Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»

ЛАБОРАТОРНЫЙ ПРАКТИКУМ «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ПЛАТФОРМЕ ARDUINO»

Выпускная квалификационная работа
по направлению подготовки 44.03.04 Профессиональное обучение
(по отраслям)
профилю подготовки «Энергетика»
профилизации «Компьютерные технологии автоматизации и управления»

Идентификационный номер ВКР: 315

Екатеринбург 2017

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ
Заведующая кафедрой ИС
_____ Н. С. Толстова
«____» _____ 2017 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЛАБОРАТОРНЫЙ ПРАКТИКУМ «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ПЛАТФОРМЕ ARDUINO»

Исполнитель:
обучающийся группы № КТэ-402 К. П. Ушанов
Руководитель:
старший преподаватель Г. Л. Нечаева
Нормоконтролер:
старший преподаватель Т. В. Рыжкова

Екатеринбург 2017

АННОТАЦИЯ

Выпускная квалификационная работа состоит из лабораторного практикума «Основы программирования на платформе Arduino» и пояснительной записки на 61 странице, содержащей 36 рисунков, 4 таблицы и 25 источников литературы, а также же 1 приложение на 2 страницах.

Ключевые слова: ЛАБОРАТОРНЫЙ ПРАКТИКУМ, ПРОГРАММИРОВАНИЕ, СРЕДА РАЗРАБОТКИ, ПЛАТФОРМА ARDUINO.

Ушанов, К. П. Лабораторный практикум «Основы программирования на платформе Arduino»: Выпускная квалификационная работа / К. П. Ушанов; Рос. гос. проф.-пед. ун-т, Ин-т инж.-пед. образования, Каф. Информационных систем и технологий. — Екатеринбург, 2017. — 61 с.

В работе рассмотрены основы изучения программирования Arduino; разработка лабораторного практикума; список использованных источников; приложение.

Целью работы является разработка лабораторного практикума «Основы программирования на платформе Arduino».

В соответствие с поставленной целью в работе решены следующие задачи: проведено сравнение сред разработки Arduino IDE и Turbo Pascal; сравнен синтаксис программного кода; представлены примеры написания программного кода в среде Arduino IDE и Turbo Pascal; изучены основы программирования на платформе Arduino; рассмотрены требования, к созданию лабораторного практикума; разработана структура и содержание лабораторного практикума; реализован лабораторный практикум.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Появление первых микропроцессоров ознаменовало начало новой эры в развитии микропроцессорной техники. Наличие в одном корпусе большинства системных устройств, сделало микроконтроллер подобным обычному компьютеру. Раньше они назывались однокристальные микро-ЭВМ [15]. Чтобы собрать устройство и микроконтроллер, необходимо знать основы схемотехники, устройство и работу конкретного процессора, уметь программировать на ассемблере и изготавливать электронную технику. В настоящее время, все изменилось. Сейчас существует такое устройство, как проект Arduino.

Arduino представляет собой наборы, состоящие из готового электронного блока и программного обеспечения. Электронный блок — это печатная плата с микроконтроллером и элементами, которые необходимы для работы. Вторая часть — это программное обеспечение для создания программ, включающее в себя простую среду разработки и язык программирования C/C++ [15].

Актуальность выбранной темы заключается в обеспечении необходимости доступа к программированию и разработке робототехнических устройств не только профессионалам, но и заинтересованным обучающимся. Постановление Правительства РФ от 15 апреля 2014 г. № 295 «Об утверждении государственной программы Российской Федерации «Развитие образования» на 2013-2020 годы», одним из аспектов которого является развитие робототехники. Так же актуальность робототехники в современное время. Широкий набор учебно-методических материалов, готовых модулей и библиотек программ позволит мотивировать начинающих пользователей к созданию различных робототехнических устройств в рамках данной программы.

Особенностью лабораторного практикума является ориентация его на старшеклассников и студентов младших курсов, обучающихся по программе дополнительного профессионального образования.

В данном лабораторном практикуме будут рассмотрены такие лабораторные работы:

1. Знакомство со средой разработки Arduino IDE. Установка и начало работы.
2. Управление светодиодами средствами виртуальной среды Autodesk Circuits с применением условного оператора if.
3. Управление светодиодами с помощью кнопок средствами виртуальной среды Autodesk Circuits» с применением условного оператора выбора case.
4. Управление RGB-светодиодом средствами виртуальной среды Autodesk Circuits» с применением цикла for.
5. Управление жидкокристаллическим дисплеем LCD 16x2 с применением оператора цикла while.
6. Мини-проект «Ультразвуковой дальномер».

По итогам прохождения лабораторного практикума проверкой полученных знаний является самостоятельный проект с использованием представленных электронных компонентов.

Объект исследования — процесс изучения основ программирования на платформе Arduino.

Предмет исследования — лабораторный практикум «Основы программирования на платформе Arduino».

Цель работы — разработать лабораторный практикум «Основы программирования на платформе Arduino».

Педагогический адрес — практикум разработан для системы дополнительного образования школьников старших классов и студентов младших курсов колледжей и техникумов, обучающихся в учебно-техническом центре ООО «Омега-1» г. Екатеринбурга.

В соответствие с поставленной целью в работе определены следующие **задачи**:

1. Провести сравнение сред разработки Arduino IDE и Turbo Pascal.
2. Сравнить синтаксис программного кода, представить примеры написания программного кода в среде Arduino IDE и Turbo Pascal.
3. Изучить основы программирования на платформе Arduino.
4. Рассмотреть требования, предъявляемые к созданию лабораторного практикума.
5. Разработать структуру и содержание лабораторного практикума.
6. Реализовать лабораторный практикум.

Новизна выпускной квалификационной работы лабораторный практикум «Основы программирования на платформе Arduino», которая будет применяться в системе дополнительного образования, состоит в том, что был разработан лабораторный практикум «Основы программирования на платформе Arduino» в сравнении с курсом «Основы программирования Turbo Pascal» и адаптирован для учебно-технического центра «Омега-1».

1 ОБУЧЕНИЕ ОСНОВАМ РОБОТОТЕХНИКИ

Робототехника — область науки и техники, ориентированная на создание роботов и робототехнических систем [4]. Робототехника возникла на основе мехатроники и кибернетики подразумевает знание механики, электроники, программирования. Является универсальным инструментом для образования. Вписывается в дополнительное образование, в преподавание предметов школьной программы, в четком соответствии с ФГОС [21].

Предмет робототехники — это создание и применение роботов и других средств робототехники различного назначения. Возникнув на основе кибернетики и механики, робототехника в свою очередь породила новые направления развития и самих этих наук. Для кибернетики это связано, прежде всего, с интеллектуальным управлением, которое требуется для роботов, а для механики с – многозвенными механизмами типа манипуляторов [25].

Робототехника является одним из важнейших направлений научно-технического прогресса (НТП), в котором проблемы механики и новых технологий соприкасаются с проблемами искусственного интеллекта [3]. В настоящее время проблемы робототехники рассматриваются более масштабно. Происходит внедрение в учебный процесс, организовываются соревнования, конкурсы, обмен новыми идеями, знаниями, технической информацией и поддерживается государственной программой «Развитие образования» на 2013-2020 годы.

Современные общество старается внедрить роботов в повседневную жизнь путем применения роботов в различных сферах жизни, а так же заменить, усовершенствовать и модернизировать различные процессы жизни. А это значит, что люди, обладающие знаниями робототехники,

программирования остаются востребованными на рынке труда. Следовательно, вопрос внедрения робототехники актуален.

Применяются различные специальные робототехнические комплексы [17]:

- Lego Mindstorms. Специальный конструктор нового поколения;
- Конструктор Fischertechnik. Данный конструктор является развивающим. Он подходит как для детей, так и для подростков и студентов;
- Scratch Board;
- Arduino;
- Конструкторы УМКИ. Такие модули оснащены микропроцессором, а также наборами датчиков.

Лабораторный практикум «Основы программирования на платформе Arduino» адаптирован для системы дополнительного образования и предназначен для школьников старших классов и студентов младших курсов колледжей и техникумов, обучающихся в учебно-техническом центре ООО «Омега-1» г. Екатеринбурга и предполагает последовательное изучение языков программирования, начиная от курса «Основы программирования Turbo Pascal», до языков объектно-ориентированного программирования.

1.1 Изучение основ программирования Arduino

Особенности преподавания программирования в вузах, учебных центрах требуют достаточного и прочного усвоения базовых знаний.

В настоящее время профессиональная подготовка в области программирования сталкивается с проблемой недостатка учебного времени, слабой подготовкой и высокими требованиями на рынке труда. Путь решения проблемы — это формирование достаточной мотивации у обучающихся [12].

С течением времени интерес к программированию продолжает расти, так как специальность программиста является очень востребованной не только в России, но и во многих других странах.

Программа — это описание процесса обработки информации. При выполнении программы рассчитывается совокупность выходных значений исходя из совокупности переменных или постоянных входных значений. Цель выполнения программы — сбор данных либо получение отклика на входные значения [18].

Программа состоит из строк текста. Каждая строка содержит один или несколько арифметических или управляющих операторов.

Компьютерная программа — это четко формализованный план, состоящий из команд для контроллера (системы принятия решений). Контроллер поочередно читает команды и исполняет [10]. Команды внутри любого цифрового устройства, например, Arduino, закодированы нулями и единицами, а называется это *двоичным представлением чисел*, то есть вся информация перед поступлением в контроллер перекодируется из привычной для нас десятичной системы счисления в двоичную. Таким образом, при выполнении логических или арифметических операций контроллер сравнивает, делит, вычитает, выполняет прочие действия именно над двоичными числами. Эти числа хранятся в последовательных ячейках памяти, имеющие определенные адреса.

При поступлении питания на контроллер Arduino автоматически начинается выполнение той программы, которая была в него загружена, если же программа отсутствует или написана некорректно, то происходит сбой, который либо останавливает выполнение команд, либо приводит к *зависанию* программы. Номер выполняемой программа хранится в специальной ячейке памяти, которая называется *счетчиком команд* [10]. Этот номер изменяется на следующий при выполнении арифметической операции, но может измениться и на любой адрес, если выполнялась

логическая команда, результатом которой стал переход на некоторый пункт плана, отличный от следующего по порядку.

Для визуального представления компьютерных программ при изучении курса «Основы программирования Turbo Pascal» учащиеся изучают их графическую запись, называемую *блок-схемой алгоритма*.

Алгоритмом называют упорядоченную определенным образом последовательность инструкций программы [18]. Алгоритм, не имеющий логических блоков и выполняемый строго по пунктам от начала и до конца, называется линейным (рисунок 1.1).

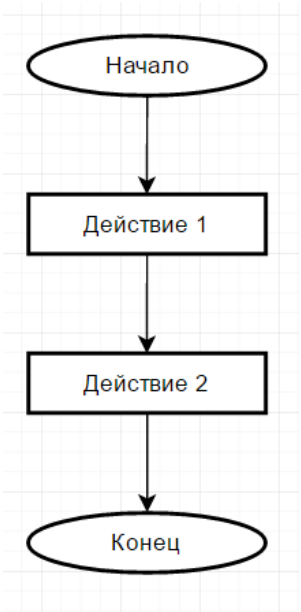


Рисунок 1.1 — Линейный алгоритм

Если алгоритм после достижения последнего пункта подразумевает бесконечное повторение, его называют *циклическим* (рисунок 1.2).

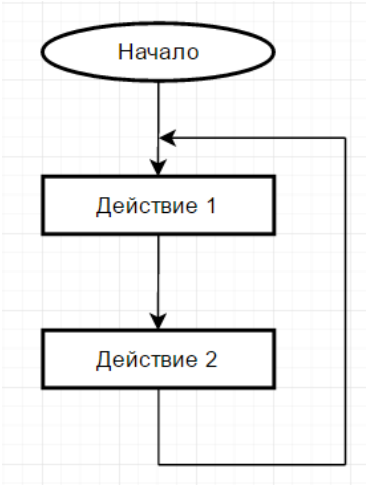


Рисунок 1.2 — Циклический алгоритм.

При наличии в алгоритме анализа некоторого значения и принятия решения в результате такого анализа, этот алгоритм называется *разветвляющимся* (рисунок 1.3).

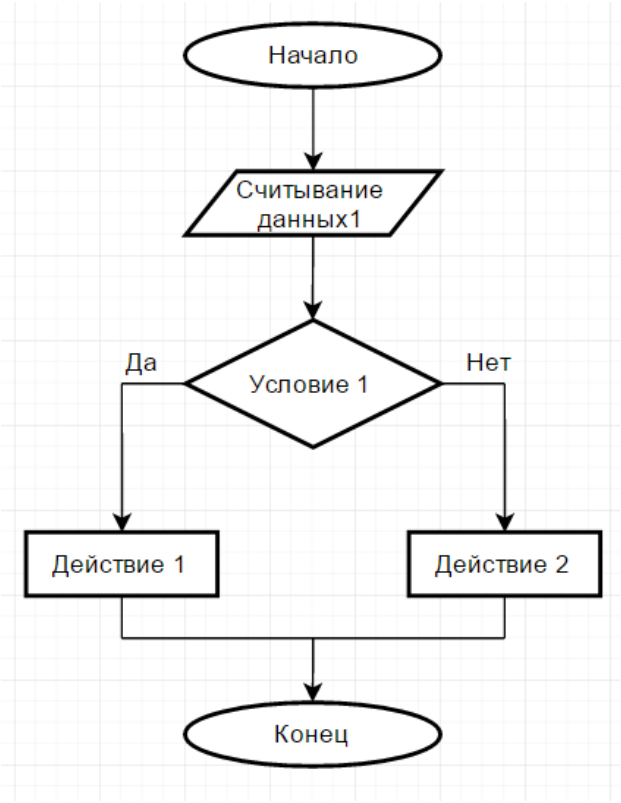


Рисунок 1.3 — Разветвляющийся алгоритм

При изучении темы учащимся предлагается разработать и проанализировать алгоритм задачи. Навыки алгоритмизации, полученные

учащимися в учебно-техническом центре «Омега-1», позволяют описывать не только компьютерные программы, но и любые планы.

Алгоритмы можно записывать и в простой текстовой форме:

Действие 1.

Действие 2.

А для разветвляющегося и циклического алгоритмов:

Считывание данных о наличии каких-либо изменений в окружающей среде с датчика.

Если изменения есть, то перейти на пункт 4.

Действие 2.

Перейти на пункт 6.

Выполнение действия 1.

Перейти на пункт 1.

Текстовая форма алгоритма более похожа на компьютерную программу, но визуально воспринимается хуже, в ней труднее найти логические ошибки, особенно если переходов много.

Для того, чтобы создать компьютерную программу, необходимо ясно понять, что она должна делать и главное, как она это будет делать. Если некоторые вопросы вами еще не проработаны, а вы уже пишете серьезную программу, то, скорее всего, она не будет работать так, как вы рассчитываете. Поэтому нужно идти от простого к сложному, разбивать программы на маленькие простые блоки, добиваться их работоспособности, а только затем собирать из них полую программу.

1.2 Среда разработки Arduino IDE

Для эффективного изучения основ программирования в среде Arduino необходимо провести сравнение с курсом «Основы программирования Turbo Pascal», сравнить синтаксис программного кода,

и изучить реализацию основных алгоритмов в среде Arduino и представить примеры написания программного.

В основу языка программирования, используемого в проектах Arduino, положен C++ — один из самых широко используемых языков программирования, поддерживающий как работу с низкоуровневыми командами, так и построение сложных объектов [10]. Программирование контроллеров Arduino удобно осуществлять в специальной среде Arduino IDE, поскольку в нее включен основной функционал для работы с ними.

1.2.1 Установка Arduino IDE.

Изучать основы программирования на платформе Arduino можно как на линейке физических устройств, так и в виртуальной онлайн среде Electronics Lab, предоставляемой компанией Autodesk на сайте https://circuits.io.

Arduino IDE можно скачать в Интернете по адресу основного сайта проекта: www.arduino.cc/en/Main/Software, программное обеспечение свободно распространяется и для скачивания нужно выбрать в разделе **Download the Arduino IDE** вариант **Windows** Installer (рисунок 1.4), а затем **JUST DOWNLOAD** (рисунок 1.5).

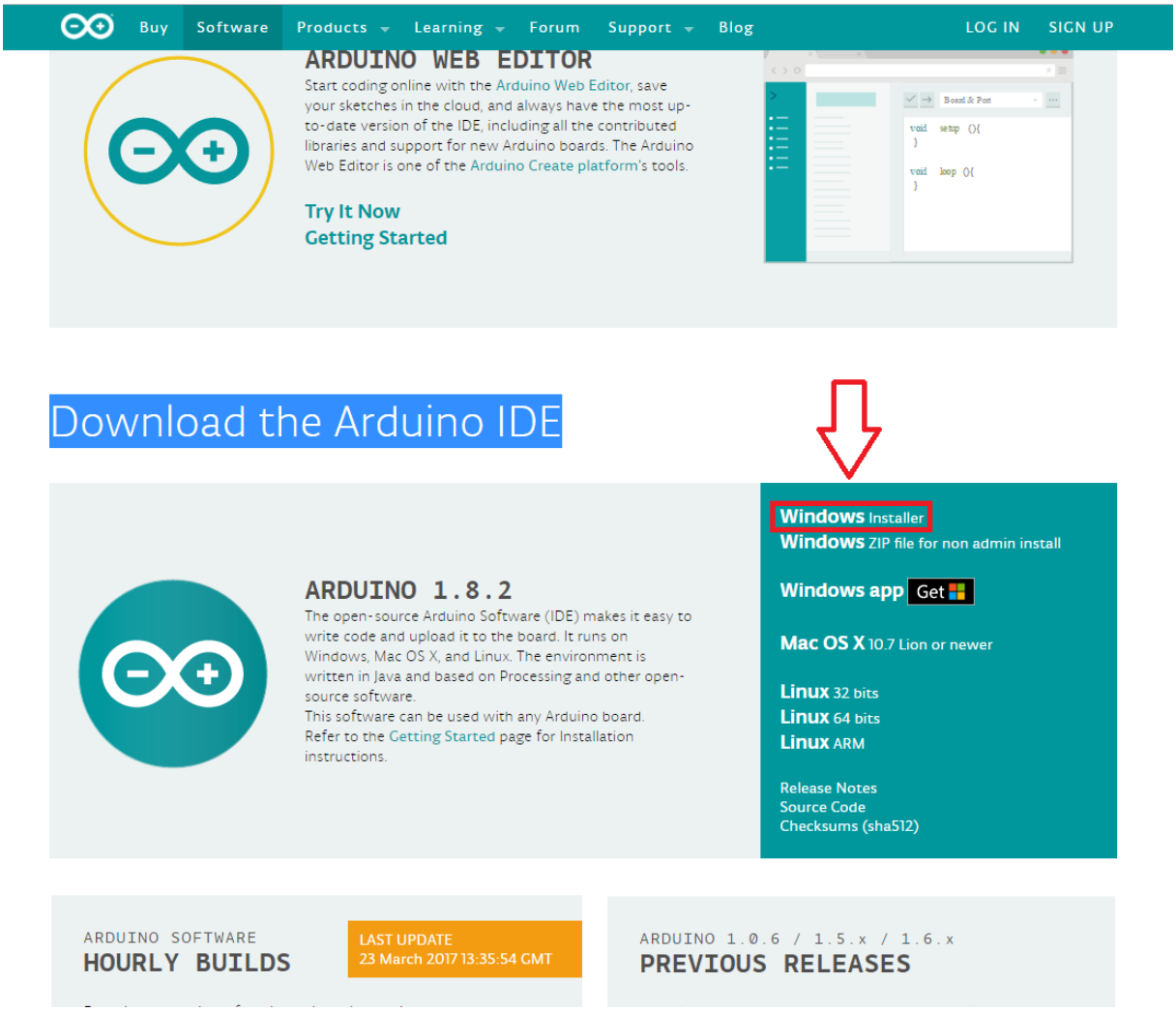


Рисунок 1.4 — Окно скачивания программы Arduino IDE

Рисунок 1.5 — Кнопка загрузки установочного файла Arduino IDE

На компьютер необходимо инсталлировать программу Arduino IDE — в настоящий момент это файл *arduino-1.8.2-windows.exe*. Его надо запустить на выполнение с административными полномочиями, принять условия лицензии GNU LESSER GENERAL PUBLIC LICENSE и согласиться с предложенным вариантом установки.

На все предупреждения Windows в процессе установки следует отвечать утвердительно (продолжать установку). Когда установщик предложит установить драйверы порта, также ответить утвердительно.

1.2.2 Начало работы с Arduino IDE

Среда разработки имеет интуитивно понятный русскоязычный интерфейс. При запуске установленной Arduino IDE откроется окно (рисунок 1.6), в котором уже содержится заготовка программы. Она состоит из двух функций: *setup* и *loop*. Функция setup содержит команды, выполняемые одни раз при включении Arduino, — это установка номеров портов ввода/вывода для управления мониторами и установки скорости обмена данными между Arduino и компьютером. Функция *loop* выполняется бесконечное число раз — до тех пор, пока мы не отключим питание, фактически она зациклена, алгоритмически это изображено на рисунке 1.7.

Интерфейс позволяет реализовать ранее изученные базовые алгоритмы.

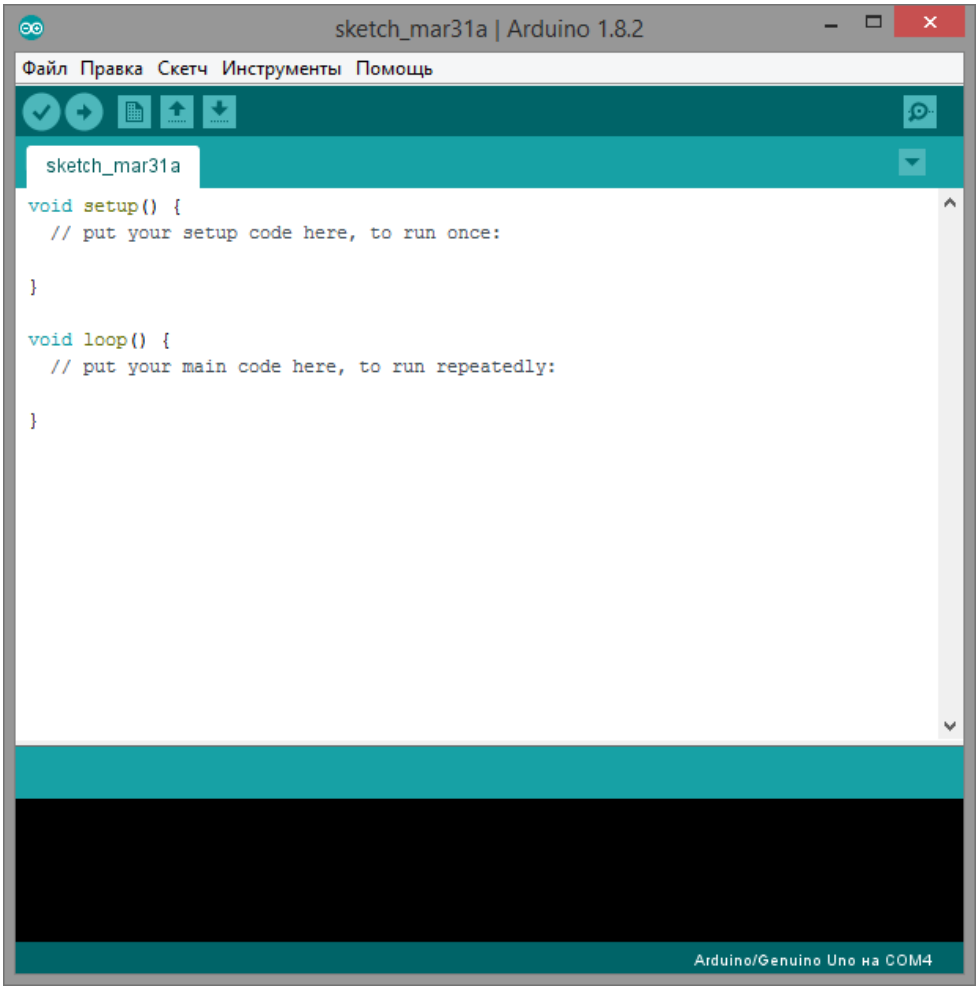


Рисунок 1.6 — Окно Arduino IDE

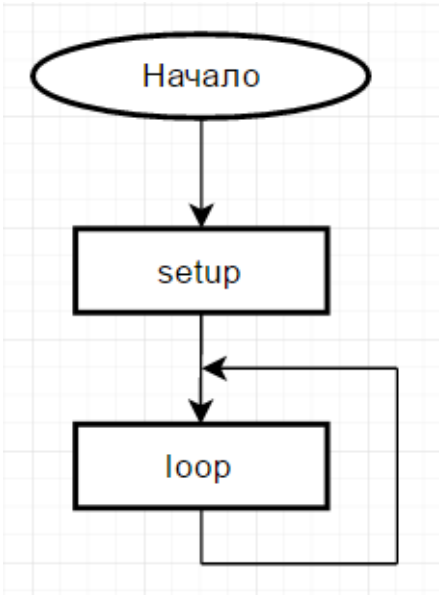


Рисунок 1.7 — Типовой алгоритм программы Arduino IDE

1.2.3 Подключение контроллера Arduino к ПК

Возможности среды позволяют в зависимости от контроллера Arduino использовать, разные USB-кабели (рисунок 1.8) [1]:

- для Arduino UNO R3 и Mega — это кабель с разъемом под USB-принтер с одной стороны и стандартным USB-разъемом с другой;
- для контроллера Nano требуется кабель с разъемом mini-USB;
- для Arduino Micro — это micro-USB, а для программирования контроллера Arduino Mini и Pro Mini потребуется программатор, так как у него отсутствует стандартный интерфейс для подключения его к компьютеру.

Рисунок 1.8 — Варианты USB-кабелей: а) USB для Arduino UNO и Mega; б) кабель для Arduino Nano; в) micro-USB для Arduino Mini

Некоторые контроллеры требуют для своей работы нестандартный драйвер. Они требуют установки отдельного драйвера, не входящего в комплект Arduino IDE, — его название *ch341ser.exe*. Найти его не трудно, после установки драйвера проблема будет решена.

Контроллер подключается к ПК через USB интерфейс и устанавливает связи между ним и оболочкой Arduino IDE. Для этого нужно задать номер порта, к которому подключен контроллер (рисунок 1.9). Если портов много, и найти нужный сложно, рекомендуется запомнить все имеющиеся, а затем физически отсоединить Arduino от кабеля, и снова проанализировать список портов, — тот, который исчез, и есть нужный. После подключения необходимо выбрать нужный порт — для этого нужно

установить соответствующий ему флажок. Иногда для появления порта в списке требуется некоторое время, за которое операционная система компьютера анализирует и проверяет подключенное устройство, так что следует немного подождать до завершения этого процесса.

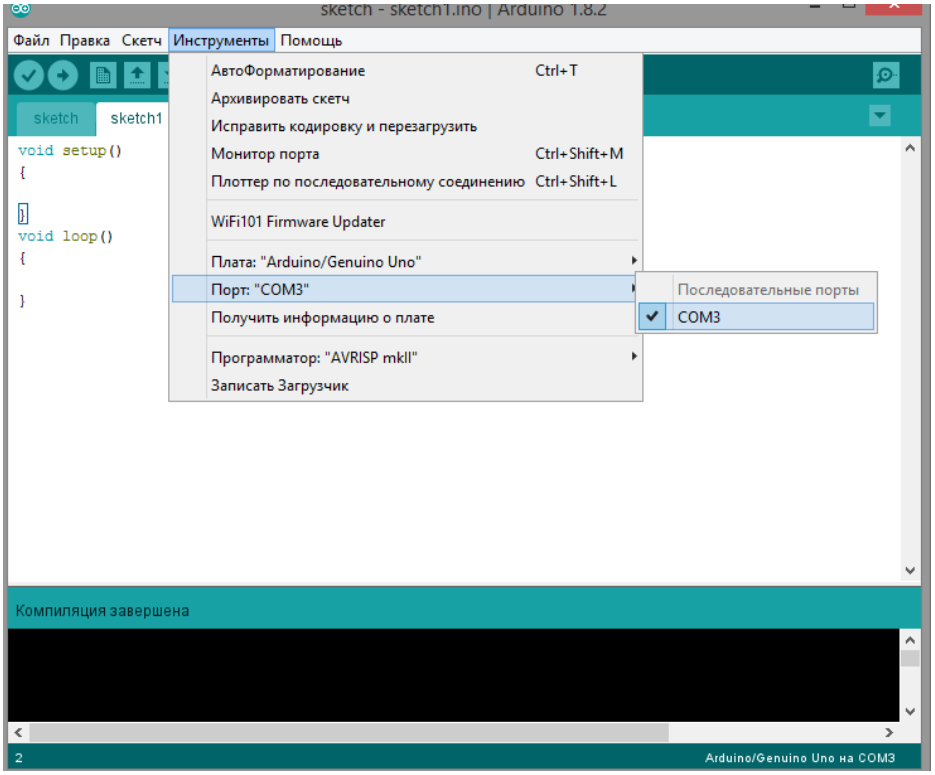


Рисунок 1.9 — Выбор нужного порта

Затем учащиеся выбирают тип контроллера Arduino. На рисунке 1.10 можно видеть, что выбран Arduino/ Genuino Uno.

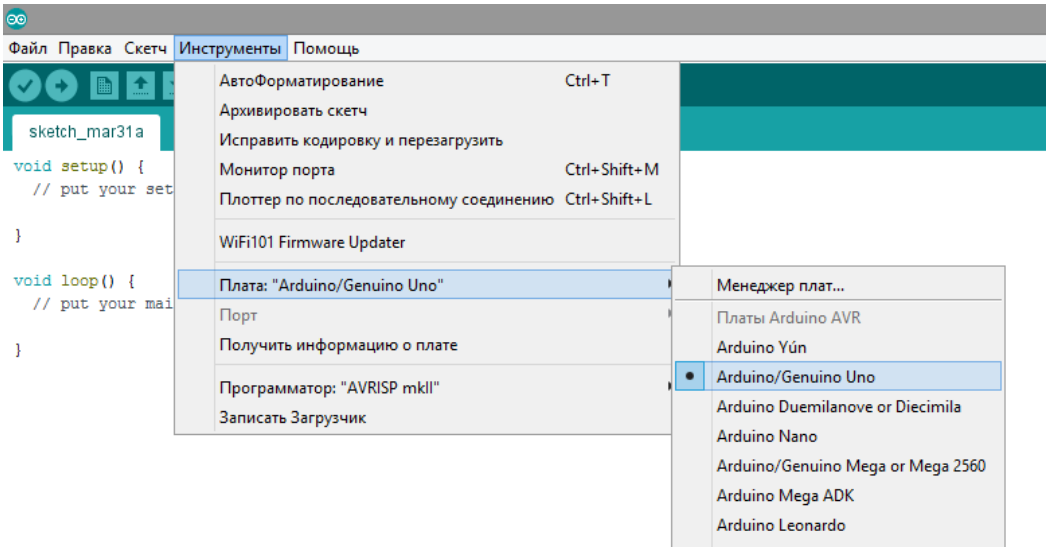


Рисунок 1.10 — Выбор типов контроллера Arduino

Если контроллер выбран системой автоматически, необходимо проверить правильность этого выбора. Для некоторых контроллеров необходимо еще выбрать подвид микроконтроллера, на котором реализована плата Arduino. Название микроконтроллера можно найти на самой его микросхеме — это, как правило, самая большая микросхема платы и расположена она в ее центре.

1.3 Базовые правила синтаксиса языка C\C++

Прежде чем начать изучение языка C\C++, необходима интегрированная среда разработки — IDE.

Язык C\C++ представляет собой набор команд. Этот самый набор команд иными словами называется кодом или исходным кодом.

1.3.1 Загрузка программы

Написанная программа в Arduino IDE загружается в контроллер нажатием кнопки со стрелкой вправо  (см. рисунок 1.6). Оболочка проверит программу на наличие ошибок, а затем переведет ее в двоичный код данных и команд выбранного микроконтроллера и запишет в Arduino.

Если после появления слова **Загрузка** в нижней строке окна Arduino IDE замигали светодиоды TX и RX на плате Arduino, то загрузка программы в плату началась успешно. Если после этого фраза **Загрузка завершена** не появилась, то вероятнее всего неверно выбран тип платы Arduino. Если же диоды TX и RX не замигали вообще, то проблемы заключаются в выборе порта платы Arduino.

Если все пойдет штатно, появится надпись **Загрузка завершена**, и программа автоматически начнет выполняться. Если загружена пустая программа, то ничего происходить не будет, — пустой код будет бесконечно повторяться, пока к плате подключено электропитание.

Сказанное выше можно продемонстрировать на примере: заставим Arduino мигать встроенным светодиодом на 13-м порту (рисунок 1.11). Диод — это электронный элемент, который пропускает электрический ток, только в одном направлении, а светодиод — это диод, который начинает светиться при протекании по нему тока. На плате контроллера Arduino, как правило, уже установлен один светодиод — именно на 13-м порту (ножке).

```
void setup() // настройка
{
  pinMode (13, OUTPUT); // перевод 13-го порта в состояние вывода информации
}

void loop() // основная программа
{
  digitalWrite (13, 1);  // включение светодиода на плате
  delay (1000);          // задержка в 1 секунду
  digitalWrite (13,0);   // выключение светодиода на плате
  delay (1000);          // задержка в 1 секунду
}
```

Рисунок 1.11 — Программа мигания светодиода

Цифровые порты могут получать и передавать информацию только в виде последовательности нулей и единиц. В приведенном примере как раз и передается на порт 13 подобная последовательность — визуально это отслеживается по морганию светодиода на плате контроллера. Когда светодиод горит, на порту высокое электрическое напряжение 5В, а когда потушен — низкое (0В).

1.3.2 Мониторинг работы программы

Мониторинг программы позволяет выполнить отладку устройства. Для того чтобы контроллер мог общаться с компьютером во время выполнения программы записывается в функции setup команда *Serial.begin(9600)*, которая указывает, с какой скоростью происходит обмен информацией с ПК. Для обмена информацией требуется, чтобы приемник и передатчик осуществляли обмен на одинаковой скорости, а так

как контроллер Arduino достаточно медленный, то наиболее подходящей для обмена с ним является скорость 9600 бод. Обмен информацией на более высоких скоростях также возможен, но задействует большее количество ресурсов контроллера, при этом чаще происходят прерывания текущей программы, и основной код может выполняться медленнее.

Команда *Serial.println()* передает с контроллера на ПК значение, указанное в скобках. Здесь это значение, возвращаемое функцией *millis (),* которая возвращает время работы в миллисекундах. Результат работы приведен на рисунке 1.12. Для получения на экране компьютера полученных результатов после запуска программы потребуется открыть вкладку **Инструменты | Монитор порта** и скорректировать скорость порта (9600 бод) — в окне **Монитор порта** скорость задается в правом нижнем углу выбором из всплывающего списка.

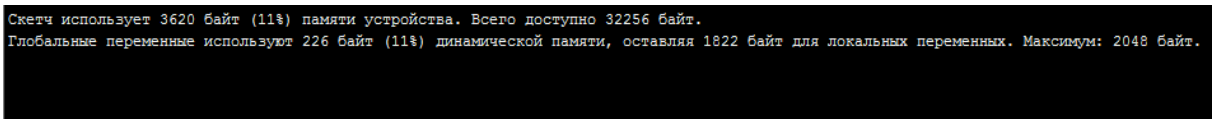


Рисунок 1.12 — Протокол загрузки программы

1.3.3 Переменные

Важной составляющей синтаксиса языков программирования являются переменные. Переменные — это области памяти, имеющие имя, которое иначе называют идентификатором. До начала работы с переменными их нужно объявить, как и в любой другой среде программирования, например Turbo Pascal. В языке C/C++ все переменные должны быть объявлены. Это означает, что, во-первых, в начале каждой программы или функции должен быть приведен список всех используемых переменных, а во-вторых, указан тип каждой из них [14].

```
// переменные видны в программе везде
int a;
float b;

// переменная видна только внутри функции setup
void setup() // настройка
{
  int c;
  c=10;
  Serial.begin(9600); // скорость порта связи Arduino - ПК
  a=c*5; //значение переменной a изменится на 10*5
}

void loop() // основная программа
{
  // следующая строка сожержит ошибку
  c=a+5;        // переменная c не может быть видна за пределами функции setup
  delay(a*100); // задержка 10*5*100 миллисекунд
  Serial.println(millis()); // передача на ПК время работы в миллисекундах
  delay(1000); // задержка 1 секунду
}
```

Рисунок 1.13 — Определение переменных. Зоны видимости. Сообщение об ошибке

Рисунок 1.13 в демонстрационных целях содержит специально добавленную в него ошибку объявления переменной, и при попытке его компилировать об этой ошибке будет выведено сообщение: **‘c’ was not declared in this scope**.

Переменные могут отличаться по типу данных, для хранения которых созданы (таблица 1.1).

Таблица 1.1 — Типы данных

Обозначение в программе	Название	Принимаемые значения	Размер в памяти
boolean	Логический	True/false, 1/0	1 байт
char	Символьный	-128/+127	1 байт
byte	Короткое беззнаковое	0-255	1 байт

Окончание таблицы 1.1

int	Целое число	-32768/32767	2 байта
long	Длинное число	-2147483648/214748367	4 байта
float	Число с плавающей точкой	$-3{,}4028235 \cdot 10^{38}/3{,}4028235 \cdot 10^{38}$	4 байта

При работе с переменными следует понимать, что они не являются идеальным хранилищем информации — так, например, целочисленные переменные могут переполняться. Это происходит в тех случаях, когда значение, которое следует записать в переменную, больше максимально возможного для этого типа данных. Переменные с плавающей точкой подвержены другой проблеме — они округляют свои значения при сложении большого числа с малым. Так же Arduino IDE не всегда корректно работает с преобразованием типов данных.

На следующем примере можно продемонстрировать небольшую программу, осуществляющую получение данных от компьютера через порт ввода/вывода (рисунок 1.14).

```
void setup() // настройка
{
 Serial.begin(9600); //скорость порта связи Arduino - ПК
}

void loop() // основная программа
{
 char symbol; // переменная символьного типа данных
 if (Serial.available() > 0)
 {
  symbol=Serial.read(); // считывание символов из переменной
  Serial.println(symbol); // вывонд на ПК то, что было считано
 }
}
```

Рисунок 1.14 — Получение данных от компьютера через порт ввода/вывода данных

Скомпилировав и загрузив эту программу, можно ввести в верхней части окна **Монитор порта** слова «Hello, World!» и **отправить** их на порт. Программа отправит сообщение обратно на ПК посимвольно (рисунок 1.15).

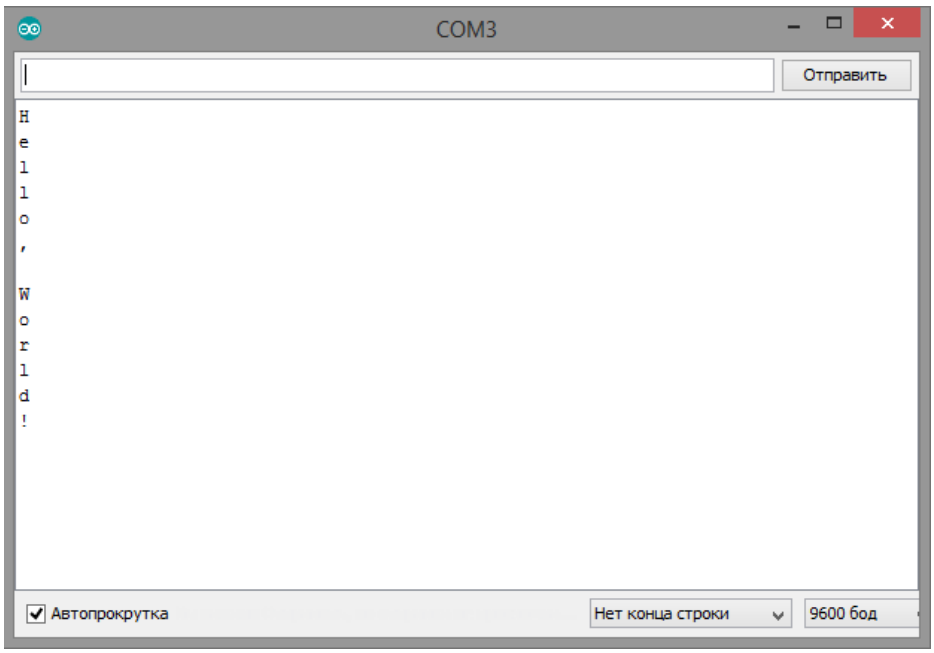


Рисунок 1. 15 — Вывод сообщения на ПК

1.3.4 Условные операторы, операторы выбора, операторы циклов

Ниже приведено описание и реализация ранее изученных учащимися алгоритмов: условных, операторов выбора и циклов.

Оператор *if… else* графически, в виде блок-схемы алгоритма, представлен на рисунке 1.16, а в виде фрагмента кода — на рисунок 1.17.

Рисунок 1.16 — Представление условного оператора

```
void setup() // настройка
{

 Serial.begin(9600); //скорость порта связи Arduino - ПК
}

void loop() // основная программа
{
 char symbol; // переменная символьного типа данных
 if (Serial.available() > 0)
 {
  symbol=Serial.read(); // считывание символов из переменной
  Serial.println(symbol); // вывонд на ПК то, что было считано
 }
else
 {
  Serial.println("Введите свой текст"); // вывонд на ПК текста в ("_____")
  delay(7000);
}
  }
```

Рисунок 1.17 — Применение оператора *if… else* при отправке сообщения на порт

Реализация данного фрагмента программы предлагает пользователю ввести свой текст каждые 7 секунд.

Следующий пример — программа, которая изменяет частоту моргания встроенного светодиода на 13-м порту (рисунок 1.18) в зависимости от посланной с компьютера команды. Код программы представлен на рисунке 1.19. Блок-схема алгоритма это программы представлена на рисунке 1.20.

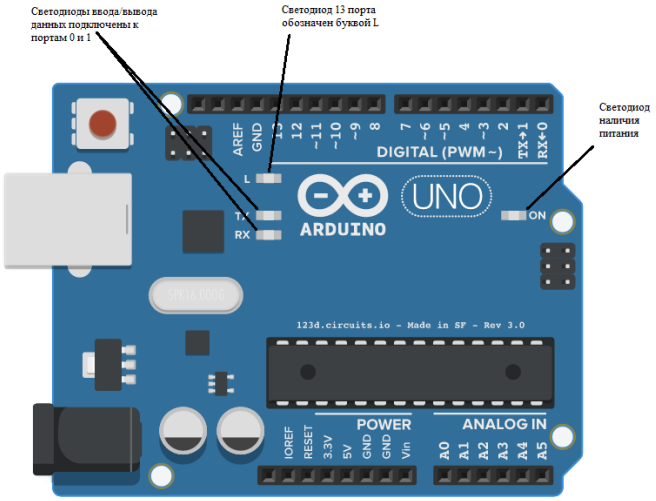


Рисунок 1.18 — Светодиоды на плате Arduino UNO

Программа не завершается, пока есть электропитание, в ней присутствуют три условных блока: один проверяет наличие данных на порту ввода/вывода, а последующие два сравнивают полученный символ с условиями, в зависимости от которых изменяют величину переменной *time_pick*, задающую время задержки при мигании светодиода.

```
int time_on; // переменная хранения времени свечения диода
void setup () // настройка
{
  Serial.begin(9600); // скорость порта связи Arduino - ПК
  pinMode (13, OUTPUT); // 13-й порт в состояние вывода информации
  time_on=200; // время свечения диода
}
void loop()
{
  char dannie; // переменная символьного типа данных
  if (Serial.available()>0) // проверка поступления данных
    {
      dannie = Serial.read();
      if (dannie=='1') // если нажата клавиша "1"
        {
        time_on=1000; // задержка мигания 1 секунда
        }
  else
    {
      if(dannie=='2') // если нажата клавиша "2"
        {
        time_on=500; // задержка мигания 0,5 секунд
        }
      else
        {
        time_on=100; // задержка мигания 0,1 секунд
        }
    }
    }

  digitalWrite (13, 1); // зажигание светодиода на плате
  delay(time_on); // задержка
  digitalWrite (13, 0); // выключение светодиода
  delay(time_on); //задержка
}
```

Рисунок 1.19 — Программа управления миганием светодиода с ПК

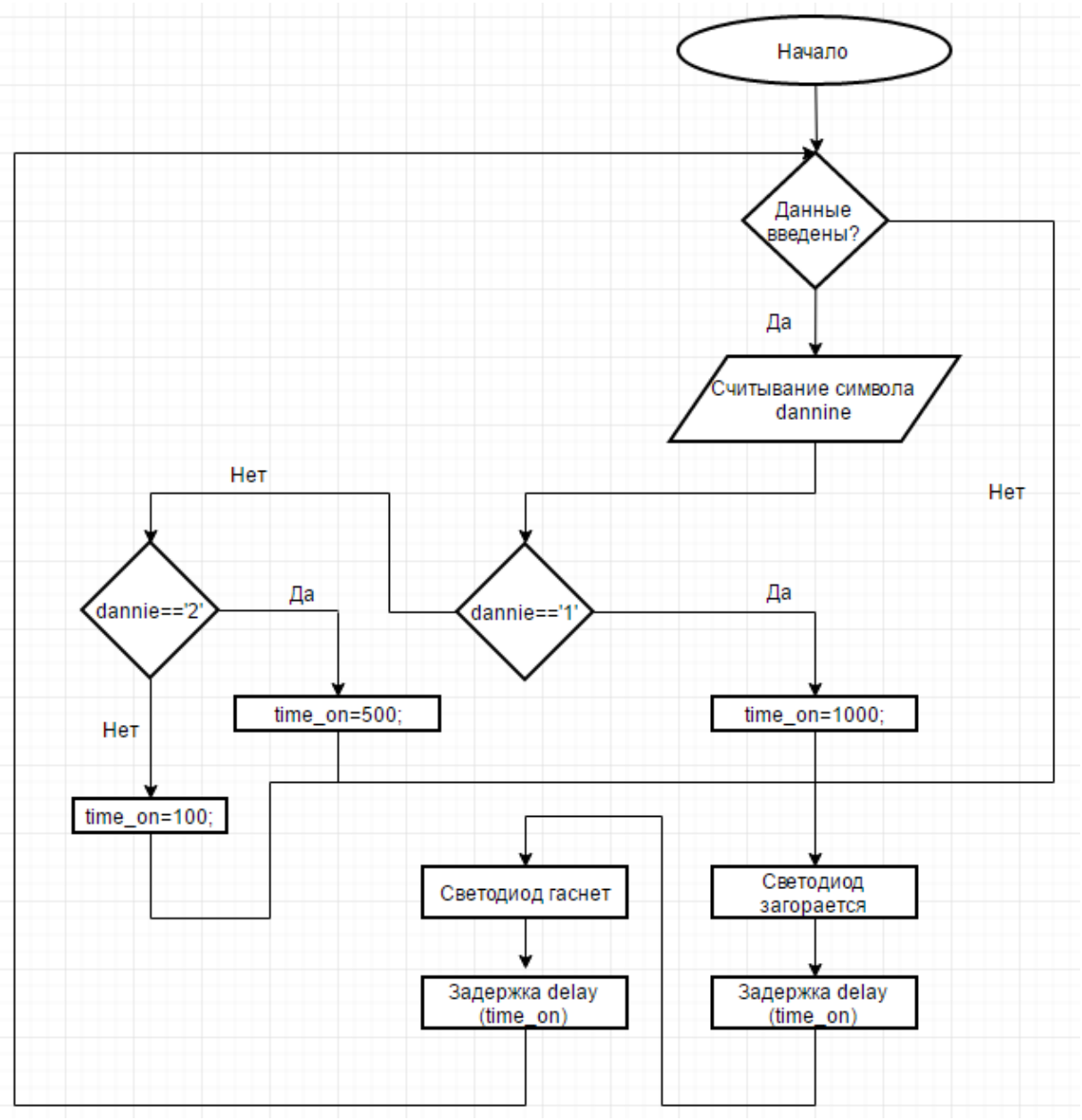


Рисунок 1.20 — Алгоритм изменения частоты светодиода

Следующий оператор *switch… case* позволяет быстро сделать выбор из набора определенных значений, программа с его использованием выглядит намного проще, ее легче анализировать (рисунок 1.21).

```
int time_on; // переменная хранения времени свечений светодиода
void setup() // настройка
{
  Serial.begin(9600); // скорость связи порта Arduino - ПК
  pinMode(13,OUTPUT); // 13-й порт переведен в состояние вывода информации
  time_on=200; // время свечения диода
}
void loop()
{
  char dannie;
  if (Serial.available()>0) // проверка поступления данных
  {
    dannie = Serial.read(); // считывание нажатого символа
    switch (dannie)
    {
      case '1': // выборка нажатого значения
      time_on=1000;
      break;
      case'2':
      time_on=500;
      break;
      default:
      time_on=200;
    }
  }
  digitalWrite(13,1);
  delay (time_on);
  digitalWrite(13,0);
  delay (time_on);
}
```

Рисунок 1.21 — Программа управления миганием светодиода с ПК с использованием оператора *switch… case*

Далее приводится описание работы операторов организации циклов. Два оператора: *while* и *for*.

Рассмотрим цикл *while* на примере программы с рисунка 1.11, заменив функцию ожидания *delay ()* на оператор цикла по условию (рисунок 1.22). Применим при этом функцию *millis ()*, которая возвращает количество миллисекунд от старта программы. В результате работы этой программы светодиод включается на 1 сек. и выключается на 1 сек.

```
void setup() // настройка
{
  pinMode(13,OUTPUT); // 13-й порт переведен в состояние вывода информации
}
void loop() // основная программа
{
  unsigned long sec1;
  unsigned long sec2;
  unsigned long sec3;
  sec1=millis(); // начальное значение - количество миллисекунд
  sec2=sec1+1000; // +1 секунда
  sec3=sec2+2000; // +2 секунды
    while (sec1<sec2)
    {
      digitalWrite(13,1); // включение светодиода
      sec1=millis(); // текущее время в sec1
    }
    while (sec1<sec3)
    {
      digitalWrite(13,0); // выключение светодиода
      sec1=millis(); // текущее время в sec1
    }
}
```

Рисунок 1.22 — Программа управления миганием светодиода с ПК с использованием оператора цикла *while*

Оператор цикла *for* хорошо подходит тогда, когда требуется выполнить определенное количество повторений. При построении алгоритмов в виде блок-схем для него есть отдельная фигура (рисунок 1.23). Для примера на рисунке 1.24 в коде программы светодиод моргает 500 раз с увеличением времени свечения на 1 миллисекунду каждый повтор.

Рисунок 1.23 — Представление оператора *for*

Рисунок 1.24 демонстрирует программу, управляющую миганием встроенного светодиода на 13-м порту, при этом частота мигания уменьшается по мере приближения к концу цикла.

```
void setup() // настройка
{
  pinMode(13,OUTPUT); // 13-й порт переведен в состояние вывода информации
}
void loop() // основная программа
{
  int k; // переменная счетчика
  for (k=0;k<500;k++)
  {
    digitalWrite(13,1); // включение светодиода на плате
    delay(k); // задержка по k миллисекунд
    digitalWrite(13,0);
    delay(k);
  }
}
```

Рисунок 1.24 — Пример использования оператора *for* на 500 повторений

1.3.5 Функции

Функции являются поименованными блоками команд, которые по их имени можно вызвать из любого места программы. Функции нужно применять там, где существует необходимость одинаковый программный код повторять много раз, или если нагромождение команд в основной программе делает ее нечитаемой [10].

Функция — это блок кодов программы, у которых есть имя и ряд команд, которые запускаются при вызове функции. Например, функции void setup и void loop [18].

Существуют встроенные функции — они уже присутствуют в библиотеке функций Arduino IDE — и функции, созданные в рамках текущей программы. Также существуют библиотеки, которые можно подключать к текущей программе и пользоваться имеющимися в них функциями. Подключаются библиотеки из пункта оболочки **Скетч | Подключить библиотеку**. После подключения той или иной библиотеки,

можно использовать содержащиеся в ней готовые программные решения, написанные для различный устройств и датчиков.

На рисунке 1.25 приведен пример написания и использования простой функции — она сравнивает два числа и возвращает наибольшее из сравниваемых. Для функции, которые могут возвращать значения, при их описании в строке перед именем функции устанавливается тип возвращаемого значения, на данном случае — это целое число: *int*. Для возврата значения из функции используется ключевое слово *return*, за которым следует возвращаемое значение.

При описании функции, не возвращающих значения, в строке перед их именем должно стоять слово *void*.

Действия, производимые функцией, заключаются в фигурные скобки и называются *телом функции*.

```
void setup()
{
}
void loop() // основная программа
{
  int x=100;
  int y=200;
  int s=sravnit(x,y); // вызов функции sravnit()
}
  int sravnit(int a, int b) // выбор типа возвращаемого значения, имя функции и информацию о аргументах
  {                         // тело функции
    if (a<b) return b;
    else return a;
  }
```

Рисунок 1.25 — Пример простой функции

1.3.6 Элементы объектно-ориентрованного программирования

В отличие от Turbo Pascal, который является структурным языком, язык C++, составляющий основу для программирования в проекте Arduino, является объектно-ориентированным. Объекты — это сложные структуры, которые имеют оригинальное название, могут включать в себя как переменные, так и функции. Фактически, объекты созданы для удобного обращения к сложным структурам. Например, когда программно

подключается сервомотор, то сначала подключается библиотека (*#include <Servo.h>*), а затем создается объект *servomotor* (рисунок 1.26).

```
#include <Servo.h> // подключение библиотеки для управления севомоторами
Servo servomotor; // создали объект сервомотор
void setup()
{
  servomotor.attach(12); // подключение сервомотора к 12 пину Arduino
}
void loop()
{
  servomotor.write(10); // поворот сервомотора в положение 10 градусов
  delay(500); // задержка, что бы сервомотор успел повернуть
  servomotor.write(150); // поворот сервомотора в положение 150 градусов
  delay(500); // задержка, что бы сервомотор успел повернуть
}
```

Рисунок 1.26 — Пример управления сервомотором

Естественно, что при этом контакт сервомотора должен быть физически подключен к порту Arduino, и на мотор подано электропитание (рисунок 1.27).

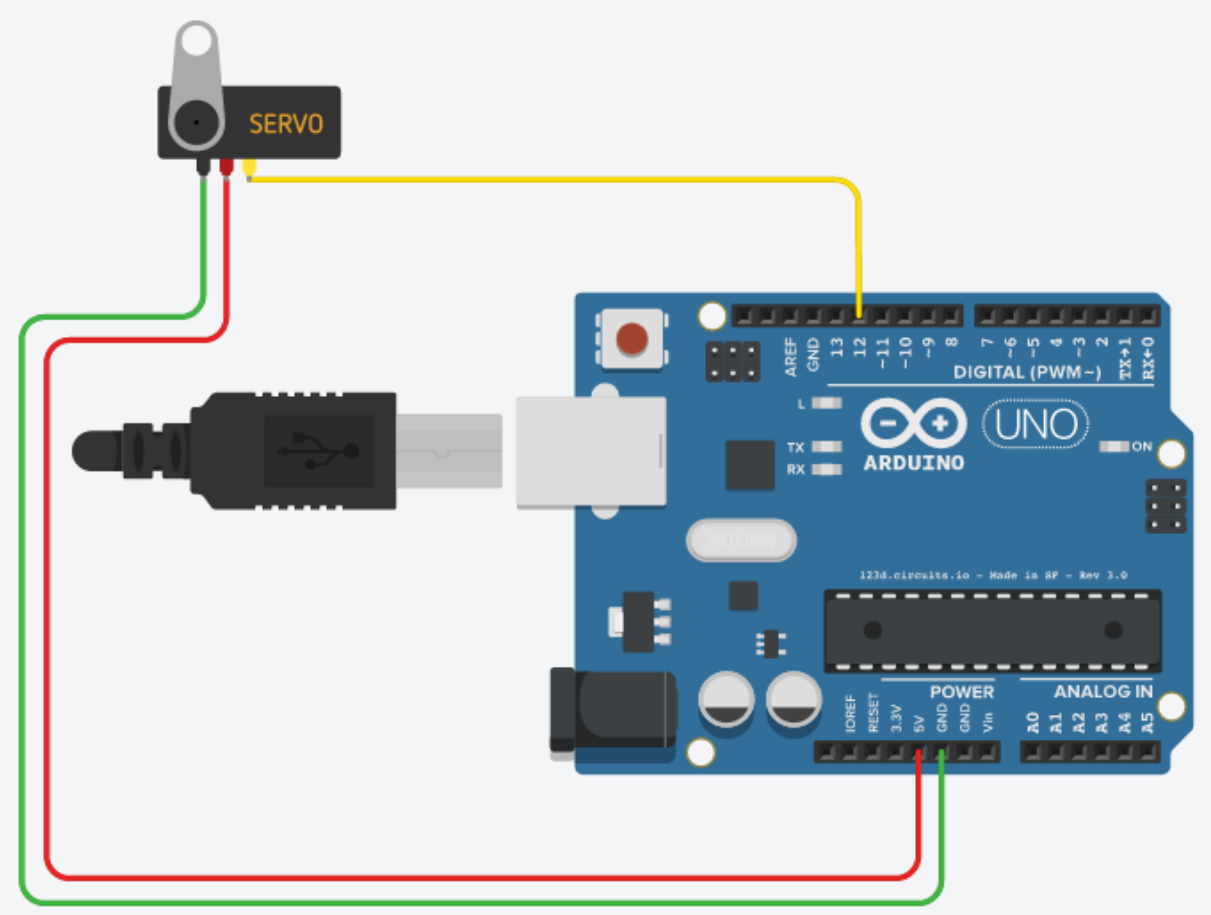


Рисунок 1.27 — Подключение сервомотора к контроллеру Arduino

1.4 Применение Arduino в обучении прикладному программированию

Обучение прикладному программированию является одним из базовых курсов в процессе подготовки студентов и учащихся различных учебных заведений, входя в дисциплины вариативной части, математического и естественно-научного цикла. Целью изучения – является создание у учащихся необходимых представлений о современных программных продуктах, средах разработки и языках программирования, а также применение знаний, умений и навыков, приобретенных в процессе обучения, на практике.

Наилучший результат практического применения знаний, умений и навыков достигается путем решения задач, несущих полезную нагрузку. Это требование полностью может выполнить применение платформы Arduino в обучении [9].

Программная часть Arduino состоит из открытой программной оболочки (IDE) для написания программ, их компиляции и программирования аппаратуры. Аппаратная часть является набором печатных плат, продающимся как официальным производителем, так и сторонними производителями. Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Arduino [9]. Имея базовые знания по основам программирования в среде Turbo Pascal можно обеспечить оптимальный переход на изучения языков высокого уровня.

Польза от применения Arduino в обучении прикладному программированию выражается следующими факторами:

Экономическая доступность представляет несколько моделей: Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Nano, Arduino Mini.

Простота освоения среды разработки — это интуитивно понятный интерфейс, который позволяет в кратчайшие сроки освоить среду разработки.

Платформа Arduino поддерживает очень большой диапазон периферийных устройств.

Использование в разработке программного обеспечения языка программирования на основе C++, языка высокого уровня.

Технологичность процесса разработки устройств, а именно наличие последовательного выполнения инструкций.

Большой выбор дополнительной литературы и электронных ресурсов.

При написании данной работы был проведен поиск научной и учебно-методической литературы, статьей в изданиях Российской Федерации и интернет источников по выбранной тематике.

Основными источниками, раскрывающими теоретическую часть, являются работы Петина В. А. «Проекты с использованием контроллера Arduino» [15], Момота М. В. «Мобильные роботы на базе Arduino» [10], Соммера У. «Программирование микроконтроллерных плат Arduino / Freeduino» [18], Эргановой Н. Е. «Методика профессионального обучения» [24], Стариковой Л. Д. «Методика профессионального обучения» [19] и др.

Среди электронных ресурсов, можно выделить такие, как «Проект Амперка» (http://amperka.ru/), CraftDuino (http://robocraft.ru/), Ардуино (http://arduino.ru/) и не менее интересный портал Хабрахабр (https://habrahabr.ru/), на которых можно встретить различные интересные проекты, разработки, статьи по тематике выпускной квалификационной работы.

Кроме того, имеется широкий набор публикаций в журналах научно-методической направленности. Можно выделить, например, научные статьи из следующих научных источников: статья «Анализ проблемы профессиональной подготовки программиста и пути ее решения» под

авторством Осадчего В.В., Осадчей Е.П. [12], опубликованная в журнале «Образовательные технологии и общество» и статья «Современное образование: робототехника в школе» с электронного ресурса Techno-guide [17].

На кафедре информационных систем и технологий имеются подобные разработки, а именно лабораторный практикум «Разработка интерактивных устройств на аппаратно-программной платформе Arduino» и «Электронные устройства на аппаратно-программной платформе Arduino».

Исходя из представленных выше факторов, можно сделать вывод, что использование Arduino в обучении прикладному программированию позволит предоставить студентам возможность применить знания, умения и навыки, приобретенные в процессе обучения, на аппаратном обеспечении, что даст мотивацию к дальнейшему обучению и закрепит теоретическую часть обучения на практике.

Актуальность изучения робототехники в учебно-техническом центре «Омега-1» заключается в необходимости изучения основ робототехники в современном обществе и заинтересованности учащихся в изучении этой тематики. Лабораторный практикум «Основы программирования на платформе Arduino» призван обеспечить непрерывный процесс обучения прикладному программированию средних и высших учебных заведениях.

2 РАЗРАБОТКА ЛАБОРАТОРНОГО ПРАКТИКУМА «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ПЛАТФОРМЕ ARDUINO»

2.1 Лабораторный практикум в дополнительном образовании

Актуальность изучения робототехники в учебно-техническом центре «Омега-1» учитывается в широком распространении робототехники в современном обществе и заинтересованности учащихся в изучении этой тематики.

Лабораторный практикум «Основы программирования на платформе Arduino» предназначен для дополнительного образования школьников старших классов и студентов младших курсов колледжей и техникумов, обучающихся в учебно-техническом центре ООО «Омега-1» г. Екатеринбурга.

Новизна выпускной квалификационной работы состоит в том, чтобы был разработан лабораторный практикум «Основы программирования на платформе Arduino» в сравнении с курсом «Основы программирования Turbo Pascal» и адаптирован для учебно-технического центра «Омега-1» и который будет применяться в системе дополнительного образования.

Лабораторный практикум — это система содержательно и методически разработанных обучающих занятий либо по научному вопросу, усвоение которого сопряжено с овладением умениями и навыками, либо по целостному учебному курсу прикладного характера, который исследует прикладную сторону профессии [23].

Практикум содействует формированию необходимых профессиональных умений [23].

В Федеральном законе «Об образовании в Российской Федерации» (№ 273-ФЗ, от 29.12.12) используется понятие дополнительного

образования. Дополнительное образование — как вид образования, который направлен на всестороннее удовлетворение образовательных потребностей человека в интеллектуальном, духовно-нравственном, физическом и (или) профессиональном совершенствовании и не сопровождается повышением уровня образования [6].

Современное дополнительное образование характеризуется как процесс освоения добровольного избранного человеком вида деятельности или области знаний, выходящих за рамки стандарта обязательного образования [11].

Качество образования зависит от возможностей системы образования. Дополнительное образование в сочетании с основным составляет единое образовательное пространство [5].

Программы дополнительного образования реализуются учреждениями дополнительного образования. Например, учебно-техническим центром ООО «Омега-1» г. Екатеринбурга, для которого разработан лабораторный практикум «Основы программирования на платформе Arduino».

2.2 Требования к разработке лабораторного практикума

К разработке лабораторных практикумов предъявляют дидактические и технологические требования [24]:

Дидактические:

- требование научности, то есть достаточная глубина, корректность, научность, достоверность учебного материала;
- требование доступности, а именно степень сложности освоения учебного материала в зависимости от возрастных и личностных характеристик учащихся;
- требование наглядности, что упрощает восприятие учебного материала;

- требование последовательности и систематичности в усвоении учебного материала, это означает структурированность учебного материала, последовательность усвоения, связь с практикой.

Технологические [24]:

- открытость — возможность изменять лабораторный практикум;
- лояльный интерфейс к пользователю;
- надежная работа практикума;
- наличие банка заданий;
- удобное перемещение.

Требования к проведению лабораторного практикума.

Лабораторные работы решают одну из важнейших задач дидактики — связь теории с практикой. Лабораторные занятия имеют большое воспитательное значение, способствуют развитию мышления и приобретению профессиональной уверенности у обучающихся [7].

К организации и проведению лабораторных работ практикума предъявляются следующие требования [19]:

1. Лабораторные работы должны быть целесообразны и эффективны.
2. Сочетание лабораторных работ с другими методами обучения.
3. Следует учитывать специфику обучающихся, уровень их подготовленности.
4. В лабораторных работах должно быть продумано оценивание проделанных работ.

Лабораторные работы включают [20]:

- вводную часть (тема, цель работы, информационно-теоретический блок, перечень оборудования);
- содержание хода работы и последовательность действий;
- рекомендации к оформлению;
- заключительная часть (анализ результата работы, рефлексия собственной деятельности [19].

2.3 Способ реализации лабораторного практикума

Лабораторный практикум разработан посредством программы Microsoft Office Word. Для удобного использования в конечном итоге представляется в PDF-формате. При помощи программы Adobe Acrobat Reader текстовый документ переводится в PDF-документ.

Формат PDF является самым и популярным форматом для полиграфической продукции, огромное количество материалов и документов хранится и распространяется в данном формате. Файлы PDF занимают мало места и удобны в использовании [13].

Acrobat Reader от Adobe Systems — это решение, принятое открытым стандартом, подкрепляемым International Organization for Standardization (ISO) [16].

ISO — это Международная организация по стандартизации.

Acrobat Reader — это программное обеспечение, которое может использоваться для просмотра, печати, комментирования документов в формате PDF (рисунок 2.1).

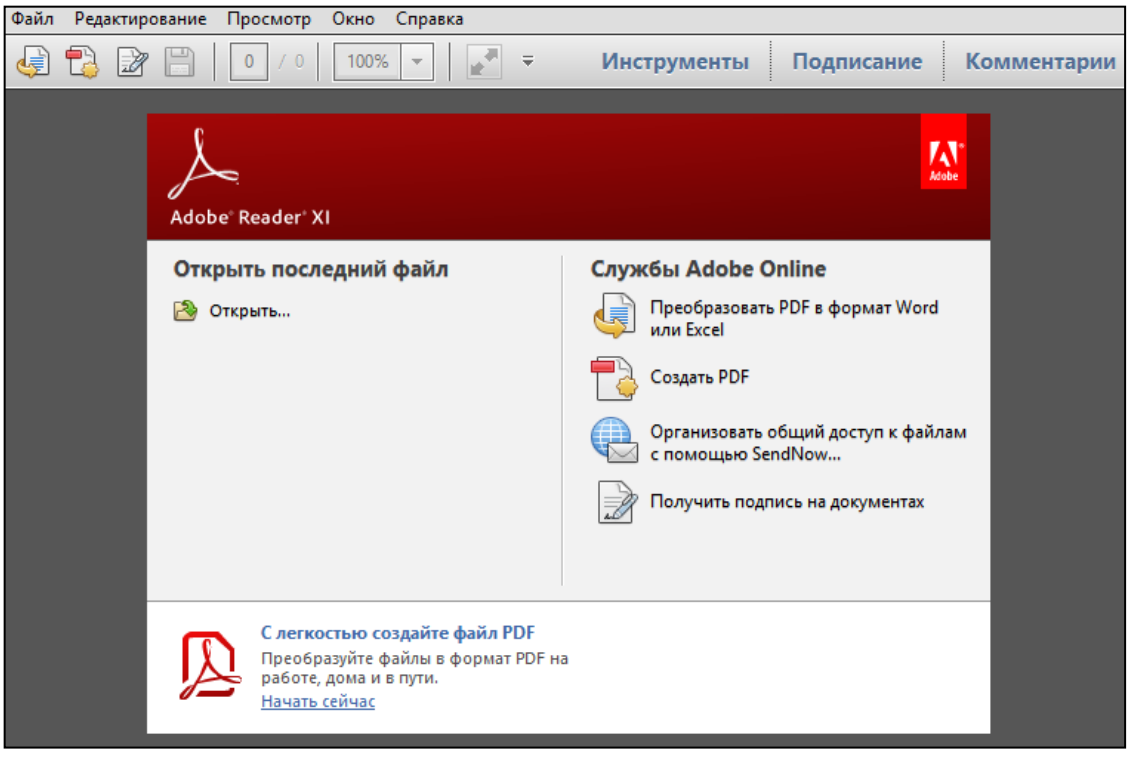


Рисунок 2.1 — Adobe Reader XI

При запуске программы появляется меню (см. рисунок 2.1).

Кнопка **«Создать PDF»** служит для создания PDF-файла путем преобразования документа в PDF. После ее нажатия появляется меню, где можно указать путь к файлу, который нужно преобразовать (рисунок 2.2)

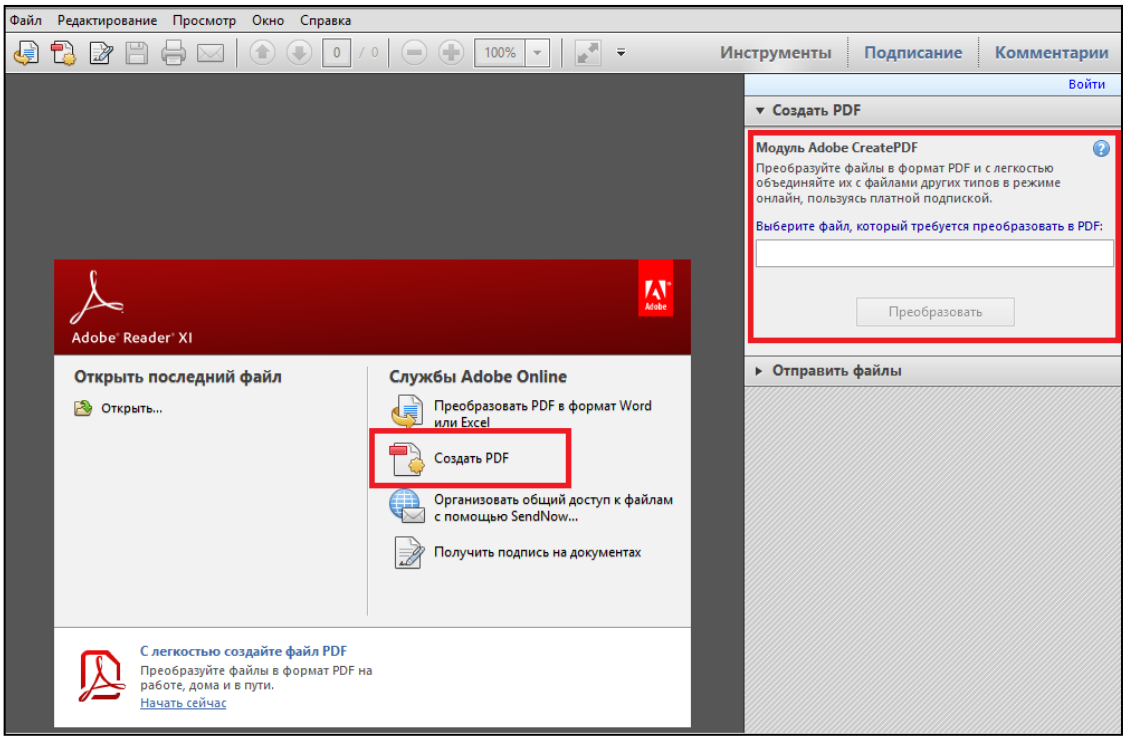


Рисунок 2.2 — Меню выбора файла

Основные особенности заключается в том, что Acrobat Reader лоялен к любой операционной системе и аппаратным требованиям, способен преобразовать документ в PDF-формат практически из любого документа, а это значит, что данное программное обеспечение «дружит» с другим программным обеспечением.

Есть возможность внедрения музыкальных файлов, видеоматериала, интерактивных кнопок, гиперссылок, оглавления и системы закладок, перекрестных ссылок и многого другого. Отсутствует, разве что, способность использования анимации внутри документа [16].

Преимущества PDF-документа перед остальными:

- первозданный вид, то есть документ остается таким же, как и до преобразования;

- соответствие стандартам ISO;
- независимость от операционной системы и аппаратных требований персонального компьютера;
- возможность адаптации PDF-документа к пользователю, а именно обеспечить поиск, навигацию и удобное использование.

Как результат формат PDF подходит для создания и хранения электронных публикаций [16].

2.4 Структура и содержание лабораторного практикума

Практикум включает в себя введение, практический блок, средства контроля, глоссарий и полезные ссылки.

Введение содержит краткое описание назначения лабораторного практикума, педагогический адрес, цель и задачи.

Практический блок состоит из шести лабораторных работы и контрольного задания.

Каждая лабораторная работа содержит задания самостоятельной работы и контрольные вопросы для проверки знаний и умений, полученных в ходе выполнения лабораторных работ.

В лабораторных работах сохранена хронология выполнения, имеются иллюстрации.

Элементы лабораторных работ:

- название лабораторной работы и ее порядковый номер;
- формулировка темы, цели;
- установка времени выполнения и необходимого оборудования;
- небольшой блок теоретических сведений;
- блок задания;
- технологию выполнения;
- блок самоконтроля;
- блок контрольных вопросов.

Контрольное задание состоит из следующих элементов:

- название контрольного задания;
- формулировка темы и цели;
- установка времени выполнения;
- блок задания поставленной задачей.

Лабораторный практикум «Основы программирования на платформы Arduino» является результатом выпускной квалификационной работы профилизации «Компьютерные технологии автоматизации и управления.

Наполнение содержания лабораторного практикума

Лабораторный практикум «Основы программирования на платформе Arduino» состоит из шести лабораторных работ и контрольного задания.

Работы включают в себя:

- блок теоретических сведений;
- формулировку задания;
- хронологи выполнения;
- технологию выполнения;
- самостоятельное задание;
- контрольные вопросы.

Проделанная лабораторная работа демонстрируется преподавателю, затем оценивается. В качестве контроля имеются самостоятельные задания и контрольные вопросы. По результатам прохождения лабораторно практикума «Основы программирования на платформе Arduino» необходимо выполнить контрольное задание, подразумевающее самостоятельно создать устройство из предложенного перечня компонентов.

Далее следует подробный пример одной из лабораторных работ, и краткое описание последующих лабораторных работ и контрольного задания.

Лабораторная работа№1

Тема: «Знакомство со средой разработки Arduino IDE. Установка и начало работы».

Цель: познакомиться со средой разработки Arduino IDE, научиться осуществлять настройку ПО перед началом работы.

Длительность: 180 минут.

Оборудование: персональный компьютер, среда разработки Arduino IDE, микроконтроллер Arduino UNO, USB-кабель.

Теоретические сведения

Arduino (Ардуино) — аппаратная вычислительная платформа, основными компонентами которой является плата ввода-вывода и среда разработки.

Алгоритм — набор последовательных действий/команд, соблюдающих логичный порядок, который позволяет решать какие-либо задачи.

Для графического представления алгоритмов используются, так называемые *блок-схемы. Блок-схемы* — это графическое представление алгоритма, состоящее из набора геометрических фигур. Каждая фигура изображает определенную операцию или действие (рисунок 2.3).

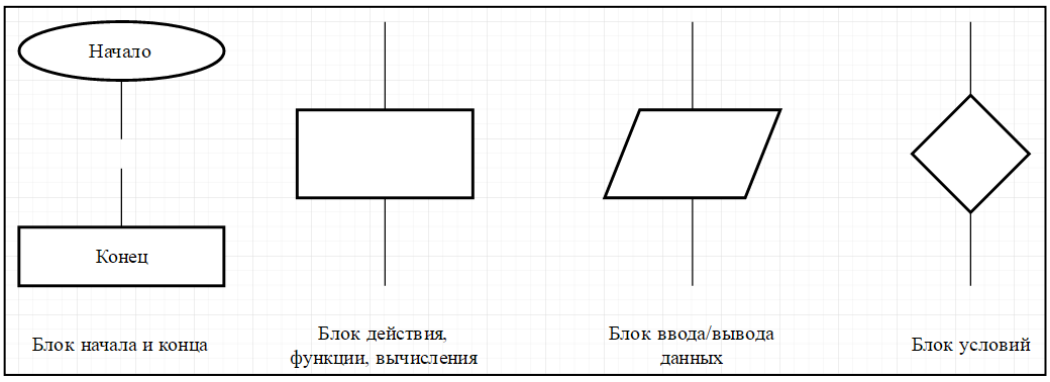


Рисунок 2.3 — Графическое представление алгоритма

Алгоритмы бывают нескольких типов: линейные, циклические, разветвляющиеся.

Среда разработки Arduino IDE — в основу языка программирования в данной среде положен C++ — один из самых популярных языков программирования.

Структура простой программы, написанной в среде Arduino IDE, в сравнении с Turbo Pascal (Таблица 2.1).

Таблица 2.1 — Сравнение простых программ

Arduino IDE	Turbo Pascal

int a; //обозначение переменных
int b;
void setup () { //настройки программы
Serial.begin(9600); //скорость связи данных
}
void loop() { // основная программа
b=5;
a=b*10;
Serial.println(a); //вывод на ПК ответ
delay(5000); // задержка 5 секунд
}
	PROGRAM sum; {заголовок программы}
	USES CRT; { подключение библиотек}
	VAR a,b: integer;{объявление переменных}
	BEGIN {начало блоков операторов}
	CLRSCR; {очистка экрана}
	b:=5; {оператор 1}
	a:=b*10; {оператор 2}
	WRITELN(a); {Вывод a}
	END. {конец блока операторов}

Переменные отличаются по типу данных (таблица 2.2).

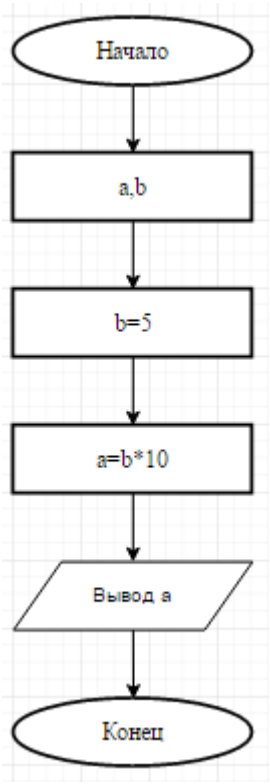
Таблица 2.2 — Типы данных в сравнении с Turbo Pascal

Arduino IDE	Turbo Pascal	Принимаемые значения
boolean	BOOLEAN	True/False
char	CHAR	-128/+127
byte	*BYTE*	0-255
int	*INTEGER*	-32768/32767
long	*LONGINT*	-2147483648/214748367
float	RE-AL/SINGLE/DOUBLE/EX TENDED	$-3{,}4028235\cdot10^{38}/3{,}4028235\cdot10^{38}$

Программы с линейным алгоритмом (таблица 2.3). Сравниваемые части программ выделены жирным курсивом.

Таблица 2.3 — Примеры линейной программы Arduino IDE в сравнении с Turbo Pascal

Arduino IDE	Блок-схема	Turbo Pascal

int a;
int b;
void setup ()
{
Serial.begin(9600);
}
void loop()
{
b=5;
a=b*10;
Serial.println(a);
delay(5000);
}
		PROGRAM sum;
		USES CRT;
		VAR a,b: integer;
		BEGIN
		CLRSCR;
		b:=5;
		a:=b*10;
		WRITELN(a);
		END.

Задание: установить среду разработки Arduino IDE, настроить оборудование перед началом работы.

1. Для начала зайдем на официальный сайт проекта Arduino в Интернете по адресу **www.arduino.cc/en/Main/Software**. Для того, чтобы скачать инсталлятор выберем в разделе **Download the Arduino IDE** вариант **Windows** Installer (рисунок 2.4), затем нажмите **JUST DOWNLOAD** (рисунок 2.5).

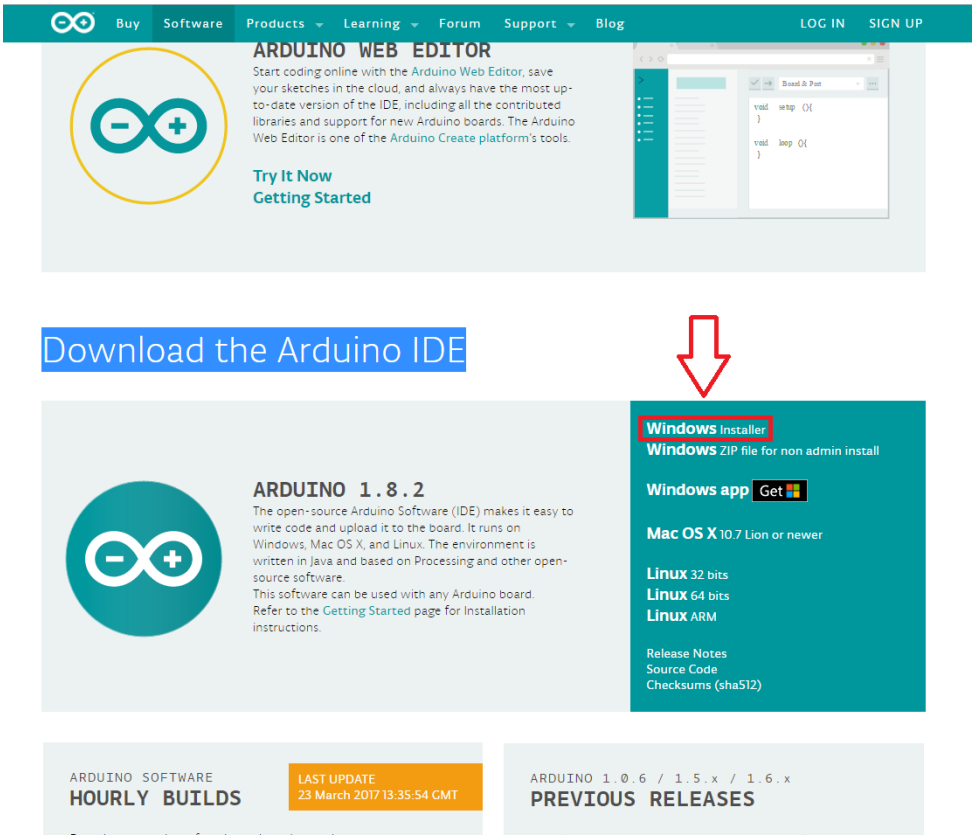


Рисунок 2.4 — Окно скачивания программы Arduino IDE

Рисунок 2.5 — Кнопка загрузки установочного файла Arduino IDE

2. После загрузки файла *arduino-1.8.2-windows.exe* запустите его с административными полномочиями, примите условия лицензионного соглашения и принимайте последующие варианты установки.

Установщик будет предлагать установить драйвера порта, на что следует отвечать утвердительно.

3. После установки запустите на рабочем столе ярлык **Arduino.exe**. При запуске появится окно (рисунок 2.6), в котором содержится заготовка

программы. В окне имеется заготовка, которая состоит из функций setup и loop. Функция setup содержит команды, выполняемые одни раз при включении — это так называемые настройки программы. А loop включает в себя основную программу. Она выполняется бесконечное число раз, до тех пор, пока мы не отключим питание.

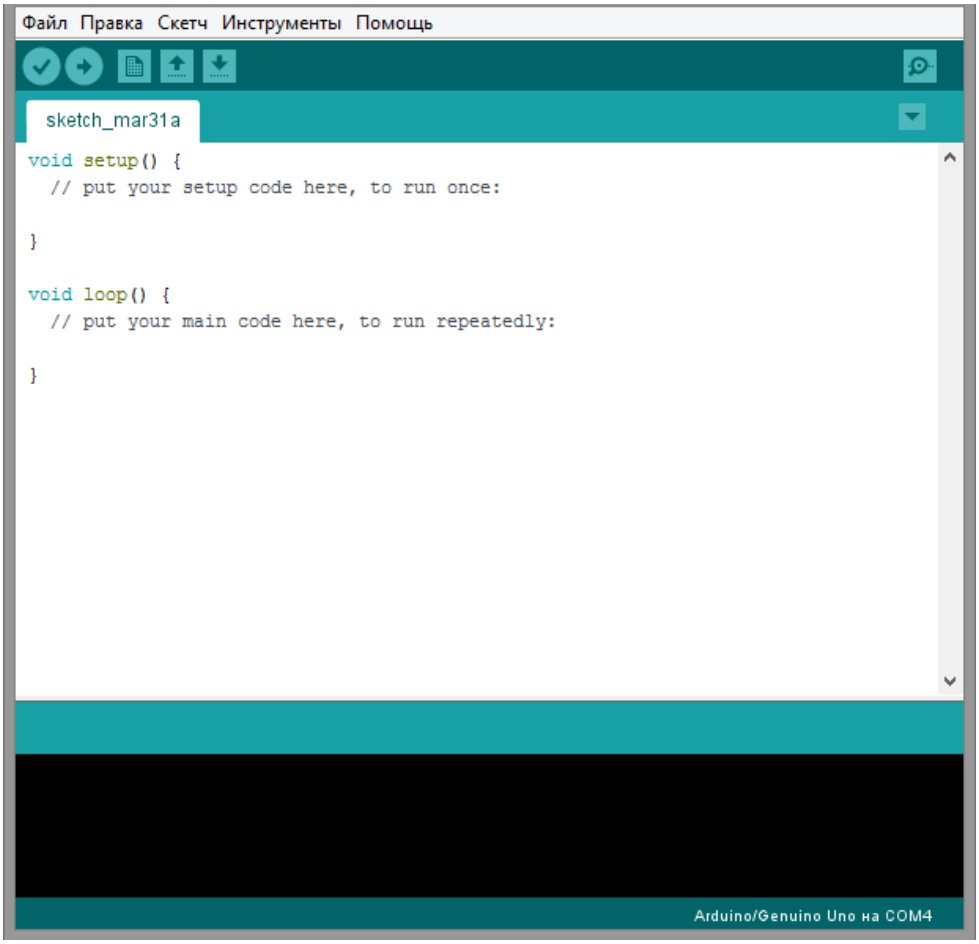


Рисунок 2.6 — Окно программы Arduino IDE

4. Для подключения контроллера используются USB-кабели. Подключите контроллер Arduino к компьютеру кабелем. После того как физический контакт ПК с контроллером установлен, нужно установить связи между ним и средой разработки Arduino IDE. Для этого необходимо выбрать номер порт (рисунок 2.7). Если портов много, рекомендуется запомнить все имеющиеся, а затем физически отсоединить Arduino от кабеля, и снова проанализировать список портов, — тот, который исчез, и есть нужный. Далее установите флажок на нужном порте.

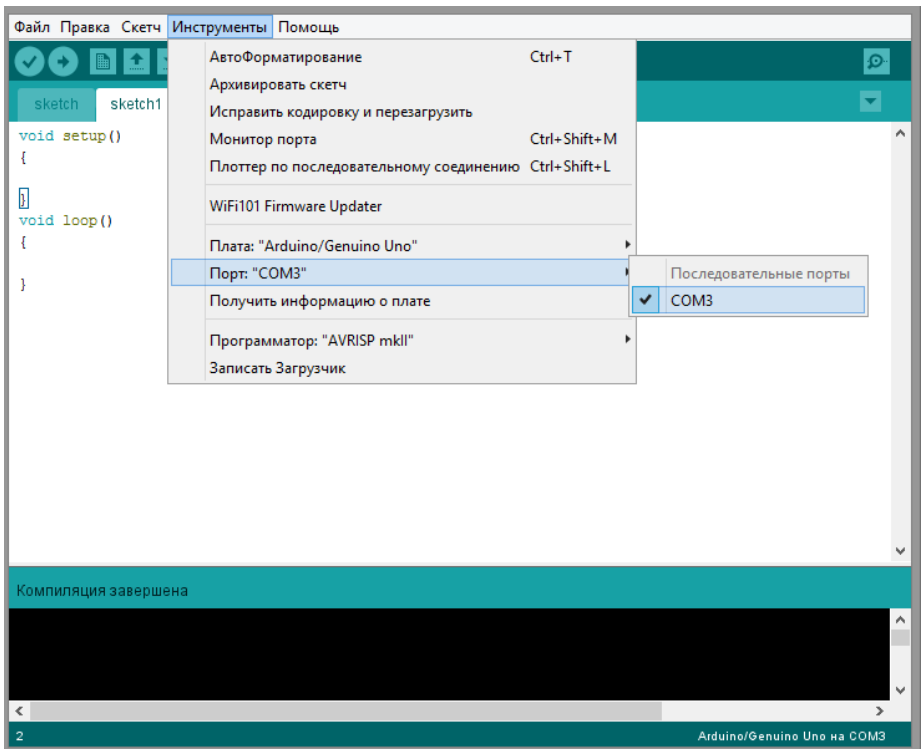


Рисунок 2.7 — Выбор порта

У пользователей Windows XP может возникнуть проблема — драйвер порта не установился автоматически. Решение проблемы описано на сайте по адресу: http://arduino.ru/Guide/Windows.

5. Далее выберите тип вашего контроллера. На рисунок 2.8 можно видеть, что выбрать Arduino/Genuino Uno.

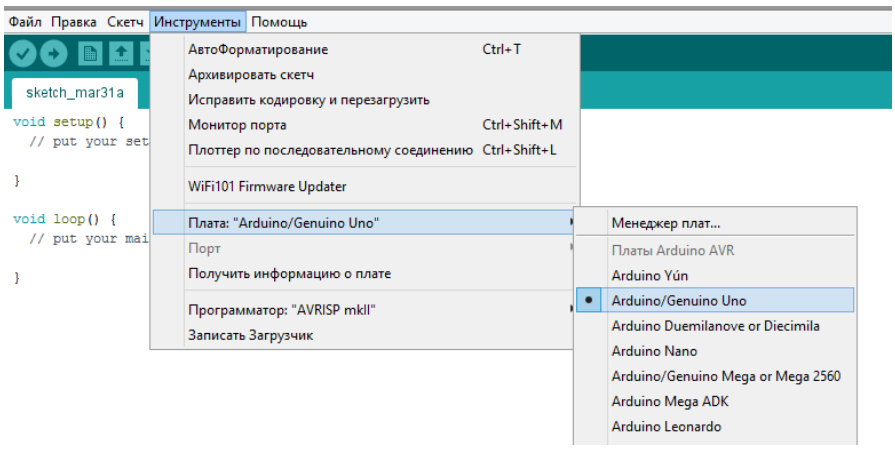


Рисунок 2.8 — Выбор типов контроллера Arduino

Если контроллер выбран автоматически, то убедитесь в правильности выбора.

6. Настройка закончена. Чтобы загрузить программу необходимо нажать кнопку  (см. рисунок 2.6). Оболочка проверит программу на наличие ошибок, а затем переведет ее в двоичный код данных и команд выбранного микроконтроллера и запишет в Arduino.

7. В качестве примера, заставим Arduino мигать встроенным светодиодом на 13-м порту (рисунок 2.9).

Диод — это электронный элемент, который пропускает электрический ток, только в одном направлении.

Светодиод — это диод, который начинает светиться при протекании по нему тока. На плате контроллера Arduino, как правило, уже установлен один светодиод — на 13-м порту (ножке).

```
void setup() // настройка
{
  pinMode (13, OUTPUT); // перевод 13-го порта в состояние вывода информации
}

void loop() // основная программа
{
  digitalWrite (13, 1);  // включение светодиода на плате
  delay (1000);          // задержка в 1 секунду
  digitalWrite (13,0);   // выключение светодиода на плате
  delay (1000);          // задержка в 1 секунду
}
```

Рисунок 2.9 — Программа мигания светодиодом

Аналогичным образом перепишем программу и загрузим ее в микроконтроллер. На выходе получим мигающий на плате светодиод с задержкой в 1 секунду (рисунок 2.10).

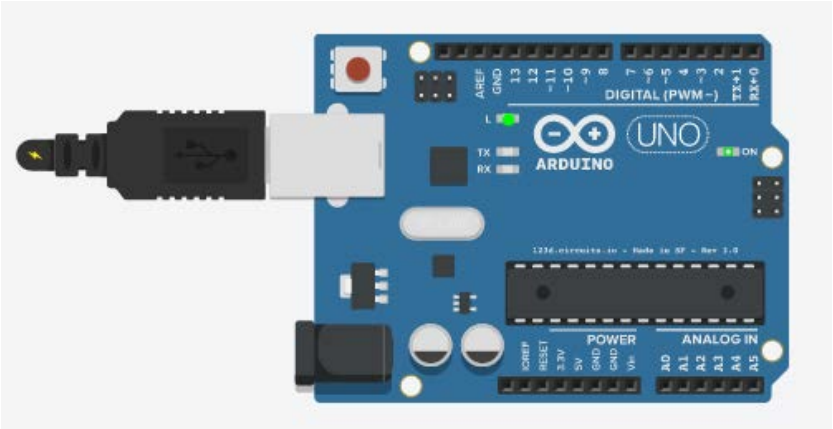


Рисунок 2.10 — Результат работы светодиода (L — светодиод)

8. Для подключения монитора порта, необходимо открыть вкладку **Инструменты | Монитор** порта и скорректировать скорость порта (9600 бод) — в окне Монитор порта скорость задается в правом нижнем углу выбором из всплывающего списка.

Команда Serial.println( ) передает с контроллера на ПК значение, указанное в скобках.

Самостоятельная работа

Измените код программы так, чтобы светодиод мигал с каждым разом быстрее в течении 10 секунд, путем добавления команд digitalWrite() и delay ().

Контрольные вопросы

1. Что такое Arduino?
2. Что такое алгоритм? Какие алгоритмы Вы знаете? Дайте определение понятию блок-схема?
3. Какие типы данных из курса «Язык программирования Turbo Pascal» Вы помните?
4. Для чего служат так называемые «боды»?

Лабораторная работа №2

Тема: «Управление светодиодами средствами виртуальной среды Autodesk Circuits с применением условного оператора if».

Цель: средствами виртуальной среды Autodesk Circuits научиться подключать светодиоды и программировать ее через компилятор.

Длительность: 180 минут.

Оборудование: персональный компьютер, браузер, виртуальная среда Autodesk Circuits (Arduino Uno, макетная плата, светодиоды 6 шт, резисторы 6шт 220 Ом).

После выполнения лабораторной работы обучающиеся будут уметь подключать и программировать светодиоды с применением условного оператора if.

Лабораторная работа №3

Тема: «Управление светодиодами с помощью кнопок средствами виртуальной среды Autodesk Circuits» с применением оператора выбора case».

Цель: средствами виртуальной среды Autodesk Circuits научиться подключать и программировать кнопки для управления светодиодами.

Длительность: 180 минут.

Оборудование: персональный компьютер, браузер, виртуальная среда Autodesk Circuits.

После выполнения лабораторной работы обучающиеся будут уметь управлять и программировать кнопки с использованием условного оператора выбора case.

Лабораторная работа №4

Тема: «Управление RGB-светодиодом средствами виртуальной среды Autodesk Circuits» с применением цикла for».

Цель: средствами виртуальной среды Autodesk Circuits научиться подключать и программировать RGB-светодиод.

Длительность: 180 минут.

Оборудование: персональный компьютер, браузер, виртуальная среда Autodesk Circuits.

После выполнения лабораторной работы обучающиеся будут уметь подключать и программировать RGB-светодиод с применением цикла for.

Лабораторная работа №5

Тема: «Управление жидкокристаллическим дисплеем LCD 16x2 с применением оператора цикла while».

Цель: научиться подключать и программировать LCD дисплей с применением оператора цикла while.

Длительность: 180 минут.

Оборудование: персональный компьютер, Arduino Uno, USB-кабель, LCD-дисплей, макетная плата, перемычки, соединительные провода.

После выполнения лабораторной работы обучающиеся будут уметь подключать и программировать LCD-дисплей с использованием оператора цикла while.

Лабораторная работа №6

Тема: Мини-проект «Ультразвуковой дальномер»

Цель: подключить схему и запрограммировать компоненты для функционирования ультразвукового дальномера.

Длительность: 180 минут.

Оборудование: персональный компьютер, Arduino Uno, макетная плата, диоды 3 шт., резисторы 3 шт. 220 Ом, LCD-дисплей 16x2, ультразвуковой датчик HC-SR04.

После выполнения лабораторной работы обучающиеся будут уметь подключать схему и программировать компоненты устройства «Ультразвуковой дальномер».

Контрольное задание

Тема: Самостоятельный «мини-проект».

Цель: самостоятельно собрать устройство и запрограммировать его.

Длительность: 360 минут.

Оборудование: микроконтроллер Arduino, светодиод, фотодиод, кнопка, потенциометр, пьезодинмаик, RGB-светодиод, светодиодная шакала, инфракрасный датчик, датчик температуры, датчик влажности, датчик давления, датчик наклона, датчик ультразвука, PIR-датчик (датчик движения), фоторезистор, сервомотор, дисплей LCD 16x2, клавиатура 4x4, 7 сегментный индикатор.

Обучающиеся самостоятельно собирают устройство, программируют его. По результатам проделанной работы составляют блок-схему устройства, комментируют код программы и презентуют свой проект, то есть защищают его.

2.5 Методические рекомендации к проведению лабораторного практикума

Лабораторный практикум «Основы программирования на платформе Arduino» предназначен для дополнительного образования школьников старших классов и студентов младших курсов колледжей и техникумов, обучающихся в учебно-техническом центре ООО «Омега-1» г. Екатеринбурга.

Задачи лабораторного практикума «Основы программирования на платформе Arduino»:

1. Познакомить с виртуальной средой Autodesk Circuits и средой разработки Arduino IDE.
2. Понимать и уметь воспроизводить схемы подключения в виртуальной среде или на макетной плате с использованием электронных компонентов.
3. Изучить основы программирования на платформе Arduino на представленных примерах схем и программ.
4. Научить самостоятельно, изменять и дополнять программный код в зависимости от поставленной задачи и условий.
5. Овладеть умением проектировать, собирать и программировать устройство самостоятельно.

Для решения задач практикум применим в дополнительном образовании и разработан для школьников старших классов и студентов младших курсов колледжей и техникумов, обучающихся в учебно-техническом центре ООО «Омега-1» г. Екатеринбурга.

Используются такие средства обучения, как программная оболочка Arduino IDE, виртуальная среда разработки Autodesk Circuits, технологические карты.

Деятельность обучающихся основывается на первом, втором и третьем уровне, где первый — это попытка воспроизведения без ошибок, вто-

рой — применение с изменением первоначальных условий, третий — самостоятельная деятельность, а именно сборка и программирование устройства.

Лабораторный практикум «Основы программирование на платформе Arduino» проводится в условиях самостоятельной деятельности обучающихся. Обучающиеся следуя содержанию лабораторных работ выполняют их, затем демонстрируют их преподавателю, выполняют самостоятельное задание и отвечают на контрольные вопросы.

Как только обучающиеся выполнили все этапы лабораторной работы — преподаватель ставит баллы: 10 за одну лабораторную работу, 60 за шесть лабораторных работ, 15 за контрольное задание. Итого 75 — максимальное количество баллов. Минимальное количество баллов, которые можно набрать за одну выполненную лабораторную работу — 6, 10 за контрольное задание. Итого 46 баллов минимум.

Обучающимся необходимо выполнить лабораторные работы, самостоятельные задания, ответить на контрольные вопросы.

ЗАКЛЮЧЕНИЕ

Выпускная квалификационная работа лабораторный практикум «Основы программирования на платформе Arduino» для дополнительного образования школьников старших классов и студентов младших курсов колледжей и техникумов, обучающихся в учебно-техническом центре ООО «Омега-1» г. Екатеринбурга, включает в себя шесть лабораторных работ, которые в свою очередь включают в себя задания самостоятельной работы и контрольные вопросы. По итогу выполнения лабораторного практикума выполняется контрольное задание.

Результат проделанной работы таков:

1. Проведено сравнение сред разработки Arduino IDE и Turbo Pascal.
2. Проведено сравнение синтаксиса программного кода и представлены примеры написания программ в этих средах, изучены основы программирования на платформе Arduino.
3. Рассмотрены и соблюдены требования к разработке лабораторного практикума.
4. Разработана структура и содержание лабораторного практикума, включающее в себя шесть лабораторных работ и контрольное задание.

По итогу выполнения лабораторного практикума «Основы программирования на платформе Arduino» обучающиеся изучат основы программирования на платформе Arduino в сравнении со знаниями, полученными из лабораторного практикума «Язык программирования Turbo Pascal» от учебно-технического центра ООО «Омега-1» г. Екатеринбурга.

Цель достигнута — практикум разработан, задачи выполнены. Лабораторный практикум «Основы программирования на платформе Arduino» может использоваться в системе дополнительного образования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аппаратная платформа Arduino[Электронный ресурс]. — Режим доступа: http://arduino.ru/Hardware (дата обращения 24.04.17).

2. Белослудцева Л. И., Прончев Г. Б. Курс робототехники для дополнительного образования [Текст] // Проблемы и перспективы развития образования: материалы II Междунар. науч. конф. (г. Пермь, май 2012 г.). — Пермь: Меркурий, 2012. — С. 102-104.

3. Гайсина И. Р. Развитие робототехники в школе [Текст] // Педагогическое мастерство: материалы II Междунар. науч. конф. (г. Москва, декабрь 2012 г.). — М.: Буки-Веди, 2012. — С. 105-107.

4. Готлиб Б. М. Введение в специальность «Мехатроника и робототехника» [Текст]: курс лекций / Б. М. Готлиб, А. А. Вакалюк. — Екатеринбург: УрГУПС, 2012. — 134 с.

5. Золотарева А. В. Дополнительное образование детей: история и современность [Текст]: учеб. пособие для СПО / А. В. Золотарева, А. Л. Пикина, Н. А. Мухамедьярова, Н. Г. Тихомирова; отв. ред. А. В. Золотарева. —2-е изд., испр. и доп. — М.: Издательство Юрайт, 2016. — 353 с. — (Профессиональное образование).

6. Золотарева А. В. Методика преподавания по программам дополнительного образования детей [Текст]: учебник и практикум для СПО / А. В. Золотарева, Г. М. Криницкая, А. Л. Пикина. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2016. — 399 с. — (Профессиональное образование).

7. Кругликов Г. И. Методика профессионального обучения с практикумом [Текст]: учеб. пос. для студ. высш. учеб. зав-ий / Г. И. Кругликов. — 2-е изд., стер. — М.: Издательский центр «Академия», 2007. — 288 с.

8. Курс «Arduino для начинающих» [Электронный ресурс]. — Режим доступа: http://edurobots.ru/kurs-arduino-dlya-nachinayushhix (дата обращения 10.05.17).

9. Максимов П. В. Анализ одноплатных компьютеров, потенциально пригодных для использования в обучении [Текст] / П. В. Максимов, Ю. В. Корнилов // Педагогическое мастерство и педагогические технологии: материалы VI Междунар. науч.-практ. конф. (Чебоксары, 27 нояб. 2015 г.). В 2 т. Т. 2 / Редкол.: О. Н. Широков [и др.]. — Чебоксары: ЦНС «Интерактив плюс», 2015. — №4 (6). — С. 244-246.

10. Момот М. В. Мобильные роботы на базе Arduino [Текст] / М. В. Момот. — СПб.: БХВ-Петербург, 2017. — 288 с.: ил.

11. Морозова Н. А. Дополнительное образование — многоуровневая система в непрерывном образовании России [Текст] / Н. А. Морозова. — М.: МГУП, 2001. — 277 с.

12. Осадчий В. В., Осадчая Е. П. Анализ проблемы профессиональной подготовки программиста и пути ее решения [Текст]: журнал «Образовательные технологии и общество» / В. В. Осадчий, Е. Е. Осадчая. — Республика Татарстан, Казань: ФГБОУ ВПО «КНИТУ», 2014, том 17, выпуск № 3. — С. 362-377.

13. Осин А. В. Мультимедиа в образовании: контекст информатизации [Текст] / А. В. Осин. — М.: Агентство «Издательский сервис», 2004. – 320 с.

14. Переменные, типы переменных, объявление переменных, константы [Электронный ресурс]. — Режим па: https://myrobot.ru/stepbystep/pr_variables.php (дата обращения 19.04.17).

15. Петин В. А. Проекты с использованием контроллера Arduino [Текст] / В. А. Петин. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2016. — 464 с.: ил.

16. Родионов В. И. Подготовка электронных публикаций в InDesign CS6 [Текст] / В. И. Родионов. — СПб.: БХВ-Петербург, 2013. — 224 с.: ил.

17. Современное образование: робототехника в школе [Электронный ресурс]: Techno-guide. Technologies of the future — Режим па: http://techno-guide.ru/robototekhnika/sovremennoe-obrazovanie-robototekhnika-v-shkole.html (дата обращения: 05.05.2017).

18. Соммер У. Программирование микроконтроллерных плат Arduino / Freeduino [Текст] / У. Соммер. — СПб.: БХВ — Петербург, 2012. — 256 с. ил.

19. Старикова Л. Д. Методика профессионального обучения [Текст]: практикум / Л. Д. Старикова, Ю. С. Касьянова. — Екатеринбург: изд-во Рос. гос. проф.-пед. ун-та, 2013. — 131 с.

20. Степанова-Быкова А. С. Методика профессионального обучения [Электронный ресурс]: курс лекций / А. С. Степанова-Быкова, Т. Г. Дулинец. — Режим доступа: http://files.lib.sfu-kras.ru/ebibl/umkd/1513/u_lecture.pdf (дата обращения: 28.04.2017).

21. Филиппов С. А. Робототехника для детей и родителей [Текст]: учеб. пособие / С. А. Филиппов. — СПб.: Наука, 2013. – 319 с.

22. Халамов В. Н. Образовательная робототехника на уроках информатики и физики в средней школе [Текст]: уч.-метод. пособие / В. Н. Халамов. — Челябинск: Взгляд, 2011. — 160 с.

23. Чернилевский Д. В. Дидактические технологии в высшей школе [Текст]: учебное пособие для вузов / Д. В. Чернилевский. Москва: ЮНИТИ-ДАНА, 2002. — 437 с.

24. Эрганова Н. Е. Методика профессионального обучения [Текст]: учебное пособие / Н. Е. Эрганова. — 3-е изд., испр, и доп. — М: Академия, 2007. — 160 с.

25. Юревич Е. И. Основы робототехники [Текст] / Е. И. Юревич. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2005. — 416 с.

ПРИЛОЖЕНИЕ

Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Российский государственный профессионально–педагогический университет»

Институт инженерно–педагогического образования
Кафедра информационных систем и технологий
Направление подготовки 44.03.04 Профессиональное обучение
Профиль «Энергетика»
Профилизация Компьютерные технологии автоматизации и управления

УТВЕРЖДАЮ
Заведующая кафедрой ИС

_____ Н.С. Толстова
(подпись) (Фамилия И.О.)
« _____ » _____ 20 17г.

ЗАДАНИЕ
на выполнение **выпускной квалификационной работы** бакалавра
(дипломная работа)

студента (ки) **4** курса группы **КТэ-402**

Ушанова Кирилла Павловича
(фамилия, имя, отчество полностью)

1. Тема **Лабораторный практикум «Основы программирования на платформе Arduino»**

утверждена распоряжением по факультету от « ___ » _____ 20 17 г. № _____

2. Руководитель Нечаева Галина Лиминариевна
(фамилия, имя, отчество полностью)

_____ _____ старший преподаватель каф. ИС
(ученая степень) (ученое звание) (должность) (место работы)

3. Место преддипломной практики **УТЦ ООО «Омега-1»**

4. Исходные данные к ВКР
1. Петин В. А. Проекты с использованием контроллера Arduino
(список основной литературы)
2. Момот М. В. Мобильные роботы на базе Arduino
3. Соммер У. Программирование микроконтроллерных плат Arduino / Freeduino
4. Эрганова Н.Е. Методика профессионального обучения

5. Содержание пояснительной записки ВКР (перечень подлежащих разработке вопросов)
1) Изучение основ программирования
2) Разработка лабораторного практикума (требования, структура, содержание, реализация)
4) Список используемых источников литературы

5) Приложение

6. Перечень графических и демонстрационных материалов
Презентация, выполненная средствами Microsoft PowerPoint

7. Календарный план выполнения выпускной квалификационной работы

№ п/п	Наименование этапа дипломной работы	Срок выполнения этапа	Процент выполнения ВКР	Отметка руководителя о выполнении
1	Поиск информации по теме ВКР Работа над теоретическим разделом ВКР Сдача зачета по преддипломной практике	10.04.17– 19.05.17		(подпись)
2	Выполнение работ по разрабатываемым вопросам, их изложение в пояснительной записке ВКР:			(подпись)
	Выполнение и оформление теоретического раздела ВКР	15.04.17		(подпись)
	Работа над практическим разделом ВКР			(подпись)
	Выполнение и оформление практического раздела ВКР	21.04.17- 16.05.17		(подпись)
	Выполнение и оформление методического раздела	18.05.17		(подпись)
	Оформление ПЗ согласно требованиям	01-10.06.17		(подпись)
3	Оформление демонстрационных материалов: электронная презентация (плакаты) и подготовка доклада к предварительной защите	13.06.17		(подпись)
4	Допуск руководителя к защите	15.06.17		(подпись)
5	Допуск нормоконтроля	16.06.17		(подпись)
6	Предварительная защита	16.06.17		(подпись)
7	Получение рецензии, подготовка к защите			(подпись)
8	Защита ВКР	28.06.17		

9. Консультанты по разделам выпускной квалификационной работы

Наименование раздела	Консультант	Задание выдал		Задание принял	
Методическая часть		(подпись)	(дата)	(подпись)	(дата)
Нормоконтроль		(подпись)	(дата)	(подпись)	(дата)
Предварительная защита		(подпись)	(дата)	(подпись)	(дата)

Руководитель (подпись) (дата) Задание получил (подпись) (дата)

10. Пояснительная записка дипломной работы и все материалы проанализированы
Считаю возможным допустить **Ушанова К.П.** к защите выпускной квалификационной работы в государственной экзаменационной комиссии

Руководитель (подпись) (дата)

11. Допустить **Ушанова К.П.** (фамилия и.о. студента) к защите выпускной квалификационной работы в государственной экзаменационной комиссии (протокол заседания кафедры от « » 20 17 г., №)

Заведующая кафедрой (подпись) (дата)