

Лабораторная работа № 18

Тема: Знакомство с Arduino IDE.

Цель работы: Ознакомиться с Arduino IDE, освоить основные функции и возможности среды разработки для программирования микроконтроллеров Arduino.

Arduino - это открытая платформа для создания прототипов электронных устройств, основанная на простой и доступной аппаратной и программной инфраструктуре. Вот основные характеристики Arduino:

1. **Микроконтроллеры:** Arduino использует микроконтроллеры, такие как ATmega328P, ATmega2560 и другие, как основу для своих плат. Эти микроконтроллеры предоставляют возможность управления различными внешними устройствами и сенсорами.
2. **Простота использования:** Arduino предлагает простой и интуитивно понятный интерфейс как для аппаратной, так и для программной разработки. Это делает Arduino доступным даже для новичков в области электроники и программирования.
3. **Расширяемость:** Существует широкий выбор различных плат Arduino, каждая из которых имеет свои особенности и характеристики. Кроме того, существует множество дополнительных модулей (шлемов) и периферийных устройств, которые могут быть подключены к платам Arduino для расширения их функциональности.
4. **Программирование:** Программирование на Arduino осуществляется с использованием среды разработки Arduino IDE, которая базируется на языке C/C++. Arduino IDE обладает простым и интуитивно понятным интерфейсом и обеспечивает простую загрузку программ на микроконтроллеры Arduino.
5. **Широкое применение:** Arduino находит применение в различных областях, таких как робототехника, автоматизация, интернет вещей (IoT), хобби-проекты и многое другое. Благодаря своей доступности и простоте использования, Arduino стал популярным инструментом для создания различных электронных устройств.

Arduino IDE

Среда разработки Arduino IDE (Integrated Development Environment) представляет собой интегрированную среду для программирования микроконтроллеров Arduino. Вот обзор основных характеристик и функций этой среды:

1. **Простота использования:** Arduino IDE разработана с учетом простоты и легкости использования. Интерфейс прост и интуитивно понятен даже начинающим пользователям.
2. **Кроссплатформенность:** Arduino IDE доступна для различных операционных систем, включая Windows, macOS и Linux, что позволяет использовать ее на широком спектре компьютеров.
3. **Поддержка различных плат Arduino:** Среда разработки Arduino IDE поддерживает большинство плат Arduino, включая стандартные модели, такие как Arduino Uno, Arduino Nano, Arduino Mega, а также множество других плат на базе AVR и ARM.
4. **Примеры и библиотеки:** Arduino IDE поставляется с обширной библиотекой примеров и стандартных библиотек, что упрощает начало работы с платформой Arduino и позволяет быстро разрабатывать прототипы проектов.
5. **Встроенный текстовый редактор:** В Arduino IDE есть встроенный текстовый редактор с подсветкой синтаксиса, автоматическим форматированием и подсказками кода, что делает процесс написания кода более удобным и эффективным.
6. **Серийный монитор:** Arduino IDE включает в себя встроенный серийный монитор, который позволяет взаимодействовать с платой Arduino через последовательный порт, отображая вывод программы и отправляя данные на Arduino.
7. **Программирование на языке Wiring:** Arduino IDE использует язык программирования Wiring, который основан на языке C/C++, но имеет упрощенный синтаксис и API для работы с платформой Arduino.
8. **Загрузка программ на Arduino:** Arduino IDE позволяет компилировать и загружать программы (скетчи) на плату Arduino через USB-порт компьютера.
9. **Расширяемость:** Arduino IDE поддерживает расширение функциональности с помощью дополнительных библиотек и плагинов, что позволяет адаптировать среду разработки под конкретные потребности пользователей.
10. **Совместимость с другими IDE:** Пользователи могут также использовать другие среды разработки, такие как Visual Studio Code, PlatformIO и Eclipse, для программирования Arduino с использованием соответствующих плагинов и расширений.

Совместимость Arduino и Atmega

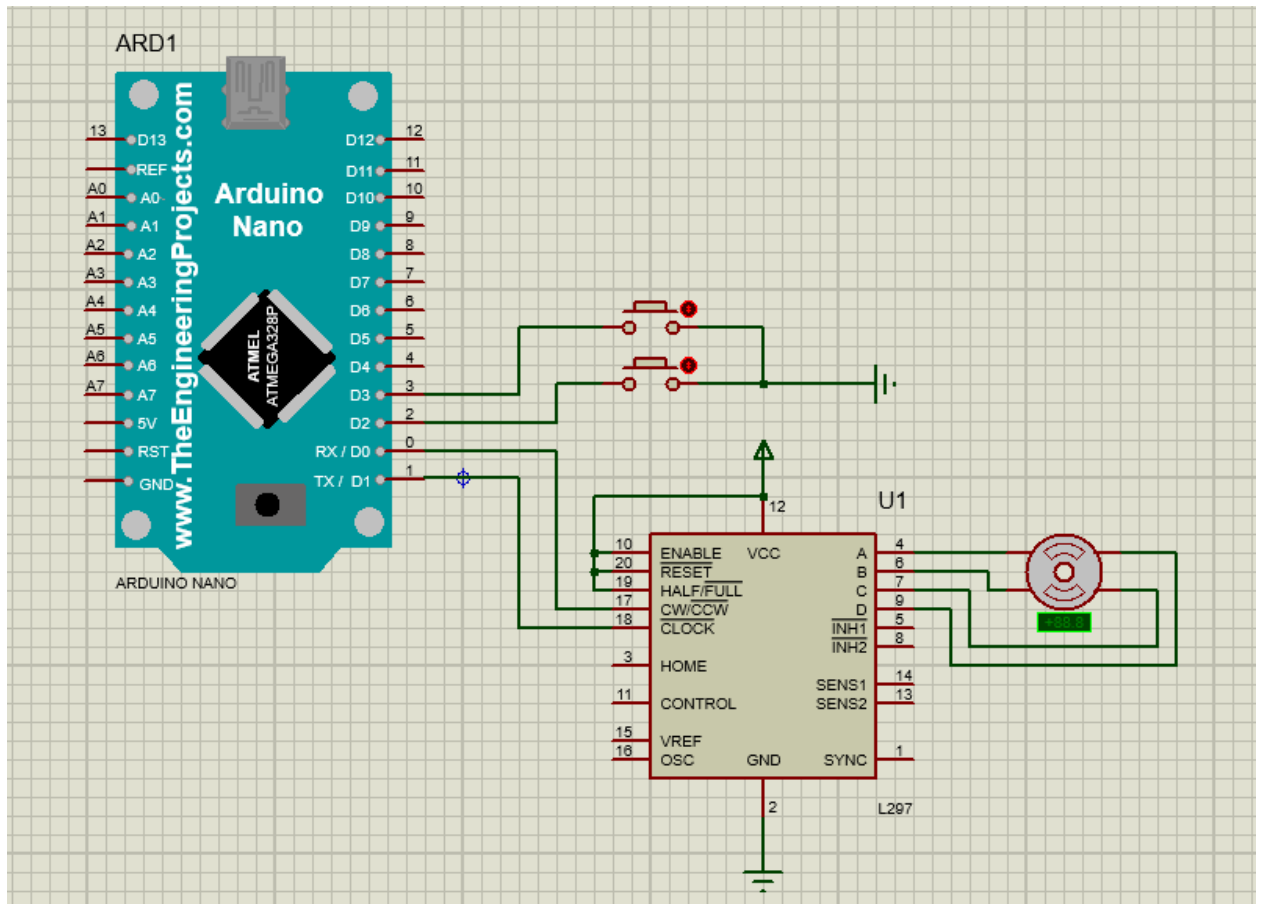
Arduino и микроконтроллеры Atmega тесно связаны друг с другом. Фактически, многие платы Arduino основаны на микроконтроллерах Atmega, таких как Atmega328P, Atmega2560 и других. Вот некоторые

ключевые аспекты совместимости между Arduino и микроконтроллерами Atmega:

1. **Аппаратная совместимость:** Многие платы Arduino, такие как Arduino Uno, Arduino Nano, Arduino Mega и другие, используют микроконтроллеры Atmega в качестве основного компонента. Поэтому скетчи, написанные для плат Arduino, обычно совместимы с соответствующими микроконтроллерами Atmega.
2. **Язык программирования:** Arduino использует язык программирования Wiring, который основан на языке C/C++. Скетчи Arduino, написанные на Wiring, могут быть скомпилированы и загружены на микроконтроллеры Atmega с использованием среды разработки Arduino IDE.
3. **Библиотеки и примеры:** Arduino IDE поставляется с обширной библиотекой примеров и стандартных библиотек, предназначенных для работы с микроконтроллерами Atmega. Эти библиотеки предоставляют простые и удобные интерфейсы для работы с периферийными устройствами, такими как датчики, дисплеи и другие.
4. **Поддержка платформы:** Arduino поддерживает различные модели микроконтроллеров Atmega, что позволяет разработчикам выбирать подходящий для своих проектов микроконтроллер и платформу Arduino.
5. **Порты и пины:** Многие функции и возможности платформы Arduino, такие как цифровые и аналоговые входы/выходы, прерывания, UART, SPI и I2C, поддерживаются микроконтроллерами Atmega. Поэтому код, написанный для Arduino, может легко переноситься на микроконтроллеры Atmega.
6. **Прошивка и загрузка программ:** Arduino IDE обеспечивает простой способ компиляции, загрузки и прошивки программ на микроконтроллеры Atmega через USB-порт компьютера. Это делает процесс разработки и отладки проектов на микроконтроллерах Atmega быстрым и удобным.

В целом, Arduino и микроконтроллеры Atmega очень совместимы друг с другом, что делает платформу Arduino привлекательной для широкого круга разработчиков, начиная от новичков до опытных профессионалов.

В Proteus создадим новый проект, добавим новый элемент – Arduino nano, соберем схему как в предыдущей работе (Что бы в Proteus появились платы Arduino нужно скопировать файлы из папки **arduino library\Proteus Library Files** в папку **C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\DATA\Library**):



В Arduino IDE создадим новый проект, плату прототипирования выберем Arduino Nano, напомним и скомпилируем скетч с использованием библиотеки `arduino.h`, (она более удобна, но недоступна в AtmelStudio):

```
#include <Arduino.h>

#define DIR_PIN 0 // 0-й бит порта С управляет направлением вращения двигателя.
#define STEP_PIN 1 // 1-й бит порта С управляет отдельным шагом двигателя.
#define OPEN_BUTTON_PIN 2 // "Открыть" шторы.
#define CLOSE_BUTTON_PIN 3 // "Заккрыть" шторы.

#define OPEN_STEPS 10 // Количество шагов для открытия
#define CLOSE_STEPS 10 // Количество шагов для закрытия

#define OPENED_FLAG 1 // Флаг открытого состояния
#define CLOSED_FLAG 0 // Флаг закрытого состояния

int current_state = CLOSED_FLAG; // Текущее состояние (открыто/закрыто)

void setup() {
```

```
// Настройка пинов управления L297
pinMode(DIR_PIN, OUTPUT);
pinMode(STEP_PIN, OUTPUT);

// Настройка пинов входов для кнопок
pinMode(OPEN_BUTTON_PIN, INPUT_PULLUP);
pinMode(CLOSE_BUTTON_PIN, INPUT_PULLUP);
}

void stepMotor(int steps, int direction) {
    digitalWrite(DIR_PIN, direction == 1 ? HIGH : LOW);

    int steps_taken = 0;
    while (steps_taken != abs(steps)) {
        digitalWrite(STEP_PIN, HIGH);
        delay(50);
        digitalWrite(STEP_PIN, LOW);
        delay(50);

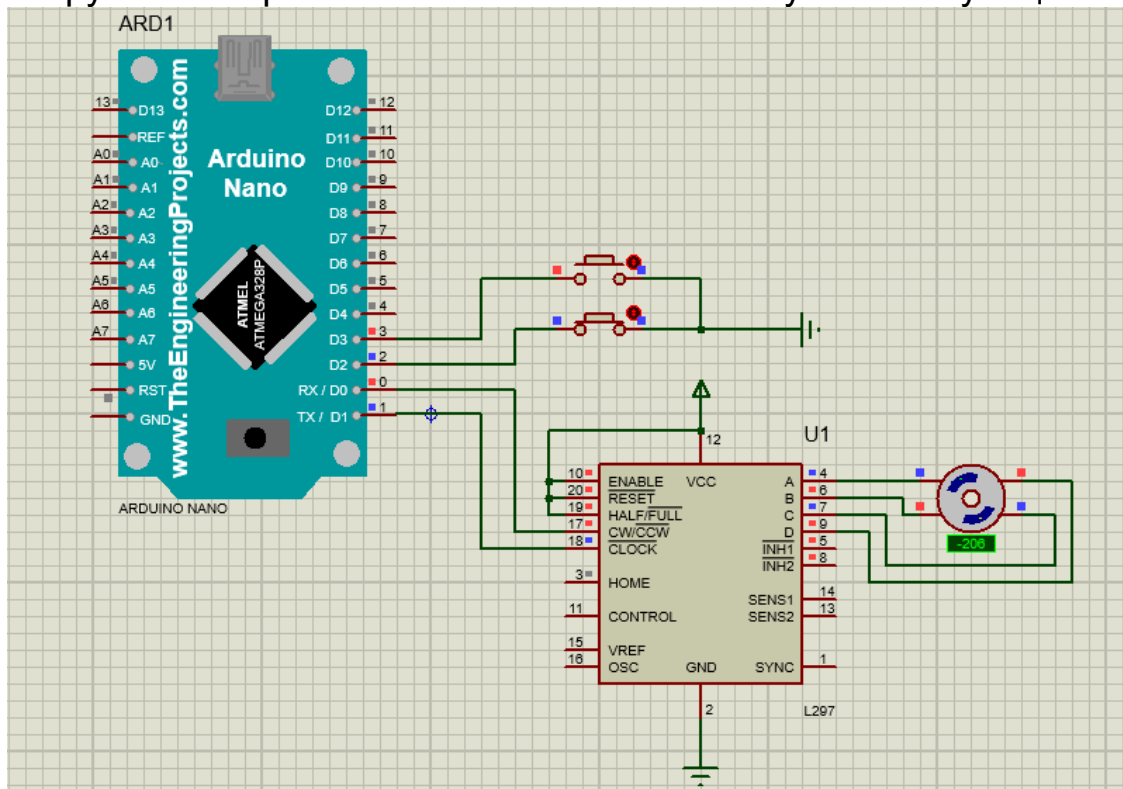
        steps_taken++;
    }
}

void handleManualControls() {
    // Проверка нажатия кнопки "Открыть шторы"
    if (digitalRead(OPEN_BUTTON_PIN) == LOW) {
        if (current_state == CLOSED_FLAG) {
            stepMotor(OPEN_STEPS, 1);
            current_state = OPENED_FLAG;
        }
    }

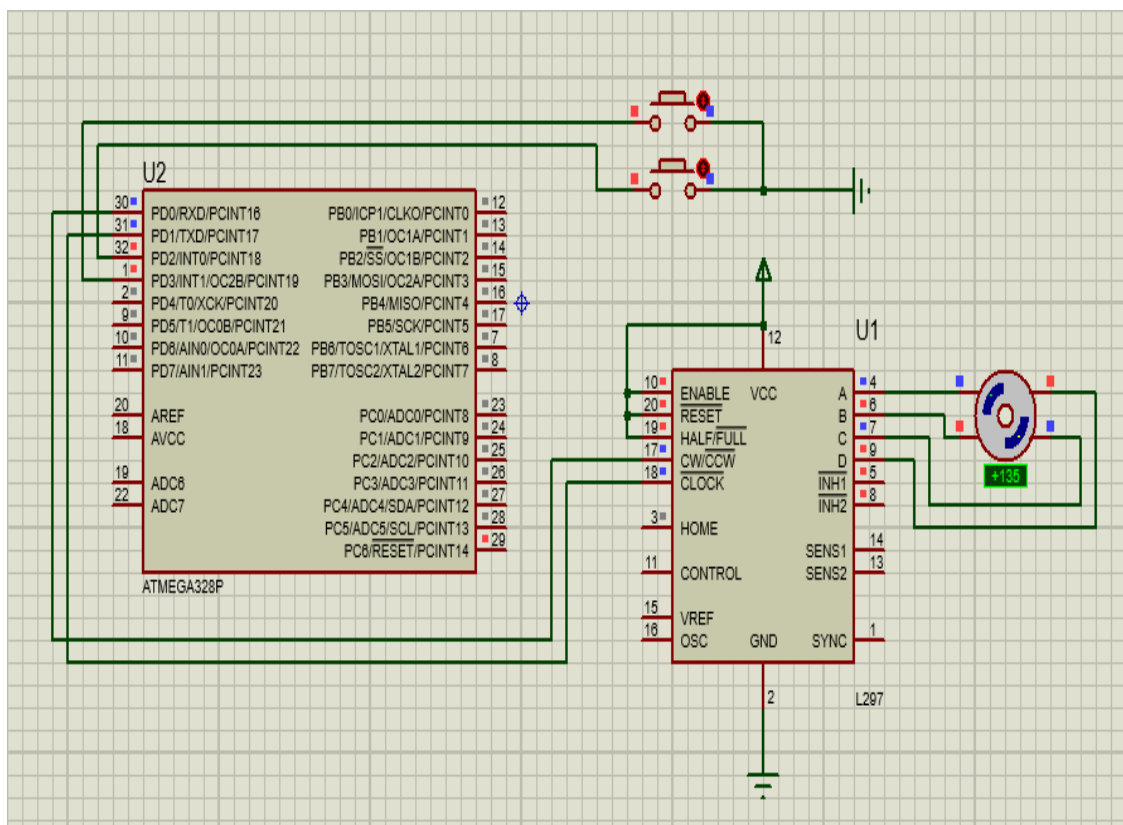
    // Проверка нажатия кнопки "Заккрыть шторы"
    if (digitalRead(CLOSE_BUTTON_PIN) == LOW) {
        if (current_state == OPENED_FLAG) {
            stepMotor(CLOSE_STEPS, -1);
            current_state = CLOSED_FLAG;
        }
    }
}

void loop() {
    handleManualControls();
    delay(100);
}
```

Загрузим hex-файл в Arduino в Proteus и запустим симуляцию:



Теперь заменим Arduino на Atmega328, загрузим в него ту же прошивку и проверим:



Задание:

1. Собрать схему в Proteus с Arduino, датчиком освещенности, двигателем.
2. Написать скетч в Arduino IDE для управления приводом штор.
3. Проверить в симуляции.