

**Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа № 41 имени В.В.Сизова»
города Курска**

ИТОГОВЫЙ ИНДИВИДУАЛЬНЫЙ ПРОЕКТ

**Тема:
«Arduino»**

**Выполнил:
Воронков Дмитрий Николаевич,
ученик 9 класса Г**

**Куратор проекта:
Склярова Елена Александровна,
учитель информатики**

Допуск к защите: _____

Подпись куратора: _____

Курск, 2024 г.

Оглавление

Введение.....	4
1 Платформа Arduino	5
1.1 Язык программирования C++	5
1.2 Программирование Arduino	6
1.2.1 Работа с Arduino IDE	6
1.2.2 Программа для Arduino	8
2 Создание пианино	9
Заключение	11
Список использованных источников и литературы	12
Приложение 1	13

Введение

Тема проекта: «Arduino».

Тема проекта создания пианино на базе Arduino имеет высокую **актуальность** по нескольким причинам:

1. Интерес учащихся к технологиям: проект на основе Arduino позволяет ученикам познакомиться с программированием, электроникой и механикой, что может стимулировать их интерес к техническим наукам и инженерии.

2. Интеграция предметов: создание пианино на Arduino объединяет знания из различных предметных областей, таких как физика, математика, информатика, музыка. Это способствует целостному обучению и развитию универсальных компетенций.

3. Развитие творческого мышления: проектирование и программирование пианино позволяет учащимся проявить свою творческую индивидуальность, экспериментировать и находить нестандартные решения.

4. Современные технологии: использование Arduino, платформы с открытым исходным кодом, позволяет учащимся освоить современные технологии и методы работы, которые востребованы на рынке труда.

5. Возможность индивидуализации обучения: каждый ученик может создать уникальное пианино, настроенное под свои предпочтения и потребности, что способствует индивидуальной работе и развитию личностных качеств.

Таким образом, проект создания пианино на Arduino представляет собой увлекательную и актуальную практическую задачу для учащихся, способствующую развитию разносторонних навыков и знаний.

Проблема проекта: Создание пианино через платформу Arduino, имея малый опыт работы с ней.

Цель проекта: данная работа заключается в изучении языке C++, программировании на нем, путем создания практического устройства - Пианино.

Для успешного достижения цели следует выполнить ряд **задач**:

1. Изучить платформу Arduino и язык C++ для работы с ней.
2. Узнать о создании подобных устройств с помощью набора Arduino
3. Создать объект проекта - Пианино на плате Arduino UNO

Тип проекта: практико-ориентированный

Проектным продуктом является: пианино, реферат, презентация

План работы над проектом:

1. Сбор необходимых источников (сентябрь 2023г.).
2. Изучение функционала языков (октябрь-ноябрь 2023г.)
3. Сборка и написания кода для пианино (декабрь-январь 2024г.).
4. Написание проекта (февраль 2024г.).
5. Создание презентации для защиты (март 2024г.).

1 Платформа Arduino

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

Свою популярность Arduino приобрела благодаря простоте и дружелюбности. Даже полный ноль в программировании и схемотехнике может освоить основы работы с Arduino за пару часов. Этому способствуют тысячи публикаций, учебников, заметок в интернете и отличная серия видеоуроков по Arduino на русском языке.

Программы для Arduino пишутся на обычном C++, дополненным простыми и понятными функциями для управления вводом-выводом на контактах. Если вы уже знаете C++ - Arduino станет дверью в новый мир, где программы не ограничены рамками компьютера, а взаимодействуют с окружающим миром и влияют на него. Если же вы новичок в программировании — не проблема, вы с лёгкостью научитесь, это просто.

Для удобства работы с Arduino существует бесплатная официальная среда программирования Arduino IDE, работающая под Windows, Mac OS и Linux. С помощью неё загрузка новой программы в Arduino становится делом одного клика, только лишь подключите плату к компьютеру через USB.

Вам не понадобится паяльник. Полноценные устройства можно собирать, используя специальную макетную доску, перемычки и провода абсолютно без пайки. Конструирование ещё не было таким быстрым и простым¹.

1.1 Язык программирования C++

Программирование микроконтроллеров Arduino осуществляется на языке программирования C++. Этот язык является низкоуровневым, поэтому считается сложным и имеет высокий порог вхождения. Но для программирования Arduino используется упрощенная версия этого языка программирования. Так же для упрощения разработки прошивок существует

¹ Введение в Arduino. URL: <https://alexgyver.ru/lessons/before-start/>

множество функций, классов, методов и т.п. Благодаря этому работать с этими микроконтроллерами очень удобно и легко.

1.2 Программирование Arduino

Для работы с Ардуино нам понадобится компьютер, USB data-кабель и некоторый софт.

Компьютер с Windows 7 или выше. На WinXP новые версии Arduino IDE либо работают очень плохо, либо не работают вовсе. MacOS и Linux-подобные оси тоже подходят, так как среда разработки Arduino IDE кроссплатформенная. Также программировать Ардуино можно с планшета или даже смартфона под Андроид при помощи официального приложения.

USB кабель должен быть именно data-кабелем, по такому кабелю передаются данные (например, таким подключается к компьютеру смартфон/фотоаппарат/плеер). Есть три типа USB штекера, которые используются на платах Ардуино: mini-USB (Arduino Nano), micro-USB (Arduino micro) и USB type B (Arduino UNO, MEGA), бывают и китайские поделки на USB type C².

Arduino IDE.

Программирование Arduino осуществляется в приложении Arduino IDE. Среда разработки Arduino IDE, скачивается с официального сайта и устанавливаем актуальную версию для своей операционной системы. В окне загрузки жмём JUST DOWNLOAD, чтобы начать загрузку без пожертвования.

1.2.1 Работа с Arduino IDE

Рассмотрим основную программу, с которой будем работать – Arduino IDE. «IDE» расшифровывается как интегрированная среда разработки, и в нашем случае представляет собой блокнот, в котором мы пишем код[12], препроцессор и компилятор, которые проверяют и компилируют код, и инструменты для загрузки, которые загружают код выбранным способом. IDE написана на языке java.

Описание того, в чем могут возникнуть сложности при изучении программы:

«Интерфейс».

Сначала рассмотрим общий вид программы, т.е. как она выглядит после запуска. В самом центре – блокнот, то самое место, где пишется код. По умолчанию уже написаны два стандартных блока, setup и loop.

² Ведение в Arduino. URL:<https://alexgyver.ru/lessons/before-start/>

Проверить – компиляция (сборка, проверка на ошибки...) кода без его загрузки в плату. То есть код можно написать и проверить на ошибки даже не подключая плату к компьютеру

Загрузить – компиляция и загрузка прошивки в плату

Создать/открыть/сохранить – тут всё понятно

Монитор порта – кнопка открывает монитор последовательного порта для общения с платой

Меню вкладок – работа с вкладками, о них поговорим ниже

Текущее состояние – тут выводится краткая информация о последнем действии³.

Конфигурация оборудования – выводится название выбранной платы, версии микроконтроллера и номер выбранного СОМ порта.

«Меню».

Пункты меню, которые могут вызвать трудности в понимании:

Папка со скетчами – список скетчей, которые сохранены туда, куда программа предлагает их сохранять по умолчанию (Документы/Arduino)

Примеры – список установленных библиотек с подсписком примеров к каждой. Весьма полезная штука

«Настройки».

Куча интересных настроек на свой вкус. Из них отмечу:

Размещение папки скетчей – куда по умолчанию сохраняются скетчи

Показать подробный вывод – показывает подробный лог при компиляции и загрузке, нужно при поиске багов и непонятных глюков/ошибок.

Проверить/компилировать – то же, что кнопка галочка

Загрузка – то же, что кнопка загрузка

Подключить библиотеку – подключает в код библиотеку, с директивой include, всё как нужно.

Управлять библиотеками... – открывает менеджер библиотек, из которого можно установить библиотеки из официального списка.

Добавить файл... – подключает в код внешний файл

Архивировать скетч – сохраняет скетч как .zip архив

Исправить кодировку и перезагрузить – полезная штука, если код был предварительно скачан из интернета и сбилась кодировка.

Плата – выбор платы, куда хотим загрузить прошивку

Процессор – выбор процессора, который стоит на плате. Обычно он один на выбор, но иногда есть варианты

Порт – СОМ порт, к которому подключена плата. Это всегда порт, отличный от СОМ1 (системный порт)⁴.

³Фаблаб «Робототехника Ардуино». URL: https://roboteh.club/course/robototexnika_arduino/?ysclid=ltxf68gtqb878918017

⁴Официальный сайт Arduino. URL: <https://www.arduino.cc/>

1.2.2 Программа для Arduino

Так выглядит программа в приложении Arduino IDE.

Пояснение команд кода:

setup()

Функция `setup()` вызывается, когда скетч начинает исполняться. Используется для определения режимов работы выводов, запуска используемых библиотек и т.д. Функция `setup` выполняется только один раз, после каждой подачи питания или перезагрузки платы Arduino.

PinMode ()

Конфигурирует режим работы указанного вывода: как вход либо как выход.

`pinMode(pin, mode)`

`pin`: номер вывода, режим работы которого будет конфигурироваться.

`mode`: принимает значения INPUT, OUTPUT

digitalRead()

Отправляет на цифровой вывод значение [HIGH](#) или [LOW](#).

Если функцией `pinMode()` вывод сконфигурирован как выход (OUTPUT), то при выполнении функции `digitalWrite()` его напряжение будет изменено на соответствующее значение: 5 В (либо 3.3 В для плат, работающих от 3.3В) при отправке HIGH, 0 В (земля) — при LOW⁵.

Оператор if()

if (условие){}

`if` проверяет условие и если оно верно то выполняется следующий блок кода, заключенный в фигурные скобки. Если блок кода состоит из одной команды то фигурные скобки можно опустить.

`buzzer`

предназначен для генерации звукового сигнала в роботах

В коде используется библиотека «IRremote.h». Для ее загрузки перейдем в Инструменты/Управлять библиотеками.... В менеджере библиотек находим IRremote и устанавливаем. Команды из этой библиотеки пояснены в коде. Подробное описание библиотеки⁶.

⁵ Фаблаб «Робототехника Ардуино». URL:<https://roboteh.club/course/robototexnika-arduino/?ysclid=ltxf68gtqb87891801>

⁶ Официальный сайт Arduino. URL:<https://www.arduino.cc/>

2 Создание пианино

Для начала, нам понадобятся детали и устройства:
Провода (12 штук); Кнопки (4 штуки); Пьезоэлемент (1 штука);
Резисторы (4 штуки) и самые главные устройства: макетная плата и
плата Arduino UNO.

Ставим все устройства как показано на рисунке 1.

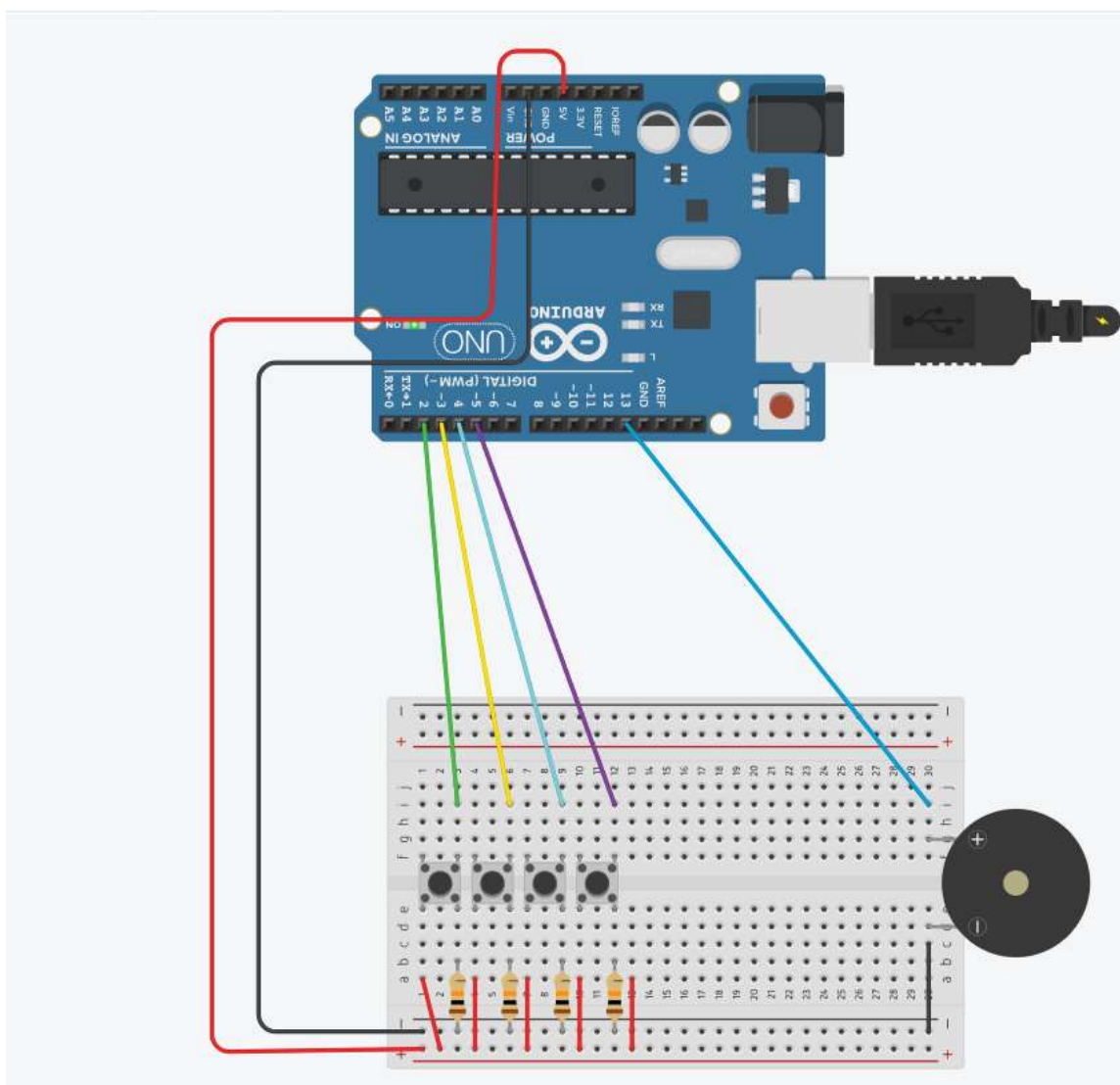


Рисунок 1 – Сбор и установка необходимых элементов

Далее вводится во вкладке «Код» наш код (рисунок 2). Если всё сделано верно, то из пьезоэлемента будут идти звуки разных тонов, нажимая на кнопки.


```

1 int but1 = 2;
2 int but2 = 3;
3 int but3 = 4;
4 int but4 = 5;
5
6
7 int buzzer = 10;
8
9 void setup()
10 {
11
12   pinMode(but1, INPUT);
13   pinMode(but2, INPUT);
14   pinMode(but3, INPUT);
15   pinMode(but4, INPUT);
16
17   pinMode(buzzer, OUTPUT);
18
19 }
20
21
22 void loop()
23 {
24
25   int b1 = digitalRead(but1);
26   int b2 = digitalRead(but2);
27   int b3 = digitalRead(but3);
28   int b4 = digitalRead(but4);
29
30
31   if( b1 == 1 ){
32     tone(buzzer, 300, 100);
33   }
34   if( b2 == 1 ){
35     tone(buzzer, 400, 100);
36   }
37   if( b3 == 1 ){
38     tone(buzzer, 500, 100);
39   }
40   if( b4 == 1 ){
41     tone(buzzer, 600, 100);
42   }
43 }
44
45
46 delay(10);

```

Рисунок 2 – Код для пианино

Заключение

В ходе работы над проектом по созданию пианино на Arduino я нашел наиболее интересным и запоминающимся то, как можно объединить программирование, электронику и музыку в одном проекте. Меня поразило, как можно создать функциональное музыкальное устройство с использованием простых компонентов и технологий.

В процессе работы над проектом я научился основам программирования на Arduino, понял принципы работы с электроникой и изучил основы создания музыкальных проектов. Также я улучшил свои навыки в планировании и организации работы, а также в решении технических проблем и ошибок.

Что касается оценки результатов моего проекта, я доволен тем, что мне удалось создать рабочий прототип пианино на Arduino. Однако, я столкнулся с некоторыми техническими трудностями, которые потребовали дополнительного времени и усилий для их решения.

Я считаю, что явным успехом было то, что я смог добиться звучания музыки при нажатии клавиш.

Если бы я начал свой проект заново, я бы уделил больше времени изначальной концепции и планированию. Также я бы более детально изучил документацию и примеры кода, чтобы избежать некоторых ошибок и ускорить процесс разработки. В целом, я считаю, что проект был успешным, но есть место для улучшения в организации работы и технической подготовке.

Список использованных источников и литературы

1. Введение в Arduino. URL:<https://alexgyver.ru/lessons/before-start/> (дата обращения: 26.09.2023г.).
2. Официальный сайт Arduino. URL:<https://www.arduino.cc/> (дата обращения: 26.09.2023г.).
3. Программирование Arduino. URL:<https://all-arduino.ru/programmirovanie-arduino/> (дата обращения: 26.10.2023г.).
4. Фаблаб «Робототехника Ардуино». URL:<https://roboteh.club/course/robototekhnika-arduino/?ysclid=ltxf68gtqb878918017> (дата обращения: 30.02.2024г.).

Приложение 1

Пояснения терминов, встречающийся в тексте

1. Макетная плата — универсальная печатная плата для сборки и моделирования прототипов электронных устройств. Макетные платы подразделяются на два типа: для монтажа посредством пайки и без таковой.
2. Периферийное устройство (англ. peripheral) — аппаратура, которая позволяет вводить информацию в компьютер или выводить её из него.
3. Окно прозрачности (англ. Transmission Window, Telecom Window) — диапазон длин волн оптического излучения, в котором имеет место меньшее, по сравнению с другими диапазонами, затухание излучения в среде, в частности — в оптическом волокне.
4. 4 Фотодиод — приёмник оптического излучения, который преобразует попавший на его фоточувствительную область свет в электрический заряд за счёт процессов в p-n-переходе.
5. Печатная плата (англ. printed circuit board, РСВ, или printed wiring board, РWB) — пластина из диэлектрика, на поверхности и/или в объёме которой сформированы электропроводящие цепи электронной схемы.
6. Пин — контакт (ножка) в электронике для соединения двух элементов схемы.
7. Софт - программное или математическое обеспечение.
8. Код — взаимно однозначное отображение конечного упорядоченного множества символов, принадлежащих некоторому конечному алфавиту, на иное, не обязательно упорядоченное, как правило более обширное множество символов для кодирования передачи, хранения или преобразования информации.
9. Библиотека (от англ. library) в программировании сборник подпрограмм и ли объектов, используемых для разработки программного обеспечения (ПО).
10. Подробное описание библиотеки <https://github.com/Arduino-IRremote/Arduino-IRremote#converting-your-2x-program-to-the-3x-version>