

Муниципальное общеобразовательное бюджетное Учреждение Лицей № 6
муниципального района Мелеузовский район Республики Башкортостан

Учебно-методическая работа
на тему

**Создание управляемых устройств
на базе вычислительной платформы Arduino
и интеграция этого модуля в курс информатики**

Автор: учитель информатики и ИКТ
Габитов Артур Гаязович

Мелеуз 2016

Оглавление

Введение	3
Платформа Arduino.....	5
Мини-проекты с Arduino.....	12
Интеграция в курс информатики	21
Заключение.....	28
Список использованной литературы	30

Введение

Одной из ключевых проблем в России является её недостаточная обеспеченность инженерными кадрами, а также низкий статус инженерного образования при выборе будущей профессии выпускниками школ. Сейчас в средней школе необходимо активно начинать популяризацию профессии инженера. Детям нужны образцы для подражания в области инженерной деятельности. Конструирование и проектирование управляемых устройств на платформе Arduino направлено на внедрение и распространение лучших практик по профориентации талантливой молодёжи на инженерно-конструкторские специальности. Модульные занятия позволят почувствовать себя инженером, занимаясь, дети начинают чувствовать творческий путь от «идеи» до её «реализации», то есть весь производственный цикл. Микроэлектроника является эффективным методом для изучения важных областей науки, технологии, конструирования и математики.

Интерес подрастающего поколения к микропроцессорам и желание освоить современную мировую радио- и микроэлектронику делают педагогически целесообразным ознакомление учащихся с основами знаний в этих областях, используя технологию современного мирового уровня.

В кружковой деятельности 4-х классов уже наработан опыт по сборке проектов робототехнических устройств, поэтому МОБУ Лицей №6 города Мелеуз предполагает реализацию инновационного технического проекта "Школьная лаборатория микроэлектроники".

Цель данного проекта – создание современной высокоинтеллектуальной межпредметной образовательной среды для учащихся среднего и старшего звена для углубленного изучения основ знаний в самостоятельном создании управляемых устройств с дальнейшим применением их в общественной деятельности. Основными задачами проекта являются:

1. Изучение платформы Arduino;

2. Ознакомление с накопленной базой проектов, разработанных на Arduino;
3. Интеграция в курс информатики;
4. Обмен полученным опытом на муниципальном и республиканском уровне.

Наличие в МОБУ Лицей №6 15 мобильных компьютеров ученика позволяет легко использовать оборудование в разных учебных кабинетах и создать среду «1 ученик - 1 компьютер», тем самым, наиболее эффективно развивать самостоятельность учащихся в выполнении экспериментальных работ.

Основа учебного оборудования – это открытая платформа Arduino: микроконтроллер и его среда программирования. Arduino легко соединяется с разными электронными компонентами, позволяя создавать различные устройства автоматики. Научившись программировать, учащиеся смогут самостоятельно создавать интересные и полезные электронные устройства.

Платформа Arduino

Arduino – это небольшая плата с собственным процессором и памятью, на ней находятся контакты, к которым можно подключать всевозможные компоненты: лампочки, датчики, моторы, чайники, роутеры, магнитные дверные замки и многое другое, что работает от электричества.



В процессор Arduino можно загрузить программу, посредством подключения к компьютеру через интерфейс usb, которая будет управлять всеми этими устройствами по заданному алгоритму. Таким образом, можно создать бесконечное количество уникальных инновационных гаджетов, сделанных своими руками и по собственной задумке.

Arduino является прототипированной платформой с открытым исходным кодом на основе аппаратных средств и программного обеспечения. Плата умеет считывать входы: датчик света, палец на кнопке или сообщение в Twitter – и превращать их в выходной сигнал – запуск двигателя, включение светодиода, публикация в сети Интернет, непосредственно посылая команды микроконтроллеру на плате. Для этого используется язык программирования Arduino на основе Wiring и программное обеспечение Arduino IDE, основанное на обработке данных.

Долгое время Arduino был мозгом тысячи проектов: от простых бытовых предметов до сложных научных приборов. Всемирное сообщество создателей: студентов, любителей, художников, программистов и профессионалов – собрало вокруг этой платформы с открытым исходным кодом базу с невероятным количеством доступных знаний, которые могут быть большим подспорьем как для новичков, так и для экспертов.

Изначально Arduino простой инструмент для быстрого прототипирования, направленный на студентов без каких-либо знаний в

области электроники и программирования. Как только он достиг более широкого сообщества, плата Arduino начала меняться, чтобы адаптироваться к новым потребностям и задачам, дифференцируя свое предложение от простых 8-разрядных плат к продуктам для ИТН-приложений, пригодных для носки, 3D - печати и встраиваемых средах. Все Arduino платы полностью с открытым исходным кодом, расширенными правами и возможностями для пользователей. Они могут создавать их самостоятельно и в конечном итоге адаптировать к конкретным потребностям. За счет взносов пользователей по всему миру программное обеспечение используется с открытым исходным кодом.

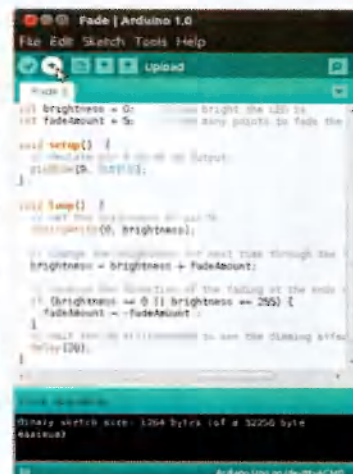
Простой и доступный пользовательский опыт Arduino используется в различных проектах и приложениях. Программное обеспечение Arduino является простым в использовании для начинающих, но достаточно гибким для продвинутых пользователей. Оно работает на Mac, Windows и Linux. Преподаватели и студенты используют его для создания научных инструментов с низкой стоимостью. Например, дизайнеры и архитекторы создают интерактивные прототипы, музыканты и художники используют его для работы с новыми инструментами. Arduino является ключевым инструментом для познания и создания новых вещей. Микроплата Arduino рассчитана на широкий круг заинтересованных пользователей. Уникальность этого проекта в том, что можно начать мастерить сразу же после изучения пошаговой инструкции или после обмена идеями онлайн с другими членами сообщества Arduino.

Есть много других микроконтроллеров и микроконтроллеров платформы, доступных для физических вычислений. Подобную функциональность предлагают не только Параллакс Basic Stamp, NETMEDIA в BX-24, Phidgets, Handyboard Массачусетского технологического института, но и многие другие.

Arduino упрощает процесс работы с микроконтроллерами и обладает некоторыми преимуществами по сравнению с другими системами:

- ❖ Относительно низкая стоимость: Arduino доски относительно недорого стоят по сравнению с другими платформами микроконтроллера. Наименее дорогая версия модуля Arduino может быть собрана вручную, и даже предварительно собранные модули Arduino имеют стоимость менее 50\$
- ❖ Кроссплатформенность: платформа Arduino программного обеспечения (IDE) работает на Windows, Mac OS и Linux операционных системах. Большинство систем микроконтроллера ограничены Windows.
- ❖ Простота и ясность среда программирования: платформа Arduino программного обеспечения (IDE) является простой в использовании для начинающих, но достаточно гибкой для продвинутых пользователей. Для учителей это удобство основано на среде программирования Processing, поэтому учащиеся, умеющие писать программы в этой среде, будут знакомы с тем, как работает Arduino IDE.
- ❖ Открытость исходного кода и расширяемость программного обеспечения: программное обеспечение Arduino выпускается в виде открытых источников, инструментов, доступных для расширения опытными программистами. Язык может быть расширен с помощью библиотек C++, при желании понять технические детали, можно сделать скачок от Arduino на языке программирования AVR C, на котором он основан. При этом можно добавить код AVR-C непосредственно в Arduino программы.
- ❖ Открытость исходного кода аппаратно-расширяемого модуля: платы Arduino публикуются под лицензией Creative Commons, поэтому опытные дизайнеры могут сделать свою собственную версию модуля, расширяя и улучшая его. Относительно неопытные пользователи могут строить версию макета модуля для того, чтобы понять, как это работает, и сэкономить деньги.

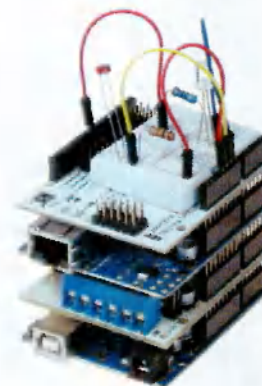
Свою бешеную популярность Arduino приобрела благодаря своей простоте и дружелюбности. Даже человек, не понимающий ничего в программировании и схемотехнике, может освоить основы работы с Arduino за пару часов, этому поспособствуют тысячи публикаций, учебников, заметок в интернете и отличная серия видеоуроков по Arduino на русском языке.



Программы для Arduino пишутся на обычном C++, дополненным простыми и понятными функциями для управления вводом/выводом на контактах. Если знать C++ – Arduino станет дверью в новый мир, где программы не ограничены рамками компьютера, а взаимодействуют с окружающим миром и влияют на него. Если же не понимать в программировании ничего – не проблема, можно с лёгкостью научиться.

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, нужно лишь только подключить плату к компьютеру через USB. Для более опытных возможна работа и через Visual Studio, Eclipse, другие IDE или командную строку.

Паяльник здесь не понадобится. Полноценные устройства можно собирать, используя специальную макетную доску, перемычки и провода абсолютно без пайки. Конструирование является быстрым и простым.



Ещё одной отличительной особенностью Arduino является наличие плат расширения, так называемых shields, или просто «шилдов». Это дополнительные платы, которые ставятся подобно слоям бутерброда поверх Arduino, чтобы дать ему новые возможности. Например, существуют платы расширения для подключения к локальной сети и интернету (Ethernet Shield), для управления

мощными моторами (Motor Shield), для получения координат и времени со спутников GPS (модуль GPS) и многие другие.

Arduino – это сердце конструктора, в котором нет конечного, определённого набора деталей и нет ограничений в разнообразии того, что можно собрать. Всё ограничено лишь фантазией. Это новый мир, убойное хобби и отличный подарок. Десятки тысяч людей в мире уже поняли это.

Для изучения понадобится сам Arduino. Arduino Uno – это самая популярная платформа в настоящий момент. Её последняя модель – Arduino Rev3, оригинальная, произведённая в Италии, выполнена на базе процессора ATmega328p с тактовой частотой 16 МГц, обладает памятью 32 кб и имеет 20 контролируемых контактов ввода и вывода для взаимодействия с внешним миром, из них 14 контактов (pins) могут быть использованы для цифрового ввода и вывода. Какую роль исполняет каждый контакт, зависит от программы. Все они работают с напряжением 5 В и рассчитаны на ток до 40 мА. Также каждый контакт имеет встроенный, но отключённый по умолчанию резистор на 20 - 50 кОм. Некоторые контакты обладают дополнительными ролями:

- ❖ Serial: 0-й и 1-й. Используются для приёма и передачи данных по USB.
- ❖ Внешнее прерывание: 2-й и 3-й. Эти контакты могут быть настроены так, что они будут провоцировать вызов заданной функции при изменении входного сигнала.
- ❖ PWM: 3-й, 5-й, 6-й, 9-й, 10-й и 11-й. Могут являться выходами с широтно-импульсной модуляцией (pulse-width modulation) с 256 градациями.
- ❖ LED: 13-й. К этому контакту подключен встроенный в плату светодиод. Если на контакт выводится 5 В, светодиод зажигается; при нуле светодиод гаснет.

Помимо контактов цифрового ввода/вывода на Arduino имеется 6 контактов аналогового ввода, каждый из которых предоставляет разрешение в 1024 градации. По умолчанию значение меряется между землёй и 5 В,

однако возможно изменить верхнюю границу, подав напряжение требуемой величины на специальный контакт AREF. Кроме этого на плате имеется входной контакт Reset. Его установка в логический ноль приводит к сбросу процессора. Это аналог кнопки Reset обычного компьютера.

Arduino Uno может питаться как от USB подключения, так и от внешнего источника: батарейки или обычной электрической сети. Источник определяется автоматически. Платформа может работать при наличии напряжения от 6 до 20 В. Однако при напряжении менее 7 В работа может быть неустойчивой, а напряжение более 12 В может привести к перегреву и повреждению, поэтому рекомендуемый диапазон 7–12 В. Также понадобится USB-кабель, макетная доска, перемычки, резисторы, транзистор и ещё десяток подручных вещей.

Также Arduino Uno обладает несколькими способами общения с другими Arduino, микроконтроллерами и обычными компьютерами. Платформа позволяет установить последовательное (Serial UART TTL) соединение через контакты 0 (RX) и 1 (TX). Установленный на платформе чип ATmega16U2 транслирует это соединение через USB: на компьютере становится доступен виртуальный COM-порт. Программная часть Arduino включает утилиту, которая позволяет обмениваться текстовыми сообщениями по этому каналу. Встроенные в плату светодиоды RX и TX светятся, когда идёт передача данных между чипом ATmega162U и USB компьютера. Отдельная библиотека позволяет организовать последовательное соединение с использованием любых других контактов, не ограничиваясь штатными 0-м и 1-м.

С помощью отдельных плат расширения становится возможной организация других способов взаимодействия, таких как ethernet-сеть, радиоканал, Wi-Fi. Arduino Uno обладает предохранителем, защищающим USB-порты вашего компьютера от перенапряжения и коротких замыканий. Хотя большинство компьютеров обладают собственными средствами защиты, предохранитель даёт дополнительную уверенность. Он разрывает

соединение, если на USB-порт подаётся более 500 мА, и восстанавливает его после нормализации ситуации. Размер платы составляет $6,9 \times 5,3$ см. Гнёзда для внешнего питания и USB выступают на пару миллиметров за обозначенные границы. На плате предусмотрены места для крепления на шурупы или винты. Расстояние между контактами составляет 0,1" (2,54 мм), но в случае 7-го и 8-го контакта расстояние - 0,16".

Для более мощных проектов можно рассмотреть Arduino Mega. Существует также готовые наборы, например: «Матрёшка Х», «Матрёшка Y» или «Матрёшка Z». В них собрано всё, что нужно для начала.



Мини-проекты с Arduino

Здесь представлено серия небольших проектов базового уровня с Arduino, макетной платой и различными «сырыми» компонентами. Мини-проекты выстроены в порядке от простого к сложному, для начала лучше выполнять их по очереди.

Проект 1. Маячок

В этом проекте показано просто мигание светодиода.

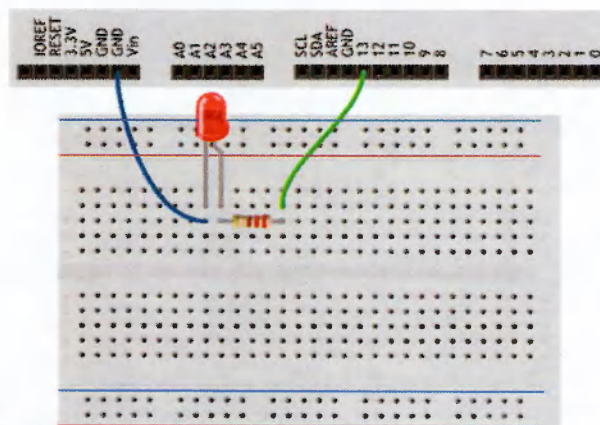
Список деталей:

- 1 плата Arduino Uno
- 1 беспаячная макетная плата
- 1 светодиод
- 1 резистор номиналом 220 Ом
- 2 провода «папа-папа»

Принципиальная схема:



Схема на макетке:



При сборке следует обратить внимание на следующее:

- Не забыть, как соединены рельсы в беспаячной макетной плате. Если на макетке красная и синяя линии вдоль длинных рельс прерываются в середине, значит проводник внутри макетки тоже прерывается.

- Катод («минус») светодиода — короткая ножка, именно её нужно соединять с землёй (GND)
- Не пренебрегать резистором, иначе светодиод выйдет из строя
- Выбрать резистор нужного номинала можно с помощью таблицы маркировки или с помощью мультиметра в режиме измерения сопротивления
- Плата Arduino имеет три пина GND, можно использовать любой из них

Скетч

```
void setup()
{
  // настраиваем пин №13 в режим выхода,
  // т.е. в режим источника напряжения
  pinMode(13, OUTPUT);
}
void loop()
{
  // подаём на пин 13 «высокий сигнал» (англ. «high»), т.е.
  // выдаём 5 вольт. Через светодиод побежит ток.
  // Это заставит его светиться
  digitalWrite(13, HIGH);
  // задерживаем (англ. «delay») микроконтроллер в этом
  // состоянии на 100 миллисекунд
  delay(100);
  // подаём на пин 13 «низкий сигнал» (англ. «low»), т.е.
  // выдаём 0 вольт или, точнее, приравниваем пин 13 к земле.
  // В результате светодиод погаснет
  digitalWrite(13, LOW)
  // замираем в этом состоянии на 900 миллисекунд
  delay(900);
  // после «размораживания» loop сразу же начнёт исполняться
  // вновь, и со стороны это будет выглядеть так, будто
  // светодиод мигает раз в 100 мс + 900 мс = 1000 мс = 1 сек
}
```

Пояснения к коду

- Процедура setup выполняется один раз при запуске микроконтроллера. Обычно она используется для конфигурации портов микроконтроллера и других настроек.

- После выполнения `setup` запускается процедура `loop`, которая выполняется в бесконечном цикле. Именно это используется в данном примере, чтобы маячок мигал постоянно
- Процедуры `setup` и `loop` должны присутствовать в любой программе (скетче), даже если не нужно ничего выполнять в них — пусть они будут пустые, просто не нужно ничего писать между фигурными скобками.

Например:

```
void setup()
{
}
```

- Каждой открывающей фигурной скобке `{` всегда соответствует закрывающая `}`. Они обозначают границы некоего логически завершённого фрагмента кода. Нужно следить за вложенностью фигурных скобок. Для этого удобно после каждой открывающей скобки увеличивать отступ на каждой новой строке на один символ табуляции (клавиша `Tab`)
- Обращать внимание на `;` в концах строк. Не стирать их там, где они есть, и не добавлять лишних. Будет понятно, где они нужны, а где нет.
- Функция `digitalWrite (pin, value)` не возвращает никакого значения и принимает два параметра:
 - `pin` - номер цифрового порта, на который отправляется сигнал
 - `value` - значение, которое отправляется на порт. Для цифровых портов значением может быть `HIGH` (высокое, единица) или `LOW` (низкое, ноль)
- Если в качестве второго параметра передать функции `digitalWrite` значение, отличное от `HIGH`, `LOW`, `1` или `0`, компилятор может не выдать ошибку, но считать, что передано `HIGH`.
- Используемые константы: `INPUT`, `OUTPUT`, `LOW`, `HIGH`, пишутся заглавными буквами, иначе компилятор их не распознаст и выдаст

ошибку. Когда ключевое слово распознано, оно подсвечивается синим цветом в Arduino IDE.

Проект 2. Светильник с управляемой яркостью.

В этом проекте меняется яркость светодиода, вращая ручку переменного резистора.

Список деталей:

- 1 плата Arduino Uno
- 1 беспаячная макетная плата
- 2 светодиода
- 2 резистор номиналом 220 Ом
- 8 проводов «папа-папа»
- 1 потенциометр

Принципиальная схема:

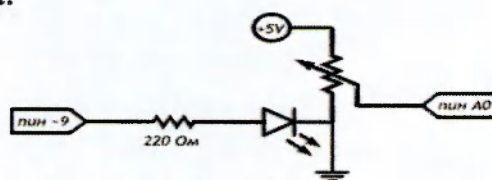
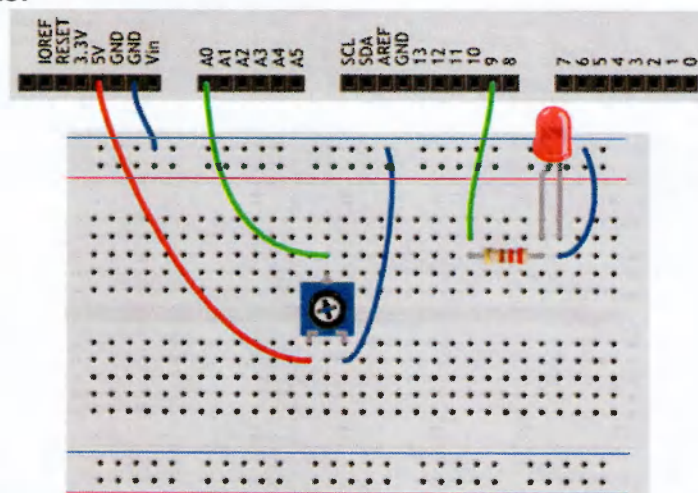


Схема на макетке:



При сборке следует обратить внимание на следующие:

- «Земля» светодиода и переменного резистора (потенциометра) подключается к длинному рельсу «-» макетной платы, и уже она соединяется с входом GND микроконтроллера. Таким образом

использовалось меньше входов и от макетки к контроллеру тянется меньше проводов.

- Подписи «+» и «-» на макетке не обязывают использовать их строго для питания, просто чаще всего они используются именно так, и маркировка в этом помогает.
- Не важно, какая из крайних ножек потенциометра будет подключена к 5 В, а какая к GND, поменяется только направление, в котором нужно крутить ручку для увеличения напряжения. Сигнал мы считывается со средней ножки.
- Для считывания аналогового сигнала, принимающего широкий спектр значений, а не просто 0 или 1, как цифровой, подходят только порты, помеченные на плате как «ANALOG IN» и пронумерованные с префиксом A. Для Arduino Uno — это A0-A5.

Скетч

```
// даём разумные имена для пинов со светодиодом
// и потенциометром (англ potentiometer или просто «pot»)
#define LED_PIN 9
#define POT_PIN A0
void setup()
{
  // пин со светодиодом — выход, как и раньше...
  pinMode(LED_PIN, OUTPUT);
  // ...а вот пин с потенциометром должен быть входом
  // (англ. «input»): мы хотим считывать напряжение,
  // выдаваемое им
  pinMode(POT_PIN, INPUT);
}
void loop()
{
  // заявляем, что далее мы будем использовать 2 переменные с
  // именами rotation и brightness, и что хранить в них будем
  // целые числа (англ. «integer», сокращённо просто «int»)
  int rotation, brightness;
  // считываем в rotation напряжение с потенциометра:
  // микроконтроллер выдаст число от 0 до 1023
  // пропорциональное углу поворота ручки
  rotation = analogRead(POT_PIN);
```



```

// в brightness записываем полученное ранее значение rotation
// делённое на 4. Поскольку в переменных мы пожелаем хранить
// целые значения, дробная часть от деления будет отброшена.
// В итоге мы получим целое число от 0 до 255
brightness = rotation / 4;
// выдаём результат на светодиод
analogWrite(LED_PIN, brightness);
}

```

Пояснения к коду

- С помощью директивы `#define` компилятору указано заменять идентификатор `POT_PIN` на `A0` - номер аналогового входа. Можно встретить код, где обращение к аналоговому порту будет по номеру без индекса `A`. Такой код будет работать, но во избежание путаницы с цифровыми портами нужно использовать индекс.
- Переменным принято давать названия, начинающиеся со строчной буквы.
- Чтобы использовать переменную, необходимо ее объявить, что делается инструкцией:

```
int rotation, brightness;
```

- Для объявления переменной необходимо указать ее тип, здесь - `int` (от англ. `integer`) — целочисленное значение в диапазоне от -32 768 до 32 767.
- Переменные одного типа можно объявить в одной инструкции, перечислив их через запятую.
- Функция `analogRead(pinA)` возвращает целочисленное значение в диапазоне от 0 до 1023, пропорциональное напряжению, поданному на аналоговый вход, номер которого мы передаем функции в качестве параметра `pinA`.
- Значение, возвращенное функцией `analogRead()` получается при помещении его в переменную `rotation` с помощью оператора присваивания `=`, который записывает то, что находится справа от него в ту переменную, которая стоит слева.

Проект 3. Светильник, управляемый по USB

В этом проекте отправляются устройству команды, как ему светит.

Список деталей:

- 1 плата Arduino Uno
- 1 беспаячная макетная плата
- 1 светодиод
- 1 резистор номиналом 220 Ом
- 2 провода «папа-папа»

Принципиальная схема:

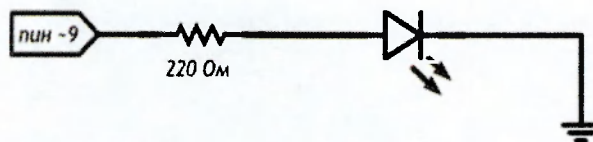
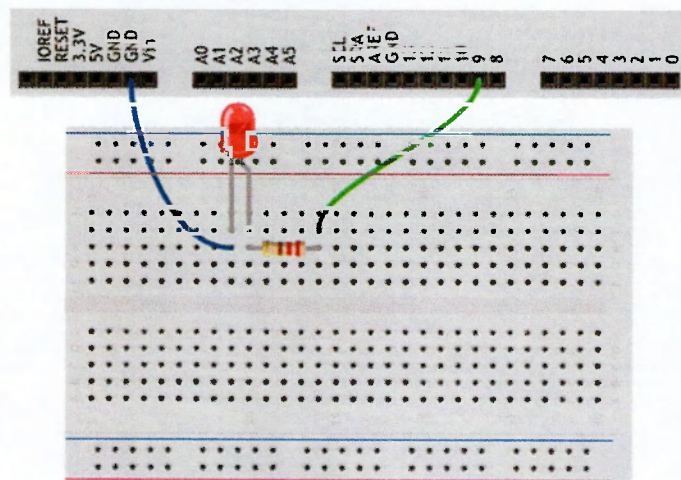


Схема на макетке:



Скетч

```
#define LED_PIN 9
// для работы с текстом существуют объекты-строки (англ. string)
String message;
void setup()
{
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  // передаваемые с компьютера данные поставляются байт за
  // байтом, в виде отдельных символов (англ. character). Нам
  // нужно последовательно их обрабатывать пока (англ. while)
```



```

// в порту доступны (англ. available) новые данные
while (Serial.available()) {
    // считываем (англ. read) пришедший символ в переменную
    char incomingChar = Serial.read();
    // не стоит путать целые числа и символы. Они соотносятся
    // друг с другом по таблице, называемой кодировкой. Например
    // '0' — это 48, '9' — 57, 'A' — 65, 'B' — 66 и т.п. Символы
    // в программе записываются в одинарных кавычках
    if (incomingChar >= '0' && incomingChar <= '9') {
        // если пришёл символ-цифра, добавляем его к сообщению
        message += incomingChar;
    } else if (incomingChar == '\n') {
        // если пришёл символ новой строки, т.е. enter, переводим
        // накопленное сообщение в целое число (англ. to integer).
        // Так последовательность символов '1', '2', '3' станет
        // числом 123. Результат выводим на светодиод
        analogWrite(LED_PIN, message.toInt());
        // обнуляем накопленное сообщение, чтобы начать всё заново
        message = "";
    }
}
// посылайте сообщения-числа с компьютера через Serial Monitor
}

```

Пояснения к коду

- В этой программе создается объект класса String. Это встроенный класс, предназначенный для работы со строками, т.е. с текстом.
- Не путать его с типом данных string, который является просто массивом символов. String же позволяет использовать ряд методов для удобной работы со строками.
- Новый вид циклов: цикл с условием while. В отличие от цикла со счетчиком for, цикл while(expression) выполняется до тех пор, пока логическое выражение expression истинно.
- Метод available() объекта Serial возвращает количество байт, полученных через последовательный порт.
- В данном проекте цикл while работает до тех пор, пока available() возвращает ненулевое значение, любое из которых приводится к true.
- Переменные типа char могут хранить один символ.

- В этом примере символ получается методом `Serial.read()`, который возвращает первый байт, пришедший на последовательный порт, или -1, если ничего не пришло.
- В `if` сравнивается не только пришедший символ с 0 и 9, но их коды. Если пришел какой-то символ, который не является цифрой, он не будет его добавлять к строке `message`.
- Объекты типа `String` позволяют производить конкатенацию, т.е. объединение строк. Это можно сделать так: `message = message + incomingChar`, но можно записать в сокращенной форме: `message += incomingChar`.
- В этой программе `if` дополняется конструкцией `else if`. Это еще один условный оператор, который проверяется только в случае ложности выражения, данного первому оператору. Несколько `else if` могут следовать друг за другом, при этом каждое следующее условие будет проверяться только в случае невыполнения всех предыдущих. Если в конце разместить `else`, он выполнится только если ни одно из условий не выполнено.
- Последовательностью `\n` кодируется символ переноса строки. Если он был передан устройству, то полученные ранее символы передаются как параметр для `analogWrite()`, которая включает светодиод.
- Используется один из методов `String`, `toInt()`, который заставляет считать строку не набором цифр, а числом. Он возвращает значение типа `long`, при этом, если строка начинается с символа, не являющегося цифрой, будет возвращен 0. Если после цифр, идущих в начале строки, будут символы не-цифры, на них конверсия остановится.
- Выпадающее меню внизу монитора порта - это чтобы устройство получало символ перевода строки, там должно быть выбрано «Новая строка (NL)»
- Пустая строка обозначается так: «`<>`». Опустошив ее, можно собирать новую последовательность символов.

Интеграция в курс информатики

Необходимость обращения к интегрированному обучению вызвана рядом объектных причин, которые обнаружились в процессе работы в школе. Одной из важнейших проблем является заметное снижение интереса учащихся к предметам физико-математического цикла, что во многом обусловлено объективной сложностью физики и математики. Уроки информатики дали возможность снять многие возникающие в процессе обучения познавательные трудности, вызвать интерес у учащихся к физическим и математическим проблемам, показать возможность их решения новыми, нестандартными методами: алгоритмизацией решения сложных задач на компьютере, возможность смоделировать и наглядно увидеть на экране дисплея физические и математические процессы и управлять этими процессами.

Комплексный подход к обучению физико-математическим предметам на основе информатики позволяют решить и проблемы обучения самой информатике. Например, предлагаемые в учебниках информатики задачи зачастую не имеют реальной практической ценности, выглядят формальным и не вызывает интереса у учащегося. Использование же компьютера по его прямому назначению (для решения практических задач, для выполнения громоздких, малоинтересных вычислений, для обработки большого объема информации и др.) усиливают практическую направленность как физики, математики, так и информатики; отражает современные методы исследования в этих отраслях научного знания, способствует устойчивому интересу учащихся к изучаемым предметам.

Интегрированное обучение позволяет наиболее эффективно показать межпредметные, метапредметные связи и естественнонаучный метод исследования, используемый на стыке наук. Целями интегрированного обучения являются:

1. Создание оптимальных условий для развития мышления учащихся в процессе обучения физике, математике, информатике на основе интеграции этих предметов.
2. Преодоление некоторых противоречий процесса обучения.
3. Повышение и развитие интереса учащихся к указанным предметам.

Интеграция знаний из различных предметов в Лицее осуществится с помощью изучения модуля «Основы программируемой микроэлектроники. Создание управляемых устройств на базе вычислительной платформы Arduino». Он позволяет с легкостью организовать полноценный учебный процесс в течении года. В теоретической части модуля изложена информация, необходимая для построения электрических схем, описываются физические принципы работы элементов, основы программирования. В практической части пошагово описывается процесс разработки проекта, начиная со сборки схемы и заканчивая написанием программы. Модуль предполагает знакомство с основами программирования на языке высокого уровня, ведь программирование формирует методологию познания. Ребенок начинает задавать вопрос: "Как оно работает?". Ответ же применяет ко всем явлениям и процессам в самых различных сферах жизни. Предметом изучения являются и методы разработки конструирования и программирования управляемых электронных устройств на базе вычислительной платформы (контроллера) Arduino.

На начальном этапе изучается среда визуального программирования Scratch for Arduino(S4A). Это творческая среда, в которой, помимо работы с Arduino, можно взаимодействовать с графикой и звуками. Программа на S4A состоит из блоков, которые окрашены в разные цвета, в зависимости от назначения они соединяются между собой, подобно элементам пазла. Программирование в данной среде развивает у детей абстрактное и логическое мышление, знакомит с основными принципами программирования и алгоритмизации. Затем учебная программа предусматривает переход на текстовое программирование средствами языка

C++, адаптированного под Arduino. Текстовое программирование позволяет использовать больше возможностей контроллера и создавать более сложные проекты.

Дальнейшее обучение посвящено наиболее перспективным областям IT - индустрии: криптография, биометрия, искусственный интеллект, Data Science. Последняя предусматривает работу с данными: сбор, обработка, визуализация больших объемов данных, а также принятие на их основе решения. Учащиеся знакомятся с мощными актуальными математическими пакетами программ, со средствами сбора и обработки данных. На данный момент отрасль IT является одной из самых востребованных и высокооплачиваемых. С роботизацией и автоматизацией мира уменьшается количество рабочих мест в сфере услуг требующих физического труда. В это время увеличивается потребность в технических специалистах.

Программа модуля направлена на развитие у детей логического мышления, на получение ими базовых знаний по информатике, программированию, схемотехнике и конструированию.

Итогами изучения модуля является знакомство с архитектурой аппаратной вычислительной платформы Arduino, умение работать с основными электронными компонентами: с цифровыми и аналоговыми датчиками, серводвигателями и шаговыми двигателями, LCD-дисплеями, семисегментными и матричными индикаторами: отработка навыков в построении алгоритмов в программировании для Arduino на визуальном языке Scratch и текстовом языке C++; получение сведений о передовых и наиболее перспективных областях IT – индустрии, такими как: криптография, искусственный интеллект, биометрия, data science; создание простых систем мониторинга и управления с использованием программного пакета для моделирования.

Учебно-тематическое планирование

№	Тема занятия	Содержание темы
1	Управление светом	Познакомиться с аппаратной платформой Arduino и визуальной средой программирования s4a. Получить базовые знания об электрическом токе, проводниках, беспаячной плате, некоторых электроэлементах. Познакомиться с правилами подключения светодиода и кнопки, научиться создавать программы для работы с ними.
2	Автоматический ночник	Получить знания о назначении, принципе работы и правилах подключения потенциометра и фоторезистора. Узнать о видах и режимах работы портов.
3	Охранная система	Получить знания о назначении, принципе работы и правилах подключения датчика движения, зуммера и RGB - светодиода. Создание охранной системы, реагирующей на движение.
4	Автоматическая поилка для домашних животных	Получить знания о назначении, принципе работы и правилах подключения серводвигателя и датчика уровня воды. Научиться использовать электроэлементы в комплексе с подручными средствами для создания простых систем автоматизации. Создать прототип автоматической поилки для домашних животных.
5	Визуальный термометр	Получить знания о назначении, принципе работы и правилах подключения датчика температуры. Научиться работать с объектами в среде S4A. Создать простую систему визуализации.
6	Дистанционный светильник	Получить знания об инфракрасной связи. Научиться работать с ИК приемником и пультом
7	Прототип игры «Кроты» с двумя кнопками	Получить навыки в создании сложных алгоритмов. Применить знания о работе с кнопками и светодиодами для создания игры. Создать прототип игры «Кроты» с двумя кнопками.
8	Игра «Кроты»	Закрепить навыки в создании сложных алгоритмов. Создать игру «Кроты» с пятью кнопками.

9	Электронное пианино	Познакомиться с IDE Arduino. Научиться создавать простейшие программы с помощью языка C++. Узнать о назначении и правилах подключения пьезоизлучателя. Создание прототипа пианино с кнопками и зуммером.
10	Часы на шаговом двигателе	Узнать о назначении, принципе работы и правилах подключения шагового двигателя. Познакомиться с такими понятиями, как переменная, библиотека, научиться создавать пользовательские функции. Освоить работу с библиотекой Stepper.h. Создать аналоговые часы с использованием шагового двигателя.
11	Сканер окружности «Sonar»	Узнать о назначении, принципе работы и правилах подключения ультразвукового дальномера. Изучить функции для приема и передачи данных через COM – порт. Освоить работу с библиотекой newping.h.
12	Инфракрасный дальномер	Узнать о назначении, принципе работы и правилах подключения инфракрасного дальномера. Освоить программную работу с ним.
13	Игра «Машинки»	Узнать о назначении, принципе работы и правилах подключения LCD – дисплея. Освоить работу с библиотекой liquidcrystal.h.
14	Игра «Рулетка»	Научиться моделировать игры, используя вычислительные способности платформы Arduino.
15	Игра «угадай число»	Узнать о назначении, принципе работы и правилах подключения семисегментного индикатора. Познакомиться с понятием динамическая индикация.
16	Будильник	Узнать о назначении, принципе работы и правилах подключения часов реального времени. Познакомиться с библиотеками, предназначенными для работы с ними.
17	«Змейка» на светодиодной матрице	Узнать о назначении, принципе работы, методах подключения и работы со светодиодной матрицей, джойстиком. Освоить библиотеку, предназначенную для управления светодиодной матрицей. Реализовать игру «Змейка».
18	Набор номера телефона с	Узнать о назначении, принципе работы и правилах подключения матричной клавиатуры. Реализовать

	использовани ем матричной клавиатуры.	симулятор телефона.
19	Создание калькулятора	Закрепить полученные знания о работе с матричной клавиатурой и LCD – дисплеем. Создание программы калькулятора.
20	Мини- метеостанция на LCD – мониторе	Узнать о назначении, принципе работы и правилах подключения датчика влажности и температуры. Познакомиться с его протоколом передачи данных. Создание мини-метеостанции на LCD – мониторе.
21	Удержание горизонтальн ого уровня	Узнать о назначении, принципе работы и правилах подключения датчика наклона. Реализовать удержание горизонтального уровня планки, прикрепленной к серводвигателю.
22	Музыкальный центр.	Научиться различать сигналы, получаемые с каждой кнопка ИК пульта. Создание управляемого с пульта музыкального центра, используя зуммер, ИК приемник и пульт.
23	Вход по пропускам.	Узнать о назначении, принципе работы и правилах подключения RFID модуля. Познакомиться с библиотекой softwareserial.h.
24	Сейф с двойной защитой	Закрепить полученные навыки работы с RFID модулем и матричной клавиатурой для создания защитной системы сейфа.
25	Азбука Морзе.	Познакомиться с историей передачи сигналов. Научиться кодировать сообщения с использованием азбуки Морзе.
26	Шифр Цезаря.	Познакомиться с понятием «криптография». Узнать о принципе шифрования и взлома шифра Цезаря. Реализовать шифр Цезаря.
27	Шифр Виженера.	Познакомиться с принципом шифрования и взлома шифра Виженера. Реализовать шифрование сообщения с использованием шифра Виженера.
28	Введение в работу с информацией.	Познакомиться с понятием «Data Science». Узнать о назначении, принципе работы и возможностях различных систем сбора и обработки данных.
29	Мониторинг уровня	Познакомиться со средой моделирования с симуляции JMCAD. Научиться строить графики по

	освещенности	накопленным значениям, которые были получены с датчиков.
30	Визуальный монитор порта.	Познакомиться со средой Scilab. Научиться принимать и визуализировать данные с порта.
31	Домашняя метеостанция.	Закрепить навыки работы с данными, получаемыми от нескольких датчиков. Реализовать мониторинг нескольких параметров.
32	Домашняя метеостанция с контролем освещенности	Изучить элементы теории автоматического управления для регулирования освещенности. Реализовать систему автоматического управления освещенностью.
33	Итоговое занятие	Закрепить навыки работы
34	Итоговый проект	Закрепить навыки проектирования и конструирования

Заключение

Arduino – это открытая платформа, которая позволяет конструировать всевозможные электронные устройства. Arduino будет интересна креативщикам, дизайнерам, программистам и всем пытливым умам, желающим собрать собственный гаджет. Устройства могут работать как автономно, так и в связке с компьютером. Всё зависит от идеи.

Электронный конструктор Arduino – это удобная платформа быстрой разработки электронных устройств. Программируется на специальном языке программирования, который основан на C/C++. Работу созданного алгоритма можно наглядно проверить на устройстве. Платформа Arduino позволяет не просто собирать всевозможные электронные устройства и их программировать, но и проводить экспериментальные и исследовательские лабораторные работы, стимулирующие познавательную активность учащихся. Это важнейшее условие эффективности образовательного процесса.

Сейчас каждого ребёнка окружает немыслимое число различных устройств, но, к сожалению, дети не имеют представления, как работают эти гаджеты. Образовательная среда на основе Arduino позволит вместе с учениками исследовать, как создаются такие устройства, как ими управлять, как их проектировать. Учащиеся придут к выводу, во-первых, что ходом физических процессов можно управлять с помощью программ, во-вторых, что в основе любого устройства находится контроллер, который реагирует на изменения одних параметров изменением других. В итоге складывается современный базовый межпредметный инженерный модуль. Создание в Лицее системы подобных инженерных модулей позволит ребятам, проявляя свои творческие способности, окунуться в мир интерактивных объектов, проводить опыты, исследования и эксперименты в метапредметной среде, развивая инновационное мышление и формируя качественно новый взгляд на сегодняшний технологический мир. Это будет взгляд инженера и дизайнера, а не потребителя.

Модуль «Основы программируемой микроэлектроники. Создание управляемых устройств на базе вычислительной платформы Arduino» предназначен для учащихся общеобразовательных учреждений. Для его освоения не требуется каких-либо специализированных знаний в электротехнике и программировании. В нем рассматривается проектирование, отлаживание и изготовление современных электронных устройств. Физические принципы работы электронных схем и различных типов радиоэлектронных компонентов иллюстрируются практическими примерами в виде моделей автоматических систем управления. На доступном уровне излагаются теоретические основы цифровой техники, иллюстрируется, как микроконтроллеры взаимодействуют с окружающим миром.

На занятиях учащиеся узнают основы электричества, изучат устройство некоторых электроприборов и их применение; получают практический опыт по сборке схем различных устройств и приборов, начиная от простейшего выключателя лампочки до цифрового дисплея; получают начальные знания в электричестве и схемотехнике; узнают об устройстве контроллера и его функциях; приобретут понимание единства информационных принципов строения и функционирования самоуправляемых систем различной природы.

Arduino прекрасно подходит для проведения занятий по темам: алгоритмизация, моделирование, информационные основы процессов управления. Государственные программы по информатике позволяют проводить экспериментальные, лабораторные и исследовательские работы в такой системе непосредственно на уроках информатики, а также на факультативных занятиях или элективных курсах.

Список использованной литературы

1. Arduino Programming Notebook / Brian W. Evans = Гололобов В. Н. Блокнот программиста Arduino: пер. на рус. / редакция и правка команда сайта <http://robocraft.ru> – USA, 2007. – 40 с.
2. Гололобов В.Н. С чего начинаются роботы. О проекте Arduino для школьников (и не только). – М.: Электронное издание, 2011. – 189 с.
3. Программирование микроконтроллерных плат Arduino / Freeduino. / Сомер У. – СПб.: БХВ Петербург, 2012. – 256 с.
4. Петин В.А. Проекты с использованием контроллера Arduino. – СПб.: БХВ Петербург, 2014. – 400 с.
5. Петин В.А. Проекты с использованием контроллера Arduino. – 2-е изд., перераб. и доп. – СПб.: БХВ Петербург, 2015. – 464 с.
6. Степанов С. Путеводитель по ARDUINO // Радио-ежегодник. – 2015. – №34. – 418 с.
7. <http://arduino.ru>
8. <https://www.arduino.cc>
9. <http://amperka.ru>
10. <http://wiki.amperka.ru>