

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 05.05.2022 22:54:51
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11aabbf738943df4a4851fda56d089

МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра нанотехнологий, микроэлектроники,
общей и прикладной физики

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова



02

2021 г.

КВАНТОВАЯ И ОПТИЧЕСКАЯ ЭЛЕКТРОНИКА
методические указания к выполнению практических работ
для студентов направления подготовки
28.04.01 «Нанотехнологии и микросистемная техника»

УДК 53

Составители: А.В. Кузько, А.Ю. Мыцких-Коробанов

Рецензент

Директор РИЦ, д.ф.-м.н., профессор А.П. Кузьменко

Квантовая и оптическая электроника: методические указания к выполнению практических работ для студентов направления подготовки 28.04.01 «Нанотехнологии и микросистемная техника» / Юго-Зап. гос. ун-т; сост.: Кузько А.В., Мыцких-Коробанов А.Ю. Курск, 2021. 32 с.: ил. 20. Библиогр.: с. 32.

Излагаются методические рекомендации по выполнению практических проектных работ, в которых предлагается осуществить подбор заданий и спроектировать устройства квантовой и оптической электроники с использованием лазерного модуля KY-008, фоторезистора LDR, RGB-светодиода, матрицы светодиодов, LCD дисплея с учетом требований менеджмента безопасности при работе с оптоэлектронными приборами.

Методические указания соответствуют требованиям Федерального государственного образовательного стандарта высшего образования и учебному плану направления подготовки 28.04.01 Нанотехнологии и микросистемная техника, степень (квалификация) – магистр. Предназначены для студентов всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60 x 84 1/16.

Усл. печ. л. 3,25. Уч.-изд. л. 0,68. Тираж экз. Заказ . Бесплатно.

Юго-Западный государственный университет.

305040 Курск, ул. 50 лет Октября, 94.

ПРАКТИЧЕСКАЯ РАБОТА № 1.

Подбор заданий и проектирование установок с использованием лазерного модуля КУ-008. Соблюдение требований менеджмента безопасности при работе с оптоэлектронными приборами, в особенности с лазерным модулем

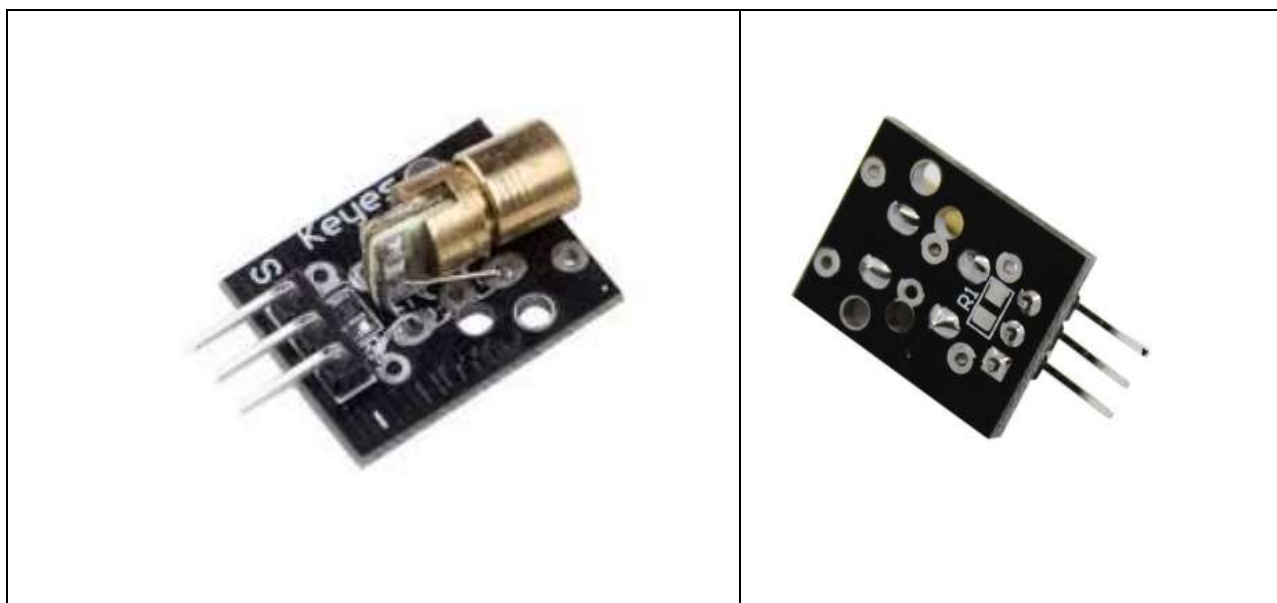
Спецификация лазерного модуля (КУ-008 datasheet):

<https://datasheetspdf.com/pdf/1415009/ETC/KY-008/1>

ОБЩИЕ СВЕДЕНИЯ

ВНИМАНИЕ!!! При работе с лазером необходимо соблюдать особые меры предосторожности! Самое главное луч лазера 650nm нельзя направлять в глаза, кроме того лазерный светодиод при работе сильно нагревается, особенно если на модуль подается питание с **большой силой тока**. При выполнении каждой лабораторной и практической работы необходимо соблюдать инструкцию по технике безопасности, которая размещена на сайте кафедры <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>

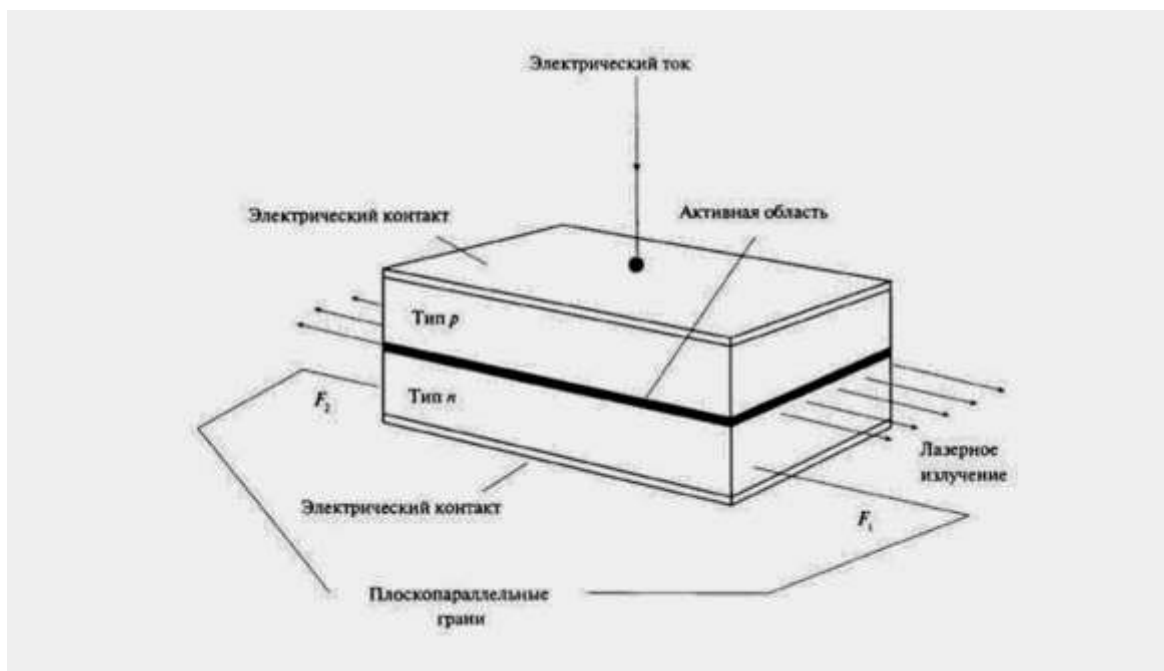
Лазерный диодный модуль КУ-008 (красный)



Принцип работы лазерного светодиода

При подключении светодиода к Arduino, во время прохождения электронов через p-n переход происходит спонтанное излучение фотонов света. В лазерном светодиоде происходит рекомбинация фотонов (вынужденное излучение фотонов с одинаковыми параметрами). Принцип работы лазера основан на том, что излучаемые фотоны света вызывают повторное излучение, которое увеличивается лавинообразно.

Интенсивность излучения передатчика KY-008 зависит от силы тока. При малых токах модуль работает, как обычный светодиод, так как происходит только спонтанное излучение фотонов. Когда сила тока превышает пороговое значение – мощность излучения резко вырастает. Лазерный диод испускает свет перпендикулярно поверхности кристалла и фокусируется с помощью оптики.

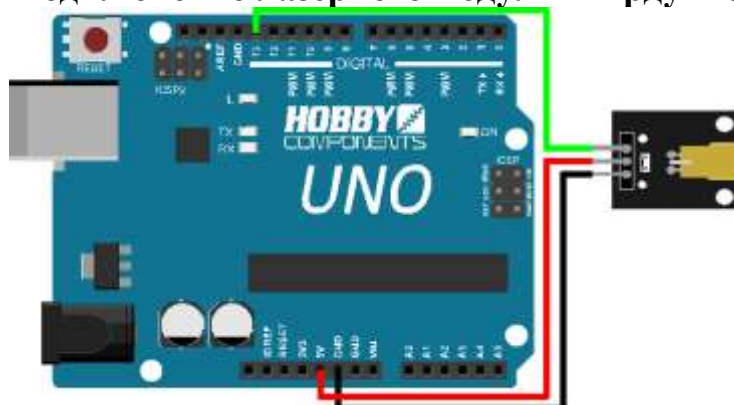


В основе модуля лазера "KY-008" лежит светодиод и два резистора на 1кОм и 91Ом, расположенных на плате с тремя выводами.

Назначение выводов модуля:

- Крайний вывод с меткой «-» → GND
- Средний вывод без метки → +5V
- Крайний вывод с меткой «S» → Управление

Подключение лазерного модуля к Ардуино



Спецификация модуля лазерного передатчика KY-008:

- Рабочее напряжение - 5 В
- Выходная мощность - 5 мВт
- Длина волны - 650 нм
- Рабочий ток - менее 40 мА

- Рабочая температура - от -10°C до 40°C [от 14 до 104° F]
- Размеры - 18,5 мм x 15 мм [0,728 дюйма x 0,591 дюйма]

Подключение лазерного модуля KY-008 к Arduino довольно простое ввиду того, что используется всего три порта.

Скетч «Мигание лазерного светодиода»

```
#define LASER 13 // задаем имя для Pin13

void setup() {
  pinMode(LASER, OUTPUT); // инициализируем Pin13 как выход
}

void loop() {
  for (int i=0; i<=5; i++) // мигание лазерным светодиодом
  {
    digitalWrite(LASER, HIGH);
    delay(500);
    digitalWrite(LASER, LOW);
    delay(500);
  }
  delay(3000);
}
```

Пояснения к коду:

- в скетче цикл for несколько раз включает и выключает лазер;
- время задержки в миллисекундах можно изменять, чем меньше задержка, тем быстрее будет происходить мигание светодиода KY-008.

Вопрос: Можно ли в данном скетче обойтись без цикла for?

Скетч плавного включения KY-008 Arduino

Пересоедините провод с пина 13 на пин 10 (ШИМ).

```
#define LASER 10 // задаем имя для Pin10

void setup() {
  pinMode(LASER, OUTPUT); // инициализируем Pin10 как выход
}

void loop() {
  for (int i=0; i<=255; i++) // плавное включение лазера
  {
    analogWrite(LASER, i);
    delay(5);
  }
  for (int i=255; i>=0; i--) //плавное выключение лазера
  {
    analogWrite(LASER, i);
    delay(5);
  }
}
```

Пояснения к коду:

- порт для включения лазерного светодиода можно менять, но только на аналоговый выход с поддержкой ШИМ (Широтно-импульсная модуляция);
- задержкой `delay(5)`; можно управлять скоростью включения/выключения.

Вопрос: Можно ли в данной задаче вывод лазерного модуля без метки (средний контакт) оставить не подключенным?

Задания

I. По описанию, которое приведено выше (ОБЩИЕ СВЕДЕНИЯ), выполните предложенные проекты (подключение и работа с лазерным модулем).

II. Выберите из представленного перечня те проекты, в которых может быть использован **лазерный модуль**:

- Лазерный счетчик
- Определить концентрацию раствора сахара
- Лазерную систему безопасности
- Лазерная связь кодом Морзе
- Лазерный тир
- Датчик света на фоторезисторе
- Терменвокс - электромузыкальный инструмент
- Ночной светильник
- Белый свет, включаемый кнопкой
- Изменение цвета RGB светодиода потенциометром
- Гирлянда с RGB светодиодами
- Концентрические окружности, которые должны увеличиваться в диаметре, двигаясь по светодиодной матрице из центра
- Секундомер на светодиодной матрице
- Двигающаяся стрелка по светодиодной матрице
- Создание метеостанции с ЖК дисплеем
- Тестер батареек
- Создание электронных часов на ЖК дисплее

III. Один из выбранных вами проектов необходимо выполнить (можно использовать информационные ресурсы) и написать руководство к его реализации (отчет с указанием

- комплектующих,
- спецификаций компонент,
- электрической схемы,
- примерной программы скетча.

IV. Напишите инструкцию по технике безопасности для спроектированного вами устройства, используя инструкцию на сайте кафедры как шаблон <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>.

ПРАКТИЧЕСКАЯ РАБОТА № 2.

Подбор заданий и проектирование установок с использованием фоторезистора как датчика освещенности

Спецификация фоторезистора (LDR GL5528 Datasheet) : <https://www.alldatasheet.com/datasheet-pdf/pdf/1131893/ETC2/GL5528.html>

ОБЩИЕ СВЕДЕНИЯ

ВНИМАНИЕ!!! Работа с электричеством даже при низком напряжении может быть опасной - внимательно следуйте схемам подключения и инструкциям и всегда обращайтесь за советом к преподавателю, если вы в чем-то не уверены! При выполнении каждой лабораторной и практической работы необходимо соблюдать инструкцию по технике безопасности, которая размещена на сайте кафедры <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>

Фоторезистор

Фоторезисторы, также известные как светозависимые резисторы (LDR) или фотоэлементы, представляют собой недорогие переменные резисторы, сопротивление которых изменяется в зависимости от количества света, попадающего на его поверхность. В темных условиях сопротивление высокое; в светлых условиях сопротивление ниже.



Из-за неточной природы фоторезисторов, они не подходят для измерения точных уровней света, но способны обнаруживать изменения. Они могут использоваться для реагирования на такие события, как переход от дневного к ночному (и наоборот) режиму для домашней автоматизации и садоводства, и часто используются для управления уличным освещением. Фоторезисторы, размещенные внутри корпуса, могут обнаруживать, когда он открывается, или обнаруживать присутствие объектов, которые блокируют датчик.

Комплектующие

Для демонстрации работы и подключения фоторезистора понадобятся:

- [Arduino Uno](#) или Arduino-совместимая плата с аналоговыми входами.

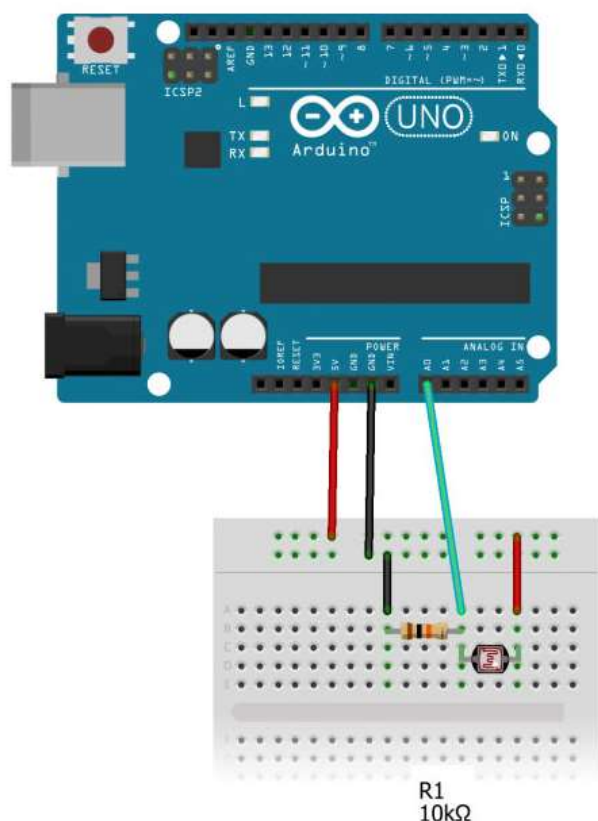
- [Arduino IDE](#) (интегрированная среда разработки).
- Фоторезистор (фотоэлемент)
- Один резистор 10 кОм.
- Один потенциометр (переменный резистор).
- Макетная плата и соединительные провода.

Фоторезистор представляет собой электронный компонент, удельное сопротивление которого изменяется в зависимости от количества получаемого света (сопротивление уменьшается при воздействии света). Фоторезистор выполнен из сульфида кадмия, полупроводник.

Когда фотоны попадают на ленты, электроны могут проходить через полупроводник. Основное использование фоторезистора - измерение силы света (камера, системы обнаружения и т.д.).

Схемы соединения

- Первый вывод -> 5 В
- Второй вывод -> A0 (сопротивление подключено к заземлению и ко второму выводу фоторезистора).



Фоторезистор и резистор 10 кОм питаются от источника питания 5 В Arduino и образуют делитель потенциала, который защищает Arduino от коротких замыканий и гарантирует, что по крайней мере какое-то сопротивление всегда присутствует на линии.

Провод от этой схемы соединен с аналоговым входом 0 на Arduino. Резисторы понижают напряжение, проходящее через них, и поэтому для считывания изменений в освещении этой цепи вы можете использовать аналого-цифровые преобразователи (АЦП) Arduino для измерения уровня напряжения на входе. АЦП преобразуют аналоговое значение в целое число в диапазоне от 0 до 1023.

Когда фоторезистор подвергается воздействию света, его сопротивление уменьшается, и поэтому показания напряжения будут выше. Когда свет блокируется, сопротивление фоторезистора увеличивается, и поэтому показания напряжения будут ниже.

Фоторезистор представляет собой простой пассивный компонент с двумя клеммами и не имеет полярности.

Коды проекта

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  int value = analogRead(A0);
  Serial.println("Analog value : ");
  Serial.println(value);
  delay(250);
}
```

Пояснения

В «void setup» мы инициализируем последовательный монитор:

```
void setup() {
  Serial.begin(9600);
}
```

Затем мы читаем аналоговое значение, поступающее от фоторезистора, и определяем его как value («значение»):

```
void loop(){
  int value = analogRead(A0);
```

И мы записываем значение на последовательном мониторе:

```
Serial.println("Analog Value :");
Serial.println(value);
delay(250);
}
```

Обнаружение изменений

На панели инструментов нажмите кнопку «Последовательный монитор» (**Serial Monitor**).

Точные значения, выводимые на последовательном мониторе в скетче выше, будут различаться в зависимости от нескольких факторов:

- Блок питания от Arduino. В частности, при питании от USB-кабеля обычно 5 В блока питания Arduino немного меньше этого идеала;
- Минимальное и максимальное значения сопротивления используемого фоторезистора;
- Точность резистора;

- Конструкция макета и используемых проводов - они имеют небольшие уровни сопротивления, которые могут повлиять на АЦП;
- И количество окружающего света в комнате.

Задания

I. По описанию, которое приведено выше (ОБЩИЕ СВЕДЕНИЯ), выполните предложенный проект (подключение и работа с фоторезистором).

II. Выберите из представленного перечня те проекты, в которых может быть использован **фоторезистор**:

- Лазерный счетчик
- Определить концентрацию раствора сахара
- Лазерную систему безопасности
- Лазерная связь кодом Морзе
- Лазерный тир
- Датчик света на фоторезисторе
- Терменвокс - электромузыкальный инструмент
- Ночной светильник
- Белый свет, включаемый кнопкой
- Изменение цвета RGB светодиода потенциометром
- Гирлянда с RGB светодиодами
- Концентрические окружности, которые должны увеличиваться в диаметре, двигаясь по светодиодной матрице из центра
 - Секундомер на светодиодной матрице
 - Двигающаяся стрелка по светодиодной матрице
 - Создание метеостанции с ЖК дисплеем
 - Тестер батареек
 - Создание электронных часов на ЖК дисплее

III. Один из выбранных вами проектов необходимо выполнить и написать руководство к его реализации (отчет) с указанием

- комплектующих,
- спецификаций компонент,
- электрической схемы,
- примерной программы скетча.

IV. Напишите инструкцию по технике безопасности для спроектированного вами устройства, используя инструкцию на сайте кафедры как шаблон <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>.

ПРАКТИЧЕСКАЯ РАБОТА № 3.

Подбор заданий и проектирование установок с использованием RGB-светодиода и матрицы светодиодов

Спецификация полноцветного (трехцветного) светодиода (RGB module KY-016 datasheet): <https://datasheetspdf.com/pdf-file/1402027/Joy-IT/KY-016/1>

Спецификация матрицы светодиодов (TZT 8x8 LED Matrix datasheet): https://components101.com/sites/default/files/component_datasheet/LED%20Matrix%20Datasheet.pdf

Спецификация сдвигового регистра - микросхемы для увеличения количества цифровых пинов (shift register 74HC595 datasheets): <https://www.diodes.com/assets/Datasheets/74HC595.pdf>

ОБЩИЕ СВЕДЕНИЯ

ВНИМАНИЕ!!! Работа с электричеством даже при низком напряжении может быть опасной - внимательно следуйте схемам подключения и инструкциям и всегда обращайтесь за советом к преподавателю, если вы в чем-то не уверены! При выполнении каждой лабораторной и практической работы необходимо соблюдать инструкцию по технике безопасности, которая размещена на сайте кафедры <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>

Устройство и назначение RGB светодиода

Для отображения всей палитры оттенков вполне достаточно три цвета, используя RGB синтез (Red — красный, Green — зеленый, Blue — синий). RGB палитра используется не только в графических редакторах, но и в сайтостроении. Смешивая цвета в разной пропорции можно получить практически любой цвет. Преимущества RGB светодиодов в простоте конструкции, небольших габаритах и высоком КПД светоотдачи.

RGB светодиоды объединяют три кристалла разных цветов в одном корпусе. RGB LED имеет 4 вывода — один общий (анод или катод имеет самый длинный вывод) и три цветовых вывода. К каждому цветовому выходу следует подключать резистор. Кроме того, модуль RGB LED Arduino может сразу монтироваться на плате и иметь встроенные резисторы.

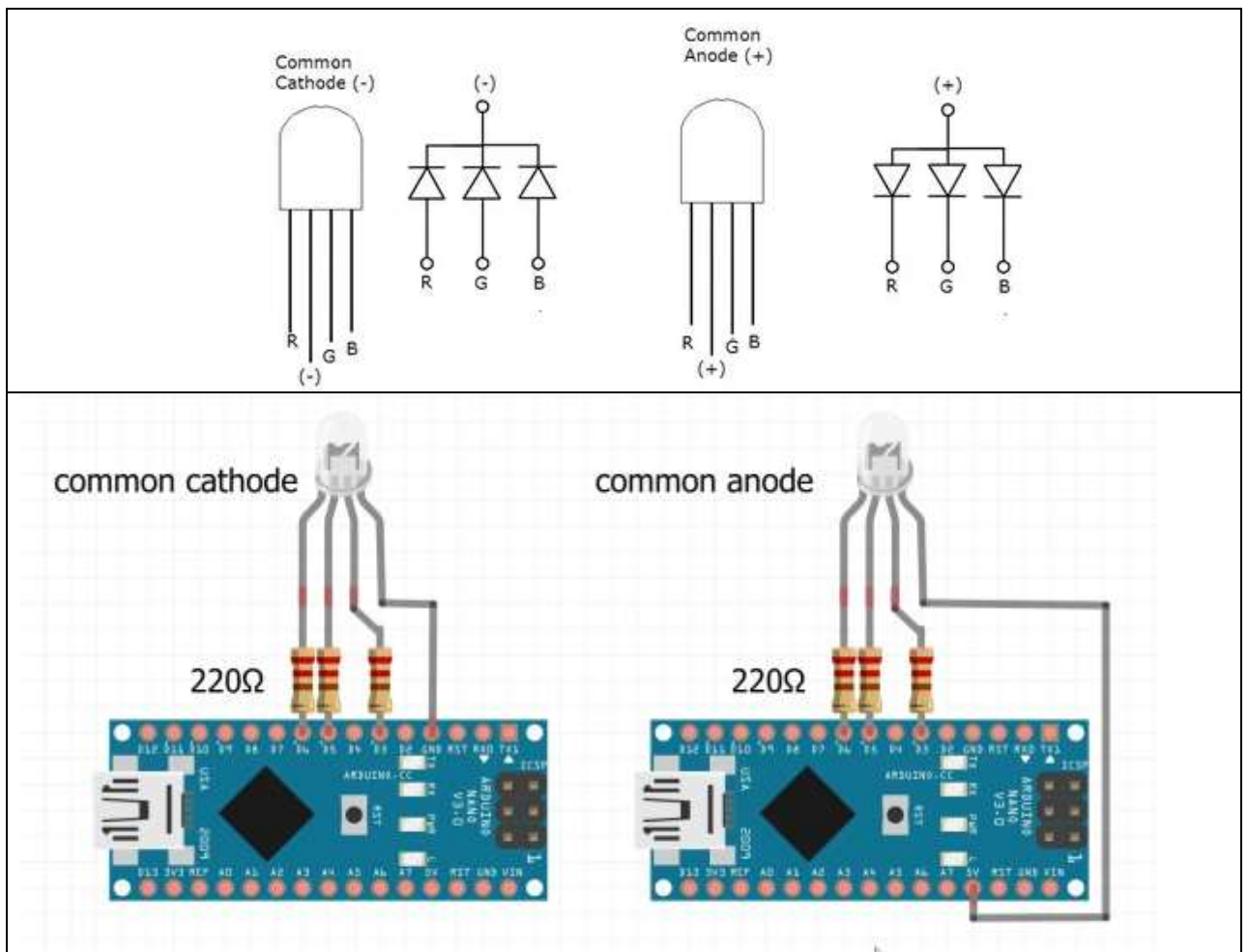


Рисунок 1 - Распиновка RGB светодиода

Для многих полноцветных (трехцветных) светодиодов необходимы светорассеиватели, иначе будут видны составляющие цвета. Далее подключим трехцветный светодиод к Ардуино и заставим его сначала мигать разными цветами, а затем плавно переливаться разными цветами с помощью «широтно импульсной модуляции».

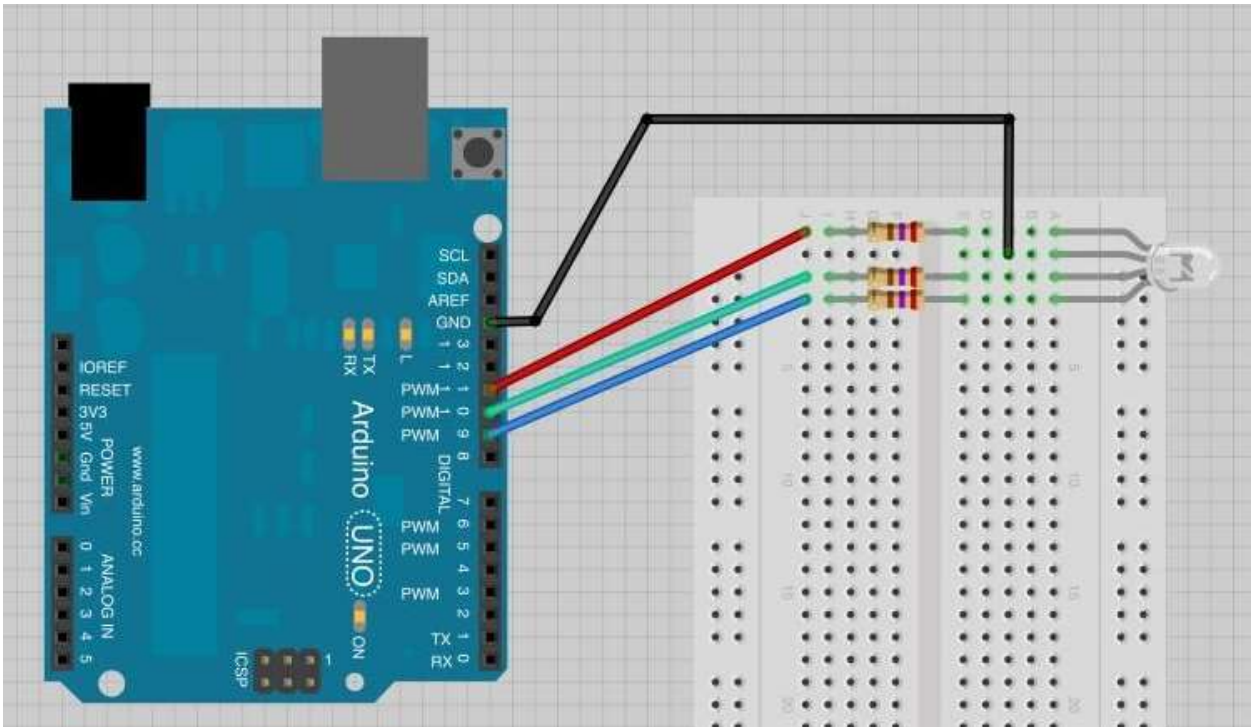


Рисунок 2 - Схема подключения RGB LED к Ардуино на макетной плате

Модуль «RGB светодиода» можно подключить напрямую к плате, без проводов и макетной платы. Подключите модуль с полноцветным RGB светодиодом к следующим пинам: Минус — GND, В — Pin13, G — Pin12, R — Pin11.

Скетч для мигания RGB светодиодов на Ардуино

```
#define RED 11 // присваиваем имя RED для пина 11
#define GRN 12 // присваиваем имя GRN для пина 12
#define BLU 13 // присваиваем имя BLU для пина 13

void setup() {
  pinMode(RED, OUTPUT); // используем Pin11 для вывода
  pinMode(GRN, OUTPUT); // используем Pin12 для вывода
  pinMode(BLU, OUTPUT); // используем Pin13 для вывода
}

void loop() {

  digitalWrite(RED, HIGH); // включаем красный свет
  digitalWrite(GRN, LOW);
  digitalWrite(BLU, LOW);

  delay(1000); // устанавливаем паузу для эффекта

  digitalWrite(RED, LOW);
  digitalWrite(GRN, HIGH); // включаем зеленый свет
  digitalWrite(BLU, LOW);

  delay(1000); // устанавливаем паузу для эффекта
```

```

digitalWrite (RED, LOW);
digitalWrite (GRN, LOW);
digitalWrite (BLU, HIGH); // включаем синий свет

delay(1000); // устанавливаем паузу для эффекта
}

```

Плавное управление RGB светодиодом

Управление RGB-светодиодом на Arduino можно сделать плавным, используя аналоговые выходы с «ШИМ». Для этого ножки светодиода необходимо подключить к аналоговым выходам, например, к пинам 11, 10 и 9. И подавать на аналоговые выходы микроконтроллера различные значения [ШИМ](#) (PWM), для этого воспользуемся циклом for, с помощью которого можно повторять нужные команды в программе.

Скетч для плавного мигания RGB светодиода

```

#define RED 11 // присваиваем имя RED для пина 11
#define GRN 10 // присваиваем имя GRN для пина 10
#define BLU 9 // присваиваем имя BLU для пина 9

void setup() {
  pinMode(RED, OUTPUT); // используем Pin11 для вывода
  pinMode(GRN, OUTPUT); // используем Pin10 для вывода
  pinMode(BLU, OUTPUT); // используем Pin9 для вывода
}

void loop() {
  // плавное включение/выключение красного цвета
  for (int i = 0; i <= 255; i++) {
    analogWrite(RED, i);
    delay(2);
  }
  for (int i = 255; i >= 0; i--) {
    analogWrite(RED, i);
    delay(2);
  }

  // плавное включение/выключение зеленого цвета
  for (int i = 0; i <= 255; i++) {
    analogWrite(GRN, i);
    delay(2);
  }
  for (int i = 255; i >= 0; i--) {
    analogWrite(GRN, i);
    delay(2);
  }

  // плавное включение/выключение синего цвета
  for (int i = 0; i <= 255; i++) {
    analogWrite(BLU, i);
    delay(2);
  }
}

```

```
    for (int i = 255; i >= 0; i--) {  
        analogWrite(BLU, i);  
        delay(2);  
    }  
}
```

Аналоговые выходы на Ардуино используют «широко импульсную модуляцию» для получения различной силы тока. Мы можем подавать на все три цветовых входа на светодиоде различное значение ШИМ-сигнала в диапазоне от 0 до 255, что позволит нам получить на RGB LED Arduino практически любой оттенок света.

Матричный индикатор Arduino

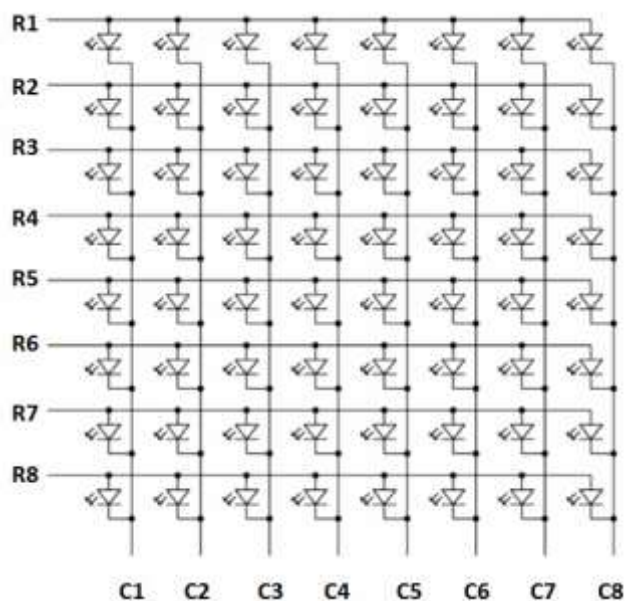
Сегментные индикаторы (шкалы или цифры) состоят из отдельных светодиодов, соединенных вместе. Например, у группы светодиодов могут быть соединены все катоды. Такой индикатор имеет приписку «с общим катодом», в противном случае — «с общим анодом».

Сетка светодиодов называется матричным индикатором или светодиодной матрицей. Разрешение матричного индикатора — это количество точек по горизонтали и вертикали. Например, самые распространенные индикаторы имеют разрешение 8×8 точек.

Если требуется светодиодная матрица с большим разрешением, то её просто-напросто составляют из нескольких 8×8 индикаторов. Как это делать, мы увидим позже. А пока разберемся как соединяются все 64 светодиода внутри матрицы.

Конечно, можно бы было как и в случае семисегментного индикатора соединить все светодиоды общим катодом или анодом. В этом случае нам бы потребовалось либо 64 вывода контроллера, либо 8 сдвиговых регистров. Оба варианта весьма расточительны.

Более правильный вариант — объединить светодиоды в группы по 8 штук с общим катодом. Пусть это будут столбцы матрицы. Затем, параллельные светодиоды в этих столбцах объединить снова в группы по 8 штук уже с общим анодом. Получится вот такая схема:



Светодиодная матрица на Ардуино, схема

Предположим, стоит задача зажечь светодиод R6C3. Для этого нам потребуется подать высокий уровень сигнала на вывод R6, а вывод C3 соединить с землей.

| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|----|----|----|----|----|----|----|----|----|
| R1 | | | | | | | | |
| R2 | | | | | | | | |
| R3 | | | | | | | | |
| R4 | | | | | | | | |
| R5 | | | | | | | | |
| R6 | | | | | | | | |
| R7 | | | | | | | | |
| R8 | | | | | | | | |

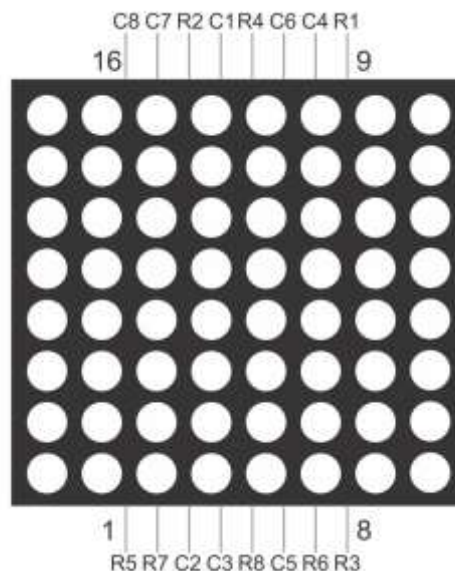
| | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 |
|----|----|----|----|----|----|----|----|----|
| R1 | | | | | | | | |
| R2 | | | | | | | | |
| R3 | | | | | | | | |
| R4 | | | | | | | | |
| R5 | | | | | | | | |
| R6 | | | | | | | | |
| R7 | | | | | | | | |
| R8 | | | | | | | | |

Не выключая эту точку, попробуем зажечь другую — R3C7. Положительный контакт питания соединим с R3 и землю с C7. Но в таком случае строки R6 и R3 будут пересекаться с колонками C3 и C7 не в двух, а в четырех местах. Следовательно и зажжется не две, а четыре точки.

Это решается с помощью динамической индикации. Если мы будем включать точки R6C3 и R3C7 по-очереди очень быстро, то сможем использовать персистентность зрения — способность интерпретировать быстро сменяющиеся изображения как одно целое.

Светодиодная матрица и сдвиговые регистры

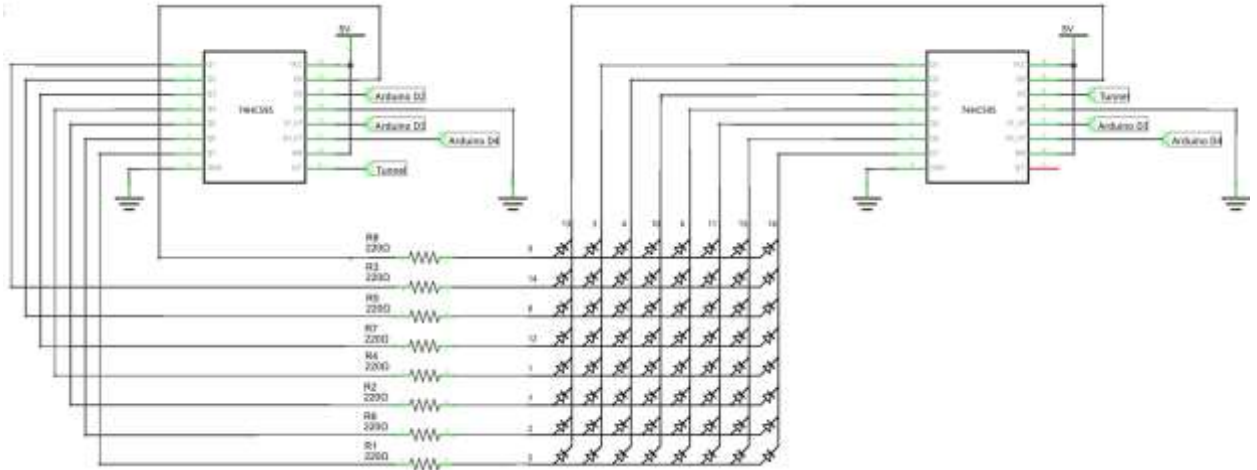
В светодиодной матрице 8×8 красного свечения нумерация выводов начинается с нижнего левого угла. При этом, нумерация ног 1-16 не связана никакой логикой с нумерацией колонок и строк C и R.



Распиновка светодиодной матрицы 8x8

Используя динамическую индикацию и 8-битные сдвиговые регистры в схеме можно управлять матричным индикатором. Один регистр подключим к выводам индикатора, отвечающим за колонки, а второй к выводам строк.

Принципиальная схема



Принципиальная схема светодиодная матрица Ардуино

Высветим на индикаторе смайлик. Как уже было сказано, для вывода изображения на матрицу воспользуемся динамической индикацией. А именно, будем высвечивать нашу картинку построчно. Сначала зажжем нужные колонки в самой верхней строке, затем во второй, в третьей, и так все 8 строк.

За колонки у нас будет отвечать первый сдвиговый регистр, а за строки второй. Следовательно, вывод строки будет состоять из двух последовательных записей в регистр: сначала передаем код строки, затем код точек в этой строке.

В этой программе мы также воспользуемся ускоренной версией функции `digitalWrite`. Это необходимо для того, чтобы процесс динамической индикации проходил очень быстро. В противном случае, мы увидим заметное мерцание матрицы.

Исходный код

```
const byte data_pin = PD2;
const byte st_pin = PD3;
const byte sh_pin = PD4;

unsigned long tm, next_flick;
const unsigned int to_flick = 500;

byte line = 0;

const byte data[8] = {
    0b00111100,
    0b01000010,
    0b10100101,
    0b10000001,
    0b10100101,
    0b10011001,
    0b01000010,
    0b00111100
```

```

};

void latchOn(){
    digitalWriteFast(st_pin, HIGH);
    digitalWriteFast(st_pin, LOW);
}

void fill( byte d ){
    for(char i=0; i<8; i++){
        digitalWriteFast(sh_pin, LOW);
        digitalWriteFast(data_pin, d & (1<<i));
        digitalWriteFast(sh_pin, HIGH);
    }
}

void setPinFast(byte pin){
    DDRD |= _BV(pin);
}

void digitalWriteFast(byte pin, byte sig){
    if( sig )
        PORTD |= _BV(pin);
    else
        PORTD &= ~_BV(pin);
}

void setup() {
    setPinFast(data_pin);
    setPinFast(st_pin);
    setPinFast(sh_pin);
    setPinFast(oe_pin);
}

void loop() {
    tm = micros();
    if( tm > next_flick ){
        next_flick = tm + to_flick;
        line++;
        if( line == 8 )
            line = 0;
        // передаем код строки
        fill( ~(1<<(7-line)) );
        // зажигаем точки в строке № line
        fill( data[line] );
        // открываем защелку
        latchOn();
    }
}

```

Массив data хранит восемь строк картинки. Для экономии памяти мы записали каждую комбинацию точек в бинарном виде.

Функция latchOn открывает защелку регистра. Это нужно делать только после заполнения обоих сдвиговых регистров.

Светодиодная матрица с разрешением 8×8 подойдет для отображения двух цифр или простого символа. Если требуется вывести на индикатор какое-то более или менее полезное изображение, необходимо объединить матрицы. Делается это с помощью добавления новых сдвиговых регистров как по вертикали, так и по горизонтали.

Задания

I. По описанию, которое приведено выше (ОБЩИЕ СВЕДЕНИЯ), выполните предложенные проекты (подключение и работа с RGB светодиодом и светодиодной матрицей).

II. Выберите из представленного перечня те проекты, в которых могут быть использованы **RGB светодиод и светодиодная матрица**:

- Лазерный счетчик
- Определить концентрацию раствора сахара
- Лазерную систему безопасности
- Лазерная связь кодом Морзе
- Лазерный тир
- Датчик света на фоторезисторе
- Терменвокс - электромузыкальный инструмент
- Ночной светильник
- Белый свет, включаемый кнопкой
- Изменение цвета RGB светодиода потенциометром
- Гирлянда с RGB светодиодами
- Концентрические окружности, которые должны увеличиваться в диаметре, двигаясь по светодиодной матрице из центра
- Секундомер на светодиодной матрице
- Двигающаяся стрелка по светодиодной матрице
- Создание метеостанции с ЖК дисплеем
- Тестер батареек
- Создание электронных часов на ЖК дисплее

III. Один из выбранных вами проектов необходимо выполнить и написать руководство к его реализации (отчет) с указанием

- комплектующих,
- спецификаций компонент,
- электрической схемы,
- примерной программы скетча.

IV. Напишите инструкцию по технике безопасности для спроектированного вами устройства, используя инструкцию на сайте кафедры как шаблон <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>.

ПРАКТИЧЕСКАЯ РАБОТА № 4.

Подбор заданий и проектирование установок с использованием LCD-дисплея

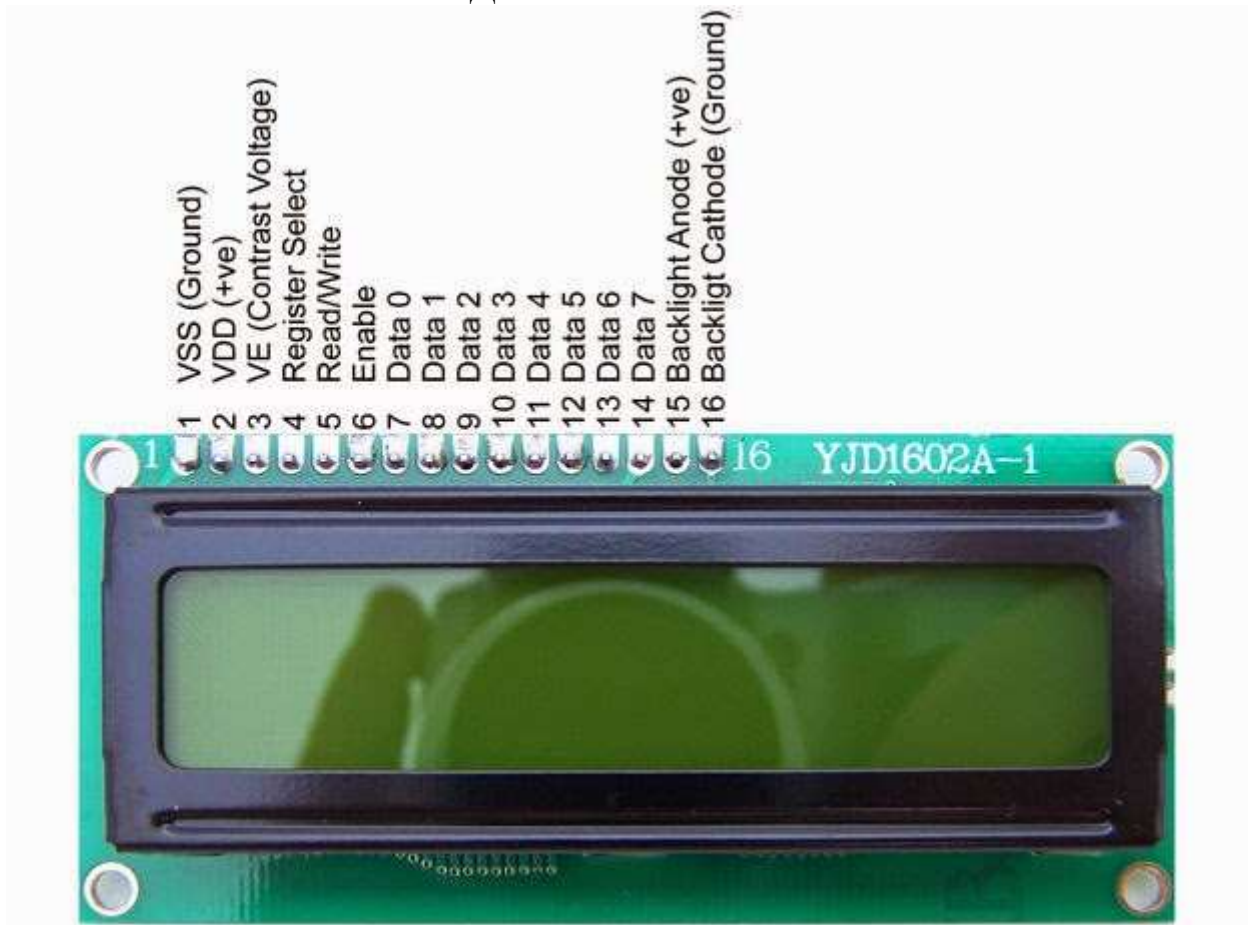
Спецификация ЖК дисплея (LCD 16x2 datasheet): <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>

Спецификация ЖК дисплея на русском языке: <http://wiki.amperka.ru/media/products/display-lcd-text-16x2:mt-16s2h-datasheet.pdf>

Спецификация модуля-посредника для уменьшения количества используемых пинов (I2C datasheet): <https://opencircuit.shop/resources/file/da88acc1702a90667728fcf4ac9c75c455475706466/I2C-LCD-interface.pdf>

ОБЩИЕ СВЕДЕНИЯ

Дисплей LCD 16x2



Каждый из выводов имеет свое назначение:

1. Земля GND;
2. Питание 5 В;
3. Установка контрастности монитора;
4. Команда, данные;

- 5. Записывание и чтение данных;
- 6. Enable;

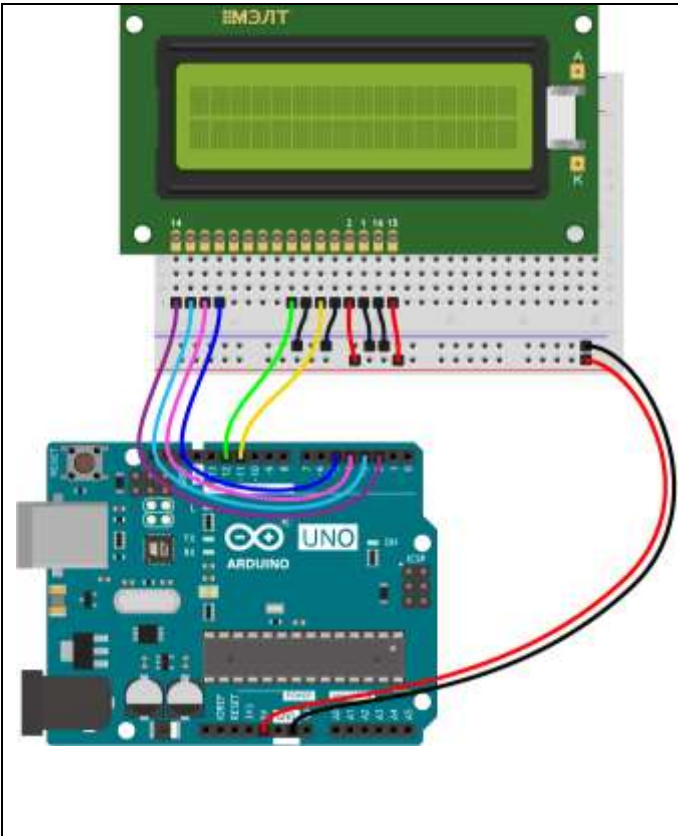
7-14. Линии данных;

- 15. Плюс подсветки;
- 16. Минус подсветки.

Технические характеристики дисплея:

- Символьный тип отображения, есть возможность загрузки символов;
- Светодиодная подсветка;
- Контроллер HD44780;
- Напряжение питания 5В;
- Формат 16x2 символов;
- Диапазон рабочих температур от -20С до +70С, диапазон температур хранения от -30С до +80 С;
- Угол обзора 180 градусов.

Подключение к Arduino



| Вывод | Обозначение | Пин Arduino Uno |
|-------|-------------|-----------------|
| 1 | GND | GND |
| 2 | VCC | 5V |
| 3 | VO | GND |
| 4 | RS | 11 |
| 5 | R/W | GND |
| 6 | E | 12 |
| 7 | DB0 | — |
| 8 | DB1 | — |
| 9 | DB2 | — |
| 10 | DB3 | — |
| 11 | DB4 | 5 |
| 12 | DB5 | 4 |
| 13 | DB6 | 3 |
| 14 | DB7 | 2 |
| 15 | VCC | 5V |
| 16 | GND | GND |

Для упрощения работы с LCD-дисплеем используйте встроенную библиотеку Liquid Crystal. В ней вы найдёте примеры кода с подробными комментариями.

Вывод текста

```

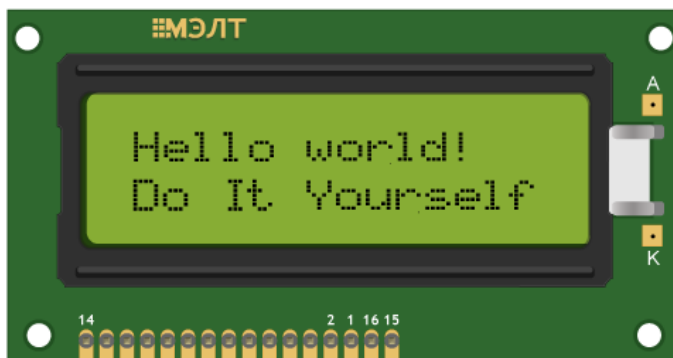
// подключаем стандартную библиотеку LiquidCrystal
#include <LiquidCrystal.h>

// инициализируем объект-экран, передаём использованные
// для подключения контакты на Arduino в порядке:
// RS, E, DB4, DB5, DB6, DB7
LiquidCrystal lcd(11, 12, 5, 4, 3, 2);

void setup() {
  // устанавливаем размер (количество столбцов и строк)
  // экрана
  lcd.begin(16, 2);
  // печатаем первую строку
  lcd.print("Hello world");
  // устанавливаем курсор в колонку 0, строку 1
  // на самом деле это вторая строка, т.к. нумерация
  // начинается с нуля
  lcd.setCursor(0, 1);
  // печатаем вторую строку
  lcd.print("Do It Yourself");
}

void loop() {
}

```



Кириллица

Существует два способа вывода кириллицы на текстовые дисплеи:

- с помощью встроенной таблицы знакогенератора
- с помощью библиотеки LiquidCrystalRus

Рассмотрим оба способа более подробно.

Дисплейный модуль хранит в памяти две страницы знакогенератора, которые состоят из различных символов и букв.

Таблица знакогенератора

Страницы «0» и «1» встроенного знакогенератора

| | Старшая цифра кода символа (в шестнадцатеричном виде) | | | | | | | | | | | | | | | | Старшая цифра кода символа (в шестнадцатеричном виде) | | | | | | | | | | | | | | | | | | |
|---|---|-----|----|---|---|---|---|---|-----|---|---|---|---|----|---|---|---|---|----|---|---|---|---|---|-----|-----|---|---|---|---|---|---|---|---|--|
| | Младшая цифра кода символа (в шестнадцатеричном виде) | | | | | | | | | | | | | | | | Младшая цифра кода символа (в шестнадцатеричном виде) | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | | | |
| 0 | x | ... | | 0 | @ | P | ' | P | ... | ± | Б | Ю | Ч | . | Д | Ж | x | ¼ | | 0 | @ | P | ' | p | i | ▶ | ■ | ° | А | Р | а | р | | | |
| 1 | x | !!! | ! | 1 | A | Q | a | q | ! | ≡ | Г | Я | Ш | Ц | Щ | Ъ | x | ½ | ! | 1 | A | Q | a | q | 1 | ◀ | ± | Б | С | б | с | | | | |
| 2 | x | ÷ | " | 2 | B | R | b | r | ! | ± | ё | б | ъ | » | Щ | Ъ | x | ¾ | " | 2 | B | R | b | r | ▶ | ± | ё | ± | В | Т | в | т | | | |
| 3 | x | → | # | 3 | C | S | s | s | ! | ◊ | Ж | В | ы | !! | Д | Ж | x | ¾ | # | 3 | C | S | s | s | ▶ | ± | ё | ◊ | Г | У | г | у | | | |
| 4 | x | ← | \$ | 4 | D | T | t | t | ! | ✓ | З | Р | ь | Ъ | Ф | И | x | ÷ | \$ | 4 | D | T | t | t | !! | ... | Ю | » | Д | Ф | д | ф | | | |
| 5 | x | \ | % | 5 | E | U | u | u | ! | ! | И | ё | э | Х | Ц | Ъ | x | ≡ | % | 5 | E | U | u | u | ... | ! | ¥ | " | Е | Х | е | х | | | |
| 6 | x | г | & | 6 | F | V | v | v | ! | ! | И | ж | У | Щ | Ъ | Ж | x | г | & | 6 | F | V | v | v | ↑ | ! | о | Ч | Ж | Ц | ж | ц | | | |
| 7 | x | н | ' | 7 | G | W | w | w | ! | ! | Л | э | Я | И | ' | Е | x | ✓ | ' | 7 | G | W | w | w | ↓ | ! | § | f | З | Ч | з | ч | | | |
| 8 | x | б | 0 | < | 8 | H | h | h | x | Р | з | П | и | « | И | ' | Е | x | Р | * | < | 8 | H | h | h | x | € | у | ё | И | Ш | и | ш | | |
| 9 | x | µ | 0 | > | 9 | I | Y | y | y | т | ° | У | й | » | ↑ | ~ | Э | x | ° | т | у | > | 9 | I | Y | y | т | ° | У | й | » | ↑ | ~ | Э | |
| A | x | ú | ≤ | * | : | J | Z | z | z | - | € | Ф | к | » | ↓ | é | Э | x | ◊ | ≤ | * | : | J | Z | z | z | ◊ | э | Е | е | К | ь | к | ъ | |
| B | x | Ю | ≥ | + | ; | K | [| k | [| ◊ | Ч | л | " | Н | Ф | К | x | ◊ | ≥ | + | ; | K | [| k | [| F | f | « | » | Л | М | л | м | | |
| C | x | ї | Ј | , | < | L | φ | l | φ |) | Ш | М | ъ | Н | Ў | Х | x | ◊ | ◊ | , | < | L | \ | l | l | К | к | « | » | М | ь | м | ь | | |
| D | x | і | ¥ | - | = | M |] | m |] | т | ь | н | Ъ | Н | * | Э | x | ◊ | р | - | = | M |] | m |] | т | ь | н | Ъ | Н | Э | н | э | | |
| E | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| F | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | О | Ю | о | ю | |
| | x | € | ≠ | . | > | N | ^ | n | ^ | ◊ | Ы | п | ф | Ъ | » | Я | x | ◊ | € | ≠ | . | > | N | ^ | n | ^ | у | у | ◊ | » | | | | | |

Используя полученную информацию выведем на дисплей сообщение «Привет»:

```
// подключаем стандартную библиотеку LiquidCrystal
#include <LiquidCrystal.h>

// инициализируем объект-экран, передаём использованные
// для подключения контакты на Arduino в порядке:
// RS, E, DB4, DB5, DB6, DB7
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // устанавливаем размер (количество столбцов и строк)
  // экрана
  lcd.begin(16, 2);
  // устанавливаем курсор в колонку 5, строку 0
  // на самом деле это первая строка, т.к. нумерация
  // начинается с нуля
  lcd.setCursor(5, 0);
  // печатаем первую строку
  lcd.print("\xA8" "p" "\xB8\xB3" "e\xBF");
  // устанавливаем курсор в колонку 3, строку 1
  // на самом деле это вторая строка, т.к. нумерация
  // начинается с нуля
}
void loop() {
}
```

Полную таблицу символов с кодами можно найти в документации к экрану.

Переключение страниц знакогенератора

Дисплейный модуль хранит в памяти две страницы знакогенератора. По умолчанию установлена нулевая страница. Для переключения между страницами используйте методы:

```
// переключение с нулевой страницы на первую
lcd.command(0b101010);
// переключение с первой страницы на нулевую
lcd.command(0b101000);
```

Дисплей не может одновременно отображать символы разных страниц.

Использование библиотеки LiquidCrystalRus

Совсем не обязательно мучаться со знакогенератором, чтобы вывести русский символ. Для решения проблемы скачайте и установите библиотеку LiquidCrystalRus.

Это копия оригинальной библиотеки LiquidCrystal с добавлением русского языка. Добавленный в библиотеку код трансформирует русские символы UTF8 в правильные коды для текстового экрана.

В качестве примера выведем «Привет» на дисплей.

```
// подключаем библиотеку LiquidCrystalRus
#include <LiquidCrystalRus.h>

// инициализируем объект-экран, передаём использованные
// для подключения контакты на Arduino в порядке:
// RS, E, DB4, DB5, DB6, DB7
LiquidCrystalRus lcd(11, 12, 5, 4, 3, 2);

void setup() {
  // устанавливаем размер (количество столбцов и строк)
  // экрана
  lcd.begin(16, 2);
  // устанавливаем курсор в колонку 5, строку 0
  // на самом деле это первая строка, т.к. нумерация
  // начинается с нуля
  lcd.setCursor(5, 0);
  // печатаем первую строку
  lcd.print("Привет");
}

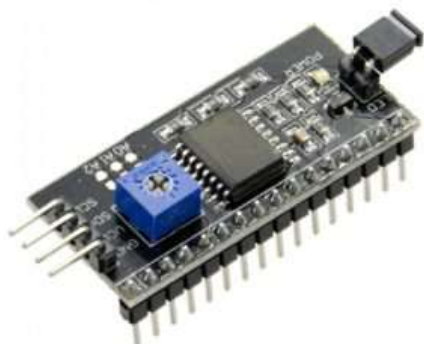
void loop() {
}
```

Подключение ЖК экрана к Ардуино по I2C Описание протокола I2C

I2C / IC(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.

Модуль I2C для LCD 1602 Arduino



Самый быстрый и удобный способ использования i2c дисплея в ардуино – это покупка готового экрана со встроенной поддержкой протокола. Но таких экранов не очень много и стоят они не дешево. А вот разнообразных стандартных экранов выпущено уже огромное количество. Поэтому самым

доступным и популярным сегодня вариантом является покупка и использование отдельного I2C модуля – переходника.

С одной стороны модуля мы видим выводы I2C – земля, питание и 2 для передачи данных. С другой стороны переходника видим разъемы внешнего питания. И, естественно, на плате есть множество ножек, с помощью которых модуль припаивается к стандартным выводам экрана.



Использование LCD 1602 модуля с уже припаянным переходником максимально упрощено. Если вы купили отдельный переходник, нужно будет предварительно припаять его к модулю.

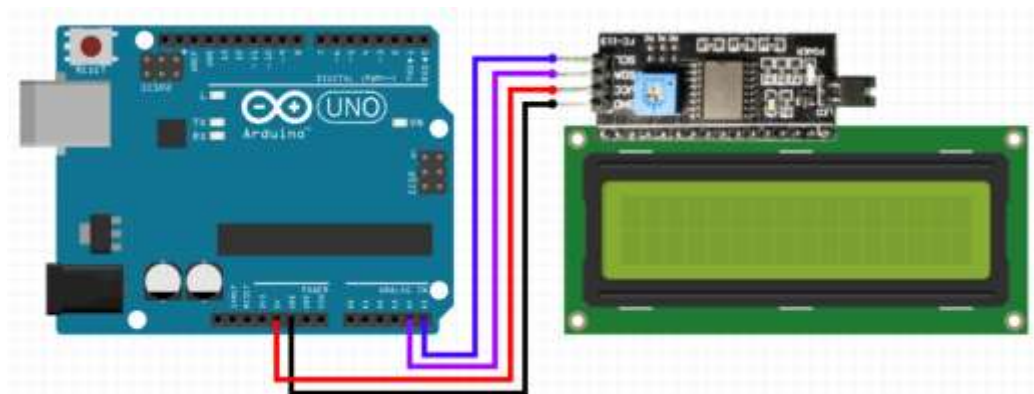
Жидкокристаллический монитор с поддержкой i2c подключается к плате при помощи четырех проводов – два провода для данных (к аналоговым входам), два провода для питания.

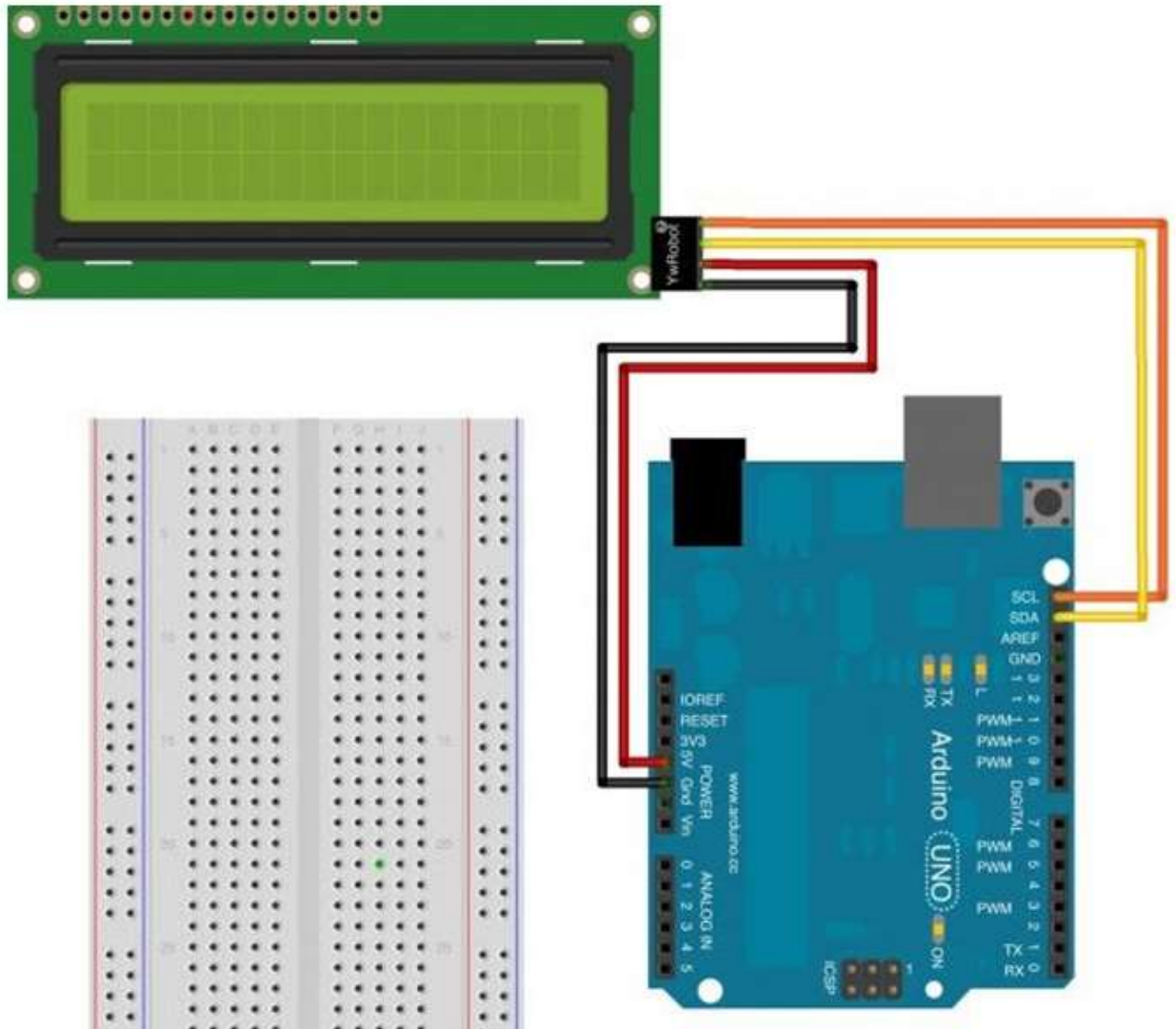
Вывод GND подключается к GND на плате.

Вывод VCC – на 5V.

SCL подключается к пину A5.

SDA подключается к пину A4.





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.

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 дисплея нужно дополнительно установить библиотеку `LiquidCrystal_I2C.h`

После подключения к скетчу всех необходимых библиотек мы создаем объект и можем использовать все его функции. Для тестирования давайте загрузим следующий стандартный скетч из примера.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h> // Подключение библиотеки
```

```

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

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("LCD+I2C"); // Набор текста на второй
строке
}
void loop()
{
  lcd.noDisplay(); // выключаем и включаем надпись на
дисплее
  delay(1000);
  lcd.display(); // включаем и включаем надпись на дисплее
  delay(1000);
}

```

Описание функций и методов библиотеки LiquidCrystal_I2C:

`home()` и `clear()` – первая функция позволяет вернуть курсор в начало экрана, вторая тоже, но при этом удаляет все, что было на мониторе до этого.

`write(ch)` – позволяет вывести одиночный символ `ch` на экран.

`cursor()` и `noCursor()` – показывает/скрывает курсор на экране.

`blink()` и `noBlink()` – курсор мигает/не мигает (если до этого было включено его отображение).

`display()` и `noDisplay()` – позволяет подключить/отключить дисплей.

`scrollDisplayLeft()` и `scrollDisplayRight()` – прокручивает экран на один знак влево/вправо.

`autoscroll()` и `noAutoscroll()` – позволяет включить/выключить режим автопрокручивания. В этом режиме каждый новый символ записывается в одном и том же месте, вытесняя ранее написанное на экране.

`leftToRight()` и `rightToLeft()` – Установка направление выводимого текста – слева направо или справа налево.

createChar(ch, bitmap) – создает символ с кодом ch (0 – 7), используя массив битовых масок bitmap для создания черных и белых точек.

Задания

ВНИМАНИЕ!!! Работа с электричеством даже при низком напряжении может быть опасной - внимательно следуйте схемам подключения и инструкциям и всегда обращайтесь за советом к преподавателю, если вы в чем-то не уверены! При выполнении каждой лабораторной и практической работы необходимо соблюдать инструкцию по технике безопасности, которая размещена на сайте кафедры <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>

I. По описанию, которое приведено выше (ОБЩИЕ СВЕДЕНИЯ), выполните предложенные проекты (подключение и работа с ЖК дисплеем).

II. Выберите из представленного перечня те проекты, в которых может быть использован **ЖК дисплей**:

- Лазерный счетчик
- Определить концентрацию раствора сахара
- Лазерную систему безопасности
- Лазерная связь кодом Морзе
- Лазерный тир
- Датчик света на фоторезисторе
- Терменвокс - электромузыкальный инструмент
- Ночной светильник
- Белый светодиод, включаемый кнопкой
- Изменение цвета RGB светодиода потенциометром
- Гирлянда с RGB светодиодами
- Концентрические окружности, которые должны увеличиваться в диаметре, двигаясь по светодиодной матрице из центра
 - Секундомер на светодиодной матрице
 - Двигающаяся стрелка по светодиодной матрице
 - Создание метеостанции с ЖК дисплеем
 - Тестер батареек
 - Создание электронных часов на ЖК дисплее

III. Один из выбранных вами проектов необходимо выполнить и написать руководство к его реализации (отчет) с указанием

- комплектующих,
- спецификаций компонент,
- электрической схемы,
- примерной программы скетча.

IV. Напишите инструкцию по технике безопасности для спроектированного вами устройства, используя инструкцию на сайте кафедры как шаблон <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>.

Список используемых источников

1. Банци М. Arduino для начинающих волшебников. – М.: Рид Групп, 2012. – 129 с.
2. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino. – Спб.: БХВ-Петербург, 2012. – 256 с.
3. Ревич Ю.В. Занимательная электроника. – Спб.: БХВ-Петербург, 2015. – 576 с.
4. Петин В.А. Проекты с использованием контроллера Arduino. – Спб.: Петербург, 2015.– 464 с.
5. Карвинен Т., Карвинен К., Валтокари В. Делаем сенсоры: проекты сенсорных устройств на базе Arduino и Raspberry Pi. – М.: ООО «И.Д. Вильямс», 2015– 432 с.
6. <https://роботехника18.рф/лазерный-модуль-ардуино/>
7. <https://arduinoplus.ru/kak-ispolzovat-fotorezistor/>
8. <https://роботехника18.рф/rgb-светодиод-ардуино/>
9. <https://robotclass.ru/tutorials/arduino-led-matrix-8x8/>
10. <https://ingener-pto.ru/2019/12/12/kak-podkljuchit-jekran-k-arduino-uno/>