

Знакомство с IoT

Первый скетч: мигаем светодиодом

Антон Парамонов
Инженер-программист АСУ ТП



Проверка связи





Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти



Поставьте в чат:

-  если меня видно и слышно
-  если нет

Антон Парамонов

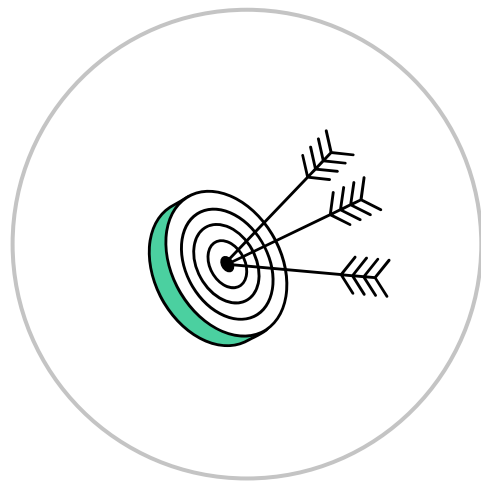
О спикере:

- Инженер-программист АСУ ТП
- С 2019 года работаю в проектах по модернизации производственных мощностей
- Занимался обслуживанием и модернизацией установок полного цикла изготовления алюминиевого каркаса двигателей Volkswagen MPI
- Имею опыт автоматизации на оборудовании Siemens, Omron, Allen-Bradley



Цели занятия

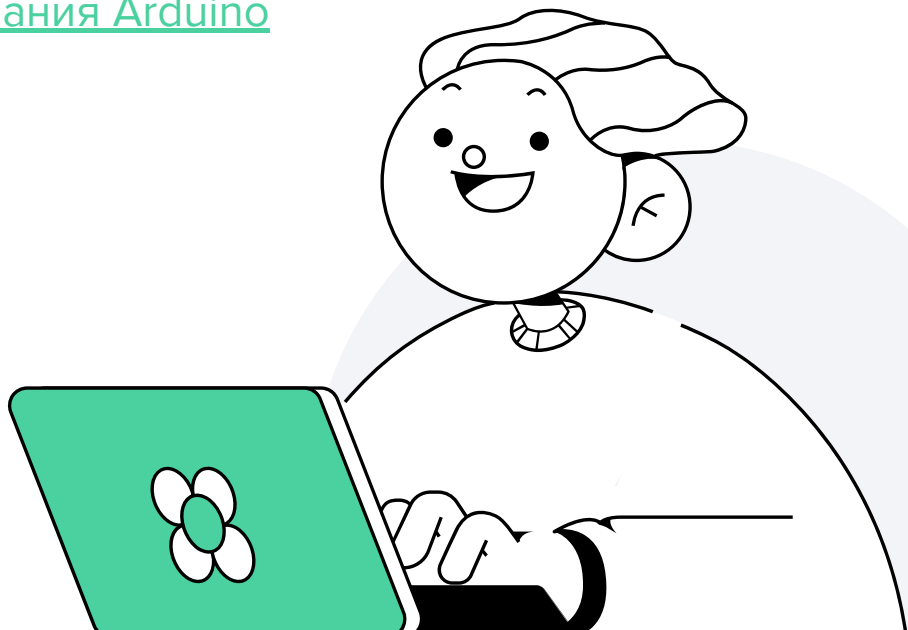
- Узнаем, что представляет собой интернет вещей (IoT)
- Познакомимся с платформой Arduino
- Узнаем, как работать в среде разработки Arduino IDE
- Познакомимся с программным симулятором WOKWI
- Подключим светодиод к плате и научимся им мигать



План занятия

- 1 [Что такое интернет вещей и зачем он нужен](#)
- 2 [Платформа Arduino](#)
- 3 [Среда разработки для программирования Arduino](#)
- 4 [Если платы нет под рукой](#)
- 5 [Подключаем внешний светодиод](#)
- 6 [Итоги](#)

*Нажми на нужный раздел для перехода

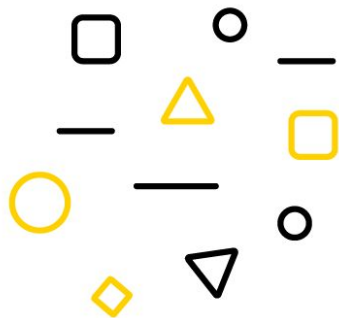


Что такое интернет вещей и зачем он нужен

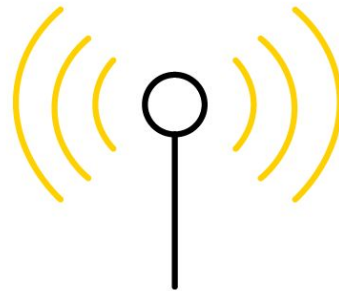


1

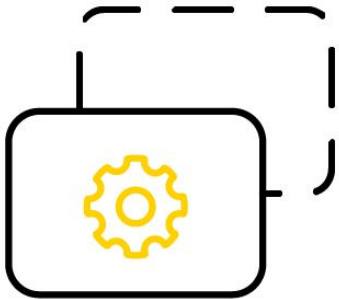
Из чего состоит интернет вещей



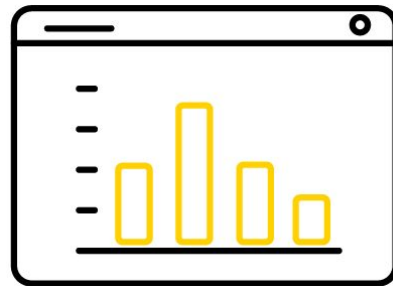
Вещи



Связь



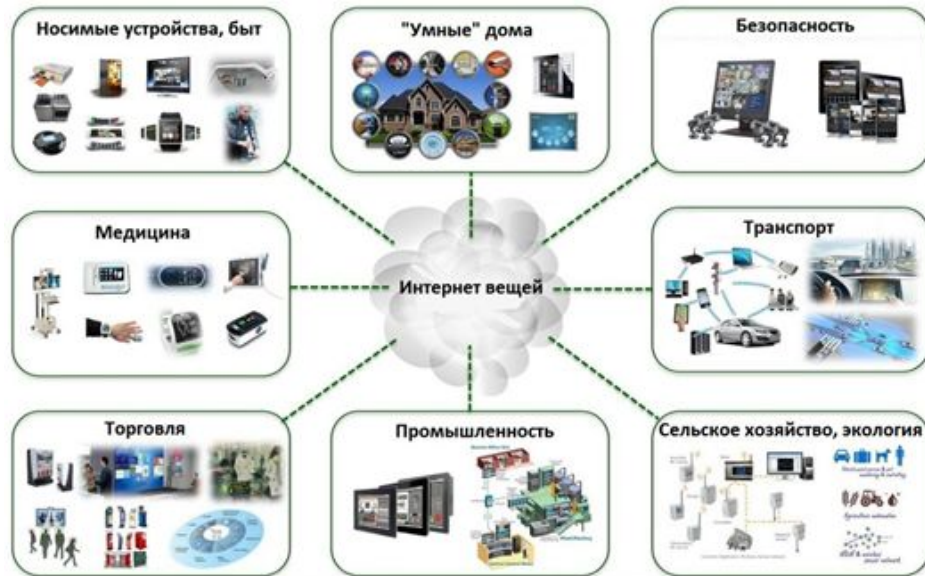
**Программное
обеспечение**



Приложение

Где применяется интернет вещей

По данным исследования IoT Analytics, в 2020 году самый высокий уровень проникновения технологии IoT наблюдался в транспорте, энергетике, ритейле, управлении жизнью города, здравоохранении и промышленности.



Плюсы интернета вещей

- Эффективность
- Автоматизация
- Снижение издержек
- Контроль качества
- Прозрачность

Минусы интернета вещей

- Совместимость
- Снижение количества рабочих мест
- Сложность
- Конфиденциальность и безопасность

Платформа Arduino

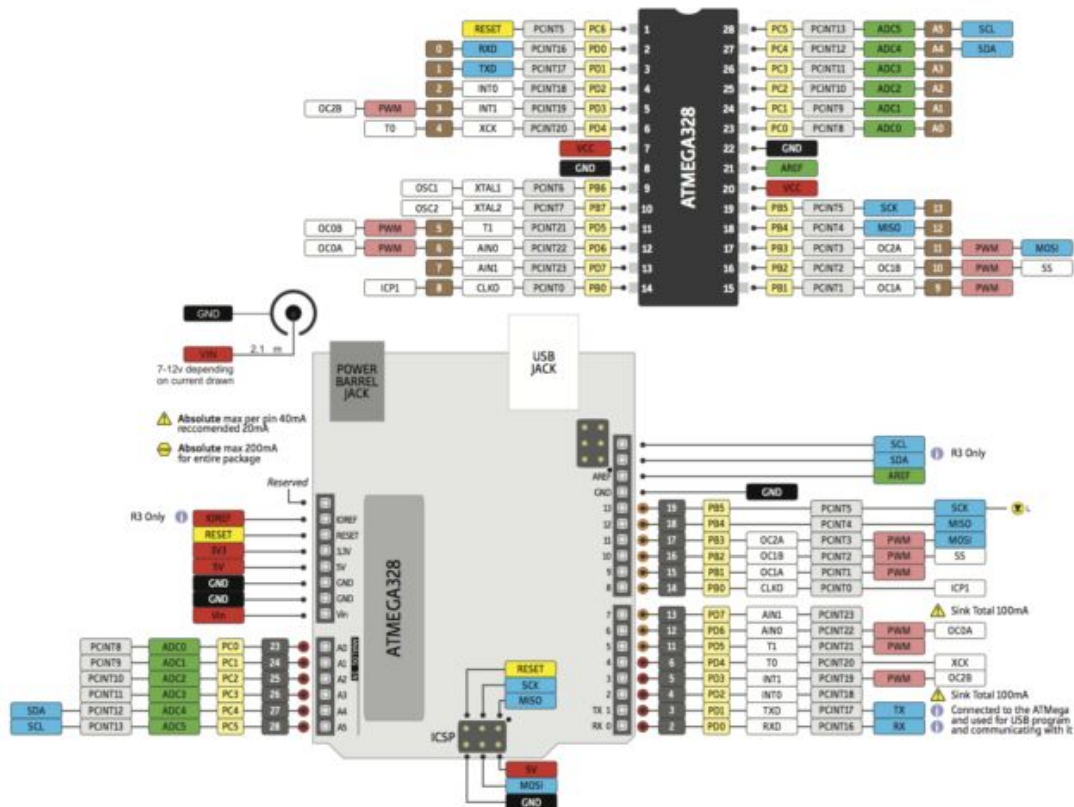


2

Преимущества Arduino

- 1 Низкая стоимость по сравнению с другими платформами
- 2 Кросс-платформенность
- 3 Большое количество плат расширения
- 4 Простая и понятная среда программирования
- 5 Большое число примеров

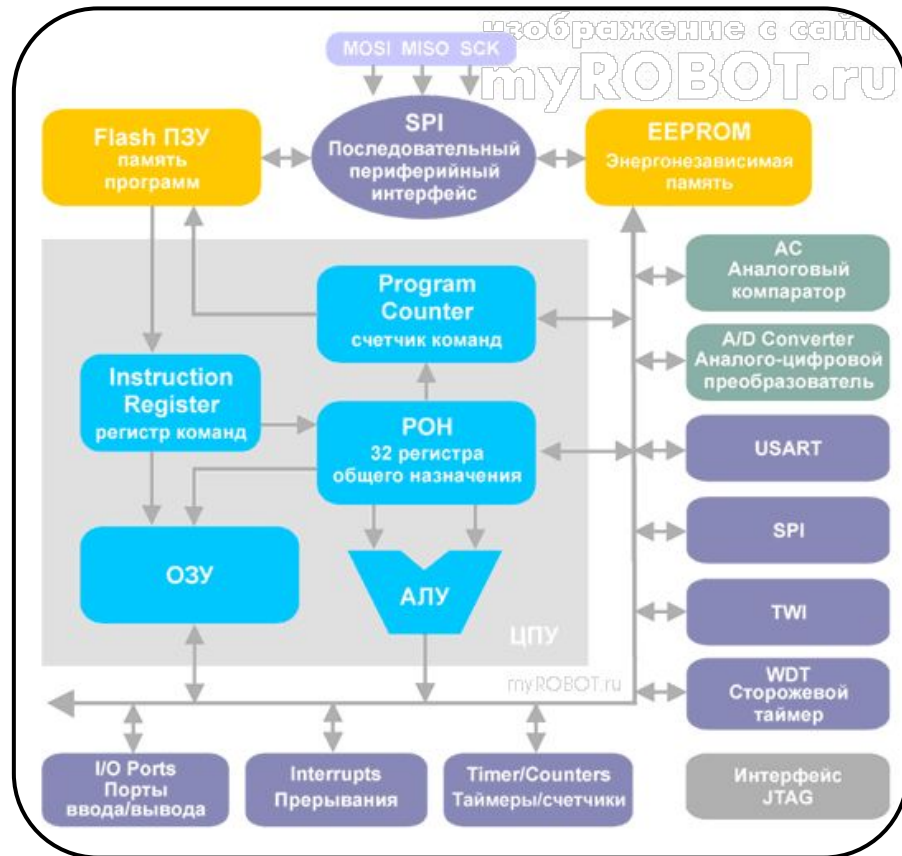
ИСТОЧНИК



Структура микроконтроллера AVR

Микроконтроллер AVR содержит:

- RISC-процессор
- два типа энергонезависимой памяти (Flash-память программ и память данных EEPROM)
- оперативную память RAM
- порты ввода/вывода
- различные периферийные модули



Типы плат Arduino

Все платы Arduino можно разделить на три категории: платы для начинающих (Basic boards), продвинутые платы (Enhanced boards) и платы для тематики интернета вещей (IoT boards)



Платы расширения (shields) для Arduino

Существует большое количество плат, позволяющих расширить функциональные возможности платформы Arduino: кнопки, индикаторы, датчики и т.п.



Среда разработки для программирования Arduino

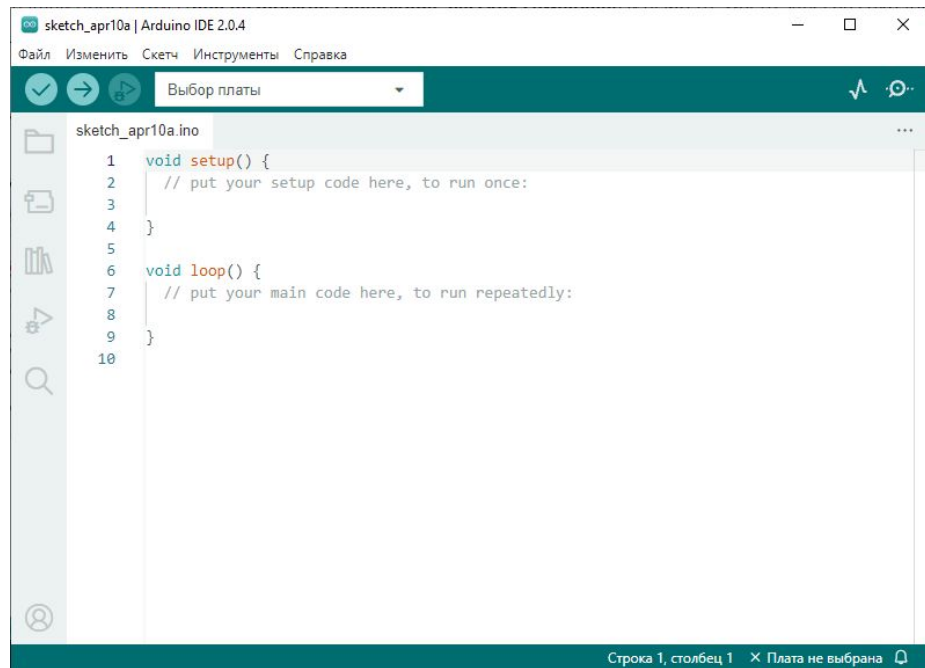


3

Среда разработки для Arduino

Основной средой разработки является Arduino IDE.

Она свободна для скачивания по ссылке: <https://www.arduino.cc/en/software>



Последовательность создания скетча

- 1 Выполнить пункт меню **Файл -> Новый**
- 2 После написания кода выполнить пункт меню **Файл -> Сохранить**
- 3 При необходимости подключить библиотеку выполнить пункт меню **Скетч -> Подключить библиотеку**
- 4 Выбрать плату в пункте меню **Инструменты -> Плата**
- 5 Выбрать тип микроконтроллера в пункте меню **Инструменты -> Процессор**
- 6 Скомпилировать скетч с помощью пункта меню **Скетч -> Проверить/Компилировать**
- 7 Выбрать порт для загрузки программы в пункте меню **Инструменты -> Порт**
- 8 Загрузить программу с помощью пункта меню **Скетч -> Загрузка**

Какой язык программирования в Arduino IDE

Язык программирования Arduino основан на Wiring

Wiring - открытый фреймворк для программирования микроконтроллеров

В действительности нет особого языка программирования Arduino, фактически программы пишутся на C/C++, а компилируются и собираются с помощью компилятора avr-gcc.

Скетч должен содержать две обязательные функции:

- `void setup ()`, которая вызывается только один раз при запуске
- `void loop()`, которая выполняется в бесконечном цикле

Какой язык программирования в Arduino IDE

Пример кода:

```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, 1);
  delay(500);
  digitalWrite(13, 0);
  delay(500);
}
```

Преобразованный код:

```
#include "WProgram.h" // определения всех Arduino ф-ий, констант
void setup();          // прототипы ф-ии setup() и loop()
void loop();
void setup()            // исходный код
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, 1);
  delay(500);
  digitalWrite(13, 0);
  delay(500);
} // конец исходного кода

int main(void)          // стандартная ф-ия main()
{
  init();                // внутренняя инициализация
  setup();                // пользовательский setup()
  for (;;)                // в бесконечном цикле вызывается loop()
    loop();
  return 0;              // а сюда вообще никогда не попадаем
}
```

Монитор последовательного порта

Между Arduino и компьютером можно обмениваться данными через последовательный порт (он же интерфейс UART)

Монитор последовательного порта может использоваться как для вывода отладочной информации от модуля Arduino, так и для других целей. Через него можно как отправлять данные в модуль Arduino, так и получать данные от него. Не забудьте выбрать порт, к которому подключен модуль Arduino, иначе монитор последовательного порта не откроется! При его открытии модуль Arduino перезагрузится!

Открыть окно монитора последовательного порта можно из пункта меню **Инструменты -> Монитор порта**

Базовые функции цифрового ввода/вывода

`void pinMode(uint8_t pin, uint8_t mode)` — устанавливает режим работы заданного вход/выхода(pin) как входа или как выхода

Параметры:

- pin: номер вход/выхода(pin), который Вы хотите установить
- mode: режим одно из двух значение - INPUT или OUTPUT, устанавливает на вход или выход соответственно

Возвращаемое значение: нет

Базовые функции цифрового ввода/вывода

`void digitalWrite(uint8_t pin, uint8_t value)` — подает HIGH (близкое к напряжению питания) или LOW (близкое к нулю) значение на цифровой вход/выход (pin). Если вход/выход (pin) был установлен в режим вход (INPUT), то функция digitalWrite со значением HIGH будет активировать внутренний подтягивающий резистор номиналом около 20 кОм

Параметры:

- pin: номер вход/выхода(pin)
- value: значение HIGH или LOW

Возвращаемое значение: нет

Базовые функции цифрового ввода/вывода

`int digitalRead(uint8_t pin)` — возвращает значение с заданного входа - HIGH или LOW

Параметры:

- pin: номер вход/выхода(pin), который Вы хотите считать

Возвращаемое значение: HIGH или LOW

Базовые функции работы со временем

`unsigned long millis(void)` — возвращает количество миллисекунд с момента начала выполнения текущей программы на плате Arduino. Это количество сбрасывается на ноль из-за переполнения приблизительно через 50 дней.

Параметры: нет

Возвращаемое значение:

- количество миллисекунд с момента начала выполнения программы

Базовые функции работы со временем

`unsigned long micros(void)` — возвращает количество микросекунд с момента начала выполнения текущей программы на плате Arduino. Значение переполняется и сбрасывается в ноль приблизительно через 70 минут

Параметры: нет

Возвращаемое значение:

- количество микросекунд с момента начала выполнения программы

Базовые функции работы со временем

`void delay(unsigned long ms)` — останавливает выполнение программы на заданное в параметре количество миллисекунд

Параметры:

- `ms`: количество миллисекунд, на которое приостанавливается выполнение программы

Возвращаемое значение: нет

Базовые функции работы со временем

`void delayMicroseconds(unsigned int us)` — останавливает выполнение программы на заданное в параметре количество микросекунд. Максимальная пауза воспроизводимая корректно — 16383

Параметры:

- us: количество микросекунд, на которое приостанавливается выполнение программы

Возвращаемое значение: нет

Практическое задание №1

Знакомство с Arduino IDE

Время выполнения:
10 минут



Задание:

1 Скачайте Arduino IDE по ссылке: <https://www.arduino.cc/en/software> и установите его

2 Создайте скетч с текстом:

```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

3 Выполните компиляцию скетча

4 Напишите в чат об удачном завершении компиляции

Если платы
нет под рукой

4



Симулятор - это программное обеспечение, которое позволяет производить отладку ПО для микроконтроллера (или микропроцессора) без наличия самой аппаратной платформы

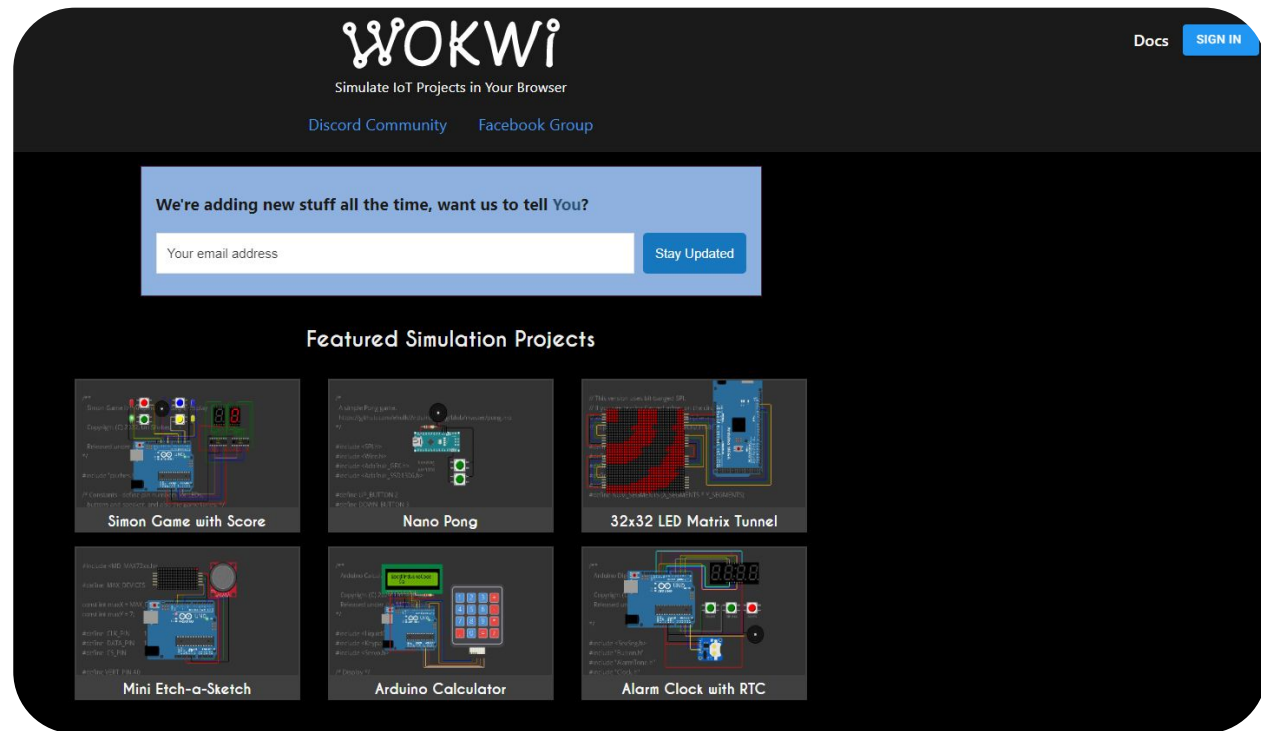
Типы симуляторов:

- Онлайн
- Не в сети
- Программы моделирования электронных схем

Симулятор Wokwi

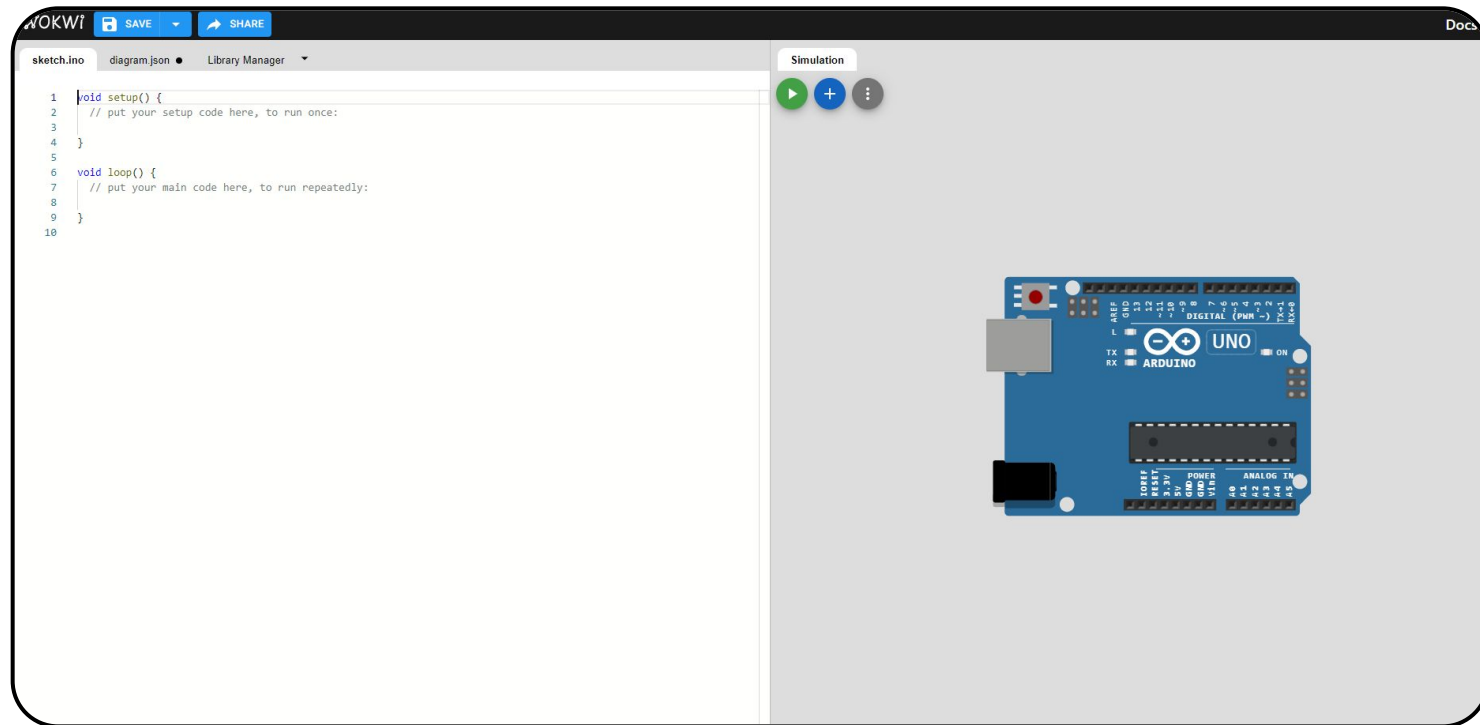
Wokwi - открытый проект с сообществом в Discord и Facebook, ссылка:

<https://wokwi.com/>



Интерфейс симулятора Wokwi

Слева - текстовый редактор, справа - симулятор



Практическое задание №2

Знакомство с симулятором Wokwi

Время выполнения:
5 минут



Задание:

1 Авторизуйтесь на сайте Wokwi по ссылке: <https://wokwi.com/> и создайте новый проект с платой Arduino Uno

2 Введите текст:

```
void setup()
{
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

3 Запустите симуляцию с помощью кнопки **Start the simulation**

4 Напишите в чат о результатах симуляции

Подключаем внешний светодиод



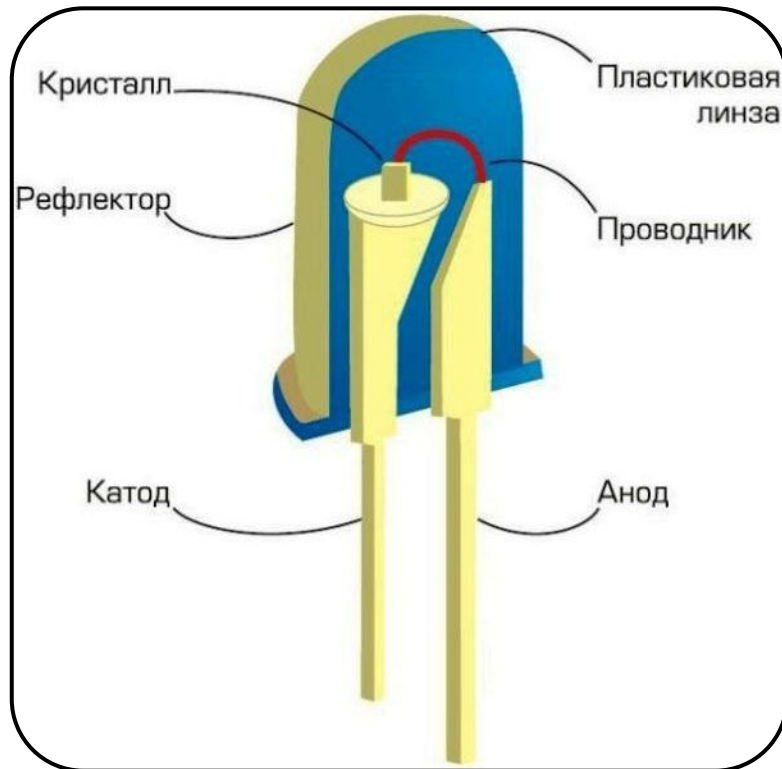
5



Светодиод — это полупроводниковый элемент, который, при прохождении через него электрического тока излучает свет

Светодиод пропускает ток только в одном направлении от анода к катоду

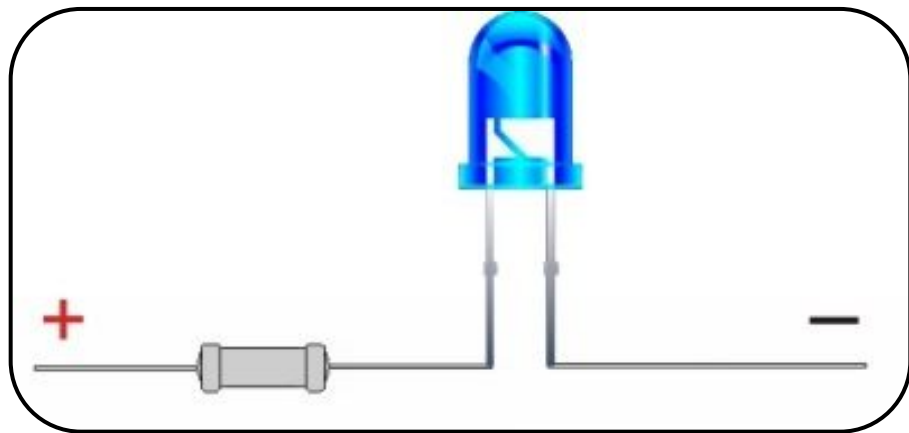
Определение светодиода



Подключение светодиода

Правила подключения:

- у светодиода есть положительный и отрицательный контакты, поэтому важно соблюдать полярность при подключении.
- у светодиодов есть ограничения по протекающему через них току, поэтому необходимо обеспечивать правильный режим электропитания



Расчет токозадающего сопротивления

Чтобы вычислить необходимый номинал резистора, нам необходимо знать характеристики источника питания и светодиода

Основные параметры:

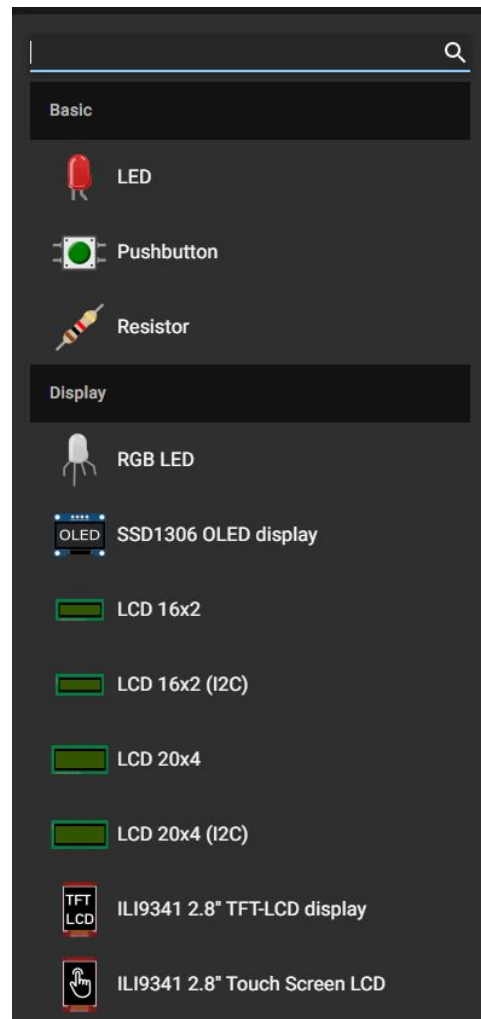
- U_{ps} — напряжение источника питания (обычно 5 В)
- U_{df} — падение напряжения на светодиоде (обычно 2 В)
- I_f — номинальный ток светодиода (обычно 10 мА)

По закону Ома:

$$R = (U_{ps} - U_{df}) / I_f = (5\text{В} - 2\text{В}) / 0.01\text{А} = 300 \text{ Ом}$$

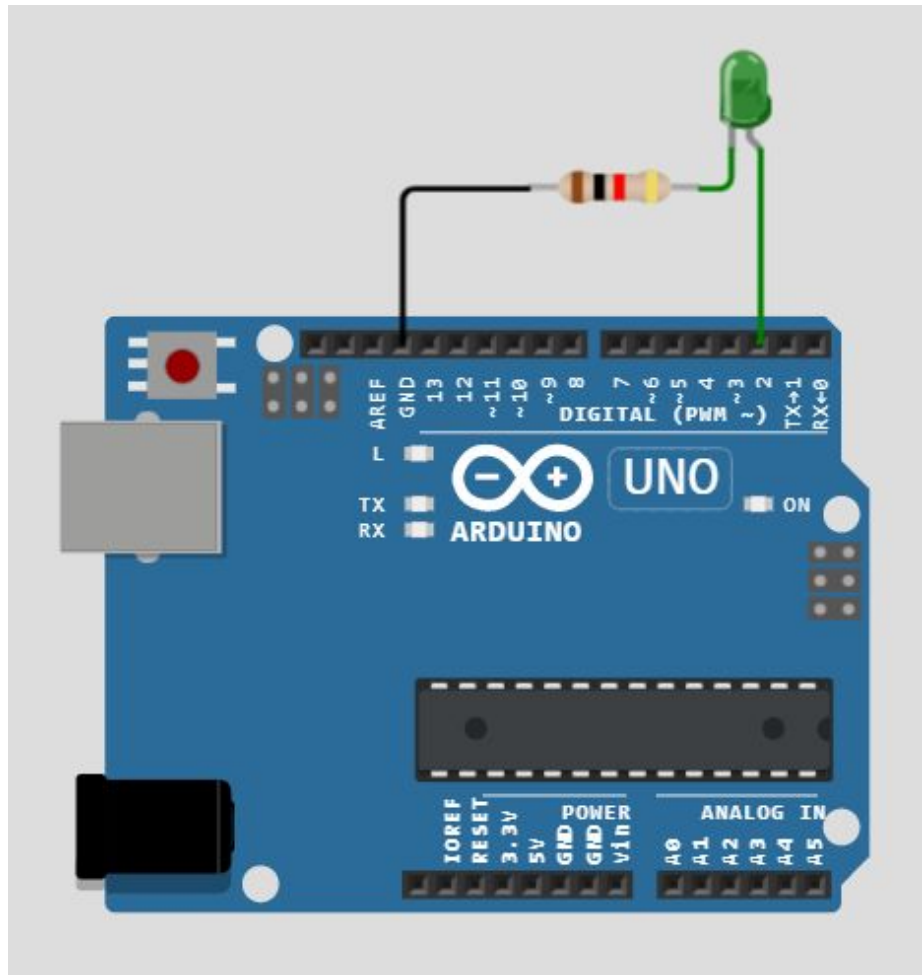
Добавление элементов в симуляторе

Для добавления какой-либо элемента необходимо нажать на кнопку "+" и откроется выпадающий список элементов



Соединение элементов в симуляторе

Для соединения пинов и контактов элементов необходимо нажать на один контакт, а затем на другой



Изменение параметров элементов

Для изменения некоторых настроек элементов в текстовом редакторе симулятора можно открыть файл `diagram.json`

Например, можно поменять номинал резистора (“value”) на 300

```
sketch.ino • diagram.json • Library Manager ▼

1 {
2   "version": 1,
3   "author": "Виталий Глазков",
4   "editor": "wokwi",
5   "parts": [
6     { "type": "wokwi-arduino-uno", "id": "uno", "top": 0, "left": 0, "attrs": {} },
7     {
8       "type": "wokwi-led",
9       "id": "led1",
10      "top": -98.4,
11      "left": 212.33,
12      "attrs": { "color": "green" }
13    },
14    {
15      "type": "wokwi-resistor",
16      "id": "r1",
17      "top": -48.85,
18      "left": 158.14,
19      "attrs": { "value": "1000" }
20    }
21  ],
22  "connections": [
23    [ "led1:C", "r1:2", "green", [ "v0" ] ],
24    [ "uno:GND.1", "r1:1", "black", [ "v0" ] ],
25    [ "led1:A", "uno:2", "green", [ "v0" ] ]
26  ],
27  "dependencies": {}
28 }
```

Практическое задание №3

Мигаем светодиодом в симуляторе

Время выполнения:
10 минут



Задание:

- 1 Создайте новый проект с платой Arduino Uno, подключите к выводу 4 зеленый светодиод, а к выводу 5 - красный
- 2 Напишите программу, которая будет мигать светодиодом зеленым светодиодом раз в секунду, а красным - раз в полсекунды.
- 3 Запустите симуляцию с помощью кнопки **Start the simulation**
- 4 Напишите в чат о результатах симуляции

Итоги



6

Итоги занятия

Сегодня мы

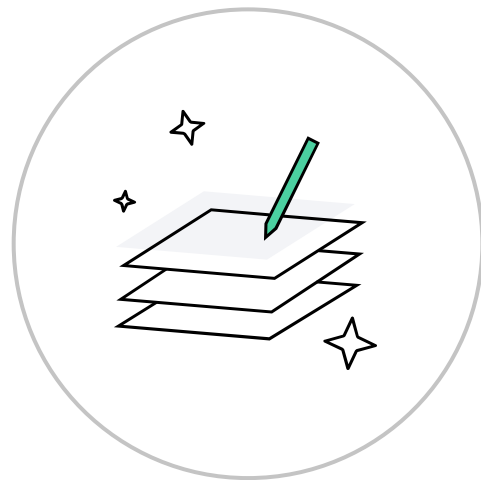
- 1 Познакомились с интернетом вещей и узнали, из каких элементов он состоит
- 2 Познакомились с платформой Arduino
- 3 Научились создавать проект в пакете Arduino IDE
- 4 Научились симулировать работу программы в симуляторе Wokwi
- 5 Рассмотрели, как подключать внешний светодиод к плате Arduino UNO



Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Дополнительные материалы

- <https://habr.com/ru/articles/422877/>
- <https://habr.com/ru/companies/yota/articles/333850/>
- Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: пер. с англ. СПб.: БХВ-Петербург, 2015. 336 с.
- Саймон Монк: Програмируем Arduino. Основы работы со скетчами. 2-е издание



**Задавайте вопросы
и пишите отзыв о лекции**

