

Государственное бюджетное профессиональное образовательное учреждение «Кингисеппский
колледж технологии и сервиса»
структурное подразделение «Детский технопарк «Кванториум»

Рассмотрено педагогическим советом ГБПОУ ЛО «ККТ и С»
Протокол от «29» августа 2023 года № 1
Согласовано: заместитель директора-руководитель «ДТ «Кванториум»
«29» августа 2023 г.

Утверждена приказом
ГБПОУ ЛО «ККТ и С» от «30» августа 2023 г. № 63-О

Дополнительная общеразвивающая программа

«IT-квантум»

Вводный уровень

Возраст обучающихся: 12-14 лет

Срок освоения программы: 72 часа

Автор- составитель:

Тельнова К.Р.

педагог дополнительного образования

г. Кингисепп

Внутренняя экспертиза проведена. Программа рекомендована к рассмотрению на педагогическом/методическом совете учреждения.

Заместитель руководителя по образовательной деятельности

Пол- О.Е. Пономарева (Подпись, ФИО)
« 25 » августа 2023г

Дополнительная общеразвивающая программа соответствует действующим федеральным, региональным нормативным документам Российской Федерации и локальным нормативным актам ГБПОУ ЛО «ККТиС».

Пояснительная записка

Дополнительная общеразвивающая программа «ИТ-квантум» Вводный уровень разработана на основании Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации», Концепции развития дополнительного образования детей до 2030 года.

Направленность программы

Техническая

Актуальность программы

Информационные технологии (ИТ) - это методы, способы, приемы и процессы обработки информации с применением средств вычислительной техники или программных и технических средств. В широком понимании ИТ охватывают все области создания, передачи, хранения и восприятия информации, не ограничиваясь только компьютерными технологиями. Главная информационная тенденция нашего времени — усложнение и интеграция всех видов информационных продуктов. Информационные технологии стремительно развиваются и будут развиваться дальше. Без них уже невозможно представить ни одну сферу жизнедеятельности, они напрямую влияют на конкурентоспособность производимых товаров и услуг. Поэтому квалифицированные ИТ-специалисты — одни из самых востребованных и высокооплачиваемых на рынке труда. В рамках реализации данной программы формируются не только необходимые навыки в области информационных технологий, но и над профессиональные навыки, позволяющие повысить эффективность и востребованность специалиста.

Программа нацелена на развитие интереса обучающихся к основам разработки программного обеспечения, использованию методологий командной работы в проекте, программированию, проектированию электронных схем и конструированию устройств на их основе. Обучение по программе позволяет подросткам получить практические навыки и знания, выходящие за рамки школьных программ по информатике.

Педагогическая целесообразность программы

Обучение по программе включает четыре основных направления деятельности:

1. Основы управления проектами, включающие методологии командной работы в проекте.
2. Получение навыков разработки программного обеспечения на основе современных и актуальных технологий.
3. Получение компетенций в области конструирования устройств, управляемых микроэлектроникой.
4. Освоение базовых принципов работы программирования, получение компетенций в области конструирования устройств, управляемых

микроэлектроникой.

Данная программа формирует компетенции, которые позволяют обучающимся успешно создавать собственные электронные устройства, заниматься разработкой программного обеспечения, программированием микроконтроллеров, а также конкурировать на рынке рабочей силы в области информационных технологий.

Программа «ИТ-квантум» направлена, в том числе, на решение профориентационных задач, обеспечивая возможность приобретения обучающимися знаний в сфере ИТ и умения применять их при решении различных инженерных задач, повышая интерес будущих специалистов к выбранному направлению. Программа также направлена на приобретение компетенций в сфере проектной, системной, организаторской и предпринимательской деятельности, расширение кругозора.

Методологической основой программы является системно-деятельностный подход, сочетающийся с различными современными образовательными технологиями, такими как технология развития понятийного мышления, технология исследовательской и проектной деятельности. Применение системно-деятельностного подхода наиболее эффективно способствует формированию универсальных учебных действий.

Цель программы

Получение базовых теоретических знаний в области устройства и функционирования современных платформ быстрого прототипирования электронных устройств на примере микроконтроллерной платформы Arduino.

1. Развитие у обучающихся чувства ответственности, внутренней инициативы, самостоятельности, тяги к самосовершенствованию.
2. Выработка у обучающихся навыков командной работы и публичных выступлений по ИТ-тематике.
3. Изучение основ алгоритмизации, построения алгоритмов и их формализации с помощью языка блок-схем.
4. Получение теоретических знаний и навыков программирования микроконтроллеров на языке C++ в среде Arduino IDE.
5. Изучение принципа действия аналоговых и цифровых датчиков, совместимых с микроконтроллерной платформой Arduino; подключение датчиков к микроконтроллерной платформе, получения и обработки показаний датчиков.
6. Получение навыков работы с электронными компонентами, совместимыми с Arduino: погружная помпа, часы реального времени, светодиодная лента и т.п.
7. Получение теоретических знаний и навыков разработки приложений для операционной системы Android.
8. Развитие логического мышления воспитанников через знакомство с основами алгоритмизации и формирование базовых знаний в области

программирования микроконтроллеров.

Задачи программы

Образовательные:

- познакомить с простейшими основами механики, с правилами техники безопасности;
- научить понимать основы алгоритмов; читать графические изображения, схемы;
- познакомить со средой программирования «Arduino IDE»;
- научить применять алгоритм на практике в программе Arduino 1.8.5;
- научить работать с различными операционными системами;
- познакомить воспитанников с принципом действия аналоговых и цифровых датчиков, совместимых с микроконтроллерной платформой Arduino;
- познакомить с платформой для разработки .NET а так же языками, подходящими для реализации проектов.
- с помощью языка C++ изучить принципы создания оконных приложений.
- научить создавать реально работающие модели действий и решений;
- обеспечить приобретение опыта создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; построения компьютерных моделей, коллективной реализации информационных проектов, преодоления трудностей в процессе интеллектуального проектирования, информационной деятельности в различных сферах, востребованных на рынке труда.

Развивающие:

- способствовать развитию образного, технического, логического мышления воспитанников;
- развивать творческие способности воспитанников;
- научить детей излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений.

Воспитательные:

- развивать у воспитанников аккуратность, силу воли, самостоятельность, внимательность, усидчивость, стремление доводить начатое дело до конца;
- формировать у воспитанников навык сохранения порядка на рабочем месте;
- формировать интерес воспитанников к программированию.

Адресат программы

Учащиеся в возрасте 12-14 лет, желающие заниматься исследованиями в области информационных технологий.

Количество обучающихся в группе

- вводный уровень - от 12 до 15 человек;

Формы обучения и виды занятий

Используются три основные формы работы:

- демонстрационная (обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах);

- фронтальная (обучающиеся синхронно работают под управлением педагога);

- самостоятельная (обучающиеся выполняют индивидуальные задания в течение части занятия или нескольких занятий),

в том числе интерактивные проблемные лекции, практическая работа, воркшопы, конференции, семинары, проведение эксперимента, исследовательская и проектная работа.

Возможны встречи с приглашенными спикерами, совместные конференции, видеоконференции или вебинары с другими квантумами и экспертами, индивидуальные и групповые консультации

По типу организации взаимодействия педагогов с обучающимися при реализации программ используются личностно-ориентированные технологии обучения (технологии проектной и исследовательской деятельности).

Реализация программы предполагает использование здоровьесберегающих технологий, реализующихся через создание безопасных условий, таких как включение в занятие динамических пауз, периодическая смена деятельности обучающихся; контроль соблюдения обучающимися правил работы на ПК; создание благоприятного психологического климата в группе.

Отличительная особенность программы

Представляемая программа основана на Методическом инструментарии наставника «IT-квантум тулкит» (Белоусова А.С., Юбзаев Т.И. – М.: Фонд новых форм развития образования, 2019 –76 с.), может иметь модульную структуру и заложенную возможность сетевого взаимодействия. Модули построены на практико-ориентированных инженерных и исследовательских проектах, направленных на решение задач прикладного и фундаментального характера.

Организационно-педагогические условия

При реализации дополнительной общеразвивающей программы используется форма, основанная на модульном принципе представления содержания образовательной программы и построения учебного плана.

Реализация программы может быть осуществлена как на собственных ресурсах кванториума, так и при поддержке сетевых и индустриальных партнеров через сетевое взаимодействие.

Совместная деятельность участников образовательного процесса выстраивается на принципах эмоциональной значимости, открытости, обратной связи и субъектности обучающегося.

Воспитательная работа

Цикл воспитательных мероприятий, изложенный в «Программе воспитания» ДТ «Кванториум» направлен на взаимодействие педагога и воспитанника, и ориентирован на сознательное овладение детьми социальным и духовным опытом, формирование у них социально-значимых ценностей и социально- адекватных приемов поведения.

Форма обучения - очная, возможно использование дистанционных образовательных технологий и электронного обучения. Занятия проводятся по группам. При реализации программы могут быть организованы и проведены массовые мероприятия для совместной деятельности обучающихся и родителей (законных представителей).

Срок освоения общеразвивающей программы определяется в учебном плане, который является приложением и может обновляться по мере необходимости.

Режим занятий

Продолжительность одного занятия – 45 минут. Количество занятий в день, неделю определяется в соответствии с учебным планом, календарным графиком.

Формы аттестации

Основой аттестации является проектная деятельность учащихся по направлению общеобразовательной программы и участием в различных соревнованиях инженерной направленности.

Промежуточная аттестация – представление проекта по итогам выполнения кейсов.

Итоговой аттестацией является разработка и защита проекта.

Системы оценки результатов освоения образовательной программы

Освоение программы на каждом уровне заканчивается защитой проектов.

Критерии оценки публичной презентации проекта:

1. Актуальность и значимость проекта (от 0 до 5 баллов).
2. Соответствие результата поставленной цели (0-5 баллов).
3. Уровень завершенности проекта (0-5 баллов).
4. Уровень самостоятельности при выполнении работы (0-3 балла).
5. Качество презентации проекта (оформление, дизайн) (0-3 балла).
6. Качество защиты проекта (устное выступление) и участие каждого в защите (0-3 балла).

7. Умение отвечать на вопросы и отстаивать свою точку зрения (0-3 балла).
8. Анализ научных и инженерных источников, конкурентных подходов к аналогичной или близкой задаче (0-3 балла).

Ученикам, успешно защитившим проект от 20 баллов и выше, посетившим 70% занятий по программе рекомендуется продолжить обучение на следующем уровне. Ученикам, набравшим по результатам защиты проекта менее 20 баллов, а также посетившим менее 70% занятий по программе рекомендуется выбрать обучение по другой дополнительной общеразвивающей программе ДТ «Кванториум».

Методическое обеспечение реализации программы

Используемые педагогические технологии:

- обучение в сотрудничестве;
- игровые технологии;
- информационно-коммуникационные технологии;
- технология проектов;
- кейсовая технология.

Используемые методы обучения:

- словесные (рассуждение, диалог, обсуждение);
- практические (работа в среде разработки, работа с программами);
- проектные методы обучения (дизайн-концепция).

Учебный план

Название	Количество часов в неделю	Количество часов всего
ИТ	4	72
Итого		72

Содержание программы

Раздел введение.

1. Вводный урок. Знакомство с детьми. Техника безопасности при работе в компьютерном классе и электробезопасность. Обзор курса вводного уровня. Оценка уровня знаний. Современное состояние робототехники и микроэлектроники в мире и в нашей стране.

2. Погружение в IT. История развития технологий и языков программирования.

Кейс 1. Платформа Arduino.

В данном кейсе обучающиеся познакомятся с платформами Arduino и детально изучат платформу Arduino UNO. Изучат теоретические основы электричества, научатся читать электрические схемы, познакомятся с резисторами, светодиодами, датчиками, сенсорами и другими элементами для

сборки проектов с участием микроконтроллеров Arduino. Получат навыки по работе в среде программирования Arduino IDE и в среде визуального программирования XOD.

Для реализации данного кейса необходимо следующее материально-техническое обеспечение:

- Компьютер ДТ «Кванториум» - 15 мест;
- Среда программирования Arduino IDE – 15 копий;
- Среда визуального программирования XOD – 15 копий.

Кейс 2. Telegram бот на Python.

При работе с данным кейсом обучающиеся ДТ «Кванториум» познакомятся с языком Python. Его базовыми основами, синтаксисом. Также обучающихся ждет краткий экскурс в историю языка и сферы его применения. Также ребята узнают, что такое Bot API и с помощью полученных знаний смогут написать собственного бота для мессенджера Telegram.

Для реализации данного кейса необходимо следующее материально-техническое обеспечение:

- Компьютеры ДТ «Кванториум» - 15 штук;
- Свободное программное обеспечение Visual Studio Code - 15 копий

Кейс 3. Оконное приложение на C++

Обучающиеся познакомятся с платформой для разработки .NET а так же языками подходящими для реализации проектов. С помощью языка C++ изучат принципы создания оконных приложения и реализуют свои знания в виде готового приложения. К концу кейса обучающие выполняют собственные проекты на платформе и защитят их.

Для реализации данного кейса необходимо следующее материально-техническое обеспечение:

- Компьютер ДТ «Кванториум» - 15 мест;

Свободное программное обеспечение Visual Studio Code - 15 копий

Учебно-тематическое планирование

№ п/п	Название урока	Количество часов всего	Теория	Практика	Форма занятия
Раздел введение.					
1	Вводный урок. ТБ. Обзор курса вводного уровня.	1	1	0	
2	Погружение в IT. История технологий и языков программирования.	1	1	0	
Кейс 1. Платформа Arduino.					
3	Знакомство с платформами Arduino. Детальное изучение Arduino Uno.	2	0,5	1,5	

4	Теоретические основы электричества. Чтение электрических схем.	2	1	1	
5	Знакомство с Arduino IDE. Кейс «Маячок». Программирование: функция digital write.	2	1	1	
Раздел свет					
6	Аналоговые и цифровые сигналы, понятие ШИМ. Знакомство с потенциометром. Кейс «Светильник с нарастающей яркостью».	2	1	1	Результат практикум а
7	Понятие алгоритм и блок-схема. Создание блок-схем в XOD IDE.	2	0,5	1,5	
8	Особенности подключения кнопки к макетной плате. Кейс «Кнопка + светодиод».	4	2	2	
9	Логические переменные и конструкции. Булевы переменные и константы, логические операции.	3	2	1	
10	Кейс «Светофор». Моделирование работы дорожного трехцветного светофора.	2	0,5	1,5	Результат практикум а
Раздел датчики и сенсоры					
11	Sheild расширения IoT модули.	2	0,5	1,5	
12	Обзор ЖК-Дисплея. Вывод данных на ЖК-Дисплей.	3	1	2	Результат практикум а
13	Кейс «Климат-контроль на Arduino».	3	0,5	2,5	
14	Завершение кейса. Подведение итогов.	1	1	0	
Всего часов:		28	11,5	16,5	
Кейс 2. Telegram bot на Python.					
3	Знакомство с языком программирования Python. История и сферы применения.	9	6	3	
4	Обсуждение функциональности и создания бота.	6	2	4	
5	Telegram bot API.	5	2	3	Результат практикум а
6	Завершение кейса. Подведение итогов.	2	1	1	
Всего часов:		22	11	11	
Кейс 3. Оконное приложение на C++					

7	Знакомство с платформой .NET. Обзор языков платформы.	3	2	1	
8	Первое оконное приложение. Программируем на C++ под Windows.	5	2	3	
9	Знакомство с языком F SHARP.	3	2	1	
10	Мультиплатформенная разработка с помощью C# и .NET	5	2	3	
12	Завершение кейса. Подведение итогов.	2	1	1	
Всего часов:		18	9	9	
19	Рефлексия по вводному курсу	2	1	1	
Итого:		72	34,5	37,5	

В результате программы обучающийся должен знать следующие ключевые понятия: напряжение, сопротивление, сила тока, микроконтроллерная платформа, датчик, сервопривод, переменная, тип переменной, область видимости переменной, функция (в программировании), оператор условного перехода (в программировании), задержка в выполнении программы, препроцессор, библиотеки встроенные, внешние, протокол связи, эксперимент, график, статистика, прогноз, мобильное приложение, закон Ома, инфракрасный свет, системы координат, объем геометрической фигуры, отношения величин, измерительная шкала, давление жидкости, объем, расстояние, система счисления. А также знать: основы языка программирования Python, принципы создания telegram бота, начальные алгоритмы создания оконных приложений на языке C++.

Планируемые результаты освоения программы представлены универсальными и предметными (техническими) компетенциями обучающихся.

Универсальные компетенции (Soft Skills):

- Умение генерировать идеи различными методами.
- Умение слушать и слышать собеседника.
- Умение аргументировано отстаивать свою точку зрения.
- Умение искать информацию в свободных источниках и

структурировать ее.

- Умение комбинировать, видоизменять и улучшать идеи.
- Навыки командной работы.
- Умение грамотно письменно формулировать свои мысли.
- Критическое мышление и умение объективно оценивать результаты

своей работы.

- Основы ораторского искусства.

Предметные компетенции (Hard Skills):

- Программирование микроконтроллерных платформ на языке C++.
- Основы алгоритмизации и формализации алгоритмов.
- Работа с датчиками и электронными компонентами на платформе Arduino.
- Проектирование интерфейсов пользователей и разработка приложений для мобильных устройств.
- Базовые навыки по работе с сетевыми устройствами и инструментами.
- Управление учетными записями пользователей и разграничение прав доступа к сетевым ресурсам.
- Основные принципы объектно-ориентированного программирования.
- Основы работы в текстовом редакторе и в программе по созданию презентаций.

Личностные и межличностные компетенции:

- Умение генерировать идеи различными методами.
- Умение слушать и слышать собеседника.
- Умение работать в команде, распределять роли и зоны ответственности.
- Умение аргументировано отстаивать свою точку зрения.
- Умение искать информацию в свободных источниках и структурировать ее.
- Умение комбинировать, видоизменять и улучшать идеи.

- Умение грамотно письменно формулировать свои мысли.
- Критическое мышление и умение объективно оценивать результаты своей работы.
- Основы ораторского мастерства.

Кейс «Маячок»

Кейс «Маячок» позволяет создать свой первый проект в среде Arduino IDE, используя практический метод.

С помощью данного кейса обучающиеся узнают:

- Как мигает встроенный светодиод на плате Arduino UNO.
- Что такое скетч и как написать программу.
- Как включить и зажечь светодиод, используя команду digitalWrite.
- Как выключить светодиод.
- Как сделать задержку в программе с помощью команды delay.

Уровень сложности – простой.

Материально-техническое обеспечение:

1. Плата Arduino UNO – 15 штук.

Пошаговая инструкция:

1. Берем плату Arduino UNO. Убеждаемся, что на ней есть встроенный светодиод.
2. Подключаем плату Arduino к компьютеру через USB-кабель. В результате должны загореться встроенные светодиоды.
3. Открываем на компьютере программу Arduino IDE.
4. Проверяем, что плата Arduino подключена правильно: найдите в меню «Инструменты», подменю «Порт». Оно должно быть активным. Нажмите на этот пункт и посмотрите список портов. Один из них должен быть отмечен галочкой. Если этого не произошло, выберите порт с самым большим номером и отметьте его.
5. Начинаем писать скетч программы:

У нас осталась часть кода и сразу обратите внимание на [два блока со словами setup и loop](#). Это две функции, которые вызываются всегда, когда запускается наш скетч. Блоки ограничены фигурными скобками – все, что внутри них, принадлежит

блоку.

```
1. // LED_BUILTIN - встроенная константа, определяющая номер пина. В Arduino
   // Uno и Nano это 13 пин.
2.
3. void setup() {
4.   pinMode(LED_BUILTIN, OUTPUT); // Установка пина в режим OUTPUT
5. }
6.
7. // Этот блок команд выполняется постоянно
8. void loop() {
9.   digitalWrite(LED_BUILTIN, HIGH); // Включение светодиода
10.  delay(1000);                      // Задержка
11.  digitalWrite(LED_BUILTIN, LOW);  // Выключение светодиода
12.  delay(1000);                      // Задержка
13.
14.  // Когда программа дойдет до этого места, она автоматически продолжится
   // сначала
15. }
```

digitalWrite – это название функции, которая отвечает за подачу напряжения на пин.

LED_BUILTIN – это название внутреннего светодиода. В большинстве плат за этим названием прячется цифра 13. Для плат Uno, Nano можно смело писать 13 вместо LED_BUILTIN.

HIGH – условное название высокого уровня сигнала. Включает светодиод. Можно заменить цифрой 1.

LOW – условное обозначение низкого уровня сигнала. Выключает светодиод. Можно заменить цифрой 0.

delay – функция, которая останавливает выполнение скетча на определенное время. В скобках мы указываем цифру – это количество микросекунд, которые нужно ждать. 1000 – это 1 секунда.

6. Проводим компиляцию скетча, с помощью встроенного функционала Arduino UNO и загружаем на плату.

Как только программа дойдет до конца, контроллер перейдет в начало блока loop и будет выполнять все команды заново. И так раз за разом, целую вечность (пока есть свет). Наш светодиод мигает без остановки.

Кейс «Маячок с мигающим светодиодом»

В данном кейсе обучающиеся научатся работать с электрической схемой на практике, подключать светодиод и токоограничивающий резистор. А также вспомнят и применят навыки из предыдущего кейса «Маячок» с встроенным светодиодом на плате Arduino UNO.

Уровень сложности – простой.

Материально-техническое обеспечение:

- Плата Arduino Uno или Nano – 15 штук;
- Макетная плата для монтажа без пайки – 15 штук;
- Резистор номиналом 220 Ом – 15 штук;
- Светодиод – 15 штук;
- Провода для соединения (папа-папа) – 2*15 штук.

Пошаговая инструкция:

1. Для подключения светодиода и токоограничивающего резистора используется макетная плата. Необходимо соединить все элементы согласно следующей схеме для Arduino UNO. Для Arduino Nano светодиод подключается по той же схеме – к пину 13.

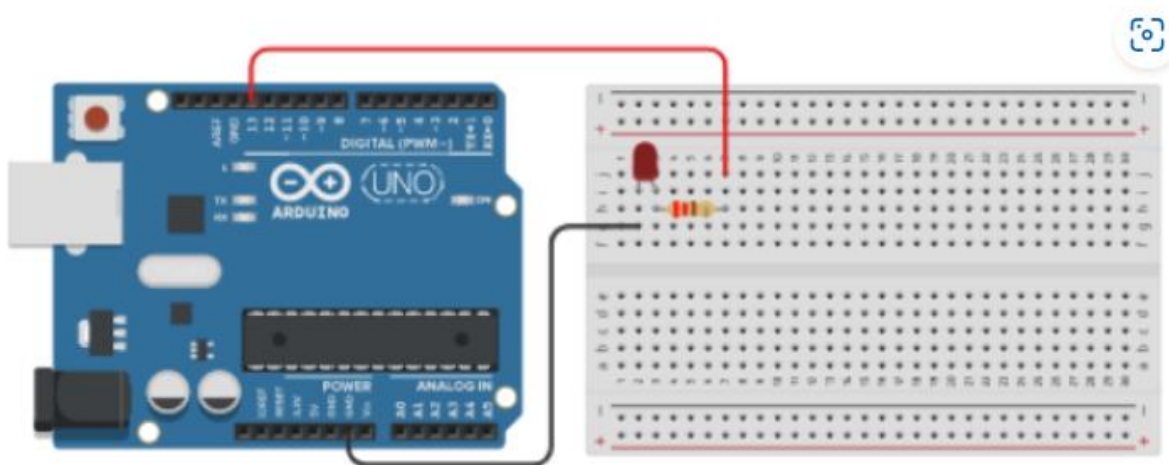


Схема подключения мигающего светодиода к Ардуино

2. Загружаем скетч (программу) для данного кейса из предыдущего кейса «Маячок» и наблюдаем за результатом.

Кейс «Светильник с нарастающей яркостью»

В данном кейсе обучающиеся познакомятся с электрическим устройством «потенциометр» и его функциями, а также с особенностями подключения к плате Arduino UNO.

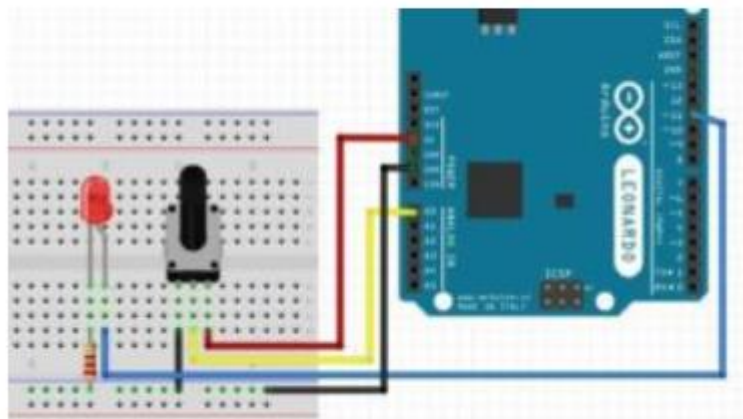
Уровень сложности – простой.

Материально-техническое обеспечение:

- Плата Arduino Uno – 15 штук;
- Макетная плата для монтажа без пайки – 15 штук;
- Резистор номиналом 220 Ом – 15 штук;
- Светодиод – 15 штук;
- Провода для соединения (папа-папа) – 6*15 штук.
- Потенциометр – 15 штук.

Пошаговая инструкция:

1. Примером реализации схемы подключения потенциометра может стать макетная плата с подключенным переменным резистором и светодиодом. При помощи потенциометра будет выполняться управление уровнем яркости свечения.



Для использования меньшего количества проводов от макетной платы к контроллеру следует подключить светодиод и потенциометр проводом земли к длинному рельсу минуса.

Изменяя положение вала подключенного потенциометра, происходит изменение параметра сопротивления, которое вызывает изменение показателя на нулевой пине платы Arduino. Считывание полученного значения напряжения аналогового импульса происходит в скетче с помощью команды [analogRead](#) ().

В плату Arduino встроен аналого-цифровой преобразователь, способный считывать напряжение и переводить его в цифровые показатели со значением от нуля до 1023. При повороте указателя до конечного значения в одном из двух возможных направлений, напряжение на пине равно нулю, и, следовательно, напряжение, которое будет генерироваться составляет 0 В. При повороте вала до конца в противоположном направлении на пин поступает напряжение величиной 5В, а значит числовое значение будет составлять 1023.

2. В этом примере важно понимать, что яркость свечения светодиода управляется не напряжением подаваемым с потенциометра, а кодом.


```

#define PIN_LED    11
#define PIN_POT    A0

void setup()
{
    // Пин, к которому подсоединяется светодиод определяем как выход
    pinMode(PIN_LED, OUTPUT);

    // Пин с переменным резистором является входом
    pinMode(PIN_POT, INPUT);
}

void loop(){
    // Определяем 2 переменные типа int
    int rotat, brightn;

    // Считывание в переменную rotat напряжения с переменного резистора:
    // микроконтроллер будет выдавать числа от 0 до 1023
    // пропорциональны положению поворота вала
    rotat = analogRead(PIN_POT);

    // Преобразуем значение в яркость. Для этого делим rotat на 4, что с учетом

    brightn = rotat / 4;

    // Запись шим значения яркости на светодиод
    analogWrite(PIN_LED, brightn);
}

```

Кейс «Кнопка + светодиод»

С помощью данного кейса обучающиеся на практике проверяют особенности подключения кнопки к плате Arduino UNO. А также научатся устранятьдребезг кнопок с помощью программного кода, использовать подтягивающий резистор и управлять светодиодом.

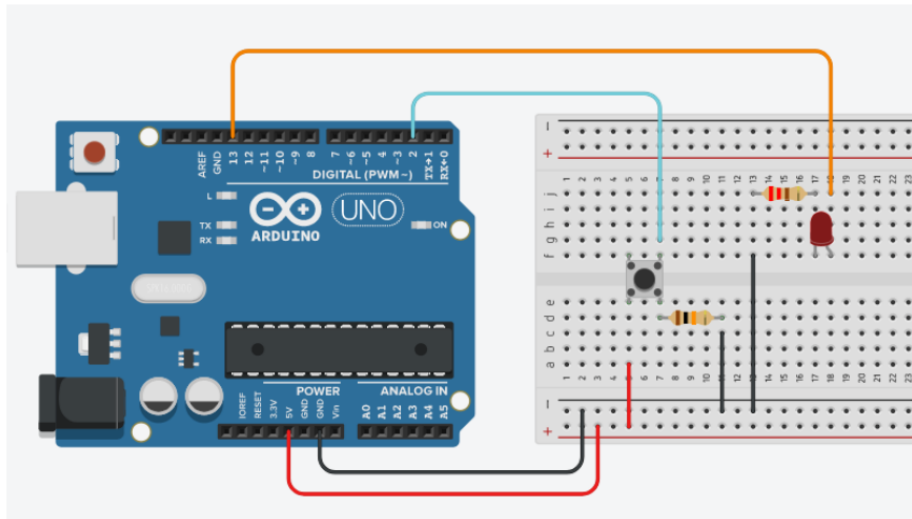
Уровень сложности – простой.

Материально-техническое обеспечение:

- Плата Arduino Uno – 15 штук;
- Макетная плата для монтажа без пайки – 15 штук;
- Резистор номиналом 10 кОм – 15 штук;
- Резистор номиналом 200 Ом – 15 штук;
- Светодиод – 15 штук;
- Провода для соединения (папа-папа) – 7*15 штук.
- Кнопка – 15 штук.

Пошаговая инструкция:

1. Необходимо собрать схему в соответствии с рисунком, представленным ниже.



2. Далее пишем скетч.

```

/*
  Скетч для схемы с использованием тактовой кнопки и светодиода
  Светодиод мигает, пока нажата кнопка.
  Кнопка подтянута к земле, нажатию соответствует HIGH на входе
*/

const int PIN_BUTTON = 2;
const int PIN_LED = 13;

void setup() {
  Serial.begin(9600);
  pinMode(PIN_LED, OUTPUT);
}

void loop() {
  // Получаем состояние кнопки
  int buttonState = digitalRead(PIN_BUTTON);
  Serial.println(buttonState);
  // Если кнопка не нажата, то ничего не делаем
  if (!buttonState) {
    delay(50);
    return;
  }

  // Этот блок кода будет выполняться, если кнопка нажата
  // Мигаем светодиодом
  digitalWrite(PIN_LED, HIGH);
  delay(1000);
  digitalWrite(PIN_LED, LOW);
  delay(1000);
}

```

Кейс «Светофор». Моделирование работы дорожного трехцветного светофора.

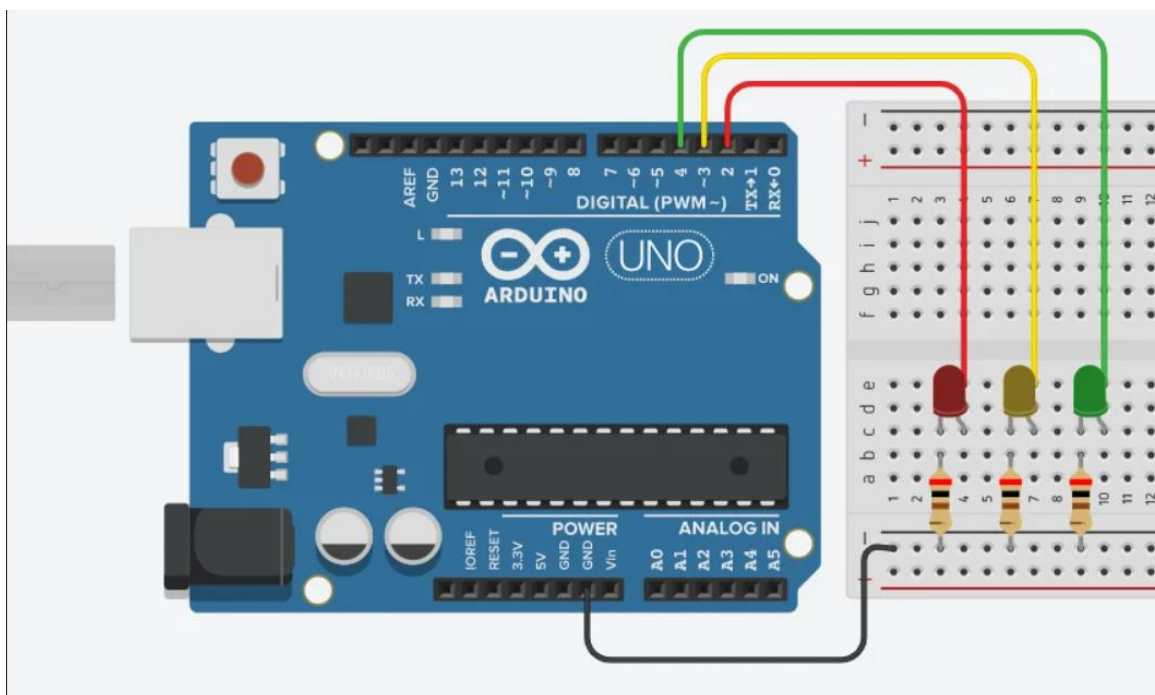
В данном кейсе обучающиеся соберут схему дорожного светофора с помощью платформы Arduino и настроят работу светофора с помощью Arduino IDE. Реализация данного кейса на [Arduino](#) популярна, благодаря наглядному примеру работы распространённого устройства, который мы встречаем ежедневно. Уровень сложности – простой.

Материально-техническое обеспечение:

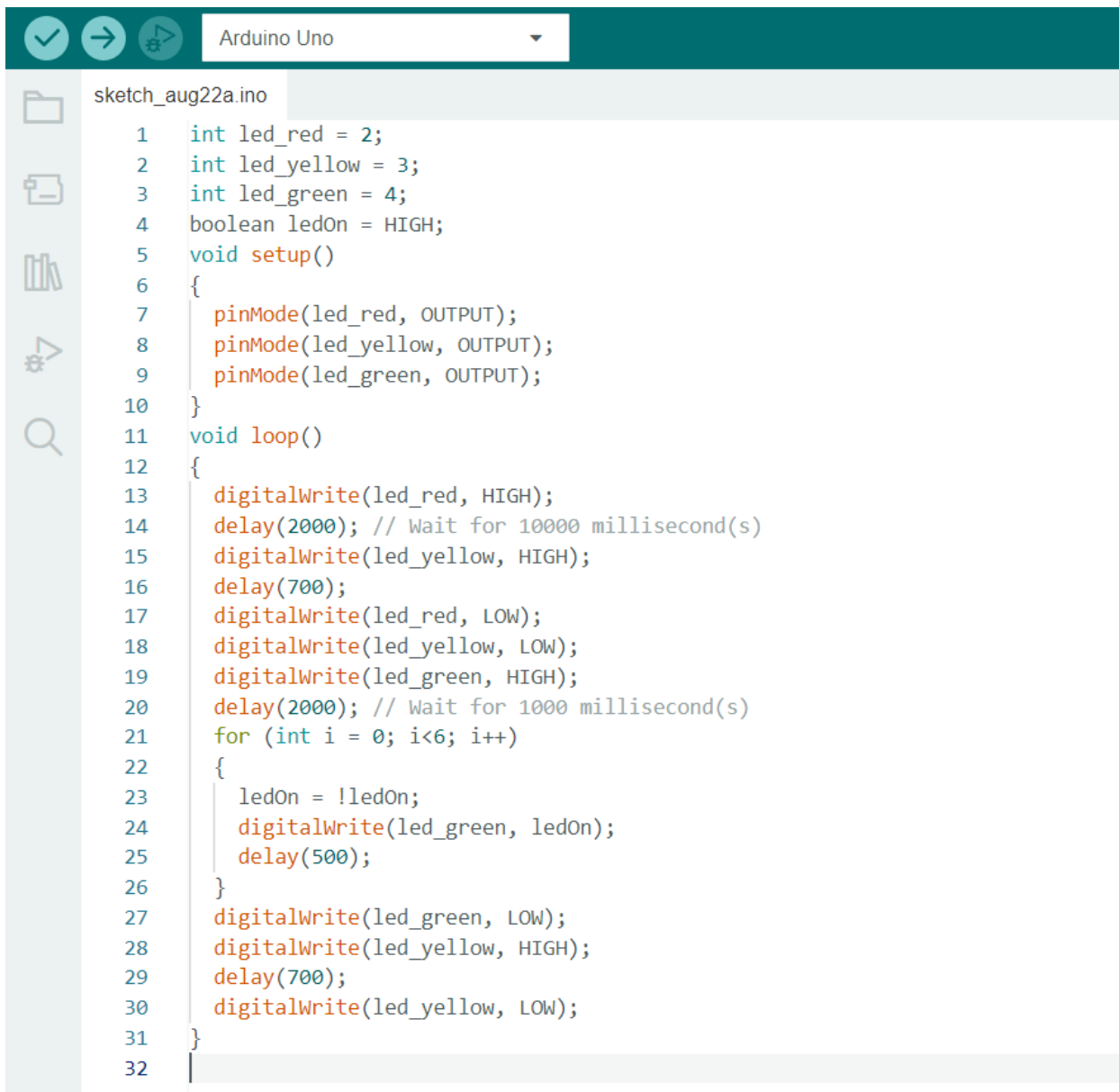
- Плата Arduino Uno – 15 штук;
- Макетная плата для монтажа без пайки – 15 штук;
- Резистор номиналом 200 Ом – 3*15 штук;
- Светодиоды (красный, желтый, зеленый) – 3*15 штук;
- Провода для соединения (папа-папа) – 4*15 штук.

Пошаговая инструкция:

1. Работу с кейсом «светофор» начинаем с подключения всех элементов к микроконтроллеру Arduino UNO, как показано на схеме ниже.



2. После сборки, запускаем среду программирования Arduino IDE и пишем скетч для данного кейса.



```
1  int led_red = 2;
2  int led_yellow = 3;
3  int led_green = 4;
4  boolean ledOn = HIGH;
5  void setup()
6  {
7      pinMode(led_red, OUTPUT);
8      pinMode(led_yellow, OUTPUT);
9      pinMode(led_green, OUTPUT);
10 }
11 void loop()
12 {
13     digitalWrite(led_red, HIGH);
14     delay(2000); // Wait for 10000 millisecond(s)
15     digitalWrite(led_yellow, HIGH);
16     delay(700);
17     digitalWrite(led_red, LOW);
18     digitalWrite(led_yellow, LOW);
19     digitalWrite(led_green, HIGH);
20     delay(2000); // Wait for 1000 millisecond(s)
21     for (int i = 0; i<6; i++)
22     {
23         ledOn = !ledOn;
24         digitalWrite(led_green, ledOn);
25         delay(500);
26     }
27     digitalWrite(led_green, LOW);
28     digitalWrite(led_yellow, HIGH);
29     delay(700);
30     digitalWrite(led_yellow, LOW);
31 }
32
```

Создаем переменные и инициализируем пины, как выходы. Далее для подключения зеленого цвета необходимо применить цикл «for». Цикл выполняется 6 раз, так как при каждом выполнении цикла мы меняем булеву переменную с HIGH на LOW и наоборот. И выполняем либо включение светодиода, либо выключение. По этой причине количество раз выполнения цикла увеличивается в 2 раза.

Кейс «Климат-контроль на Arduino»

В данном кейсе обучающиеся научатся использовать датчик температуры и давления DHT 11, подключать Дисплей [LCD 1602 по I2C](#) и управлять с помощью кнопки значениями на дисплее. По завершению кейса обучающиеся соберут метеостанцию с помощью XOD IDE.

Уровень сложности – средний.

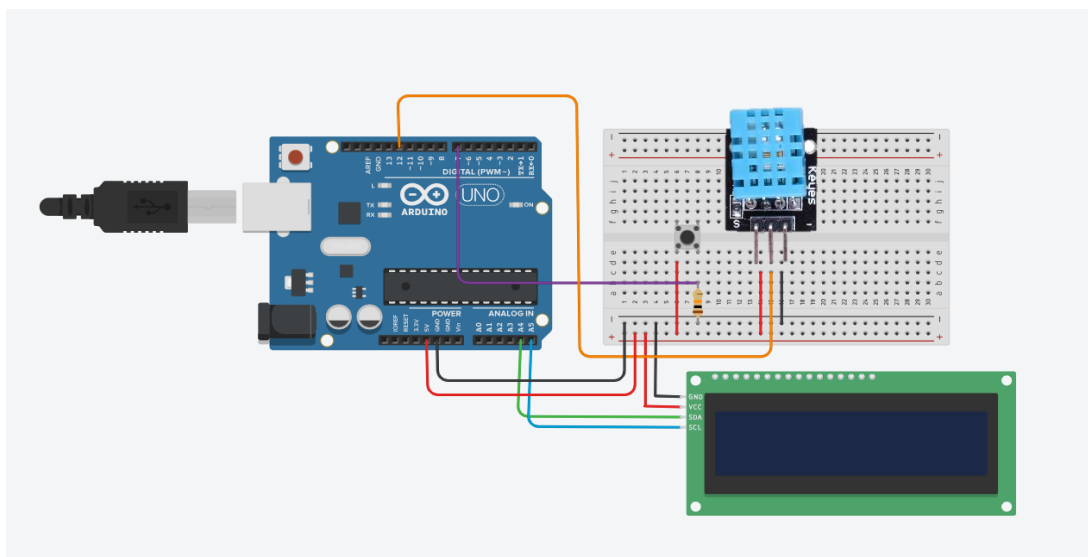
Материально-техническое обеспечение:

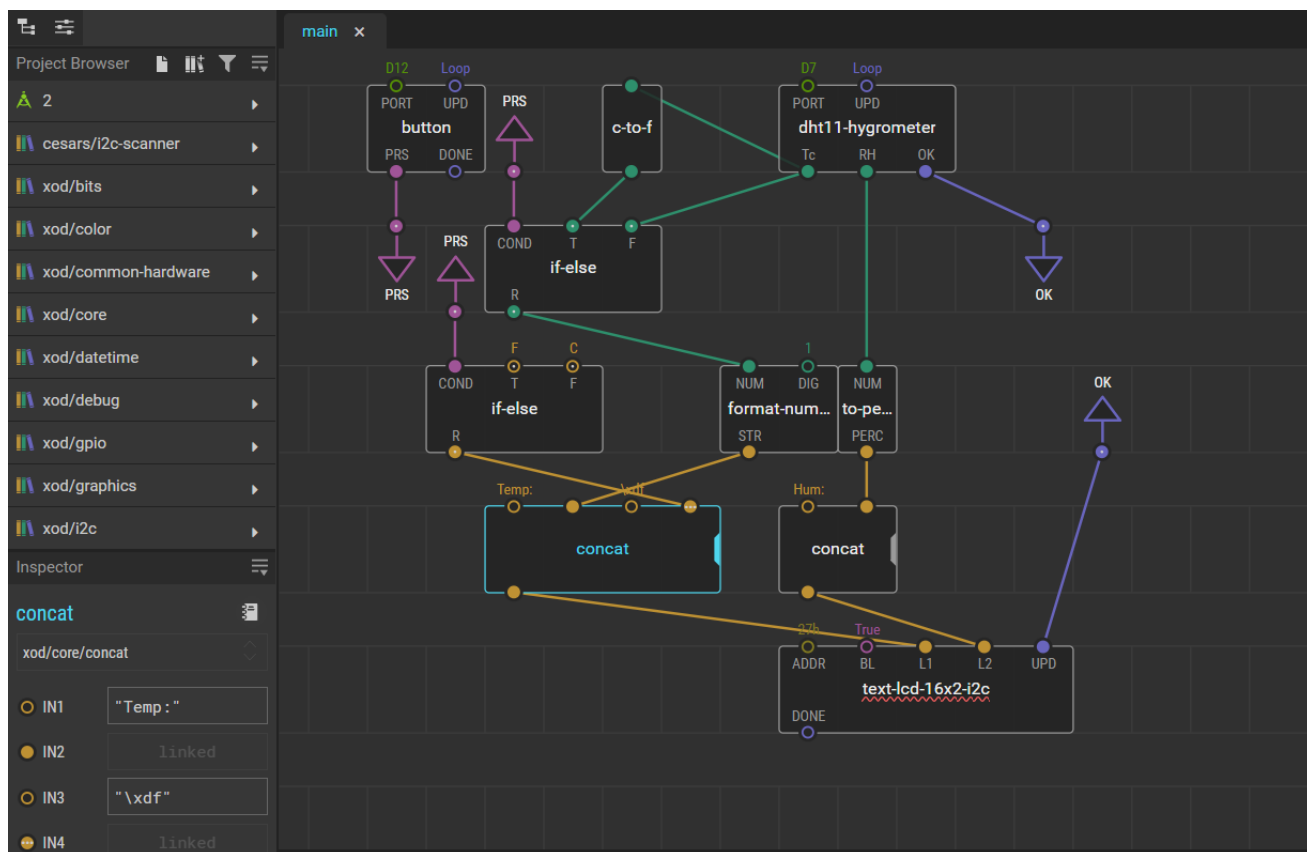
- Плата Arduino Uno – 15 штук;
- Макетная плата для монтажа без пайки – 15 штук;
- Датчик измерения температуры и влажности DHT 11 – 15 штук;
- Соединительные провода (папа-папа) – 5*15 штук.
- Соединительные провода (мама-папа) – 7*15 штук.
- Дисплей LCD 1602 по I2C – 15 штук;
- Кнопка 3х-контактная – 15 штук.

Пошаговая инструкция:

1. Необходимо собрать все элементы, как показано на схеме и подключить к плате Arduino UNO.
2. Запустить XOD IDE.

Добавить все необходимые ноды в проект Климат-контроль в среде XOD и правильно соединить между собой, как показано на рисунке ниже.





Нода «dht11-hygrometer» управляет датчиком DHT11.

Нода «Text-lcd-16x2-i2c» управляет дисплеем (L1 и L2 строки на дисплее для вывода значений с датчика DHT11).

Нода «Format-number» переводит значения с датчика в числовой вид, нода to-percent указывает значение влажности в процентах.

Нода «concat» (concatenates) позволяет соединять значения в строке на экране дисплея. Первая нода указывает значение температуры «Temp:» (L1-строка 1). Вторая нода «concat» указывает значение влажности «Hum:» (L2-строка 2).

Нода «button» управляет параметрами кнопки.

Нода «C-to-F» позволяет переводить значения температуры из градусов Цельсия в Фаренгейты.

Нода «if-else» задает условие (если кнопка зажата, состояние 1, тогда L1-значение температуры в градусах Цельсия, если кнопка в состоянии 0, тогда значение температуры отображается по Фаренгейту).

3. После того, как проект создан и все ноды соединены верно, необходимо загрузить программу на плату Arduino UNO и произвести отладку.
4. После успешной загрузки программы на плату, на дисплее должны отображаться значения температуры и влажности.

