



**Министерство науки и высшего образования РФ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО «МГТУ «СТАНКИН»)**

Институт
цифровых интеллектуальных
систем

Кафедра
измерительных информационных систем и
технологий

**Лабораторная работа №1
«Микроконтроллеры в измерительных информационных системах»**

**Москва
2022**

Микроконтроллер (далее МК) – микросхема, предназначенная для **управления** электронными устройствами. Большинство МК содержат:

1. Процессор.
2. Оперативное запоминающее устройство (далее ОЗУ).
3. Постоянное запоминающее устройство (далее ПЗУ).
4. Периферийные устройства:
 - 4.1. интерфейс ввода/вывода общего назначения GPI;
 - 4.2. интерфейсы связи такие, как UART, I²C, SPI, USB и др.;
 - 4.3. аналого-цифровые и цифро-аналоговые преобразователи (далее АЦП и ЦАП);
 - 4.4. широтно-импульсные модуляторы;
 - 4.5. таймеры;
 - 4.6. и др.

Микроконтроллеры так же, как и другая электроника могут быть выполнены в двух форм-факторах – **SMD** (surface mounted device) и **DIP** (dual in-line package). DIP формат отличается прямоугольной формой, двумя рядами выводов по длинным сторонам корпуса, технологией сквозного монтажа в отверстия или установкой на съемные площадки. SMD формат отличается квадратной формой, рядами выводов по всем сторонам корпуса, технологией поверхностного монтажа.

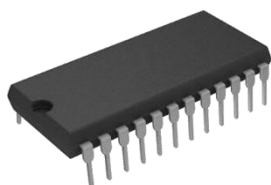


Рис. 1. DIP-корпус



Рис. 2. SMD-корпус

Arduino – это платформа аппаратно-программных средств упрощающая процесс **обучения** программирования микроконтроллеров.

Аппаратная составляющая Arduino состоит из **отладочных плат**, в которых установлены микроконтроллеры AVR и ARM.

Программная составляющая Arduino характеризуется языком программирования «**Arduino Wiring**» (видоизмененный C++) и средой разработки «**Arduino IDE**».



Рис. 3. Arduino UNO
(на базе AVR-микроконтроллере)

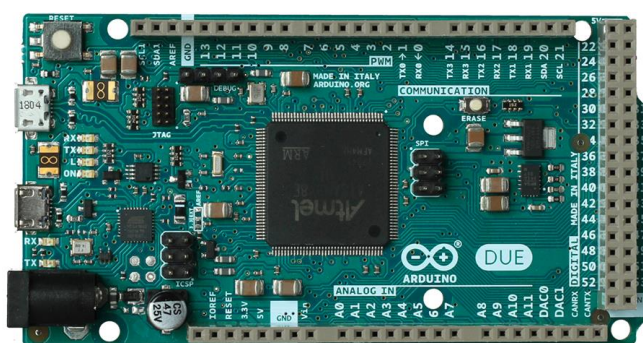


Рис. 4. Arduino DUE
(на базе ARM-микроконтроллере)

Аппаратная часть Arduino

Рассмотрим из чего состоят отладочные платы Arduino на примере одной из самых распространённых плат – Arduino UNO.

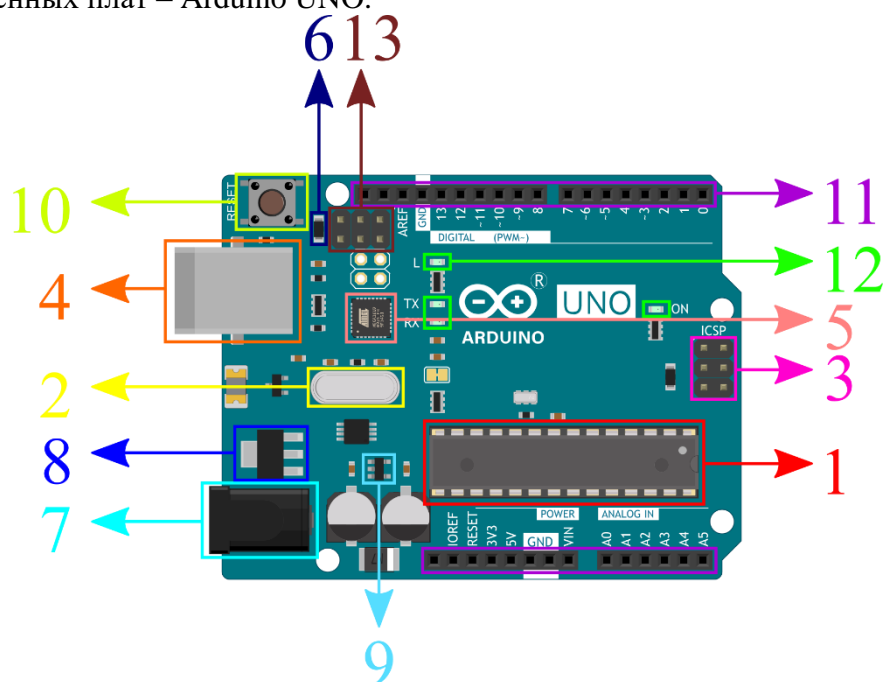


Рис. 5. Компоненты Arduino UNO

Таблица 1 – Перечень компонентов платы Arduino UNO

1	Микроконтроллер ATmega328P	8	Понижающий регулятор 5 [В]
2	Кварцевый резонатор	9	Понижающий регулятор 3.3В
3	ICSP-разъем микроконтроллера ATmega328P	10	Кнопка перезагрузки
4	Порт USB Type-B	11	Разъемы pin
5	Микроконтроллер ATmega16U2	12	Светодиодная индикация
6	Диод	13	ICSP-разъем микроконтроллера ATmega16U2
7	Разъем питания DC(постоянное напряжение)		

1. Микроконтроллер ATmega328P

Микроконтроллер ATmega328P является **основным** компонентом отладочной платы. Исполняется данный МК на архитектуре AVR с:

- тактовой частотой 16МГц,
- Flash-памяти (ПЗУ) 32 [КБ], предназначенной для хранения программы и сопутствующих ей данных.
- RAM-памяти (ОЗУ) 2 [КБ], предназначенной для хранения временных данных, например, значений переменных,
- EEPROM-памяти (ВЗУ) 2 [КБ], предназначенной для долговременного хранения данных, например, данных пользователя.

А также микроконтроллер оснащен аналого-цифровым преобразователем, широтно-импульсным модулятором, аппаратно-реализованными прерываниями и интерфейсами SPI, I²C, UART, GPIO.

2. Кварцевый резонатор

Кварцевый резонатор 16 [МГц] используется для **генерации электрических колебаний** заданной частоты. Кварцевый резонатор является аналогом колебательного LC-контура. Частота колебаний LC-контура сильно зависит от внешней температуры среды так, как емкость и индуктивность чувствительны к температурным изменениям.

Резонаторы на основе кварца обладают очень высокой температурной стабильностью, поэтому их используют для **задания тактовой частоты процессора** микроконтроллера.

3. ICSP-разъем микроконтроллера ATmega328P

Основной микроконтроллер можно «запрограммировать» напрямую через компьютер с помощью ISP программатора. Самым распространённым является USBasp который подключается к ICSP-разъёму. С помощью ICSP можно запрограммировать другую плату Arduino.

На практике для удобства подключения устройств использую платы расширений, на которых распаяны вспомогательные контроллеры, например, драйвер двигателя, дополнительные интерфейсы, цифроаналоговые преобразователи и пр. ICSP-разъём микроконтроллера ATmega328P используется для:

- передачи данных через интерфейс SPI для подключенных плат расширения (смотри рисунок 6),
- загрузки прошивки в основной микроконтроллер ATmega328P через **внешний программатор**.

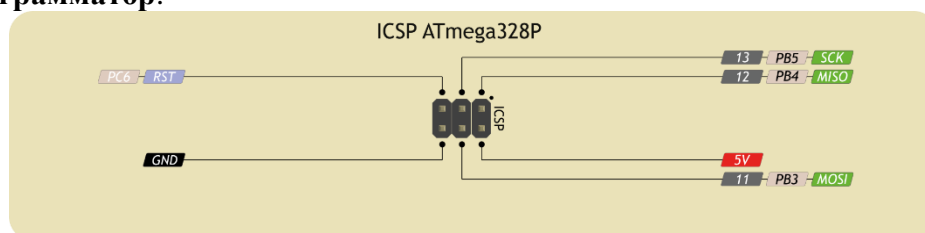


Рис. 6. Разводка выводов ICSP-разъема микроконтроллера ATmega328P

4. Порт USB Type-B

Порт USB Type-B предназначен для **подключения платы к USB порту ПК** через кабель USB (A – B).

5. Микроконтроллер ATmega16U2

Чтобы «запрограммировать» отладочную плату Arduino UNO необходимо подключить плату к компьютеру с помощью кабеля без внешних программаторов, а на компьютере запустить процесс загрузки программы.

На основном МК ATmega328P отсутствует интерфейс USB, а на ПК самой доступной технологией для передачи данных является последовательная технология, то есть USB. При подключении к ПК Arduino Uno определяется как виртуальный COM-порт. На МК ATmega16U2 загружена программа **USB-UART (USB-TTL)** преобразователя. То есть компьютер отправляет данные по USB интерфейсу микроконтроллеру ATmega16U2, а микроконтроллер ATmega16U2 отправляет данные по UART интерфейсу микроконтроллеру ATmega328P и наоборот (смотри рисунок 8).

Загрузка прошивки в основной микроконтроллер через UART возможна при условии наличия там загрузчика, который запишет принятый код во flash. Загрузчик (программа) стартует при подаче питания на МК, ждёт некоторое время, затем передаёт управление основной программе. И так происходит каждый раз при запуске МК. Загрузчик занимает во flash 0.5 [КБ] (старые версии 2 [КБ]), мигает светодиодом на 13 пине при включении, как индикация работы.

ATmega16U2 предназначен для:

- загрузки программы в основной микроконтроллер,
- обмена данных с устройствами через интерфейс USB, например, с ПК.

6. Диод

Микроконтроллеру необходимо питание, как и всей плате. Микроконтроллеры питаются в основном **постоянным напряжением** 5 [В] или 3.3 [В]. Номинальным напряжением питания ATmega328P является 5 [В], Предельно допустимыми считаются напряжения от 4 до 6 вольт. Если подать напряжение 3.5 [В], то микроконтроллер при высокой нагрузке отключится, а при питании 6.5 [В] микроконтроллер может сгореть.

Одним из способов питания **отладочной платы** является питание через USB порт. По линии питания от USB стоит диод, выполняющий **защитную функцию**: он защищает порт USB компьютера от высокого потребления тока компонентами схемы и от случайного короткого замыкания (далее КЗ). КЗ продолжительностью меньше секунды не успеет сильно навредить диоду, но продолжительное замыкание превращает диод в плавкий предохранитель. После этого плата перестаёт определяться компьютером и диод нужно заменить.

Максимальный ток, который можно получить при питании от USB – 500 [мА]. Слаботочный диод имеет ещё одну неприятную особенность: на нём падает напряжение, причем чем больше ток потребления схемы, тем сильнее. По USB плата получает ровно 5 [В], после диода остаётся ~4.7 [В].

7. Разъем питания DC

Разъем питания DC Barrel Jack предназначен для подключения **внешнего источника питания** с напряжением в диапазоне от 7 до 12 вольт.

8. Понижающий регулятор 5 [В]

Понижающий линейный преобразователь NCP1117ST50T3G обеспечивает **питание микроконтроллера** и другой логики платы при подключении питания через разъем питания DC (пункт 7) или пин Vin. Диапазон входного напряжения от 7 до 12 вольт. Выходное напряжение 5 [В] с максимальным выходным током 1 [А].

9. Понижающий регулятор 3.3 [В]

Понижающий линейный преобразователь LP2985-33DBVR обеспечивает **напряжение на пине 3V3**. Регулятор принимает входное напряжение от регулятора 5 [В] и выдаёт напряжение 3.3 [В] с максимальным выходным током 150 [мА].

10. Кнопка перезагрузки

Кнопка предназначена для ручного сброса прошивки, то есть программа начнет выполняться заново.

11. Разъемы pin

Разъемы «pin» предназначены для подключения устройств. Пины подключены к **ножкам микроконтроллера** и к другим устройствам платы (смотри рисунок 8).

Стоит отметить что микроконтроллер – **логическое устройство**, которое создано для управления другими устройствами при помощи логических сигналов. Логическое – означает не силовое, то есть питать от пина МК нельзя ничего мощнее светодиода или слабой микросхемы. Для Arduino UNO рекомендуемый ток с пинов GPIO – не более 20 [мА], максимальный ток с пина – 40 [мА], максимальный ток с порта – 200 [мА].

GPIO (general-purpose input/output) – входы-выходы общего назначения, которые позволяют читать и выдавать цифровой сигнал. К GPIO относятся порты PC, PB, PD.

12. Светодиодная индикация

На отладочной плате установлено 4 светодиода. Благодаря этим светодиодам можно определить наличие/отсутствие: питания на плате (ON), питания на пине 13 (L), передачи (TX – transmit) и приема (RX – receive) данных по интерфейсу UART. Стоит отметить что при загрузке кода, светодиоды TX и RX светятся, но почему? Ответ в пункте 5.

13. ICSP-разъем микроконтроллера ATmega16U2

ICSP-разъем предназначен для программирования (загрузки прошивки) микроконтроллера ATmega16U2.

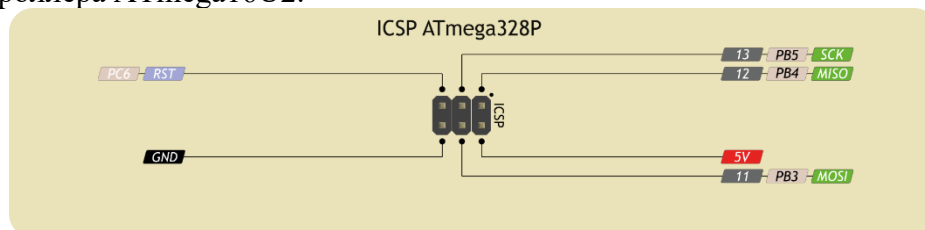


Рис. 7. Разводка выводов ICSP-разъема микроконтроллера ATmega16U2

Помимо отмеченных элементов на изображении, на плате установлены и другие компоненты, обеспечивающие стабильную и надежную работу платы. Например, конденсаторы, защищающие плату от помех и шумов по линии питания, множество токоограничивающих резисторов, предохранитель самовосстанавливающийся и др.

Взаимосвязь компонентов можно наблюдать на рисунке 8.

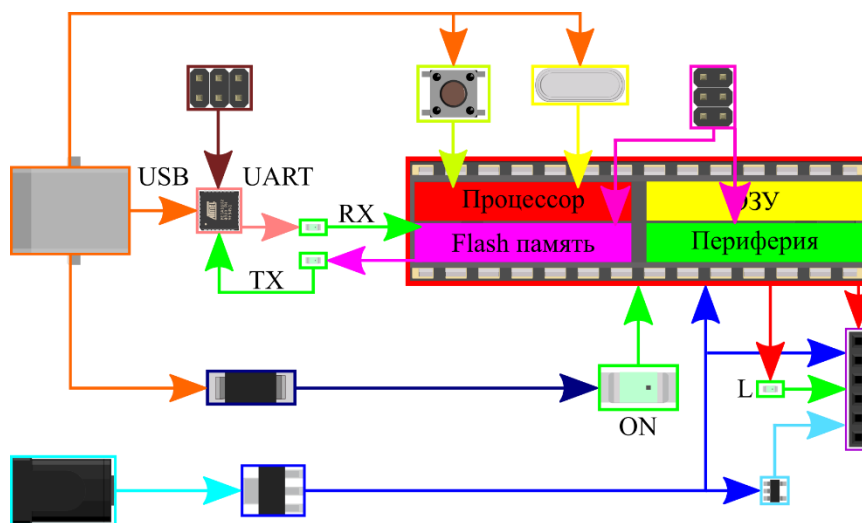


Рис. 8. Структурная схема отладочной платы

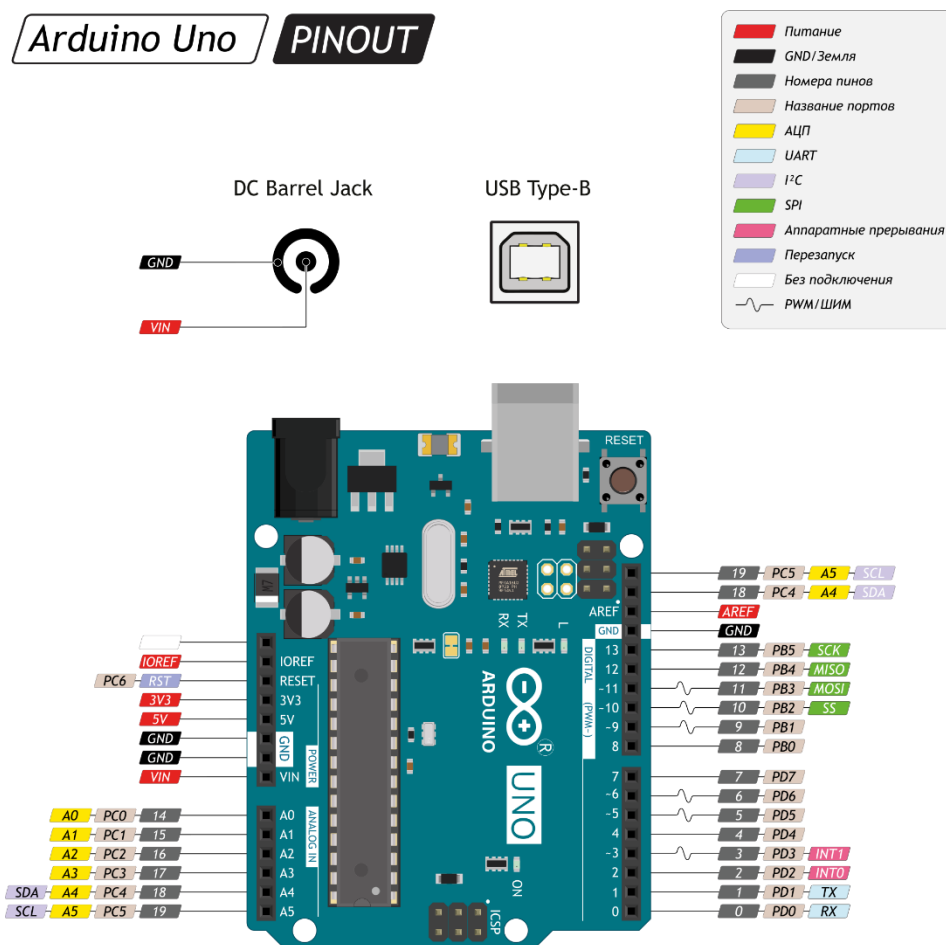


Рис. 9. Разводка выводов отладочной платы Arduino UNO

Задание №1

Подключите светодиод к плате Arduino UNO и помигайте светодиодом.

Решение:

Для сборки цепи понадобится беспаячная макетная плата (breadboard). Макетная плата упрощает процесс прототипирования цепей.

Макетная плата состоит из системы разъемов. Под разъемами расположены провода. Под верхними и нижними разъемами провода располагаются горизонтально (рис. 1а), а под центральными разъемами, провода находятся вертикально (рис. 1б). С помощью разъемов макетной платы дублируют контакты компонентов. Например, можно подключить пин «GND» к линии подписанный как «-», и все разъемы линии минус будут продолжением пина «GND». Если подключить к точке j1 катод светодиода, тогда точки i1, h1, g1, f1 будут дублировать катод светодиода.

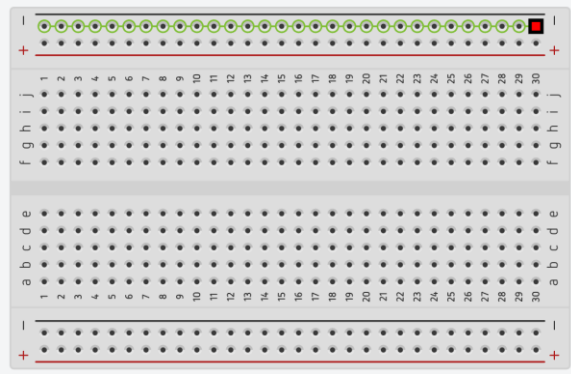


Рис. 10. Горизонтальные линии разъемов

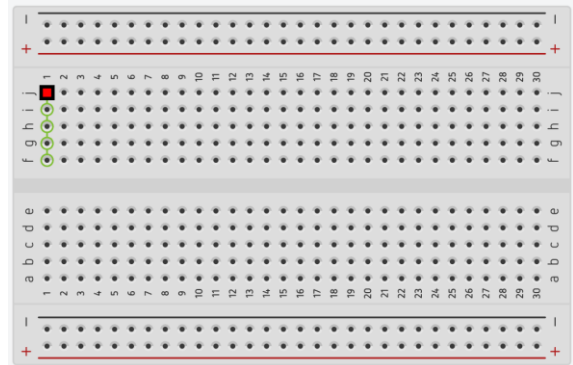


Рис. 11. Вертикальные линии разъемов

Светодиод необходимо подключить к макетной плате в любые центральные отверстия так чтобы ножки светодиода не находились на одной линии. Левая ножка светодиода на схеме является катодом (короткая ножка), а правая анодом (длинная ножка). Катод подключают к минусу, а анод к плюсу источника. Чтобы запомнить куда подключаются контакты, нужно посчитать количество букв в словах, в словах «катод» и «минус» по 5 букв, в словах «анод» и «плюс» по 4 буквы.

«Минус» на плате Uno находится на пинах «GND», а пины подписанные цифрами от 0 до 13 и от A0 до A5 могут выступать в роли «плюса» на схеме.

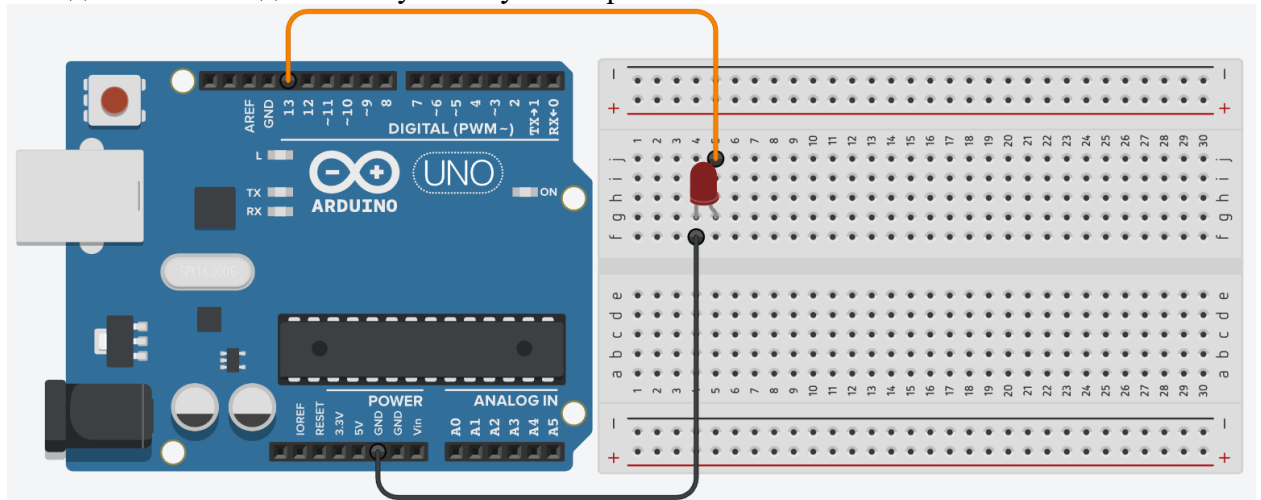


Рис. 12. Схема подключения светодиода к плате Arduino UNO

Пины от 0 до 19 включительно (A0 – это 14, а A5 – это 19) являются пинами интерфейса GPIO, которые могут выступать в роли источников двоичного цифрового сигнала. Данные пины могут как «выдавать» напряжение, так и «получать». Напряжение, которое выдает пин называется напряжением контроллера и обозначается как VCC (Voltage Collector Collector), то есть «плюс» относительно точки нулевого потенциала GND

(GROUND). VCC микроконтроллера ATmega328P зависит от качества питания. Если запитать контроллер от USB-порта, то $VCC \approx 4.7$ В.

Чтобы убедиться в том, что собранная цепь не сгорит следуют ответить себе на следующие вопросы:

1. Цепь замкнута?
2. GND является «минусом» схемы?
3. GND общий контакт или GND точка с самым низким потенциалом на схеме?

Микроконтроллер начнет мигать светодиодом если написать программу в среде разработки Arduino IDE и загрузить ее в плату.

Программа мигания светодиодом
<pre>void setup() { pinMode(13, OUTPUT); } void loop() { digitalWrite(13, 1); delay(500); digitalWrite(13, 0); delay(500); }</pre>

Любая программа, написанная на языке программирования Arduino может состоять из сколь угодно функций, но должны быть две которые называются «**setup**» и «**loop**». Процедура «**setup**» выполняется первой и один раз, далее передает управление в **loop**, который выполняется до тех пор, пока микроконтроллер не отключится.

Функция «**pinMode**» устанавливает режим работы пина, в основном на «вход» или «выход». В режиме на вход ожидается получения сигнала высокого уровня, а в режиме на выход – выдачу сигнала высокого уровня. Вызовом данной функции микроконтроллер переходит в состояние ожидания команды выдать или получить (считать) сигнал. По умолчанию все пины настроены на режим работы «**INPUT**». На светодиод необходимо выдавать сигнал, поэтому функция «**pinMode**» вызвана с аргументом «**OUTPUT**».

Функция «**digitalWrite**» сообщает микроконтроллеру команду выдать сигнал либо высокого уровня, либо низкого уровня. Сигналом высокого уровня является напряжение равное напряжению микроконтроллера, а сигналом низкого уровня – напряжение на точке с нулевым потенциалом. При вызове функции «**digitalWrite**» указывают номер пина и уровень сигнала, где 1 – сигнал высокого уровня, а 0 – сигнал низкого уровня.

Функция «**delay**» останавливает выполнение кода на указанное при вызове время. Время указывается в миллисекундах.

Написанную программу называют **скетчем** (sketch) и ее необходимо загрузить в плату. Для этого выполните следующие действия:

1. Выберите в «**Инструменты**» → «плата» → «**Arduino UNO**».
2. Выберите в «**Инструменты**» → «порт» → «**COM (Arduino UNO)**».
3. Нажмите на кнопку **загрузить** (смотри рисунок 13).



Рис. 13. Кнопка загрузки программы

Задание №2

Измерьте напряжение микроконтроллера (VCC), напряжение на светодиоде (U_c), ток протекающий через светодиод (I_c).

Решение:

Чтобы измерить напряжение микроконтроллера следует настроить плату на выдачу данного напряжения. Напишите программу и загрузите в плату.

Программа выдачи сигнала высокого уровня

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

Для измерения напряжения и силы тока используется мультиметр. Щупы мультиметра необходимо подключить ко 2ому и к 3ему разъему, а ползунок перевести в положение измерения постоянного напряжения в пределах 20 [В] (смотри рисунок 14).



Рис. 14. Мультиметр настроенный на режим измерения напряжения

Щупы следует подключить к пину №13 и к пину GND (смотри рисунок 15).

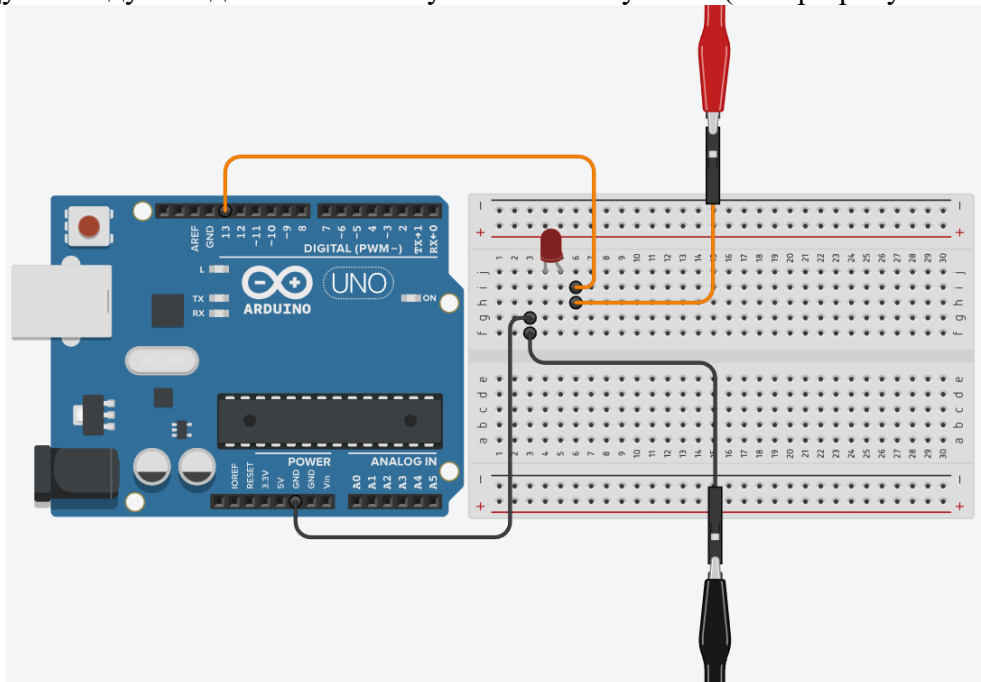


Рис. 15. Схема подключения вольтметра к 13 пину

Результаты измерения запишите в отчет. Какой параметр был измерен?
Далее подключите щупы мультиметра **параллельно** к контактам светодиода (смотри рисунок 16).

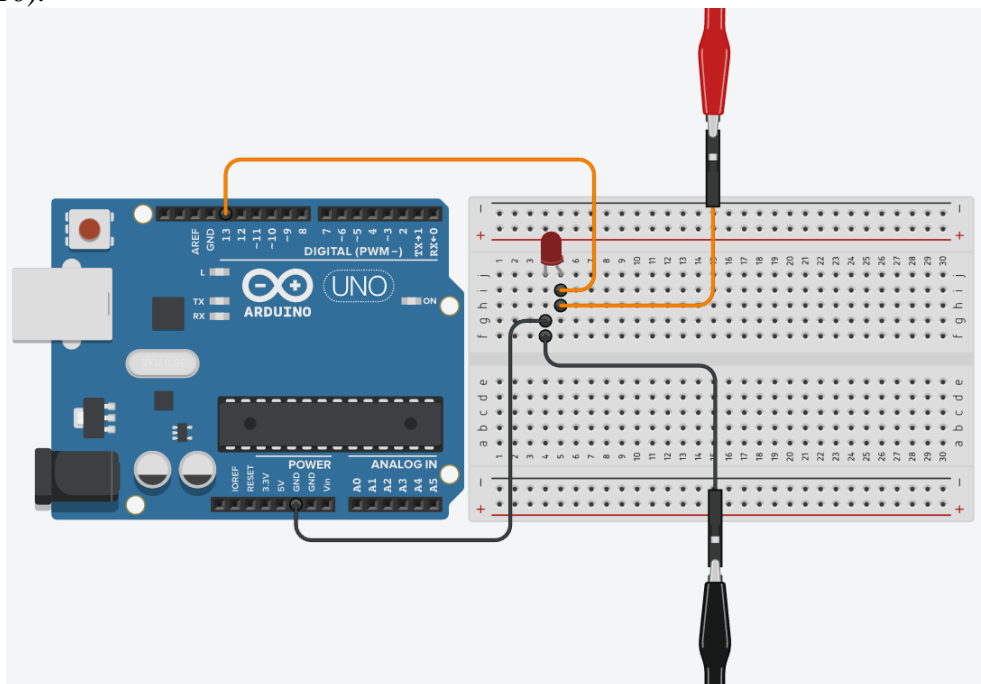


Рис. 16. Схема подключения вольтметра к светодиоду

Результаты измерения запишите в отчет. Какой параметр был измерен?
Для измерения силы тока используется мультиметр в режиме амперметра. Щупы мультиметра необходимо подключить ко 1ому и к 3ему разъему, а ползунок перевести в положение измерения постоянного тока в пределах 10 [A] (смотри рисунок 17).



Рис. 17. Мультиметр настроенный на режим измерения тока

Далее подключите щупы мультиметра **последовательно** к контактам светодиода (смотри рисунок 18).

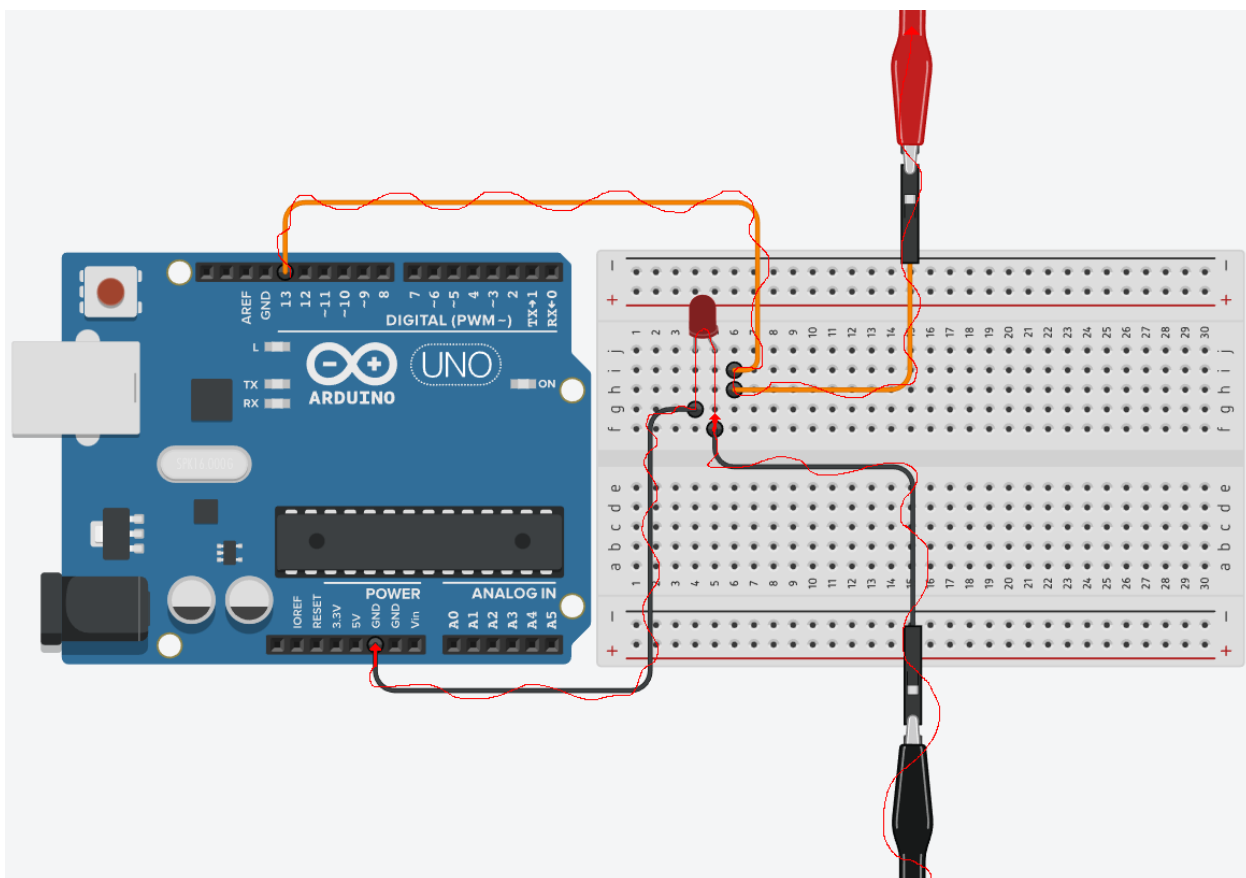


Рис. 18. Схема подключения амперметра к светодиоиду

Задание №3

Постройте график градуировочной характеристики ультразвукового дальномера

Решение:

Дальномер – устройство, предназначенное для определения расстояния от наблюдателя до объекта. Ультразвуковой (далее УЗ) дальномер. Существует много разных модификаций подобных устройств, но все они работают по принципу измерения времени прохождения отраженного звука. То есть датчик отправляет звуковой сигнал в заданном направлении, затем ловит отраженное эхо и вычисляет время полета звука от датчика до препятствия и обратно.

Скорость звука в некоторой среде величина постоянная, но зависящая от плотности среды. Зная скорость звука в воздухе (340.29 [м/с]) и время полета звука до цели, мы можем рассчитать пройденное звуком расстояние по формуле:

$$s = v * t$$

где v – скорость звука в [м/с], а t — время в секундах.

Чтобы справиться со своей задачей, дальномер имеет две важные конструктивные особенности. Во-первых, чтобы звук хорошо отражался от препятствий, датчик испускает ультразвук с частотой 40 [кГц]. Для этого в датчике имеется пьезокерамический излучатель, который способен генерировать звук такой высокой частоты. Во-вторых, излучатель устроен таким образом, что звук распространяется не во все стороны (как это бывает у обычных динамиков), а в узком направлении (смотри рисунок 19).

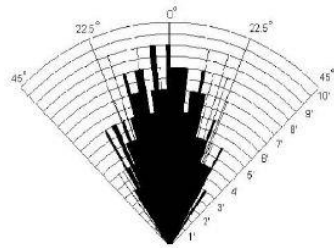


Рис. 19. Диаграмма направленности УЗ-дальномера

Как видно на диаграмме, угол обзора самого простого УЗ дальномера составляет примерно 50-60 градусов. Для данного варианта использования, когда датчик детектирует препятствия перед собой, такой угол обзора вполне пригоден. Ультразвук сможет обнаружить даже ножку стула, тогда как лазерный дальномер, к примеру, может её не заметить.

Если же мы решим сканировать окружающее пространство, вращая дальномер по кругу как радар, УЗ-дальномер даст нам очень неточную и шумную картину. Для таких целей лучше использовать как раз лазерный дальномер.

Также следует отметить два серьезных недостатка УЗ-дальномера. Первый заключается в том, что поверхности имеющие пористую структуру хорошо поглощают ультразвук, и датчик не может измерить расстояние до них. Вторым недостатком связан со скоростью звуковой волны. Эта скорость недостаточно высока, чтобы сделать процесс измерения более частым. Допустим, перед дальномером есть препятствие на удалении 4 метров. Чтобы звук слетал туда и обратно, потребуется целых 24 [мс].

Для измерения расстояния до объекта используем датчик HC-SR04. Этот популярный дальномер умеет измерять расстояние от 1-2 [см] до 4-6 метров. При этом, точность измерения составляет 1 [см].

Чтобы измерить расстояние до объекта нужно создать импульс на контре «trig» и замерить длину импульса на контакте «echo».

Мы уже выяснили ранее, нам потребуется умножить время на скорость звука:

$$s = duration * v = duration * 340 \text{ м/с}$$

Далее переводим скорость звука из [м/с] в [см/мкс]:

$$s = duration * 0.034 \text{ см/мкс}$$

Следует преобразовать десятичную дробь в обыкновенную:

$$s = duration * 1/29 = duration / 29$$

А теперь вспомним, что звук прошел два искомых расстояния: до цели и обратно и поделим всё на 2:

$$s = duration / 58$$

В результате необходимо измеренное время поделить на 58 для получения расстояния до объекта.

Соберите схему по рисунку 20.

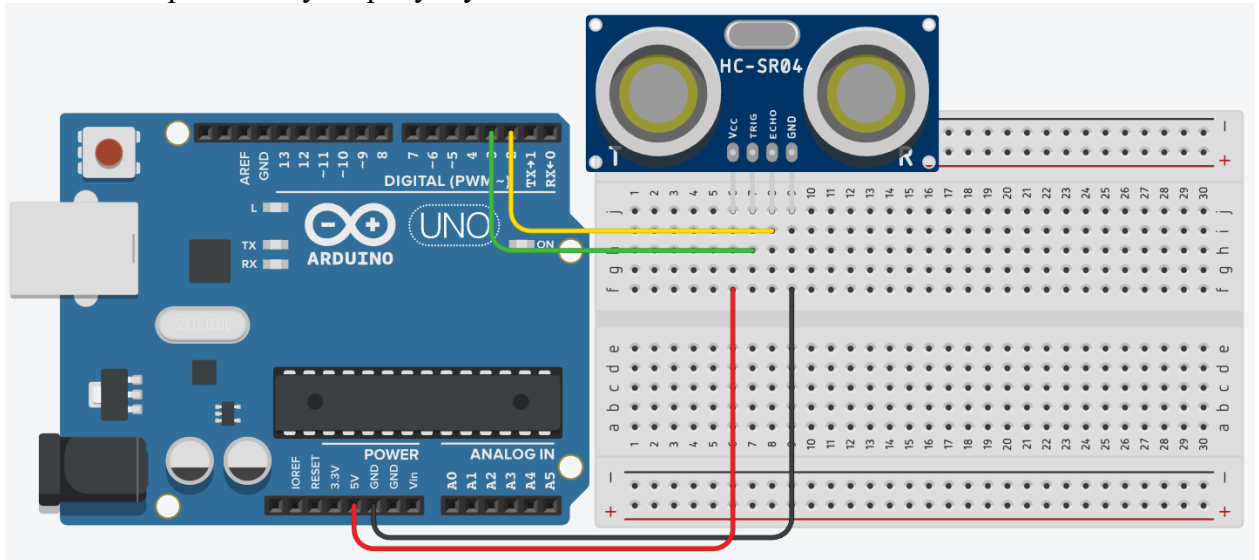


Рис. 20. Схема подключения ультразвукового датчика к Arduino UNO

Напишите программу и загрузите ее в плату.

Программа измерения расстояния до объекта

```
int echoPin = 2; //Номер echo-пина
int trigPin = 3; // Номер trig-пина

void setup() {
    Serial.begin(9600); //Открыть последовательный порт на скорости 9600 Бод
    pinMode(trigPin, OUTPUT); //Настройка trig-пина на выход
    pinMode(echoPin, INPUT); //Настройка echo-пина на вход
}

void loop() {
    int duration; //Измеренное время
    int cm; //Измеренное расстояние
    digitalWrite(trigPin, 0); //Генерация импульса
    delayMicroseconds(2); //Генерация импульса
    digitalWrite(trigPin, 1); //Генерация импульса
    delayMicroseconds(10); //Генерация импульса
    digitalWrite(trigPin, 0); //Генерация импульса
    duration = pulseIn(echoPin, 1); //Измерение времени полета сигнала
    cm = duration / 58; //Преобразование измеренного времени в расстояние
    Serial.println(cm); //Вывод в последовательный порт расстояния до объекта
}
```

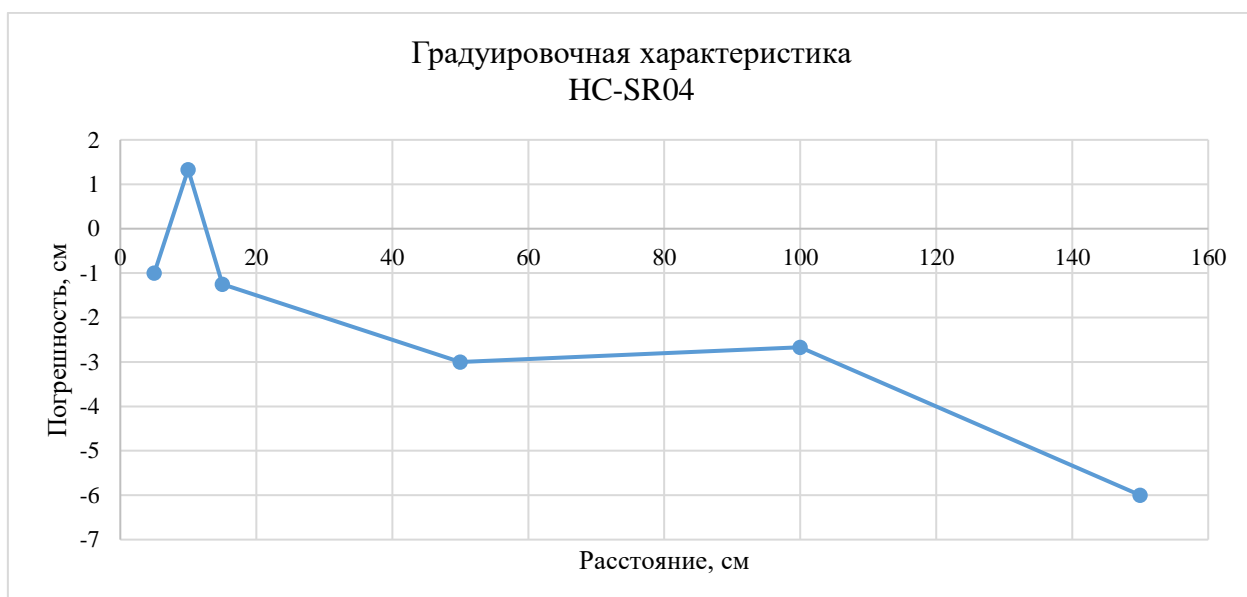
Установите объект с плоской поверхностью на расстояниях 5, 10, 15, 50, 100, 15 [см] строго перпендикулярно линии измерения. Измерьте расстояние до объекта по 3 раза на установленное расстояние. Посчитайте среднее значение измеренных расстояний. Посчитайте абсолютную погрешность. Постройте график градуировочной характеристики.

Для вывода данных с отладочной платы откройте монитор порта. Выберите в «Инструменты» → «Монитор порта». Убедитесь в том, что скорость монитора совпадает со скоростью последовательного порта отладочной платы.

Результаты измерений запишите в таблицу №2. По результатам измерений и расчетов постройте градуировочную характеристику.

Таблица 2 – Абсолютная погрешность измерительного прибора

	см.	Измеренное расстояние, см.			Среднее значение, см.	Абсолютная погрешность, см.
		№1	№2	№3		
Заданное расстояние	5	4	5	3	4	-1
	10	11	12	11	11	1
	15	13	14	13	14	-1
	50	47	46	48	47	-3
	100	97	97	98	97	-3
	150	145	144	143	144	-6



Контрольные вопросы

1. Микроконтроллер – это?
2. Arduino – это?
3. На каком языке программирования разрабатываются прошивки для отладочных плат Arduino?
4. С помощью какого компонента загружается прошивка в отладочную плату?
5. Чему равно максимальное выводимое напряжение на пинах от 0 до 19 на плате Arduino UNO?
6. Какой интерфейс микроконтроллера был использован на лабораторной работе?
7. Сколько раз выполняется процедура loop?
8. На какой режим работы по умолчанию настроены пины?
9. С каким компонентом отладочной платы связаны пины?
10. Почему светодиоды RX и TX включаются/выключаются при загрузке кода?