

Лабораторная работа №1.

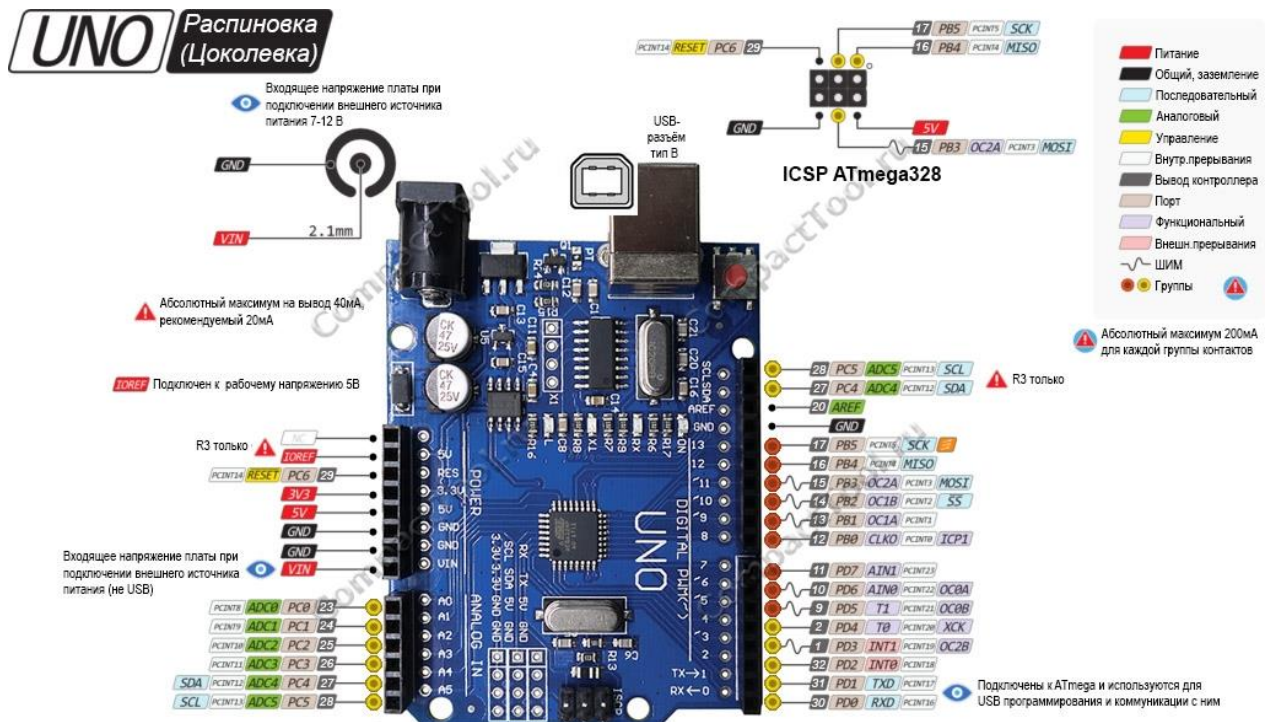
Тема: "Знакомство с Arduino и Proteus". Цифровые выходы Arduino.

Цель работы:

Освоить базовые навыки работы с Arduino IDE и Proteus: создание простой схемы с Arduino, написание и симуляция кода для управления светодиодом.

Теоретическая часть

1. **Arduino** — это платформа для разработки электронных устройств на базе микроконтроллеров. Программы (скетчи) пишутся на языке C++ в Arduino IDE.
2. **Proteus** — программа для моделирования электронных схем. Позволяет тестировать код Arduino без физического подключения устройств.
3. **Цифровые пины Arduino** могут работать как выходы (OUTPUT) для управления устройствами (например, светодиодом) или как входы (INPUT) для чтения сигналов.
4. **Функции Arduino:**
 - `pinMode(pin, mode)` — настройка режима работы пина (INPUT/OUTPUT).
 - `digitalWrite(pin, value)` — установка высокого (HIGH) или низкого (LOW) уровня на пине.
 - `delay(ms)` — задержка в миллисекундах.



Задание:

1. Создать схему в Proteus с Arduino Uno и светодиодом.
Написать программу для мигания светодиода с интервалом 1 секунда.
Провести симуляцию и проанализировать результат.
2. Соберите схему с тремя светодиодами, подключенными к разным цифровым пинам.
Напишите программу, которая будет поочередно включать и выключать светодиоды с задержкой в 500 мс.
3. Подключите 5–6 светодиодов к разным цифровым пинам (например, D2–D7). Напишите программу, которая создаёт эффект "бегущих огней":
Светодиоды загораются по очереди с интервалом 0.2 секунды.
После достижения последнего светодиода цикл начинается заново.
Дополнительное задание: Сделайте эффект двунаправленным (светодиоды загораются то слева направо, то справа налево).

Оборудование (в Proteus)

- Микроконтроллер: **Arduino Uno**.
- Светодиод: **LED** (например, LED-RED).
- Резистор: **220 Ом** (для ограничения тока).

Пошаговая инструкция

1. Создание схемы в Proteus

1. Запустите Proteus и создайте новый проект (*New Project*).
2. Добавьте компоненты:
 - **Arduino Uno**: Ищите в библиотеке по названию.
 - **Светодиод (LED)**: Компонент "LED-RED" (категория *Optoelectronics*).
 - **Резистор 220 Ом**: Компонент "RES" (категория *Resistors*).
3. Соберите схему:
 - Подключите **анод** светодиода к цифровому пину **13** Arduino.
 - Подключите **катод** светодиода (короткая ножка) к резистору 220 Ом, а резистор — к **GND** Arduino.

2. Написание кода в Arduino IDE

1. Запустите Arduino IDE и создайте новый скетч.

2. Напишите код:

```
void setup() {  
  pinMode(13, OUTPUT); // Настройка пина 13 как выхода  
}  
  
void loop() {  
  digitalWrite(13, HIGH); // Включить светодиод  
  delay(1000);           // Пауза 1 секунда  
  digitalWrite(13, LOW); // Выключить светодиод  
  delay(1000);           // Пауза 1 секунда  
}
```

Пояснение к коду.

В Arduino функции `setup()` и `loop()` являются обязательными и выполняют ключевые роли в работе программы (скетча). Вот их назначение:

1. `setup()`

Когда выполняется: Один раз при старте Arduino (после подачи питания или сброса).

Для чего используется:

- Инициализация настроек: установка режимов работы пинов (INPUT, OUTPUT).
- Подключение к внешним устройствам (например, Serial-порт, дисплеи, датчики).
- Запуск библиотек или начальных вычислений.

2. `loop()`

Когда выполняется: Бесконечно повторяется после завершения `setup()`.

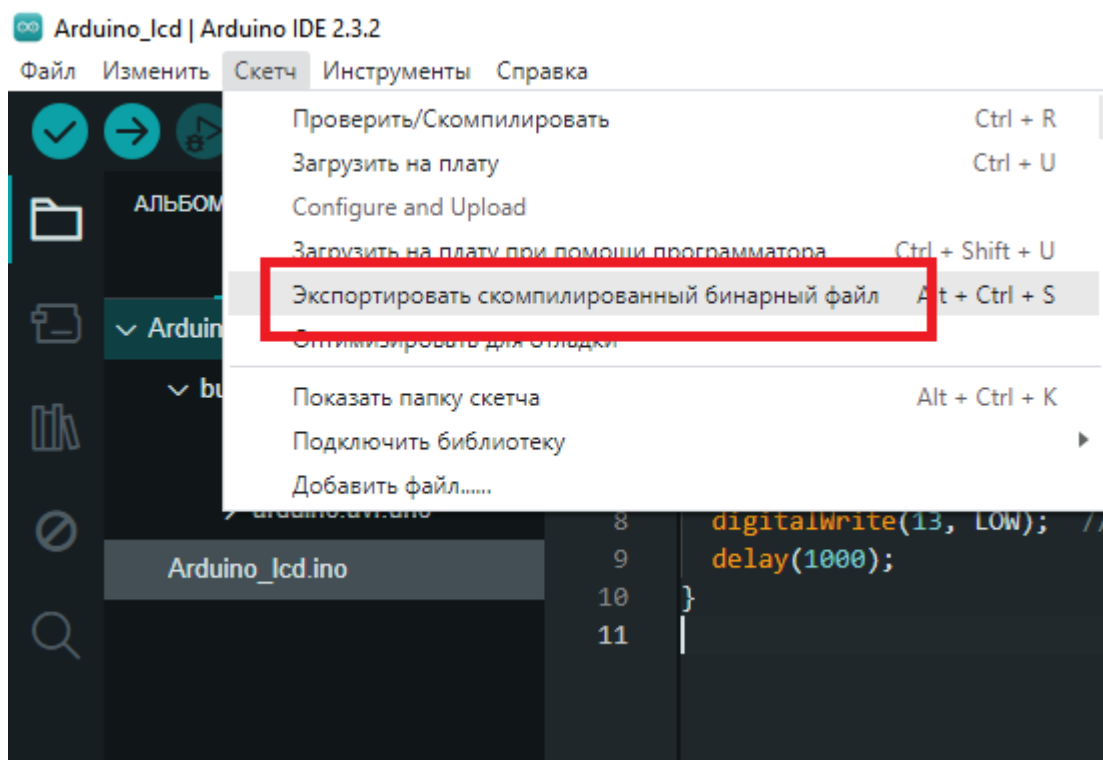
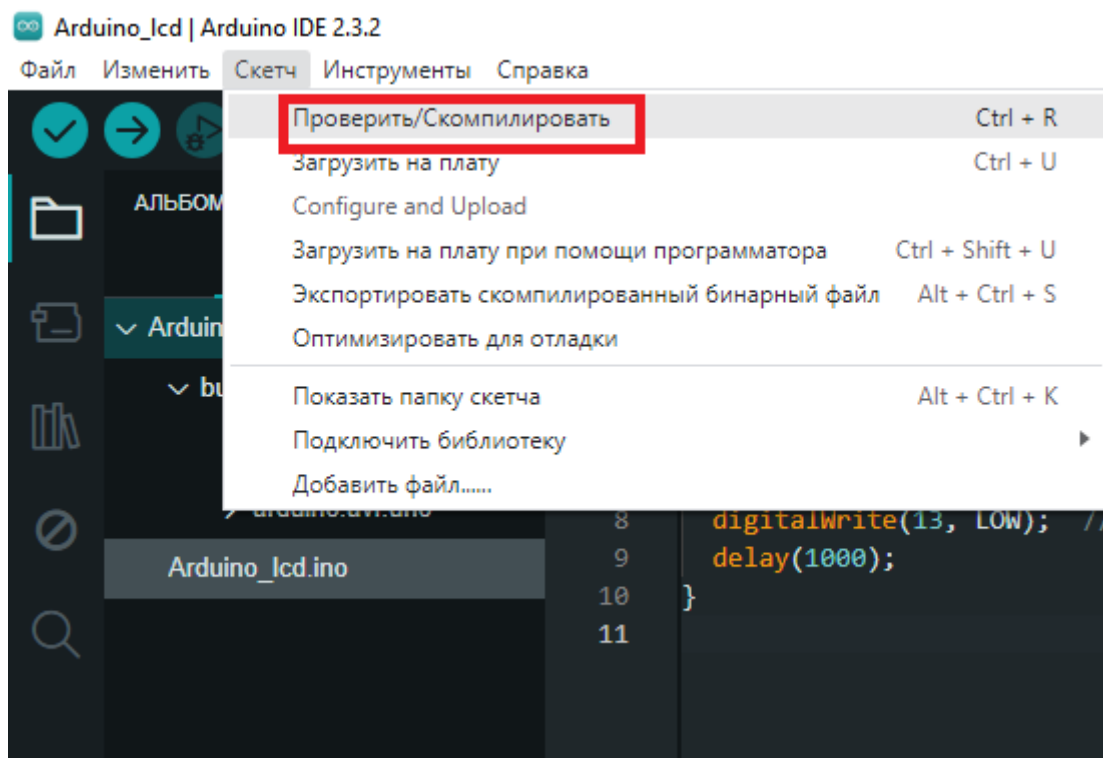
Для чего используется:

- Основная логика программы: чтение данных с датчиков, управление устройствами, обработка событий.
- Любые действия, которые должны выполняться циклически (например, мигание светодиодом).

Важные нюансы:

- Обязательность функций: Без `setup()` и `loop()` скетч не скомпилируется — Arduino IDE выдаст ошибку.
- Однократность `setup()`: Внутри `setup()` нельзя использовать бесконечные циклы (например, `while(true)`), иначе `loop()` никогда не начнется.
- Бесконечность `loop()`: Код в `loop()` выполняется снова и снова, пока Arduino включен.
- Оптимизация: Избегайте длительных задержек (`delay()`) в `loop()` — это блокирует выполнение другого кода. Для многозадачности используйте `millis()`.

Скомпилируйте код:

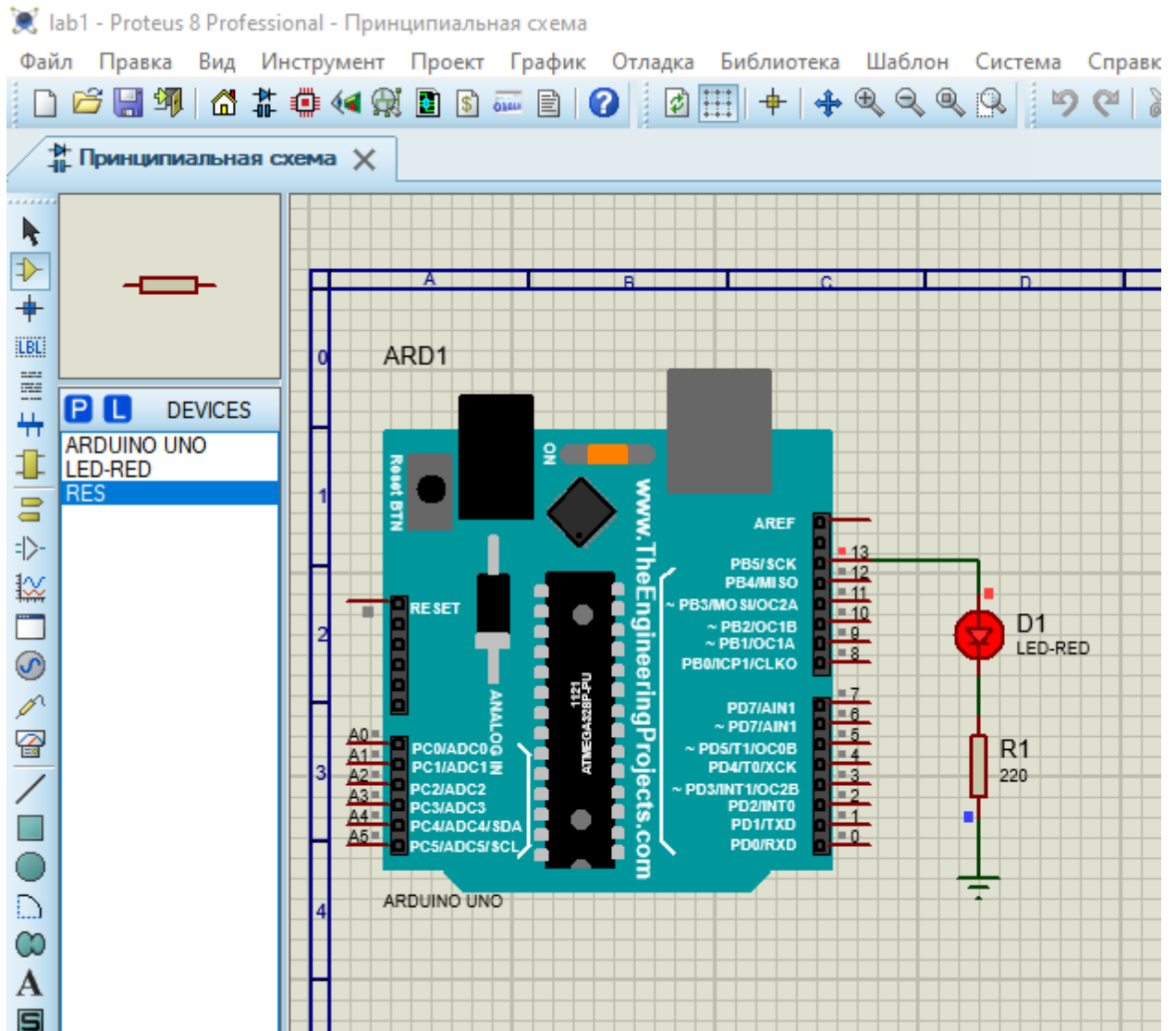


файл **.hex** в сохранится в той же папке, что и ваш файл скетча Arduino (.ino), например:
c:\Users\User\Documents\Arduino

3. Симуляция в Proteus

1. В Proteus дважды щелкните по модели Arduino Uno.
2. В разделе *Program File* укажите путь к hex-файлу.
3. Запустите симуляцию кнопкой **Play** (в нижней панели).

4. Наблюдайте за миганием светодиода.



Требования к отчёту

1. Схема подключения в Proteus (скриншот).
2. Исходный код программы с комментариями.
3. Скриншот работающей симуляции.
4. Ответы на контрольные вопросы:
 - Для чего нужен резистор в схеме со светодиодом?
 - Какие функции выполняют `setup()` и `loop()` в Arduino?
 - Как изменить частоту мигания светодиода?

Контрольные вопросы

1. Почему светодиод подключается через резистор?

2. Что произойдет, если убрать функцию delay(1000)?
 3. Можно ли использовать для подключения светодиода аналоговый пин (например, A0)?
-

Рекомендации

- Если светодиод не мигает, проверьте:
 - Правильность подключения (анод к пину 13, катод к GND).
 - Наличие резистора.
 - Корректность пути к файлу .hex в Proteus.
- Для отладки используйте **виртуальный терминал** в Proteus для мониторинга вывода данных.

Отчет должен содержать (см. образец):

- номер и тему лабораторной работы;
- фамилию, номер группы студента и вариант задания;
- скриншоты окна Proteus и Arduino IDE с исходным кодом программ и комментариями;
- скриншоты с результатами выполнения симуляции;
- пояснения, если необходимо;
- выводы.

Отчеты в формате **pdf** отправлять на email: **colledge20education23@gmail.com**