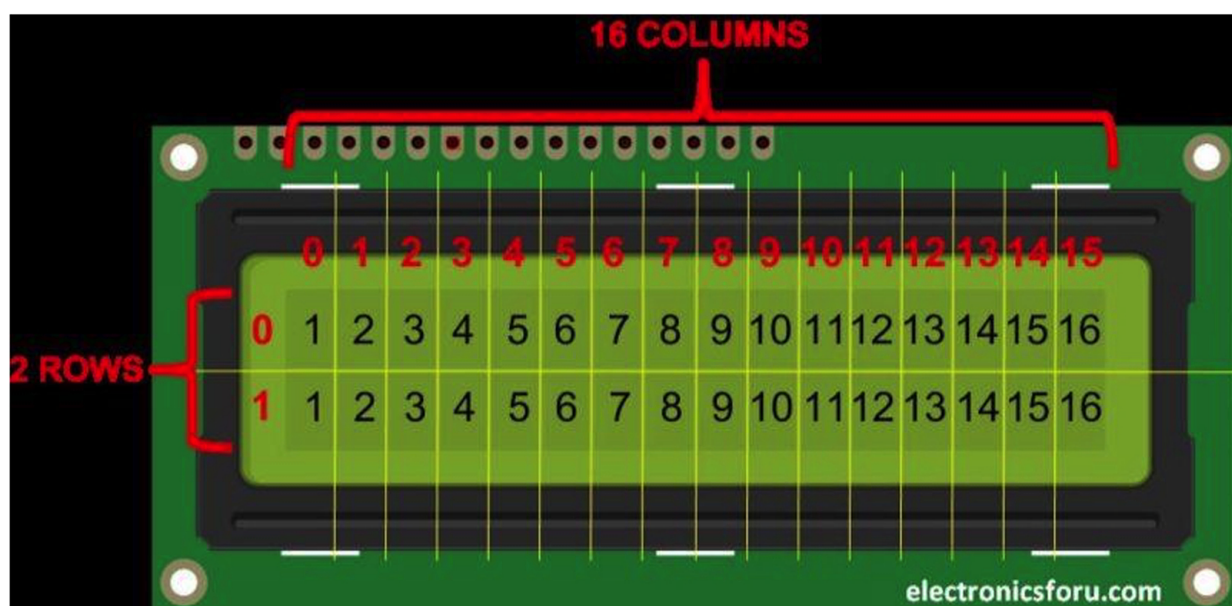


Рис.14

Практическая часть:

В этом уроке используются аналоговые порты, которые выполняют еще вторую функцию. Дисплей подключается по протоколу I2C, на плате Arduino UNO они находятся на портах A4 и A5. Подключим дисплей согласно схеме:

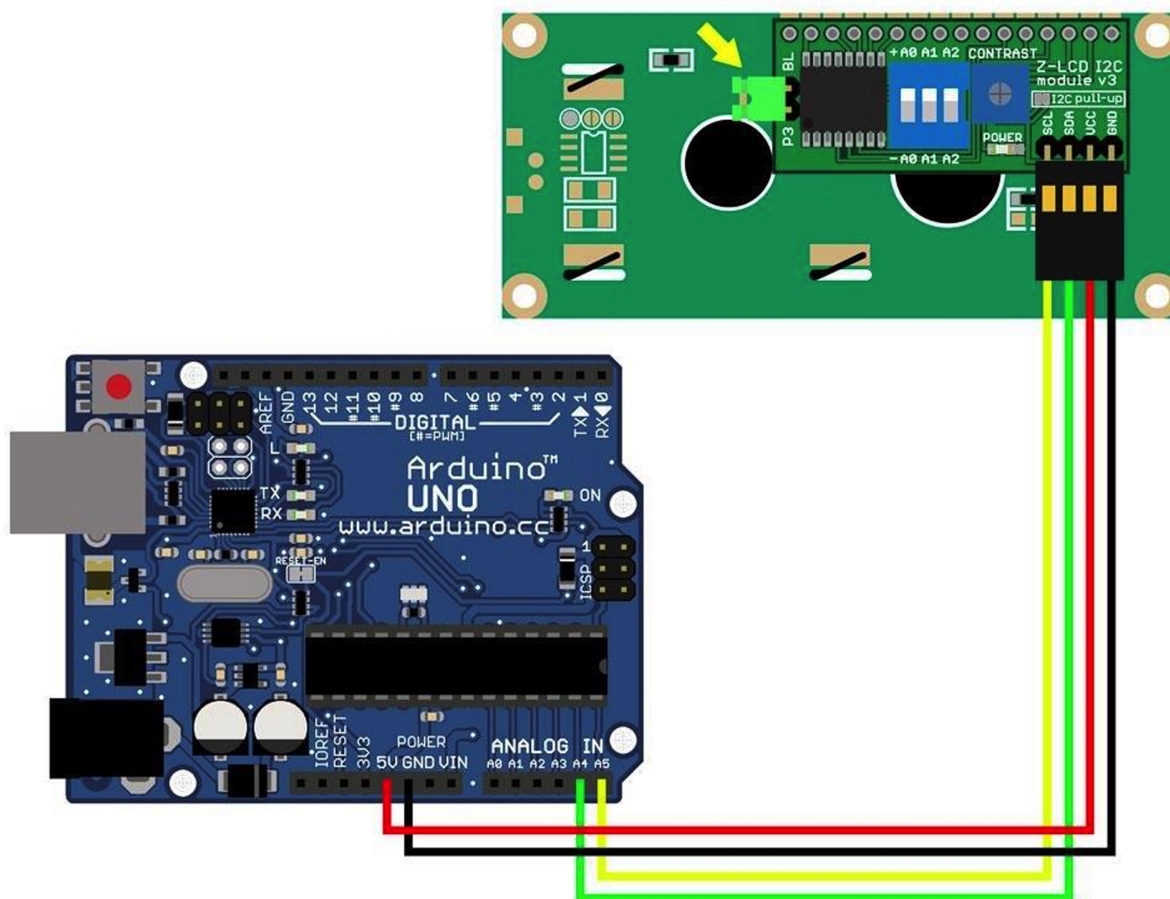


Рис.15

Дисплей, как и сервопривод нуждается в питании 5 Вольт, поэтому 2 провода подключаются на 5 вольт и разъем GND, оставшиеся 2 провода являются управляющими проводами.

Управляющий код:

```
#include <Wire.h> // Подключение библиотеки для работы с протоколом I2C
#include <LiquidCrystal_I2C.h> // Подключение библиотеки дисплея
LiquidCrystal_I2C lcd(0x27,16,2); // Устанавливаем дисплей
void setup()
{
```

```

lcd.init();
lcd.backlight();// Включаем подсветку дисплея
lcd.print("KSPU the best"); // Вывод статического текста, который не будет
меняться
lcd.setCursor(0, 1); // Устанавливаем курсор на вторую строку и нулевой
символ.
lcd.print("2021 KRSK");
}
void loop()
{
// Устанавливаем курсор на вторую строку и нулевой символ.
lcd.setCursor(0, 1);
// Выводим на экран количество секунд с момента запуска Arduino
lcd.print(millis()/1000);
}

```

Задание: Написать код вывода любого текста на экран учитывая ограничения дисплея.

## 8. Термодатчик. Виды, управление. Подключение к экрану для вывода значений.

Цель: Подключение источника ввода информации, получения с него данных.

Теоретическая часть:

Используемый цифровой датчик DS18B20 отправляет данные по Wire шине и может работать на одной линии с множеством других устройств. Каждый датчик имеет свой персональный 64-битный код, позволяющий микроконтроллеру Arduino общаться на одной шине сразу с несколькими сенсорами. Датчик преобразует температуру окружающей среды в цифровой код, т.е. для подключения не требуется дополнительного АЦП.

Датчик может быть выполнен в нескольких вариантах (смотри фото выше), от этого будет зависеть только схема подключения термодатчика к Arduino NANO или UNO. В первом случае необходимо использовать подтягивающий резистор на 4.7 кОм. Датчик, в виде готового модуля уже имеет резистор. Третий вариант — это датчик в герметичном корпусе, который можно смело использовать в горячей воде.

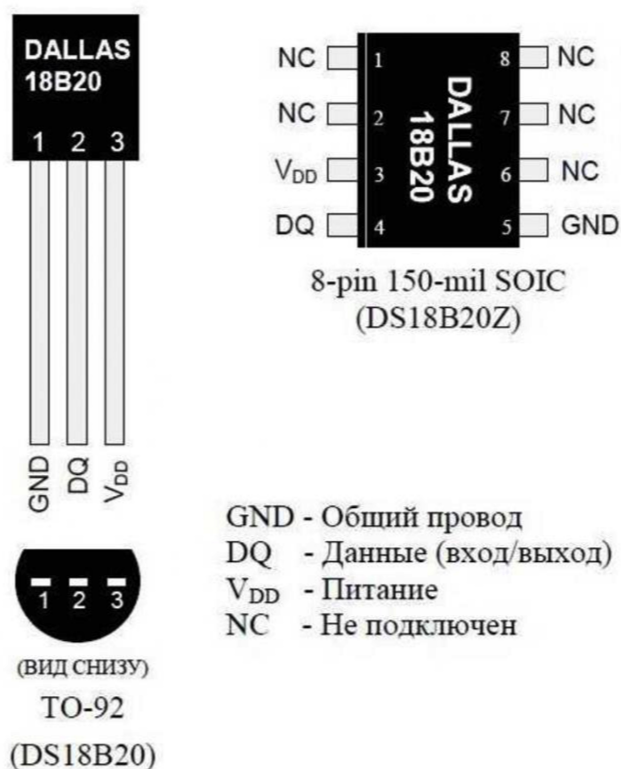


Рис.16

Практическая часть:

Подключить термодатчик согласно схеме:



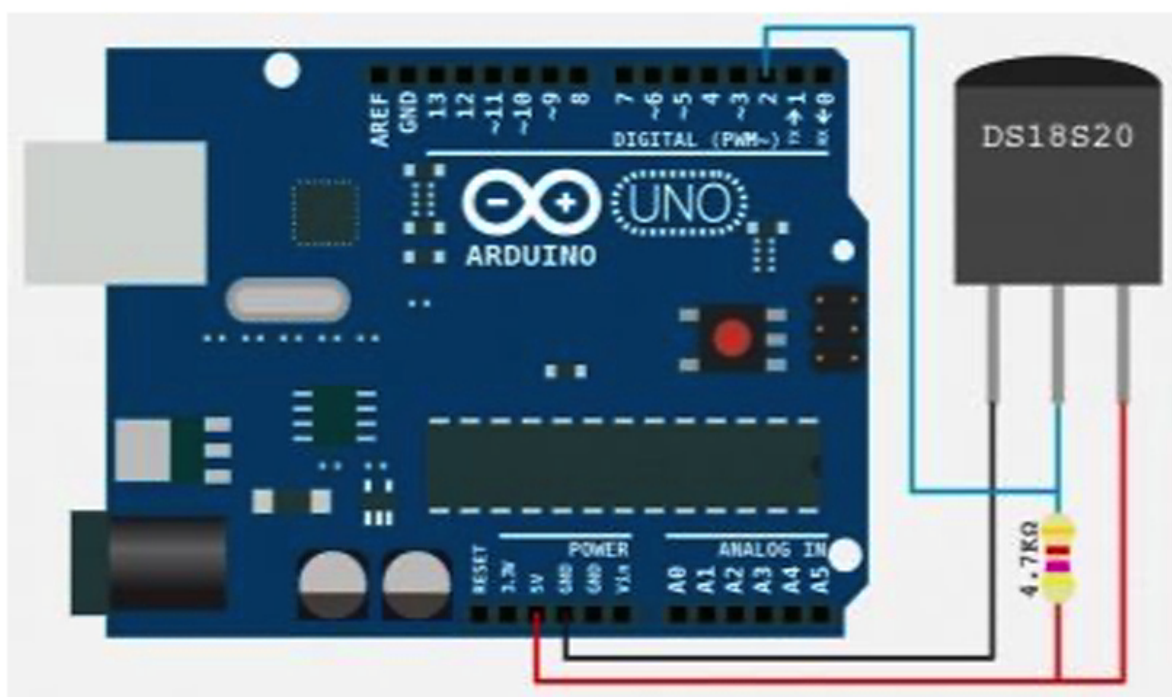


Рис.17

В предложенной схеме дополнительно подключен резистор, для более точных показаний.

Ниже представлен исполнительный код:

```
#include <OneWire.h>

OneWire ds(2); // Создаем объект OneWire для шины 1-Wire, с помощью
               // которого будет осуществляться работа с датчиком

void setup() {
  Serial.begin(9600);
}

void loop() {
  // Определяем температуру от датчика DS18B20

  byte data[2]; // Место для значения температуры

  ds.reset(); // Начинаем взаимодействие со сброса всех предыдущих команд и
             // параметров
```

```

ds.write(0xCC); // Даем датчику DS18B20 команду пропустить поиск по
адресу. В нашем случае только одно устройство

ds.write(0x44); // Даем датчику DS18B20 команду измерить температуру.
Само значение температуры мы еще не получаем - датчик его положит во
внутреннюю память

delay(1000); // Микросхема измеряет температуру, а мы ждем.

ds.reset(); // Теперь готовимся получить значение измеренной температуры

ds.write(0xCC);

ds.write(0xBE); // Просим передать нам значение регистров со значением
температуры

// Получаем и считываем ответ

data[0] = ds.read(); // Читаем младший байт значения температуры

data[1] = ds.read(); // А теперь старший

// Формируем итоговое значение:

// - сперва "склеиваем" значение,

// - затем умножаем его на коэффициент, соответствующий разрешающей
способности (для 12 бит по умолчанию - это 0,0625)

float temperature = ((data[1] << 8) | data[0]) * 0.0625;

// Выводим полученное значение температуры в монитор порта

Serial.println(temperature);

}

```

Задание: Провести эксперимент по изменению температуры, в отчете представить скриншоты из монитора порта.

9. Промежуточный контроль. Написание кода вывод температуры на экран.

Цель: Самостоятельная работа, проверка усвоенного материала.

Теоретическая часть: Объяснение задачи.

Практическая часть:

Объединение двух предыдущих уроков, необходимо написать код, который будет выполнять следующее:

Получение данных с термодатчика и вывода их на экран.

10. Транзистор. Виды, подключение.

Цель: Подключение светодиода с использованием транзистора.

Теоретическая часть:

Транзистор выступает в роли силового ключа и служит снятием нагрузки с платы Arduino. В примере будет подключен один светодиод через транзистор для примера. В будущем возможно подключения внешнего питания на транзистор, в соответствии с его характеристиками, отличным от питания Arduino 5 Вольт.

Практическая часть:

В данном уроке необходимо подключить транзистор и светодиод, транзистор позволяет подключить большую нагрузку без последствий для Arduino. При повышенном потреблении устройств, Arduino может постоянно перезагружаться из-за нехватки питания или выгореть от большой нагрузки, что является пагубным при разработке устройства.

Ниже представлена схема подключения:

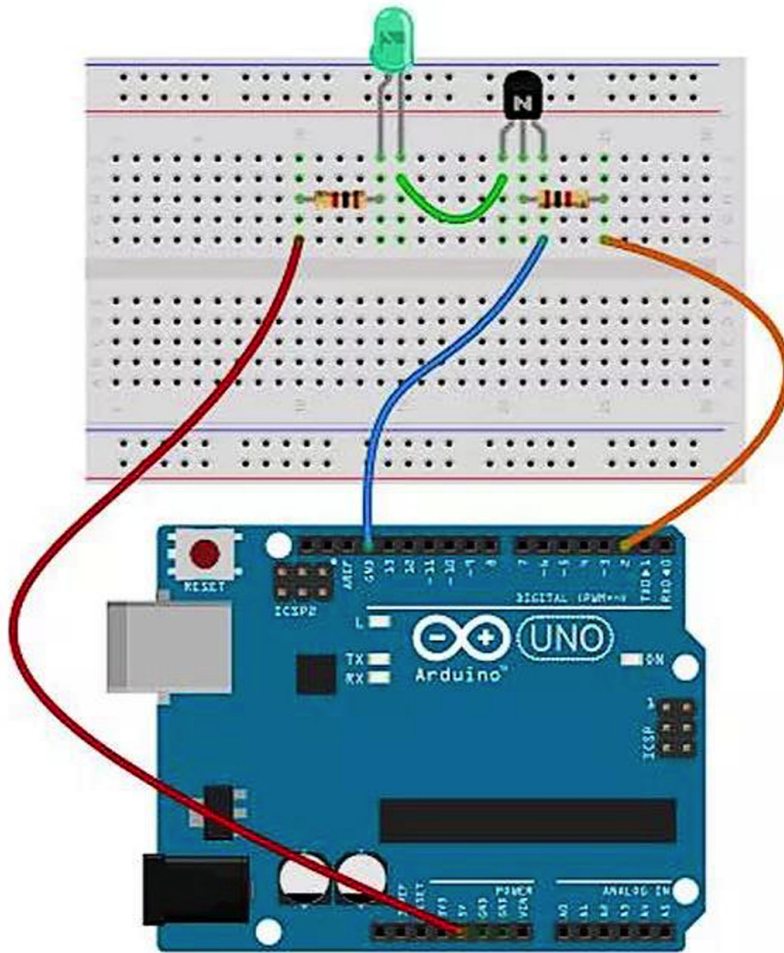


Рис.18

При подаче питания, в нашем случае на транзистор, будет загораться светодиод, в роли нагрузки может выступать любое устройство. Код полностью соответствует коду из четвертого урока “управление светодиодом”

```
int transistor = 2;  
  
void setup()  
{  
  pinMode (transistor, OUTPUT);  
}  
  
void loop()  
{
```

```
digitalWrite (transistor, HIGH);  
  
delay(1000);  
  
digitalWrite (transistor, LOW);  
  
delay(1000);  
  
}
```

Задание: Добавить в схему еще 5 светодиодов, представить фотографию и схему собранной схемы, а также код.

### 11. Матричная клавиатура. Подключение, управление.

Цель: Подключение устройства ввода информации, клавиатуры.

Теоретическая часть: В мире электроники клавиатурой называют устройство, состоящее из отдельных кнопок, которое позволяет вводить информацию в компьютер или в иной электронный прибор.

Клавиатуры имеют массу вариантов исполнения. На старых персональных компьютерах использовались клавиатуры с механическими клавишами, которые имели большой ход и были почти также удобны как клавиши на старых печатных машинках. Последние 15-20 лет широко применяются мембранные клавиатуры, которые проще и дешевле в изготовлении. Именно такую клавиатуру мы будем подключать к Ардуино.

Популярная в Ардуино проектах матричная мембранная клавиатура имеет 12 или 16 кнопок. С нижней части клавиатуры выходит шлейф, заканчивающийся dupont розеткой.

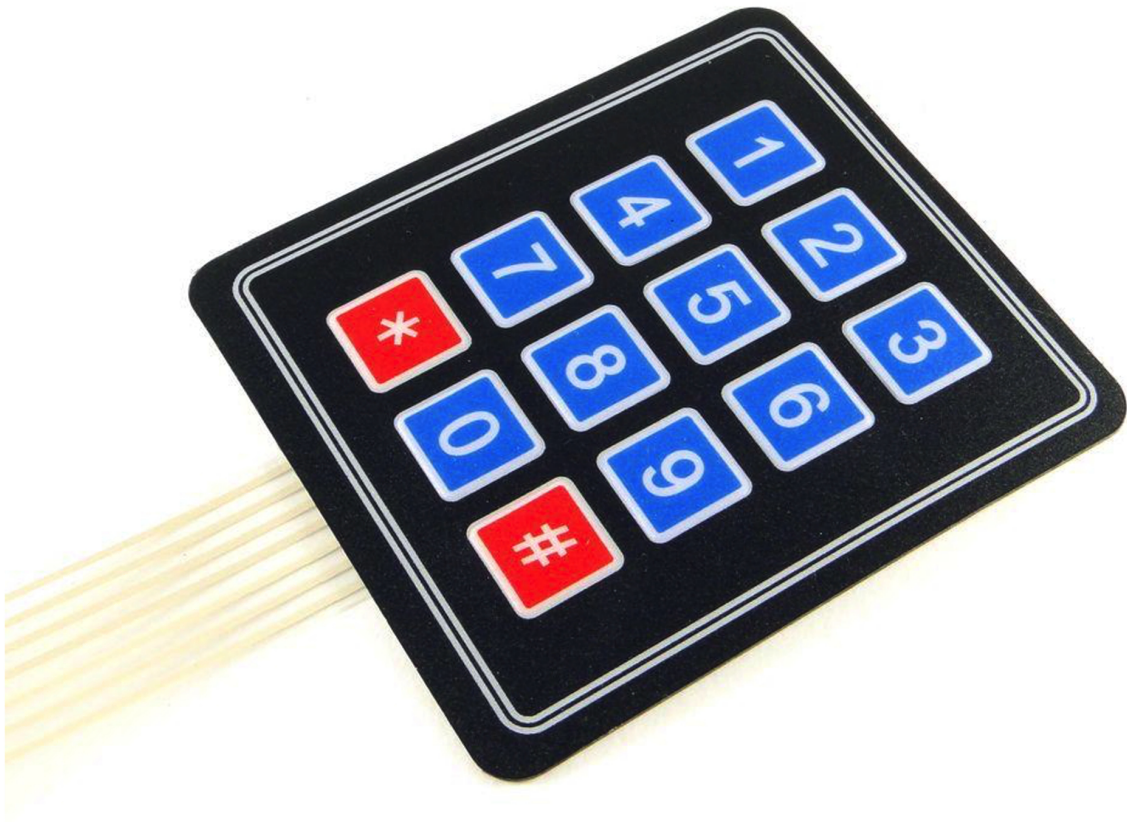


Рис.19

Матричная клавиатура состоит из кнопок, образующих матрицу  $m \times n$ , т.е. таблицу, где  $m$  — количество строк, а  $n$  — количество столбцов. Для урока мы воспользуемся клавиатурой  $4 \times 3$ . Если мы рассмотрим её шлейф, то увидим, что он состоит из 7 дорожек. Дорожки с номерами 1-4 (обозначим их Row1-Row4) — это строки матрицы с первой по четвёртую, а дорожки с номерами 5-7 (Col1-Col3) — столбцы с первого по третий.

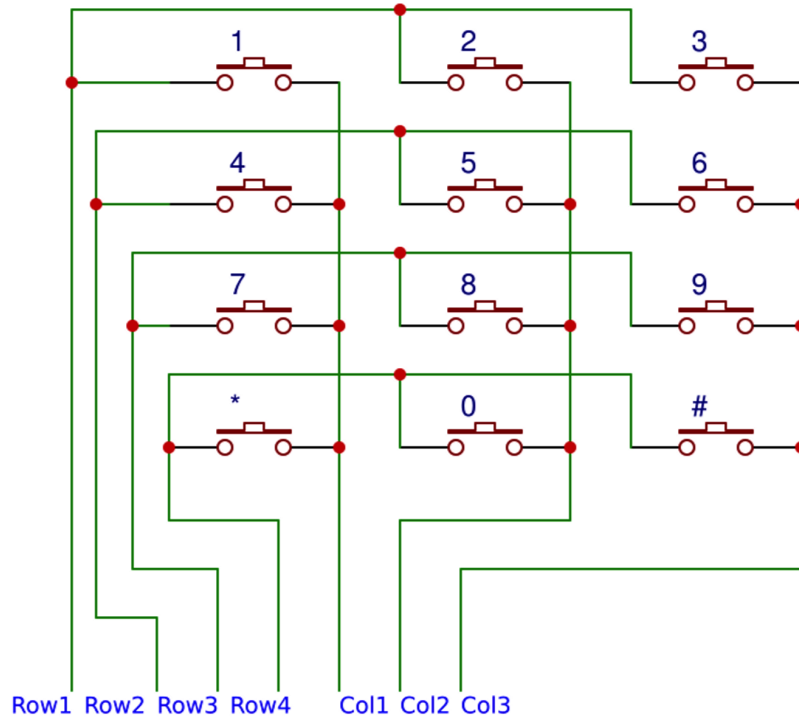


Рис.20

Каждая кнопка — это область пересечения токопроводящих дорожек. При нажатии на участок, обозначающий кнопку, происходит соединению дорожек и замыкание одного из выводов строк 1-4 с одним из выводов столбцов 1-3.

Практическая часть:

В данном уроке будет использоваться матричная клавиатура для ввода информации и дисплей для вывода информации.

Подключим все согласно схеме:

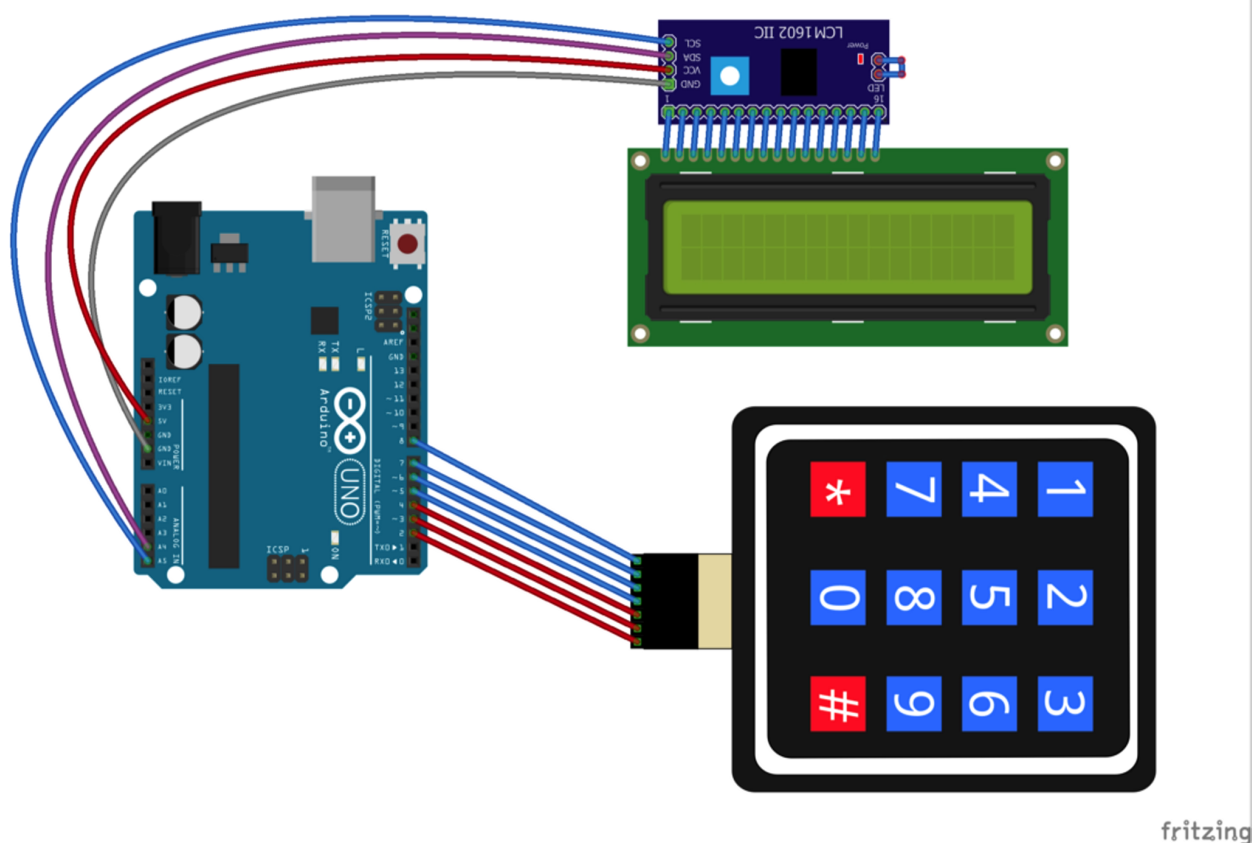


Рис.21

Ниже представленный код позволяет выводить на дисплей нажатую кнопку на клавиатуре.

```
#include <Wire.h>
#include <Key.h>
#include <Keypad.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
const byte ROWS = 4; // число строк клавиатуры
const byte COLS = 3; // число столбцов клавиатуры
char hexaKeys[ROWS][COLS] = {
    {'1','2','3'},
    {'4','5','6'},
    {'7','8','9'},
    {'*','0','#'}}
```



```

};
byte rowPins[ROWS] = {8, 7, 6, 5}; // к каким выводам подключаем управление
строками
byte colPins[COLS] = {4, 3, 2}; // к каким выводам подключаем управление
столбцами
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins,
ROWS, COLS);
void setup(){
    Serial.begin(9600);
    lcd.init(); // Инициализируем экран
    lcd.backlight();
}
void loop(){
    lcd.setCursor(0,0); // устанавливаем курсор на начало первой строки
    lcd.print(«Press any key!»);
    char customKey = customKeypad.getKey();
    if (customKey){
        Serial.println(customKey);
        lcd.setCursor(0,1); // устанавливаем курсор на начало второй строки
        lcd.print(customKey);
    }
}

```

Задание: Изменить символы выводимые кнопками, представить код.

## 12.Итоговый тест.

Цель: Самостоятельная работа, проверка усвоенного материала.

Теоретическая часть: Объяснение задачи.

Практическая часть: Результат всех пройденных уроков. Задача объединить четыре пройденных элемента и создать собственное устройство. Описать его назначение. Написание кода. Проверка работоспособности.

#### 2.4. Педагогический эксперимент

Педагогический эксперимент проводился на базе АНО ДТ “Красноярский Кванториум”.

В рамках описанного курса было проведено шесть полных курсов для учеников из разных регионов Красноярского края. Ребята выполняли все задачи в срок, проблем с задачами не возникало. Единственной проблемой возникали организационные моменты в месте проведения курса. По договоренности с местными кураторами, ребят собирали раз в неделю, в рамках уроков технологии, и решали поставленные задачи в классах информатики, позже отправляли выполненные задания. Время выполнения заданий от 40 до 120 минут, с одним перерывом 5 минут.

Перед выполнением практического задания, было предложено посмотреть видео урок, а также при необходимости ознакомиться с дополнительным материалом. В виде урока представлена ознакомительная информация и пример выполнения задания.

Перед выполнением задания учащиеся изучают требования по выполнению. При необходимости, учащиеся могут объединиться в группы не более двух человек, для выполнения поставленной задачи.

Первые задания позволяют познакомиться с платформой для программирования и самой программой. Показывается насколько просто можно войти в мир программирования, активное развитие цифровой техники в современном мире.

К концу курса, учащийся получает необходимый минимум знаний для самостоятельного продолжения изучения схемотехники и программирования.

Подводя итоги, можно сказать, что курс хорошо себя зарекомендовал и был успешно проведен в различных учебных заведениях. Реализация

осуществляется на протяжении трех лет. На основе курса были созданы образовательные программы для студентов на 108 часов и углубленная программа для учащихся кванториума.

## **Выводы по второй главе**

Практические занятия играют важную роль в образовательном процессе. Образованию приходится соответствовать современным требованиям. Интернет является неотъемлемой частью современного общества и образованию очень важно занимать в нем место.

В рамках уроков технологии важно показать учащимся, что интернет и цифровые технологии несут важную часть. Весь современный мир построен на освоении цифровых технологий. Практическое применение знаний очень важно не только для школьника, но и студента ВУЗа, важно показать, как можно применить свои знания в дальнейшем.

В рамках проведенного эксперимента, можно сказать, что материал был усвоен полностью и ученики проявили заинтересованность в дальнейшем изучении темы и личного развития в целом.

## **ЗАКЛЮЧЕНИЕ**

В ходе работы над магистерской диссертацией были проанализированы микроконтроллеры от появления до сегодняшних дней, управляемая электроника. Наиболее популярное решение для программирования в данный момент является Arduino. Платформа успешно выполняет свои задачи и активно внедряется в учебную деятельность. Приятный и понятный интерфейс программного обеспечения позволяет легко научиться программированию, а система вывода ошибок в синтаксисе дополнительно помогает как новичку, так и опытному программисту в выявлении ошибки в коде.

Проведенный эксперимент показал, что учащиеся заинтересованы в изучении программирования и схемотехники. Хотя начальный уровень сильно отличается, все учащиеся успешно прошли курс и получили сертификат о успешном прохождении курса.

Подводя итоги, можно сказать, что платформа Arduino является максимально понятной для ознакомления в любом возрасте и с любым начальным уровнем в области программирования и схемотехники.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Б.С. Воскобойников, Б.И. Зайчик, С.М. Палей. Словарь по гибким производственным системам и робототехнике. Английский. Немецкий. Французский. Нидерландский. 1991г.
2. Подураев Ю.В. Мехатроника: основы, методы, применение: учеб.пособие / Ю.В.Подураев.–2-е изд., стер.–М.: Машиностроение, 2007.–256с
3. Волович Г.И. Схемотехника аналоговых и аналого-цифровых электронных устройств. –М.: Издательский дом «Додэка-XXI», 2005. – 528с.
4. Рюмик С. М. 1000 и одна микроконтроллерная схема. –М.: Издательский дом «Додэка-XXI», 2010. – 400с
5. Хоровиц П., Хилл У. Искусство схемотехники. Том 1-3. Издательство: Мир 1993
6. Аверченков О.Е. Схемотехника: аппаратура и программы М.: ДМК Пресс, 2012. - 588 с.
7. Автушко В.П. и др. Лабораторный практикум по основам схемотехники электронных цепей. Учебное пособие. В. П. Автушко, В. М. Белопольский, В. В. Масленников. М.: МИФИ, 2005. 120 с
8. Алексенко А.Г. Основы микросхемотехники. Элементы морфологии микроэлектронной аппаратуры. М.: Изд-во "Советское радио", 1971.- 352 стр
9. Алексенко А.Г., Шагурин И.И. Микросхемотехника. 2-е издание, переработанное и дополненное Издательство: Радио и связь Год: 1990 Страниц: 496.

10. Алещенко В.И. Лекции - Микроэлектроника и схемотехника ЭВМ. Краснодар, КубГТУ, 2006г., 65 с
11. Андреевская Т.М. Полный курс основы аналоговой схемотехники. Андреевская Т. М. Кафедра РЭ, МИЭМ, 2005
12. Антипенский Р.В., Фадин А.Г. Схемотехническое проектирование и моделирование радиоэлектронных устройств. М.: Техносфера, 2007. - 128 с
13. Афонин В.В., Акулинин И.Н. Электроника. Лабораторные работы. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2005. 40 с
14. Ашанин В.Н., Исаев С.Г., Ермаков В.В. Схемотехника. Аналоговая схемотехника. Пенза: Информационно-издательский центр ПГУ, 2007. Часть 1: Аналоговая схемотехника. -268 с
15. Базарова И.А. Цифровая схемотехника. Учебное пособие. – Ухта: УГТУ, 2004. – 134 с
16. Копосов, Д. Г. Первый шаг в робототехнику. 5-6 классы. Практикум / Д.Г. Копосов. - М.: Бином. Лаборатория знаний, 2014. - 292 с.
17. Копосов, Д. Г. Первый шаг в робототехнику. Практикум для 5-6 классов / Д.Г. Копосов. - М.: Бином. Лаборатория знаний, 2012. - 292 с.
18. Робототехника для детей и родителей. С.А.Филиппов. СПб: Наука, 2010.
19. Я, робот. Айзек Азимов. Серия: Библиотека приключений. М: Эксмо, 2002.
20. Воскобойников, Б. С. Словарь по гибким производственным системам и робототехнике. Английский. Немецкий. Французский. Нидерландский / Б.С. Воскобойников, Б.И. Зайчик, С.М. Палей. - М.: Русский язык, 1991. - 392 с.
21. Иванов, А. А. Основы робототехники / А.А. Иванов. - М.: Форум, 2012. - 224 с.

22. Костров, Б. В. Искусственный интеллект и робототехника / Б.В. Костров, В.Н. Ручкин, В.А. Фулин. - М.: Диалог-Мифи, 2008. - 224 с.
23. Петров, А. А. Англо-русский словарь по робототехнике / А.А. Петров, Е.К. Масловский. - М.: Русский язык, 1989. - 494 с.
24. Овсяницкая, Л.Ю. Курс программирования робота Lego Mindstorms EV3 в среде EV3: основные подходы, практические примеры, секреты мастерства / Д.Н. Овсяницкий, А.Д. Овсяницкий. – Челябинск: ИП Мякотин И.В., 2014. – 204 с. ISBN 978-5-904593-43-8
25. Белиовская Л.Г. Узнайте, как программировать на LabVIEW. – Изд-во ДМК, 2013. – 140 с.



## Список команд в программе Arduino IDE

Операторы	Данные	Функции
<ul style="list-style-type: none"> <li>• setup()</li> <li>• loop()</li> </ul>		
	Константы	Цифровой ввод/вывод
	<ul style="list-style-type: none"> <li>• HIGH   LOW</li> </ul>	<ul style="list-style-type: none"> <li>• pinMode()</li> </ul>
Управляющие операторы	<ul style="list-style-type: none"> <li>• INPUT   OUTPUT</li> <li>• true   false</li> <li>• Целочисленные константы</li> </ul>	<ul style="list-style-type: none"> <li>• digitalWrite()</li> <li>• digitalRead()</li> </ul>
<ul style="list-style-type: none"> <li>• if</li> <li>• if...else</li> <li>• for</li> <li>• switch case</li> <li>• while</li> <li>• do... while</li> <li>• break</li> <li>• continue</li> <li>• return</li> <li>• goto</li> </ul>	<ul style="list-style-type: none"> <li>• Константы с плавающей запятой</li> </ul>	Аналоговый ввод/вывод
	Типы данных	
	<ul style="list-style-type: none"> <li>• boolean</li> <li>• char</li> <li>• byte</li> <li>• int</li> <li>• unsigned int</li> <li>• word</li> <li>• long</li> <li>• unsigned long</li> <li>• float</li> <li>• double</li> <li>• string - массив СИМВОЛОВ</li> </ul>	<ul style="list-style-type: none"> <li>• analogRead()</li> <li>• analogReference()</li> <li>• analogWrite()</li> </ul>
Синтаксис		Дополнительные функции ввода/вывода
<ul style="list-style-type: none"> <li>• ; (semicolon)</li> <li>• {} (curly braces)</li> <li>• // (single line comment)</li> <li>• /* */ (multi-line comment)</li> </ul>		<ul style="list-style-type: none"> <li>• tone()</li> <li>• noTone()</li> <li>• shiftOut()</li> <li>• pulseIn()</li> </ul>
		Работа со временем
		<ul style="list-style-type: none"> <li>• millis()</li> <li>• micros()</li> <li>• delay()</li> <li>• delayMicroseconds()</li> </ul>

Арифметические операторы	<ul style="list-style-type: none"> <li>• String - объект класса</li> <li>• массив (array)</li> <li>• void</li> </ul>	Математические функции
<ul style="list-style-type: none"> <li>• = (assignment)</li> <li>• + (addition)</li> <li>• - (subtraction)</li> <li>• * (multiplication)</li> <li>• / (division)</li> <li>• % (modulo)</li> </ul>	Преобразование типов данных <ul style="list-style-type: none"> <li>• char()</li> <li>• byte()</li> <li>• int()</li> <li>• long()</li> <li>• float()</li> </ul>	<ul style="list-style-type: none"> <li>• min()</li> <li>• max()</li> <li>• abs()</li> <li>• constrain()</li> <li>• map()</li> <li>• pow()</li> <li>• sq()</li> <li>• sqrt()</li> </ul>
Операторы сравнения <ul style="list-style-type: none"> <li>• == (equal to)</li> <li>• != (not equal to)</li> <li>• &lt; (less than)</li> <li>• &gt; (greater than)</li> <li>• &lt;= (less than or equal to)</li> <li>• &gt;= (greater than or equal to)</li> </ul>	Область видимости переменных и квалификаторы <ul style="list-style-type: none"> <li>• Область видимости</li> <li>• static</li> <li>• volatile</li> <li>• const</li> </ul>	Тригонометрические функции <ul style="list-style-type: none"> <li>• sin()</li> <li>• cos()</li> <li>• tan()</li> </ul>
Логические операторы <ul style="list-style-type: none"> <li>• &amp;&amp; (И)</li> <li>•    (ИЛИ)</li> <li>• ! (Отрицание)</li> </ul>		Генераторы случайных значений <ul style="list-style-type: none"> <li>• randomSeed()</li> <li>• random()</li> </ul>
Унарные операторы <ul style="list-style-type: none"> <li>• ++ (increment)</li> <li>• -- (decrement)</li> </ul>		Внешние прерывания <ul style="list-style-type: none"> <li>• attachInterrupt()</li> <li>• detachInterrupt()</li> </ul>
		Функции передачи данных

- += (compound addition)
  - -= (compound subtraction)
  - \*= (compound multiplication)
  - /= (compound division)
- Serial

### *Библиотеки Arduino*

Servo — библиотека управления сервоприводами.

EEPROM — чтение и запись энергонезависимой памяти микроконтроллера.

SPI — библиотека, реализующая передачу данных через интерфейс SPI.

Stepper — библиотека управления шаговыми двигателями.

Настройка курса, в которой генерируется код-приглашение, формируется папка для курса, отображение информации.

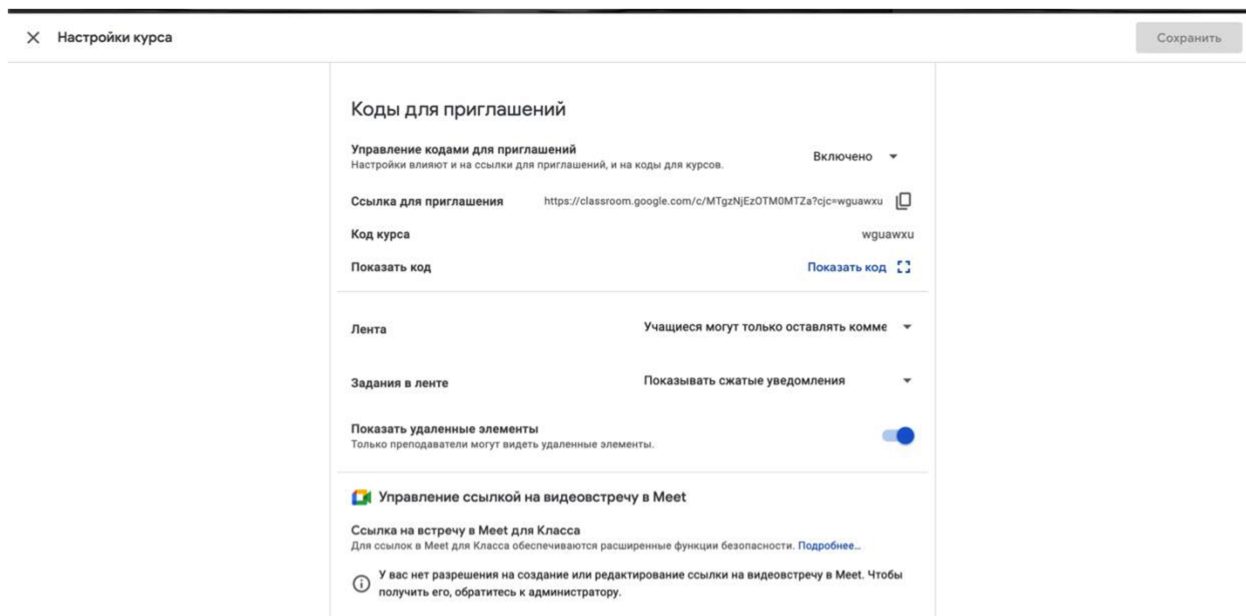


Рис.22

Главная страница, на которой можно видеть все новые задания, объявления и предстоящие задания с указанием срока сдачи

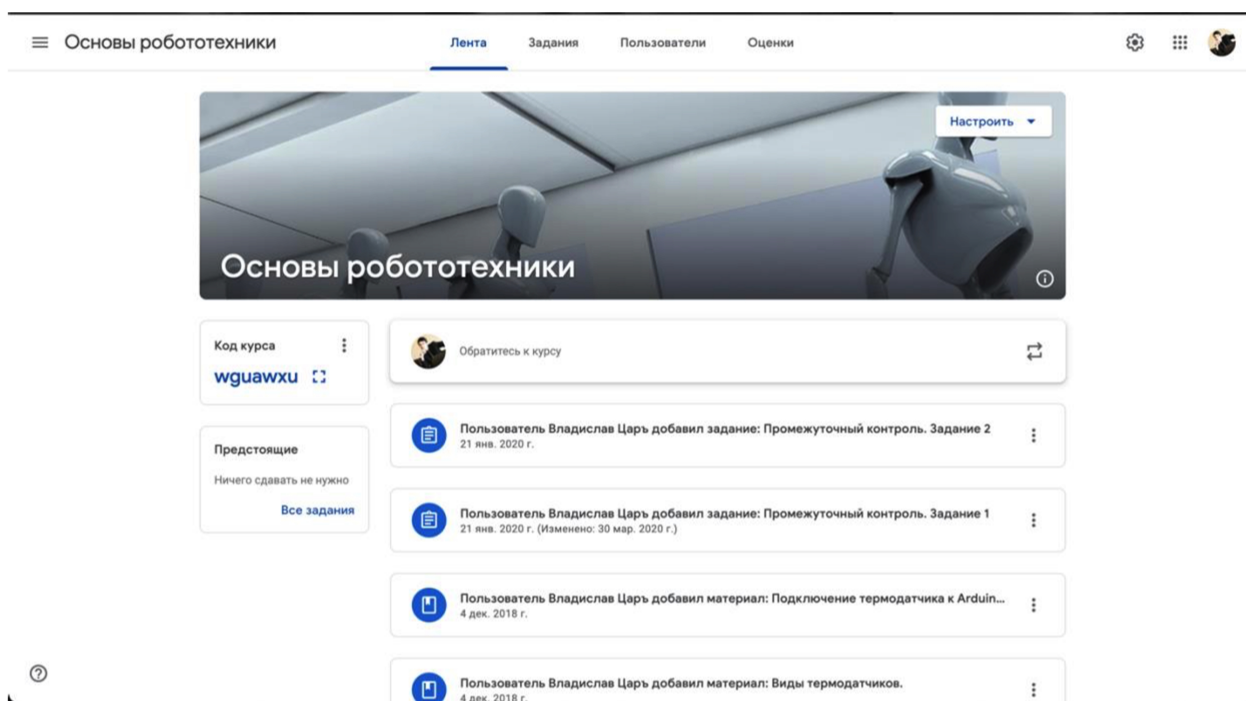
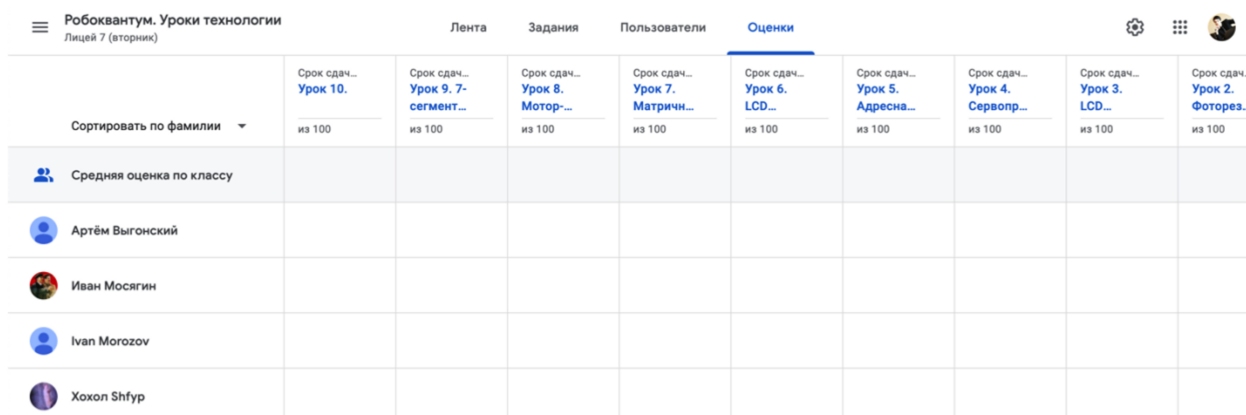


Рис.23

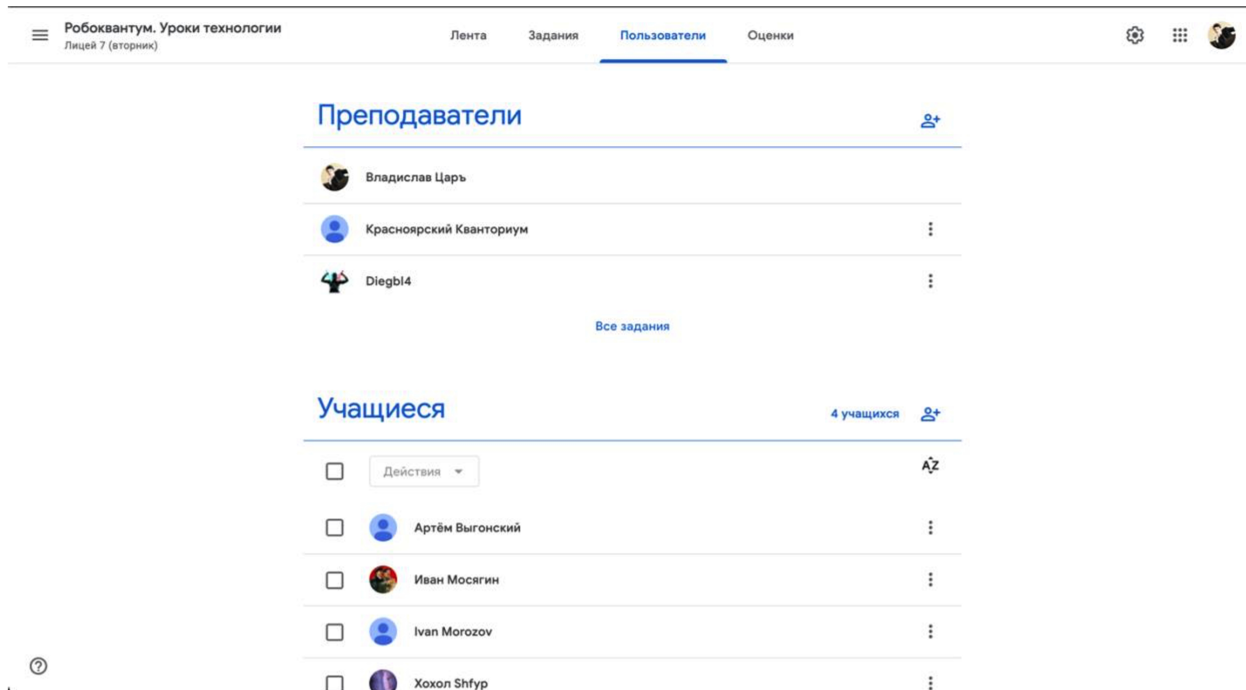
На данной странице можно отследить успеваемость каждого ученика, проверить работу и выставить оценку. Также увидеть срок сдачи задания.



	Срок сдач... Урок 10. из 100	Срок сдач... Урок 9. 7- сегмент... из 100	Срок сдач... Урок 8. Мотор... из 100	Срок сдач... Урок 7. Матричн... из 100	Срок сдач... Урок 6. LCD... из 100	Срок сдач... Урок 5. Адресна... из 100	Срок сдач... Урок 4. Сервопр... из 100	Срок сдач... Урок 3. LCD... из 100	Срок сдач... Урок 2. Фотореза... из 100
Средняя оценка по классу									
Артём Выгонский									
Иван Мосягин									
Ivan Morozov									
Хохол Shfyp									

Рис.24

На данной странице видно всех преподавателей, у кого есть доступ к курсу.



Преподаватели	
	Владислав Царь
	Красноярский Кванториум
	Diegb14

Все задания

Учащиеся	
<input type="checkbox"/>	Действия
<input type="checkbox"/>	Артём Выгонский
<input type="checkbox"/>	Иван Мосягин
<input type="checkbox"/>	Ivan Morozov
<input type="checkbox"/>	Хохол Shfyp

Рис.25

На данной странице видно сколько работ было сдано и сколько учеников еще не выполнили задание. В виде рассылки им можно напомнить о невыполненной работе.

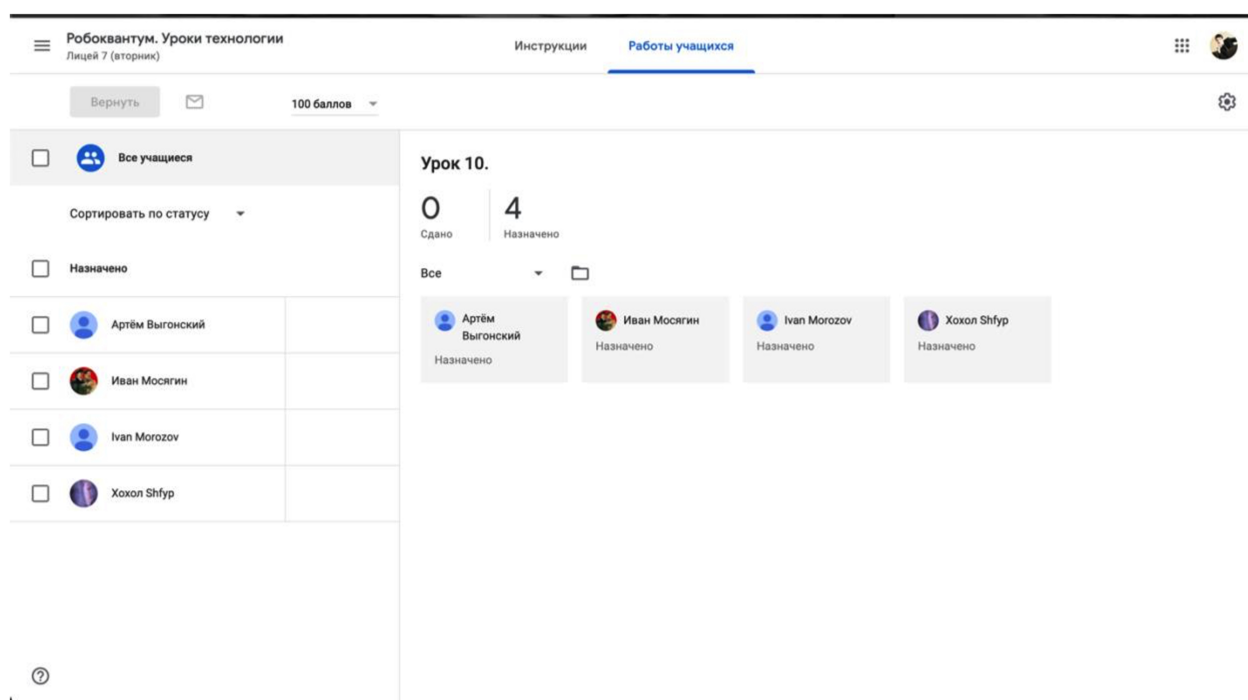



Рис.26

Итоговый тест:



### Тестирование

mikhaylov77730@gmail.com [Сменить аккаунт](#)

**\* Обязательно**

**Электронная почта \***

Ваш адрес эл. почты \_\_\_\_\_

**ФИО \***

Мой ответ \_\_\_\_\_

[Далее](#) [Очистить форму](#)

Никогда не используйте формы Google для передачи паролей.  
Компания Google не имеет никакого отношения к этому контенту. [Сообщение о нарушении](#) - [Условия использования](#) - [Политика конфиденциальности](#)

Рис.27

### Тестирование

**Зачем в схемах используются диоды? \*** 1 балл

- Для предотвращения пропускания тока в обратную сторону
- Для ограничения тока
- Для информирования о событиях

**Сколько нужно задействовать пинов Arduino для подключения матричной клавиатуры размерностью 4x4? \*** 1 балл

- 4
- 8
- 12
- 16

**Принцип работы транзисторов. \*** 1 балл

- Усиление
- Генерация
- Регулировка
- Понижение

Рис.28

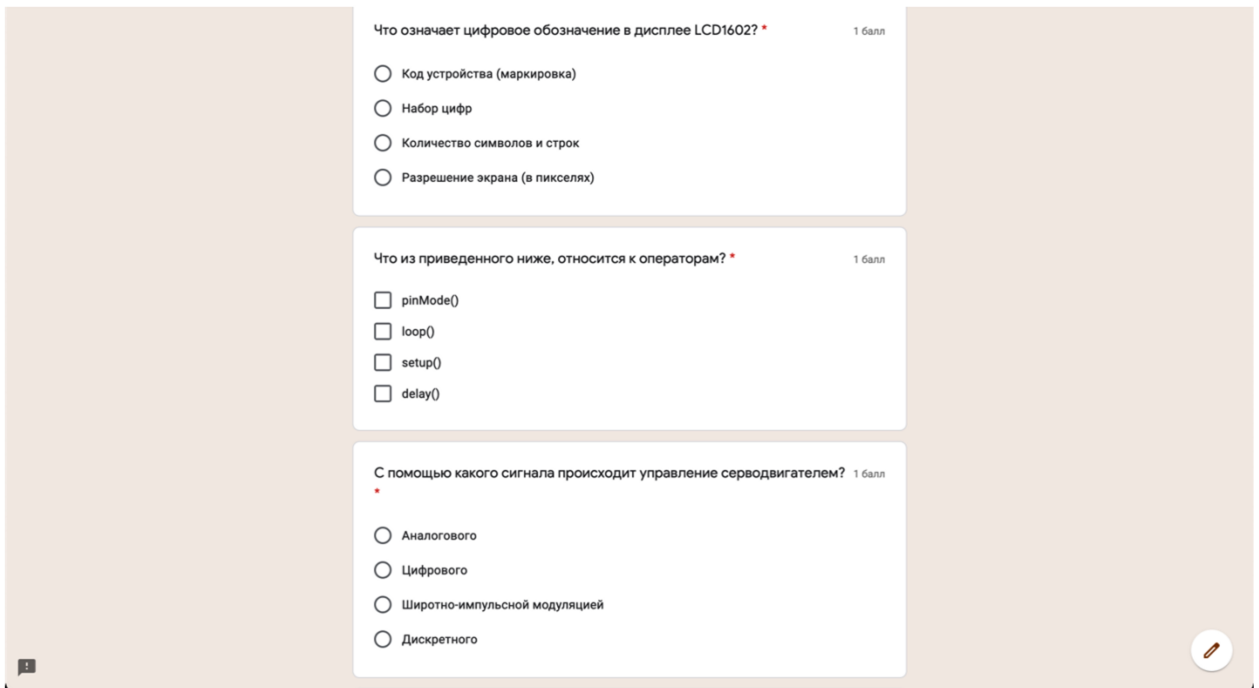


Рис.29

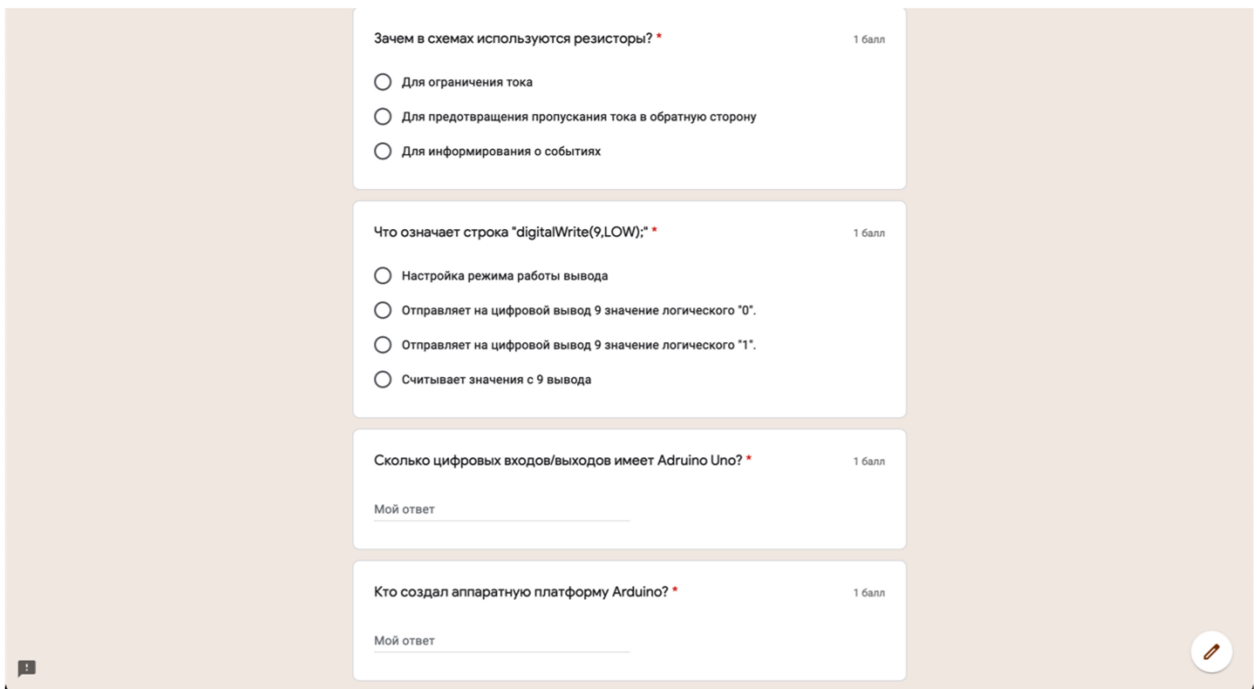


Рис.30