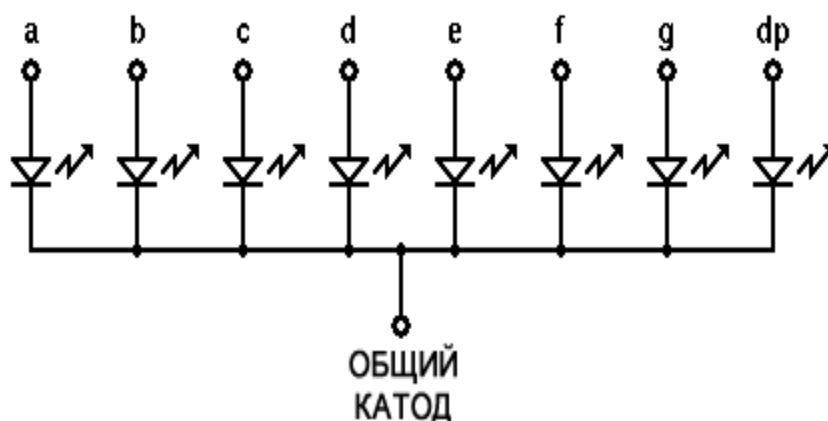


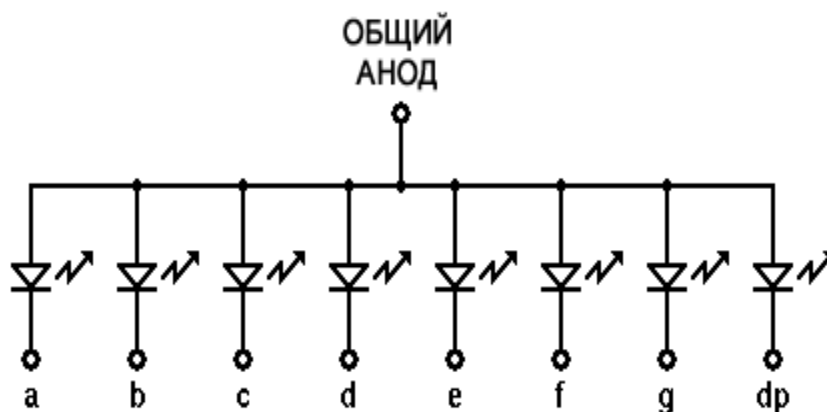
**Цель:** формирование навыков отображения данных на led матрицах и дисплеях при помощи платформы Arduino.



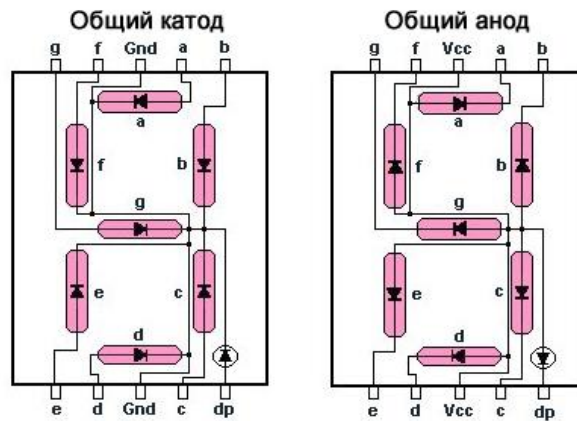
Семисегментный индикатор имеет семь светодиодов, расположенных в форме восьмерки. Он прост в использовании и экономичен в плане стоимости. Семисегментные индикаторы бывают двух типов: с общим анодом и общим катодом. Внутренняя структура обоих типов почти одинакова. Разница заключается в полярности светодиодов и общем выводе. В семисегментном индикаторе с общим катодом (такой мы использовали в экспериментах) катоды всех семи светодиодов и светодиода точки подключены к выводам 3 и 8. Чтобы использовать такой индикатор, нам необходимо подключить корпус к выводам 3 и 8 и подать +5В на другие выводы, чтобы отдельные светодиоды загорелись. Следующей схеме показана внутренняя структура семисегментного индикатора с общим катодом:



Индикатор с общим анодом является полной противоположностью. В индикаторе с общим анодом положительные выводы всех восьми светодиодов соединены вместе и подключены к выводам 3 и 8. Чтобы зажечь отдельный сегмент, вы соединяете его второй вывод с корпусом. На следующей диаграмме показана внутренняя структура семисегментного индикатора с общим анодом:



Сем сегментов обозначены как a-g, а точка как "dp", как показано на рисунке ниже:



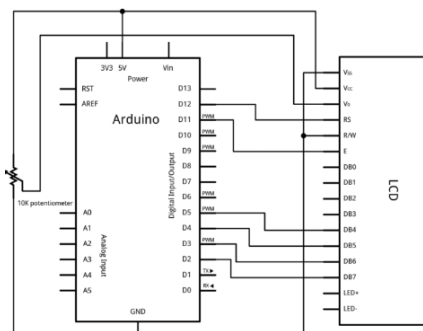
Жидкокристаллический дисплей (Liquid Crystal Display) LCD 1602 является хорошим выбором для вывода строк символов в различных проектах. Давайте посмотрим на выводы LCD1602:



Каждый из выводов имеет свое назначение:

- Земля GND;
- Питание 5 В;
- Установка контрастности монитора;
- Команда, данные;
- **Записывание** и чтение данных;
- Enable;
- 7-14. Линии данных;
- Плюс подсветки;
- Минус подсветки.

Без I2C дисплей подключается к плате следующим образом :



Из-за большого количества подключаемых контактов может не хватить места для присоединения нужных элементов. Использование I2C уменьшает количество проводов до 4, а занятых пинов до 2.

**2C / IIC(Inter-Integrated Circuit)** – это протокол, изначально создававшийся для связи интегральных микросхем внутри электронного устройства. Разработка принадлежит фирме Philips. В основе i2c протокола является использование 8-битной шины, которая нужна для связи блоков в управляющей электронике, и системе адресации, благодаря которой можно общаться по одним и тем же проводам с несколькими устройствами. Мы просто передаем данные то одному, то другому устройству, добавляя к пакетам данных идентификатор нужного элемента.

Самая простая схема I2C может содержать одно ведущее устройство (чаще всего это микроконтроллер Ардуино) и несколько ведомых (например, дисплей LCD). Каждое устройство имеет адрес в диапазоне от 7 до 127. Двух устройств с одинаковым адресом в одной схеме быть не должно.

Плата Arduino поддерживает i2c на аппаратном уровне. Вы можете использовать пины A4 и A5 для подключения устройств по данному протоколу.

В работе I2C можно выделить несколько преимуществ:

- Для работы требуется всего 2 линии – SDA (линия данных) и SCL (линия синхронизации).
- Подключение большого количества ведущих приборов.
- Уменьшение времени разработки.
- Для управления всем набором устройств требуется только один микроконтроллер.
- Возможное число подключаемых микросхем к одной шине ограничивается только предельной емкостью.
- Высокая степень сохранности данных из-за специального фильтра подавляющего всплески, встроенного в схемы.
- Простая процедура диагностики возникающих сбоев, быстрая отладка неисправностей.
- Шина уже интегрирована в саму Arduino, поэтому не нужно разрабатывать дополнительно шинный интерфейс.

Недостатки:

- Существует емкостное ограничение на линии – 400 пФ.
- Трудное программирование контроллера I2C, если на шине имеется несколько различных устройств.
- При большом количестве устройств возникает трудности локализации сбоя, если одно из них ошибочно устанавливает состояние низкого уровня.

## Оборудование



Кабель USB



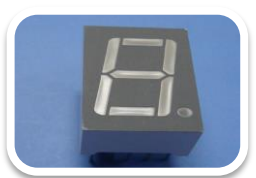
Макетная плата



Плата Arduino Uno



Комплект резисторов



Семисегментный индикатор



Комплект соединительных проводов



Lcd дисплей с I2c



Датчик температуры и влажности



Фоторезистор



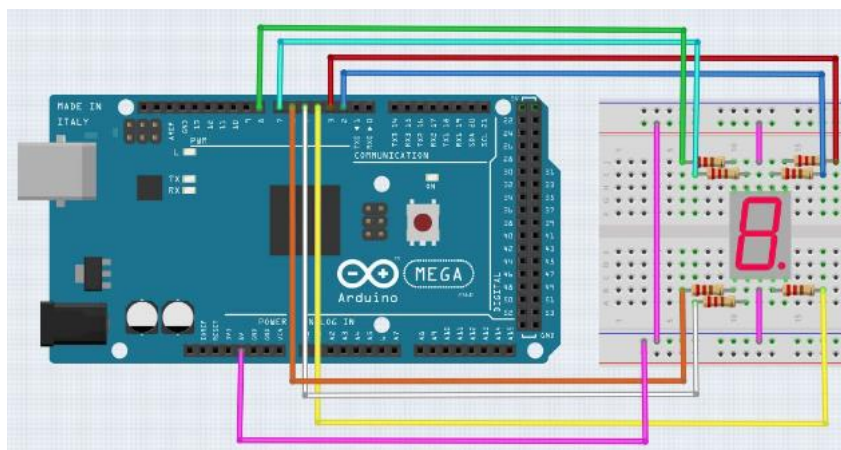
Кнопка

## Упр.1 Семисегментный индикатор

- Для подключения индикатора будем пользоваться следующей таблицей

Выводы индикатора	Выводы Arduino
1(e)	6
2(D)	5
3,8(COM)	GND
C	4
5(dp)	-
6(b)	3
7(a)	2
9(f)	7
10(g)	8

- Подключите семисегментный индикатор в соответствии с приведенной схемой.



- Подключите плату Arduino к USB порту компьютера.
- Создайте и сохраните новый проект под названием **LED7.ino**.
- Для проверки работы схемы загрузите в плату следующий код

```
void setup()
{
    // определить режимы работы выводов
    pinMode(2,OUTPUT);
    pinMode(3,OUTPUT);
    pinMode(4,OUTPUT);
    pinMode(5,OUTPUT);
    pinMode(6,OUTPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
}
void loop()
{
    // цикл для включения светодиодов сегментов индикатора
    for(int i=2; i<9; i++)
    {
        digitalWrite(i,HIGH);
        delay(600);
    }
    // цикл для выключения светодиодов сегментов индикатора
    for(int i=2; i<9; i++)
    {
        digitalWrite(i,LOW);
        delay(600);
    }
    delay(1000);
}
```

- Для отображения на индикаторе значений счетчика, уменьшающегося на 1 каждую секунду, загрузите в плату следующий код:

```
// создать массив для хранения цифр
int num_array[10][7] = { { 1,1,1,1,1,1,0 }, // 0
                        { 0,1,1,0,0,0,0 }, // 1
                        { 1,1,0,1,1,0,1 }, // 2
                        { 1,1,1,1,0,0,1 }, // 3
                        { 0,1,1,0,0,1,1 }, // 4
                        { 1,0,1,1,0,1,1 }, // 5
                        { 1,0,1,1,1,1,1 }, // 6
                        { 1,1,1,0,0,0,0 }, // 7
                        { 1,1,1,1,1,1,1 }, // 8
                        { 1,1,1,0,0,1,1 } }; // 9

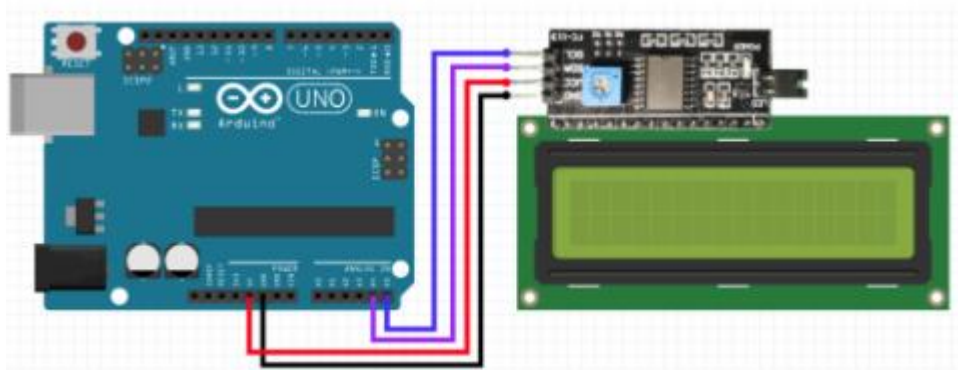
// объявление функции
void Num_Write(int);
void setup() {
    // установить режимы работы выводов
    pinMode(2, OUTPUT);
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT); }
void loop() {
    // цикл счетчика
    for (int counter = 10; counter > 0; --counter) {
        delay(1000);
        Num_Write(counter-1);
    }
    delay(3000); }

// эта функция записывает значения в выводы, подключенные к
индикатору
void Num_Write(int number)
{
    int pin= 2;
    for (int j=0; j < 7; j++)
    {
        digitalWrite(pin, num_array[number][j]);
        pin++;
    }
}
```

## Упр.2 LCD Дисплей

∞ В данном упражнении рассмотрено подключение lcd дисплея с использованием i2c интерфейса, так как подключение дисплея напрямую неудобно и занимает как минимум 6 цифровых контактов.

- Подключите дисплей к плате согласно следующей схеме



- Откройте Arduino IDE.
- Создайте и сохраните новый проект под названием **lcd.ino**.
- Для взаимодействия Arduino с LCD 1602 по шине I2C вам потребуются как минимум две библиотеки: Библиотека Wire.h для работы с I2C уже имеется в стандартной программе Arduino IDE и Библиотека LiquidCrystal\_I2C.h, которая включает в себя большое разнообразие команд для управления монитором по шине I2C и позволяет сделать скетч проще и короче.
- После подключения всех необходимых библиотек мы создаем объект и можем использовать все его функции. Для тестирования давайте загрузим следующий стандартный код из примера :

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2); // Указываем I2C адрес (наиболее
распространенное значение), а также параметры экрана (в случае LCD 1602 - 2
строки по 16 символов в каждой
//LiquidCrystal_PCF8574 lcd(0x27); // Вариант для библиотеки PCF8574
void setup() {
    lcd.init();           // Инициализация дисплея
    lcd.backlight();      // Подключение подсветки
    lcd.setCursor(0,0);   // Установка курсора в начало первой строки
    lcd.print("Hello");   // Набор текста на первой строке
    lcd.setCursor(0,1);   // Установка курсора в начало второй строки
    lcd.print("BSPU fiz-mat"); // Набор текста на второй строке
}
void loop()
{
}
```

## Задания для самостоятельного выполнения

- Добавьте в скетч `lcd.ino` функцию таймера который запускается при включении платы.
- Используя скетч упражнения 1, соберите счетчик с использованием кнопки. Каждое нажатие кнопки прибавляет единицу.
- Соберите систему определения параметров среды (температура, влажность, освещенность). Для отображения данных с датчиков используйте LCD дисплей.



Пройдите тест №5 «LED индикаторы» на странице «Тесты».