

Тема урока: Знакомство с контроллером Arduino

Цели урока:

- формирование представления о микроконтроллерах;
- знакомство со средой разработки Arduino IDE;
- создание программы для микроконтроллера Arduino.

Задачи урока:

обучающие:

- формирование у учащихся представления о микроконтроллерах;
- знакомство учеников со средой разработки Arduino IDE;
- создание школьниками своей первой программы для микроконтроллера Arduino;
- развитие ключевых компетенций обучающихся (познавательных и информационных);
- обучение учащихся самостоятельной работе с дополнительной литературой и с источниками сети Интернет;
- развитие интереса учащихся к предмету, привлечение внимания к увиденному и услышанному;

развивающие:

- развитие мыслительной деятельности учащихся посредством постановки проблемных вопросов;
- обучение сравнению;
- развитие у учащихся умения самостоятельно делать выводы;

воспитательные:

- развитие познавательного интереса обучающихся;
- усиление познавательной мотивации путем осознания учеником своей значимости в образовательном процессе;
- создание условий для развития самостоятельности, сосредоточенности, самооценки, самоконтроля, дисциплинированности, ответственности и требовательности к себе.

Класс: VIII—IX.

Место урока в учебном плане: данный урок является первым в курсе «Робототехника».

Тип урока: урок открытия новых знаний.

Педагогические технологии, методы и приемы, используемые на уроке: технологии на основе активизации и интенсификации деятельности учащихся.

Планируемые образовательные результаты:

личностные:

- проявление мотивации к изучению нового учебного материала;
- способность к самооценке на основе критерия успешности учебной деятельности;

метапредметные:

регулятивные УУД:

- постановка учебных задач под руководством учителя;
- планирование учебной деятельности;
- удержание цели и задач в течение урока;
- оценивание своей деятельности на уроке на основе критериев достижения результата учебно-познавательной задачи урока;

познавательные УУД:

- осуществление поиска, извлечения и представления информации о силе тока, напряжении и т. д.;
- моделирование работы мигания светодиода «маячок»;
- наблюдение, сравнение, умозаключение;
- установление причинно-следственных связей;

коммуникативные УУД:

- построение монологического высказывания;
- работа с возражениями оппонентов: обсуждение разных точек зрения, умение договариваться в случае разногласий, аргументировано отстаивать свою точку зрения;

предметные УУД:

- представление о микроконтроллерах;
- знакомство со средой разработки Arduino IDE;
- навыки разработки программы для микроконтроллера Arduino.

Оснащение урока:

- компьютер;
- мультимедийный проектор;
- экран;
- программа Arduino IDE;
- сайт «Амперка»: <http://wiki.amperka.ru/>
- плата Arduino;
- на каждую пару учащихся (на каждый стол):
 - ~ распечатанный план проведения эксперимента «Маячок» (см. Приложение);
 - ~ макетная плата;
 - ~ проводники;
 - ~ светодиод;
 - ~ резистор;
- видео сопровождение для физкультминутки — «Гимнастика для глаз»: <https://www.youtube.com/watch?v=68kDApw-eLU>

Список источников, использованных при подготовке урока:

1. Бачинин А., Панкратов В., Накоряков В. Основы программирования микроконтроллеров. М.: ООО «Амперка», 2013.
2. Платт Ч. Электроника для начинающих/пер. сангл. СПб.: БХВ-Петербург, 2012.
3. Сайт «Амперка»: <http://wiki.amperka.ru/>
4. Соммер У. Программирование компьютерных плат Arduino/Freduino. СПб.: БХВ Петербург, 2012.
5. Arduino//Википедия—свободная энциклопедия. <https://ru.wikipedia.org/wiki/Arduino>
6. Arduino — Compare. <http://arduino.cc/en/Products.Compare>

План урока.

1. Организационный момент — 1 мин.
2. Установка познавательной задачи — 3 мин.
3. Усвоение и закрепление новых знаний — 17 мин.
4. Физкультминутка для глаз — 1 мин.
5. Рефлексия — 15 мин.
6. Подведение итогов урока — 3 мин.

Ход урока

1. Организационный момент

Формируемые УУД:

- *личностные*: психологическая готовность учащихся к уроку; самоопределение.

Учитель приветствует обучающихся, создает положительный эмоциональный настрой на урок.

Учитель. Ежедневно мы все встречаемся с электронными устройствами. Чуть ли не для каждого действия в нашей жизни существует устройство, которое его облегчает, — телефоны, стиральные машины, телевизоры, компьютеры.

2. Установка познавательной задачи

Формируемые УУД:

- *личностные*: самоопределение (мотивация учения);
- *познавательные*: обще учебные (самостоятельное выделение и формулирование познавательной цели);
- *коммуникативные*: владение монологической и диалогической формами речи в соответствии с грамматическими и синтаксическими нормами родного языка, умение слушать;
- *регулятивные*: целеполагание (постановка учебной задачи); планирование; логические (анализ с целью выделения признаков, подведение под понятие) действия постановки и решения проблем (формулировка проблемы, самостоятельное создание способов решения проблемы).

Учитель. Что такое микроконтроллер? Микроконтроллер — это микросхема, в которой уже есть процессор, оперативная память и флеш-память.

Можно ли задавать логику поведения микроконтроллера? Да, микроконтроллер, как и компьютер, можно программировать.

Итак, сегодня мы познакомимся с микроконтроллером Arduino, со средой программирования, напишем свою первую программу — заставим Arduino мигать светодиодом.

В ходе беседы ученики отвечают на вопросы учителя, называют тему урока, ставят цель

урока, формулируют задачи урока; записывают тему в тетрадь.

3. Усвоение и закрепление новых знаний

Формируемые УУД:

- *личностные:* возможность самостоятельно осуществлять деятельность обучения;
- *познавательные:* умение классифицировать знания; умение извлекать необходимую информацию из текста;
- *коммуникативные:* умение слушать; постановка вопросов — инициативное сотрудничество в поиске и сборе информации;
- *регулятивные:* планирование своей деятельности для решения поставленной задачи; контроль полученного результата; целеполагание; коррекция — внесение необходимых дополнений и корректив в план и способ действия в случае расхождения с эталоном.

Учитель. Микроконтроллер часто является мозговым центром платы, отвечающей за определенную задачу: на него приходят все сигналы, поступающие на плату, а он в свою очередь раздает команды всем устройствам, подключенным к ней.

Микроконтроллеры используются повсеместно: от бытовых кухонных устройств до элементов управления космическим кораблем, от домофонов до систем безопасности в автомобиле, от радиоуправляемых игрушек до роботов на конвейере завода. Микроконтроллер, как и компьютер, можно программировать, т. е. задавать ему логику поведения, например, микроконтроллер будет включать светодиод при низкой внешней освещенности или будет мигать светодиодом каждую секунду.

Итак, допустим, у нас есть микроконтроллер. Как, например, заставить лампочку мигать раз в секунду? Раньше этот процесс был трудоемким, но с помощью платы Arduino его можно заметно упростить: написать

программный код и «залить» его в микроконтроллер через USB-порт.

Давайте разберемся, как управлять Arduino. Для этого нам понадобится программа — среда разработки Arduino IDE. У нас на компьютерах используется операционная система Windows, поэтому Arduino работает сразу из коробки без установки дополнительных драйверов. О подключении Arduino в Windows читаем на: [http:// wiki.amperka.ru/arduino-быстрый-старт](http://wiki.amperka.ru/arduino-быстрый-старт): start

Запустите среду Arduino IDE: *Меню приложений, Разработка, IDE, Arduino.*

Перед нами окно Arduino IDE (рис.1).

Обратите внимание мы еще не подключаем нашу плату Arduino Uno к компьютеру, а в правом нижнем углу уже красуется надпись «Arduino Uno on/ваш порт». Таким образом, программа Arduino IDE сообщает нам, что в данный момент она настроена на работу с целевой платой Arduino Uno. А когда придет время, Arduino IDE будет искать Arduino Uno на соответствующем порту. Позже мы поменяем эти настройки.

В центральную часть среды Arduino IDE пишется программный код — скетч — это текст, понятный человеку и компьютеру (рис.2). Нажав кнопку *Компилировать*, вы сможете проверить свой код на правильность. Если внизу окна вы видите надпись «Done compiling» («Компилирование выполнено»), то вы написали все верно. В противном случае вы увидите сообщение об ошибке. Чтобы написанный нами скетч перенесся с компьютера на микроконтроллер Arduino, нужно загрузить его. Это можно сделать при помощи кнопки *Upload*. Точно также при удачной загрузке внизу появится строка «Done uploading» («Загрузка выполнена»).

Учащиеся слушают объяснения учителя, отвечают на вопросы, участвуют в обсуждении, собирают цепь по схеме, пишут код на компьютере, запускают его на микроконтроллере и проверяют работоспособность установки.

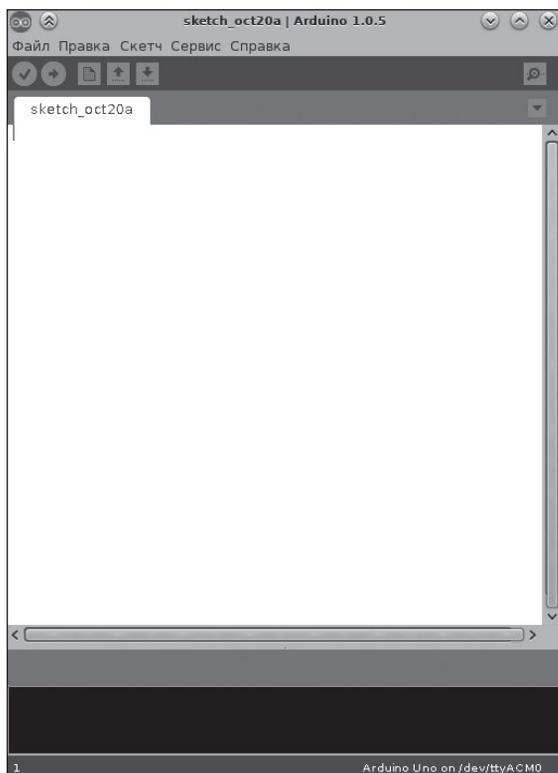


Рис.1. Окно среды разработки Arduino IDE



Рис.2. Скетч в среде разработки Arduino IDE

Учитель. Перейдем к нашему первому эксперименту «Маячок» (см. Приложение): <http://wiki.amperka.ru/конспект-arduino:маячок>.

Подсоедините вашу плату к компьютеру через USB. Вы увидите, что на Arduino загорится светодиод, сигнализирующий о готовности к работе.

Вставьте в среду скетч, предложенный в эксперименте. Давайте разберем пояснения к скетчу. Какие процедуры используются? (Ответы учащихся) Обратите внимание на то, как они записываются.

Итак, код готов. Подготовьте на макетной плате светодиод и резистор по принципиальной схеме, показанной в описании, и подключите их к плате Arduino.

Загрузите вашу первую программу на Arduino. Подождите некоторое время, и светодиод замигает, как «маячок».

Учащиеся самостоятельно выполняют задание в группах по два-три человека.

Учитель. Это успех, это ваша первая работающая программа для микроконтроллера! На следующих занятиях мы подробно разберем, почему и как это работает.

4. Физкультминутка для глаз

На экране демонстрируется физкультминутка — анимация «Гимнастика для глаз»: <https://www.youtube.com/watch?v=68kDApw-eLU>
Ребята водят глазами, следя за перемещением предметов, появляющихся на экране.

5. Рефлексия

Формируемые УУД:

- *личностные*: адекватное понимание причин успеха/не успеха в учебной деятельности;
- *познавательные*: построение речевого высказывания в устной форме; контроль и оценка процесса и результатов деятельности;
- *коммуникативные*: умение слушать и вступать в диалог; формулирование и аргументация своего мнения; контроль, коррекция, оценка действий партнеров;
- *регулятивные*: контроль и оценка своей деятельности в рамках урока.

Учитель. Ребята, давайте ответим на вопросы, указанные в описании эксперимента. (*Учащиеся отвечают на вопросы.*)

Выполните задания для самостоятельного решения.

Обучающиеся работают в группах по два-три человека, обсуждают поставленные задачи, а затем презентуют свое решение всем группам.

Учитель. Сформулируйте, соответствуют ли полученные знания вашим ожиданиям от урока.

6. Подведение итогов урока

Формируемые УУД:

- *личностные*: умение слушать; умение задавать вопросы.

Учитель. Сегодня на уроке мы познакомились с микроконтроллером Arduino. Вы написали вашу первую работающую программу для микроконтроллера. Для желающих в продолжение данной темы предлагаю прочитать дополнительно теорию по основам электричества: <http://wiki.amperka.ru/>

На ваш взгляд, пригодятся ли вам эти знания в жизни и почему? (*Ответы учащихся.*)

На следующих уроках мы будем узнавать все больше и больше о плате Arduino. И, может быть, кто-то из вас в будущем захочет стать настоящим программистом или электронщиком, чтобы создавать новых роботов. Спасибо за внимание.

Эксперимент «Маячок»

В этом эксперименте просто мигает светодиод.

Прочтите перед выполнением:

- Понятие электричества.
- Принципиальные схемы.
- Основные законы электричества.
- Управление электричеством.
- Быстрая сборка схем.
- Резистор.
- Диод.
- Светодиод.
- Начало работы с Arduino.

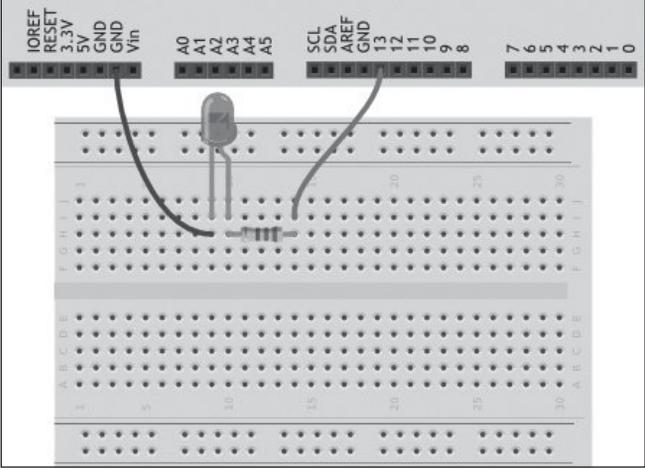
Список деталей для эксперимента:

- Плата Arduino Uno;
- 1 беспаечная макетная плата;
- 1 светодиод;
- 1 резистор номиналом 220 Ом;
- 2 провода «М—М».

Принципиальная схема:



Схема на макетке:



Обратите внимание:

- Не забудьте, как соединены рельсы в безопасной макетной плате. Если на вашей макетке красная и синяя линии вдоль длинных рельсов прерываются в середине, значит, проводник внутри макетки тоже прерывается!
- Катод («минус») светодиода — короткая ножка, именно ее нужно соединять с «землей» (GND).
- Не пренебрегайте 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, используйте любой из них.

Скетч:

```
p010_blink.ino voidsetup()
{
  //настраиваем пин №13 в режим выхода,
  //т.е. в режим источника напряжения
  pinMode(13,OUTPUT);
}

voidloop()
{
  //Подаем на пин 13 «высокий сигнал»
  //(англ. high), т. е. выдаем 5 вольт.
  //Через светодиод побежит ток.
  //Это заставит светодиод светиться
  digitalWrite(13,HIGH);

  //Задерживаем (англ. delay) микроконтроллер
  //в этом состоянии на 100 миллисекунд
  delay(100);

  //Подаем на пин 13 «низкий сигнал»
  //(англ. low), т. е. выдаем 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
  //начнет исполняться вновь, и со стороны
  //это будет выглядеть так, будто
  //светодиод мигает раз
  //в 100мс 900мс ● 1000мс ● 1сек
}

```

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 этим мы пользуемся в данном примере, чтобы маячок мигал постоянно.

- Процедуры `setup` и `loop` должны присутствовать в любой программе (скетче), даже если вам не нужно ничего выполнять в них, — пусть они будут пустые, просто не пишите ничего между фигурными скобками. Например:

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

- Запомните, что каждой открывающей фигурной скобке { всегда соответствует закрывающая }. Они обозначают границы некоего логически завершенного фрагмента кода. Следите за вложенностью фигурных скобок. Для этого удобно после каждой открывающей скобки увеличивать отступ на каждой новой строке на один символ табуляции (клавиша Tab).
- Обращайте внимание на ; в концах строк. Не стирайте их там, где они есть, и не добавляйте лишних. В скором времени вы будете понимать, где они нужны, а где нет.
- Функция `digitalWrite (pin,value)` не возвращает никакого значения и принимает два параметра:
 - 1) `pin` — номер цифрового порта, на который мы отправляем сигнал;
 - 2) `value` — значение, которое мы отправляем на порт. Для цифровых портов значением может быть HIGH (высокое, единица) или LOW (низкое, ноль).
- Если в качестве второго параметра вы передадите функции `digitalWrite` значение, отличное от HIGH, LOW, 1 или 0, компилятор может не выдать ошибку, но считать, что передано HIGH. Будьте внимательны!
- Обратите внимание, что использованные нами константы INPUT, OUTPUT, LOW, HIGH пишутся прописными буквами, иначе компилятор их не распознает и выдаст ошибку. Когда ключевое слово распознано, оно подсвечивается синим цветом в Arduino IDE.

Вопросы для самопроверки:

1. Что будет, если подключить к земле анод светодиода в место катода?
2. Что будет, если подключить светодиод с резистором большого номинала (например, 10 кОм)?
3. Что будет, если подключить светодиод без резистора?
4. Зачем нужна встроенная функция `pin Mode`? Какие параметры она принимает?
5. Зачем нужна встроенная функция `digital Write`? Какие параметры она принимает?
6. С помощью какой встроенной функции можно заставить микроконтроллер ничего не делать?
7. В каких единицах задается длительность паузы для этой функции?

Задания для самостоятельного решения:

1. Сделайте так, чтобы маячок светился полсекунды, а пауза между вспышками была равна одной секунде.
2. Измените код примера так, чтобы маячок включался на три секунды после запуска устройства, а затем мигал в стандартном режиме.