

Научно-исследовательская работа

Информатика

**ИСПОЛЬЗОВАНИЕ ПЛАТФОРМЫ ARDUINO
НА УРОКАХ ИНФОРМАТИКИ ДЛЯ ФОРМИРОВАНИЯ
НАВЫКОВ ПРОГРАММИРОВАНИЯ ПРИ СОЗДАНИИ
СОБСТВЕННЫХ ПРОЕКТОВ**

Выполнил:

Бедин Даниил Артемович

учащийся 9 класса

МБОУ Лицей № 131 “Созвездие”, Россия, г. Самара

Руководитель:

Кузнецова Елена Сергеевна

Учитель математики и информатики

МБОУ Лицей № 131 “Созвездие”, Россия, г. Самара

Введение

Современный мир полон информационных технологий, технических достижений и открытий. Развитие этого мира происходит настолько быстро, что мы даже не успеваем за ним. Но ведь не только информационные технологии должны идти вперед, еще и педагоги на уроках информатики должны соответствовать современным технологиям, а также требованиям рынка и запросам общества.

В XXI веке очень популярна отрасль робототехники. Об этом может символизировать появление открытых платформ и сред для разработки роботов, а также влияние таких гигантов индустрии, как Google, который ради развития данной отрасли приобрел себе Boston Dynamics. Но в данной статье мы не будем анализировать работу таких крупных компаний, специализирующихся на роботостроении, а постараемся рассмотреть открытую платформу Arduino в качестве средства для формирования творческого мышления, а также навыков в программировании на уроках информатики.

Плюсом выбранной платформы является то, что она полностью открыта, то есть любой человек может найти как официальные платы и расширения, так и неофициальные. С помощью данной платформы можно сделать уроки информатики по теме “Программирование” более интересными и наглядными. Данная платформа создания проектов не требует навыков работы с паяльником, что делает уроки более безопасными. Число проектов, которые можно реализовать с помощью данной платформы, ограничивается только фантазией человека. Однако, несмотря на то, что данная платформа открыта и является одной из наименее материально затратных, обучающее учреждение

может не иметь данного оборудования, а использование различных web-ресурсов никогда не сможет заменить всю полноту эмоций и ощущений от реального комплекта Arduino с различными платами и расширениями.

Использование платформы Arduino

Наверное, каждый ребенок в детстве любил конструировать различные объекты и взаимосвязи между ними с помощью металлического или пластмассового конструктора. Для нас открываются новые интересные технологии, которые на порядок сложнее, чем складывание блоков конструктора. Например, электроника или робототехника может, конечно, очень увлечь школьника, но алгоритм проектировки и разработки какого-либо робота или электронного устройства — совсем не такая простая вещь, как конструктор “LEGO”. Исходя из этой проблемы, итальянские разработчики представили всему миру свою разработку под названием Arduino, которая довольно быстро начала набирать популярность.

Arduino — это инструмент для проектирования электронных устройств, более плотно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры, которые фактически не выходят за рамки виртуальности. Arduino также является торговой маркой аппаратно-программных средств для построения простых систем автоматизации и робототехники, ориентированной на непрофессиональных пользователей.

Сама плата Arduino может быть представлена в нескольких вариантах, каждый из которых имеет определенную частоту контроллера, фиксированное количество памяти, число разъемов ввода и вывода информации. Таким

образом, каждая плата подходит под те или иные нужды пользователя для разработки им различных проектов (таблица).

Таблица

Сопоставление плат Arduino

| Arduino | CPU | | Flash memory | SRAM | Interface | Digital (PWM) | Analog | Size |
|--------------|--------------------------------|-----|-----------------|------|------------------|------------------|--------|-----------------|
| | View | MHz | Kb | Kb | Input/ Output | Output | Input | Mm |
| Nano | ATmega 168 or ATmega 328 | 8 | 16/32 | 1 | 14 | 6 | 8 | 43 x 18 |
| Diecimila | ATmega 168 | 16 | 16 | 1 | 14 | 6 | 6 | 68.6 x 53.3 |
| Uno | ATmega 328P | 16 | 32 | 2 | 14 | 6 | 6 | 68.6 x 53.3 |
| Leonardo | ATmega 32U4 | 16 | 32 | 2 | 14 | 6 | 12 | 68.6 x 53.3 |
| Mega 2560 | ATmega 2560 | 16 | 256 | 8 | 54 | 14 | 16 | 101.6 x 53.3 |
| Due | SAM3X8E ARM Cortex-M3 | 87 | 512 | 256 | 54 | 12 | 12 | 101.6 x 53.3 |

Если рассматривать варианты как desktop-версии для обучения школьников, то тут лучше всего использовать плату Arduino UNO, так как именно эта версия платы получила наибольшую популярность, к тому же, она имеет достаточно высокую частоту микроконтроллера и в ней относительно большое количество памяти, которого однозначно хватит для большинства проектов.

С одной платой мы, конечно, можем “поиграть”, но для полного понимания, изучения и наглядности программирования необходимы электронные компоненты: макетная плата (bread board) и платы расширения. К последним относятся: датчики освещенности, температуры, влажности, дисплейные модули, инфракрасные приемники, передатчики и еще многое другое.

При написании любой программы под определенную архитектуру и различные микроконтроллеры нужен язык программирования, соответствующий определенной технологии, поэтому и для Arduino был разработан Integrated Development Environment (IDE), который базируется на упрощенном языке программирования C++.

Как было указано выше, платформа Arduino является полностью открытой и свободной, как и среда ее разработки, что способствовало появлению ее альтернативных вариантов.

В данной статье мы будем рассматривать стандартную среду разработки.

Ниже представлен текст простой программы, которая обеспечивает мигание светодиода. Все строчки кода прокомментированы для лучшего понимания и усвоения материала.

Пример первой программы, написанной для Arduino:

```
int led = 13;           //Объявление переменной led, а также указываем для
нее                               порт
void setup()
{
pinMode(led, OUTPUT); //Назначаем указанной переменной ввод или вывод
данных
}
void loop()
{
digitalWrite(led, HIGH); //Подать      логическую      1      (5V)
delay(1000);           //Задержка      в      миллисекундах
digitalWrite(led, LOW); //Подать      логический      0      (0V)
delay(1000);           //Задержка      в      миллисекундах
}
```

Каждая программа, написанная для платы Arduino, называется скетч (sketch). Давайте подробно рассмотрим структуру приведенного выше скетча.

Итак, каждый скетч имеет две функции: setup и loop, в которых и происходит написание программы. Функция setup задает начальные параметры для микроконтроллера и аутентификации портов, с которыми “просыпается” Arduino при подачи питания. В данной программе мы указываем, что переменная led, которой мы присвоили 13-й порт, будет

работать в качестве вывода информации. Тем временем функция loop отвечает за те процессы, которые будет выполнять микроконтроллер на протяжении всей своей работы, пока питание платы не прекратится. Таким образом, всё, что помещено в функцию loop, “обречено” повторяться до бесконечности.

Создание собственного проекта

Мигание светодиода — это, конечно, интересно, но намного интереснее на уроке информатике создать собственный, более серьезный проект, который поможет сделать уроки по программированию более нагляднее. В данной статье мы рассмотрим примеры реальных проектов, в которых реализуем процесс ветвления с помощью оператора IF, простейшие циклы с помощью оператора FOR, потому что при изучении этих тем в разделе программирования на уроке информатики у учеников часто возникают трудности.

Проект 1. Включение светодиода при нажатии на кнопку. В данном проекте используется оператор IF, что способствует наглядному представлению ветвления и логике построения алгоритма. Для реализации данного проекта понадобится: плата Arduino, макетная плата (bread board), светодиод, токоограничивающий резистор, кнопка и соединительные кабели.

Код скетча:

```
int switchPin = 8;
int ledPin = 13;
void setup()
{
    pinMode(switchPin, INPUT);
    pinMode(ledPin, OUTPUT);
```

```

}
void loop()
{
    if (digitalread (switchPin) == HIGH);
    {
        digitalread (ledPin, HIGH);
    }
    else
    {
        digitalread (ledPin, LOW);
    }
}

```

Данный скетч наглядно демонстрирует работу оператора IF. Светодиод будет гореть тогда и только тогда, когда кнопка зажата.

Этот проект в дальнейшем можно доработать и сделать так, чтобы светодиод при нажатии загорался, а при последующем нажатии прекращал гореть. В результате ученики смогут усвоить не только тему ветвления, но и повторить тему с таким типом данных, как boolean.

Проект 2. Использование PWM для равномерного ускорения вращения мотора. В данном проекте используется широко импульсная модуляция, при которой происходит плавное ускорение вращения мотора путем подачи на него равномерно повышающегося (понижающегося) напряжения, диапазон которого находится в пределах от 0 до 5 V. Реализовать данную модель мы сможем с помощью оператора FOR, что также будет способствовать наглядному представлению использования и применения циклов в программировании. Для реализации поставленной задачи нам понадобится:

плата Arduino, макетная плата (bread board), мотор, транзистор, токоограничивающий резистор и соединительные кабели.

Код скетча:

```
int motorPin =9;
void setup ()
{
pinmode (motorPin, output);
}
void loop()
{
    for(int i=0; i<=255; i++);
    {
        analogwreat (motorPin, i);
        delay(10);
    }
delay(500);
    for(int i=255; i>=0; i--);
    {
        analogwreat (motorPin, i);
        delay(10);
    }
delay(500);
}
```

Как можно заметить, в данной программе наглядно представлена реализация цикла с помощью оператора FOR. На мотор подается напряжение, которое изменяется с помощью PWM. В программе мы можем изменять напряжение, варьируя значение параметра от 0 до 255, а также задавать собственные шаги перехода этой переменной. Ученик в ходе выполнения

данной задачи сможет понять взаимосвязь между аналоговым и цифровым сигналом, выяснить, как компьютер распознает градацию сигнала, а также осознать значимость циклов в языке программирования.

Конечно же, это далеко не все проекты, которые можно создать с помощью платы Arduino и рассматривать на уроках информатики в рамках темы программирования. В статье были рассмотрены лишь базовые проекты с использованием начальных знаний в программировании. При дальнейшем использовании Arduino на уроках информатики можно также рассмотреть работу с файлами, вывод информации на дисплейный модуль, беспроводную передачу информации как между компьютером и платой Arduino, так и с мобильными устройствами посредством GSM канала связи, а также создание роботов.

Заключение

Подводя итог, хотелось бы отметить, что нарастающая популярность Arduino не должна остаться незамеченной современными школьными преподавателями информатики. Эта платформа очень удачно вписывается в образовательный процесс обучающихся. Именно в этом мы и старались вас убедить на протяжении всей статьи. Пока у детей горит в глазах огонь интереса и жажды новой информации, учителя должны постараться дать им как можно больше знаний. И желательно, чтобы сегодняшние ученики на уроках информатики учились программированию не только через “сухую” теорию, написанную еще в прошлом веке, но активно используя самые современные технологии. И тогда, возможно, в нашей стране будет хороших инженеров-проектировщиков, а может, и великих программистов.

Список литературы

1. Омельченко Е.Я., Танич В.О., Маклаков А.С., Карякина Е.А.
КРАТКИЙ ОБЗОР И ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ
МИКРОПРОЦЕССОРНОЙ ПЛАТФОРМЫ ARDUINO [Электронный
ресурс] - Режим доступа:
<http://cyberleninka.ru/article/n/kratkiy-obzor-i-perspektivy-primeneniya-mikroprotsessornoy-platformy-arduino>
2. Официальный сайт разработчиков [Электронный ресурс] - Режим
доступа: <https://www.arduino.cc/>
3. Jeremy Blum YouTube канал [Электронный ресурс] - Режим доступа:
<https://www.youtube.com/user/sciguy14>
4. Википедия [Электронный ресурс] - Режим доступа:
<https://ru.wikipedia.org/wiki/Arduino>
5. All arduino (все о ардуино) <https://all-arduino.ru/>