

Муниципальное бюджетное общеобразовательное учреждение
«Гуманитарно-эстетическая гимназия №11
г. Дубны Московской области»
(ГИМНАЗИЯ №11)

Проектная работа
в рамках «Индивидуального проекта»

Тема: «Понимание базовых навыков в Arduino»

Автор работы:
Перельгин Сергей Викторович, 11 класс

Координатор:
Буздавина Елена Львовна, заместитель директора по УВР

2019-2020 гг.

ОГЛАВЛЕНИЕ:

ВВЕДЕНИЕ	3
ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. БАЗОВЫЕ ПОНЯТИЯ	6
1.1. Как все работает. Основные понятия и знания об Arduino	6
1.2. Основные функции в Arduino	7
ГЛАВА 2. ПРОЕКТНАЯ ЧАСТЬ	9
2.1. Начало проекта	9
2.2. Основная часть проекта	12
ЗАКЛЮЧЕНИЕ	14
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	14
ПРИЛОЖЕНИЯ	14

Введение

В последнее время все чаще можно услышать словосочетание «умный дом». Однако не все знают, что это такое. Между тем, концепция умного дома существенно упрощает быт и позволяет дистанционно управлять всей бытовой техникой. Так что же такое «умный дом»?

Домашняя автоматизация, или умный дом (англ. smart house) — система домашних устройств, способных выполнять действия и решать определённые повседневные задачи без участия человека. Таким образом, все электроприборы и бытовая техника в доме объединяются в одну единую экосистему. Технология smart house дает пользователю следующие возможности:

- дистанционно из любой точки мира управлять электрикой и электроникой в своем жилище (например, находясь в отпуске или командировке в жаркую погоду сделать работу холодильника более интенсивной);
- настроить бытовую технику таким образом, чтобы она включалась в строго определенное время (например, зажигать свет и запускать обогреватель к приходу хозяина с работы);
- настроить бытовую технику так, чтобы она включалась при наступлении определенных обстоятельств (например, чтобы при повышении определенного порога температуры начинал работать кондиционер);
- управлять бытовой электроникой с помощью голосовых команд и т.д.

Также умные дома могут включать в себя некоторые элементы безопасности. Например, сигнализация, датчики движения, датчики открывания дверей, датчики горения, видеонаблюдение и т.д.

Но как же все это работает? В этом помогает Arduino (Ардуино). Arduino — это электронный конструктор, который позволяет любому человеку создавать разнообразные электро-механические устройства. Ардуино состоит из программной и аппаратной части. Программная часть включает в себя среду разработки (программа для написания и отладки прошивок), множество готовых и удобных библиотек, упрощенный язык программирования. Аппаратная часть включает в себя большую линейку микроконтроллеров и готовых модулей для них.

Arduino может использоваться как для создания автономных объектов автоматике, так и подключаться к программному обеспечению на компьютере через стандартные проводные и беспроводные интерфейсы. Для программирования Arduino используется упрощенная версия языка C++ с предопределенными функциями.

Актуальность

Ардуино можно использовать не только в системе «умный дом», но и как централизованное управление такими механизмами автомобиля как :

1. Система запуска и остановки двигателя.
2. Система управления стеклоподъемниками.
3. Система закрывания и открывания центрального замка дверей.

Стоимость таких систем не по карману многим потребителям, да и потребность во всех функциях системы возникает не у всех. Одной из наиболее интересных функций данной системы является «Система запуска и остановки двигателя», позволяющий заводить и глушить двигатель автомобиля используя только одну кнопку.

Цель

Ознакомиться с Arduino и попробовать написать первую программу в этой среде.

Задачи

1. Изучить новый язык программирования.
2. Изучить программную среду Arduino.
3. Написать программу управления электронной платы Arduino Uno.
4. Осуществить визуализацию работы программы при помощи подключения светодиода на дополнительной монтажной плате.

ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ. БАЗОВЫЕ ПОНЯТИЯ

1.1. Как все работает. Основные понятия и знания об Arduino.

Изображение 1:



```

sketch_may04a | Arduino 1.8.9
Файл Правка Скетч Инструменты Помощь
sketch_may04a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
  
```

Void loop()

Void ..

Программную оболочку Arduino, которая предназначена для написания программ, их компиляции и программирования аппаратуры, можно бесплатно скачать с официального сайта торговой марки. При запуске программа будет выглядеть таким образом

Автоматически каждый раз при открытии нового файла можно будет увидеть два основных скетча, без которых ничего работать не будет.

Это void () setup и void loop ().

С помощью этих функций мы передадим микроконтроллеру инструкции нашего скетча.

Функция *void setup()* описывается, то, что будет происходить в программе,

а функция *void loop()* будет прокручивается бесконечное количество раз в цикле, пока на Arduino-контроллер будет подаваться питание.

Изображение 2:

```

1 // Однострочный комментарий
2 /*
3 Многострочный комментарий
4 */
5
6 int ledPin = 13;
7
8 // Команды в функции setup выполняются только 1 раз при включении или
9 void setup() {
10   pinMode(ledPin, OUTPUT);
11 }
12
13 // Инструкции в этой функции будут выполняться по кругу постоянно
14 void loop() {
15   digitalWrite(ledPin, HIGH);
16   delay(1000);
17   digitalWrite(ledPin, LOW);
18   delay(1000);
19 }

```

Задаем
переменную с

Комментарии

ledPin как выход
(OUTPUT)

Задержка на

Цифровой выход

1.2. Основные функции в Arduino

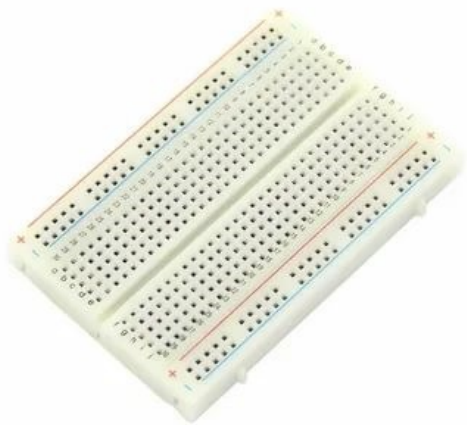
- Символы «//» означают однострочный комментарий. Это значит что строка после двух слэшей выполняться не будет. Комментарии можно использовать для описания действий программы для наглядности. Или для скрытия части кода при отладке. Для скрытия кода чаще используется многострочный комментарий начинающийся с комбинации «/*» и заканчивающийся «*/».
- Строка «int ledPin = 13;» означает, что мы объявляем переменную с типом данных int, именем ledPin и значением 13.
- **#define** это удобная директива, который позволяет дать имя константе перед тем как программа будет скомпилирована.
- **#include** используется для включения сторонних библиотек в ваш скетч. Это дает доступ к большому числу стандартных библиотек C (библиотекой называют группы предварительно написанных функций), и библиотек написанных специально для Arduino.
- Константы, характеризующие уровень напряжения на выводах, HIGH и LOW: При работе с цифровыми выводами существует всего два значения, которые они могут выводить или считывать: **HIGH**(включить) и **LOW**(выключить).

- **INPUT** (вход) и **OUTPUT** (выход).
- **pinMode ()**:
pin: номер вывода, режим работы которого будет конфигурироваться.
mode: принимает значения INPUT, OUTPUT или INPUT_PULLUP.
- **digitalWrite()** - Отправляет на цифровой вывод значение HIGH или LOW.
- **digitalRead()** - Считывает уровень сигнала HIGH или LOW с указанного цифрового вывода. Если вывод ни к чему не присоединен, функция digitalRead() может вернуть случайную величину, как HIGH, так и LOW. Выводы, являющиеся аналоговыми входами, могут также использоваться как цифровые выводы под именем A0, A1 и т.д

ГЛАВА 2. ПРОЕКТНАЯ ЧАСТЬ

2.1. Начало проекта

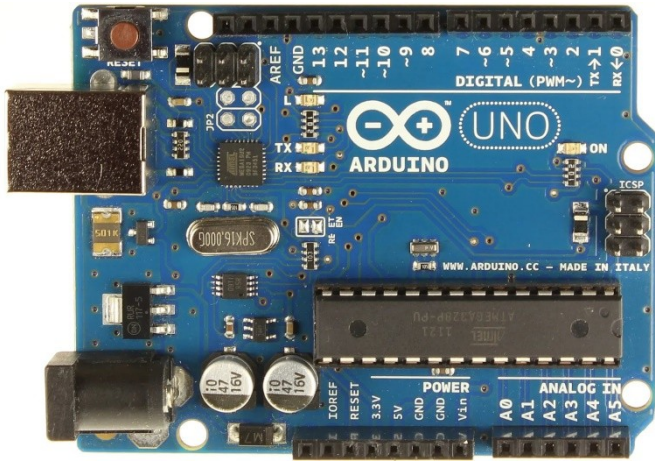
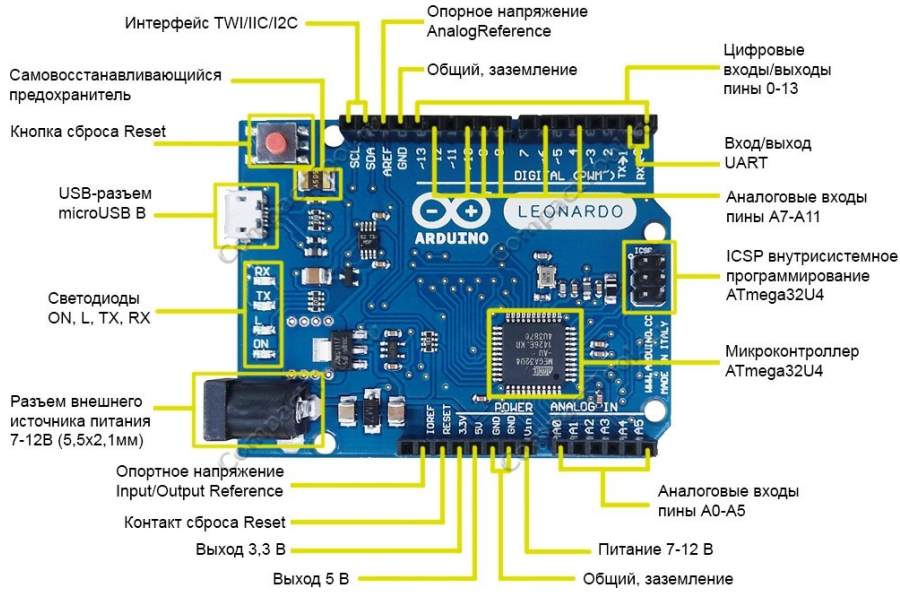
Для реализации своего проекта мне будет нужна плата Arduino Uno, некоторые провода, RGB светодиод и Breadboard (см. изображения ниже).



Breadboard(изобр



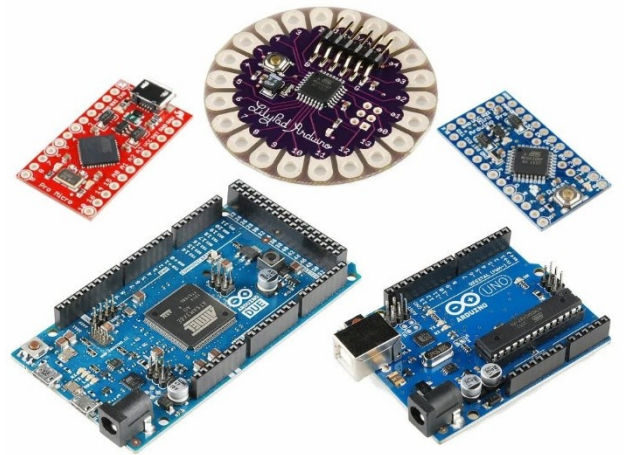
RGB светодиод(изобр.4)



Изображение



Существует огромное количество различных плат, но все они имеют схожее строение. Новичкам в этой сфере рекомендуют использовать плату Arduino Uno.



Arduino UNO

ем с

Изобр.7

простой программы, которая просто проверяет работоспособность устройства. В этой программе нам придется заставить “моргать” уже встроенный в плату светодиод. У всех Arduino есть светодиод, подключенный к контакту 13 (пин №13).

Изобр.8

- Пункт 1
- Пункт 2
- Пункт 3
- Пункт 4
- Пункт 5

```

sketch_may04a
// Горит-выключается
int ledPin = 13;
void setup() {
  pinMode(ledPin, OUTPUT);
}
void loop() {
  digitalWrite(ledPin, HIGH);
  delay(2000);
  digitalWrite(ledPin, LOW);
  delay(2000);
}

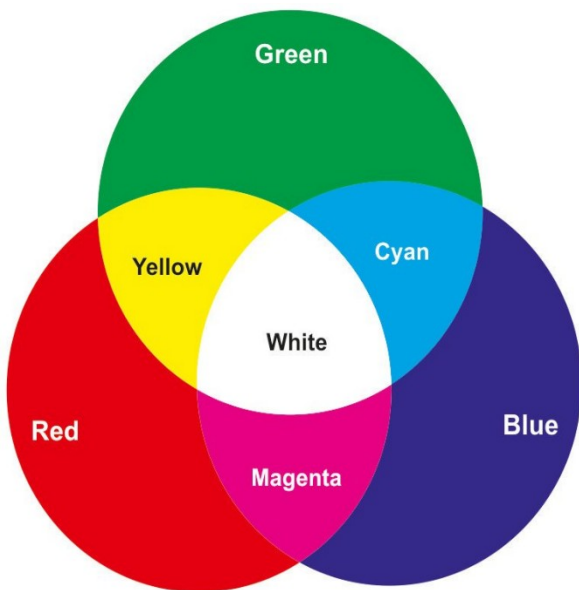
```

1. Мы присвоили 13 контакту значение ledPin.
2. Записываем две основные функции, которые обязаны присутствовать в любой программе Arduino: void setup () и void loop () между скобками.
3. Мы проинициализировали все контакты в качестве выходов. Для включения или выключения светодиода мы должны убедиться, что контакт должен быть установлен как выход (OUTPUT). Среди всех ардуино это можно сделать с помощью команды pinMode, которая принимает два аргумента: контакт, который мы хотим настроить (ledPin) и хотим ли мы, чтобы он был выходом или входом. Мы хотим, чтобы он был выходом (OUTPUT).
4. В цикле мы используем команду digitalWrite (цифровой выход), которая также принимает два аргумента: контакт, на который мы хотим повлиять, то есть ledPin, и напряжение на выходах, либо HIGH(включить), либо LOW(выключить).
5. Также, чтобы не происходило моментального перехода от HIGH к LOW, нужно использовать задержку (delay). Выражается в миллисекундах (в данной задаче задержка на 2с).
6. В плату можно вставить светодиод для визуального эффекта.

2.2. Основная часть проекта

Нам предстоит написать такую программу, чтобы RGB светодиод переливался различными цветами. Пусть всего будет семь цветов (как на изображении 9).

Изобр.9



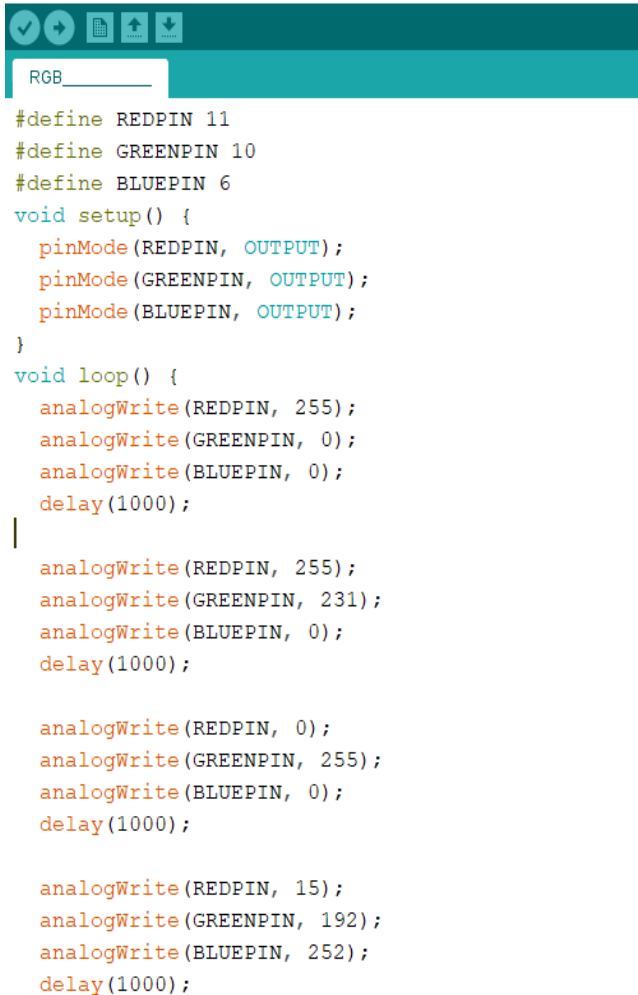
RGB - Red, Green, Blue
© Gil Dekel PoeticMind.co.uk

Изобр.10



С помощью палитры цветов можно определить любой номер нужного нам цвета. Эти номера будут нужны, чтобы передать коэффициент заполнения (255).

Изобр.11



```

#define REDPIN 11
#define GREENPIN 10
#define BLUEPIN 6
void setup() {
  pinMode(REDPIN, OUTPUT);
  pinMode(GREENPIN, OUTPUT);
  pinMode(BLUEPIN, OUTPUT);
}
void loop() {
  analogWrite(REDPIN, 255);
  analogWrite(GREENPIN, 0);
  analogWrite(BLUEPIN, 0);
  delay(1000);

  analogWrite(REDPIN, 255);
  analogWrite(GREENPIN, 231);
  analogWrite(BLUEPIN, 0);
  delay(1000);

  analogWrite(REDPIN, 0);
  analogWrite(GREENPIN, 255);
  analogWrite(BLUEPIN, 0);
  delay(1000);

  analogWrite(REDPIN, 15);
  analogWrite(GREENPIN, 192);
  analogWrite(BLUEPIN, 252);
  delay(1000);
}

```

1. Запишем глобальные переменные.
2. В void setup () проинициализируем все контакты как выходы (OUTPUT).
3. В функции analogWrite передаем вывод, на котором будет формироваться напряжение и коэффициент заполнения из палитры цветов.
4. Ставим задержку.
5. Повторяем первые четыре пункта столько

раз, сколько хотим, чтобы было цветов на светодиоде.

Заключение

1. Я ознакомился с программным продуктом Arduino и написал программу управления микроконтроллера электронной платы.
2. Я добавил светодиод в схему электронной платы для визуализации работы моей программы.
3. Мой отчет о проделанной работе может служить инструкцией для начинающего пользователя в системе Arduino.
4. Данная работа может рассматриваться как первый шаг в освоении программирования более сложных электронных систем.

Список используемой литературы

<https://all-arduino.ru/>

<https://www.arduino.cc/index.php>

<https://arduinomaster.ru/>

<https://habr.com/ru/post/357908/>

<https://ru.wikipedia.org/wiki/Arduino>

Приложения

Ссылка на видеоролики на яндекс диске: <https://yadi.sk/d/Gwp1iji8E8I1NA>