

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 07.02.2025 10:55:09

Уникальный программный ключ:

0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e245014a48511da56d089

## МИНОБРНАУКИ 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

« \_\_\_\_ » \_\_\_\_\_ 2025 г.

## ОПТОЭЛЕКТРОНИКА

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

Курск - 2025

УДК 53

Составители: А.В. Кузько, Е.В. Шельдешова, Е.А. Новиков, 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., Шельдешова Е.В., Новиков Е.А., Мыцких-Коробанов А.Ю. Курск, 2025. 66 с.: ил. 20. Библиогр.: с. 66.

Излагаются методические рекомендации по выполнению практических проектных работ, в которых предлагается обработка микроскопических изображений люминесцирующих наноразмерных материалов, освоение программного обеспечения по управлению твердотельным лазером, скрайбированием, резкой и лазерной абляцией, проектирование устройств оптоэлектроники 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 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. 3,9. Уч.-изд. л. 3,5. Тираж 30 экз. Заказ 30. Бесплатно.

Юго-Западный государственный университет.

305040 Курск, ул. 50 лет Октября, 94.

## ПРАКТИЧЕСКАЯ РАБОТА № 1

### ОБРАБОТКА МИКРОСКОПИЧЕСКИХ ИЗОБРАЖЕНИЙ ЛЮМИНЕСЦИРУЮЩИХ НАНОРАЗМЕРНЫХ МАТЕРИАЛОВ

#### Цель работы:

- 1) ознакомиться с приемами работы и возможностями программы обработки данных сканирующей зондовой микроскопии Gwyddion;
- 2) выполнить обработку АСМ изображений люминесцирующих наночастиц на кремниевой подложке;
- 3) получить распределение по размерам частиц люминесцирующих коллоидных растворов.

**Приборы и принадлежности:** программное обеспечение Gwyddion, LibreOffice Calc; АСМ изображения люминесцирующих наночастиц, осажденных на кремниевой подложке (0,5 x 0,5 мкм), образец СЭМ-изображения.

#### *ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ*

**Gwyddion** - это модульная программа анализа данных СЗМ. В первую очередь она предназначена для анализа полей высот, полученных различными техниками сканирующей зондовой микроскопии (АСМ, МСМ, СТМ, СБОМ), но в общем случае её можно использовать для анализа любых полей высот или (монохромных) изображений, например, для анализа данных профилометрии или карт толщины, построенных картирующей спектрофотометрией. **Gwyddion** является свободным программным обеспечением с открытым исходным кодом, выпущенным под лицензией [GNU General Public License](https://www.gnu.org/licenses/gpl-3.0.html) (GNU GPL).

Основной идеей при разработке **Gwyddion** было создать модульную программу для анализа двумерных данных, которую можно легко расширять отдельными модулями без пересборки ядра. Более того, поскольку программа является свободной, это позволяет предоставить исходный код разработчикам и пользователям, что существенно облегчает дальнейшее её улучшение.

**Gwyddion** в настоящий момент можно использовать на операционных системах семейства Linux/Unix (включая Mac OS X)

и Microsoft Windows. Разработка возможна на всех поддерживаемых платформах. Для создания графического интерфейса пользователя использовались библиотеки [Gtk+](#), поэтому программа может быть легко портирована на любую систему, поддерживающую Gtk+.

Разработка **Gwyddion** сейчас поддерживается [Чешским институтом метрологии \(Czech Metrology Institute\)](#). Проект был начат как часть инициативы Nanomet (проходящей по программе Euromet) в августе 2004 года.

### *Установка Gwyddion*

Исходные коды и двоичные файлы **Gwyddion** можно загрузить со страницы download сайта проекта или напрямую со страницы загрузки [SourceForge.net](#). Порядок установки слегка меняется в зависимости от используемой операционной системы и различных методов установки, описанных в следующих разделах. **Gwyddion** требует или может использовать различные программные библиотеки. Если вы устанавливаете собранные пакеты это обычно не должно вас волновать, поскольку это задача сборщика пакета убедиться, что всё нужное доступно в системе. Однако, это является важным, если вы собираете **Gwyddion** из исходного кода.

Чтобы попробовать **Gwyddion** в деле вам, возможно, понадобятся образцы файлов **Gwyddion**. Они сохранены в родном формате **Gwyddion** и представляют собой типичные данные АСМ.

*Примечание:* Отмена установки

Если вы хотите удалить **Gwyddion**, перейдите в Пуск→Панель управления→Установка и удаление программ (Пуск→Панель управления→Программы и компоненты в некоторых версиях Windows) и выберите **Gwyddion**. Этот путь может отличаться в других версиях Windows.

Ключи реестра

Программа установки создаёт следующие нужные ключи в ветке реестра  
HKEY\_LOCAL\_MACHINE\Software\Gwyddion\2.0:InstallDir  
Каталог установки, т.е. C:\Program Files\Gwyddion. Чтение этого ключа может быть полезным, чтобы определить, куда ставить дополнения.

*Locale* Язык интерфейса пользователя **Gwyddion**, выбранный при установке. Можно изменить его используя *regedit* чтобы выбрать другой язык.

### ***Запуск Gwyddion***

Запуск **Gwyddion** обычно не требует дополнительных настроек. На рабочем окне появится диалоговое окно **Gwyddion**.

Этот раздел определяет некоторые основные концепции и термины, такие, как маски и выделенные области, объясняет организацию данных в **Gwyddion** и описывает интерфейс пользователя.

Для того чтобы начать работать в **Gwyddion**. Пользовательский интерфейс **Gwyddion** достаточно интуитивен, и многое можно узнать просто играясь с настройками. Также можно получить справку о работе текущей активной функции или окна непосредственно из программы **Gwyddion**. Нажатие клавиши **F1** или кнопки Помощь в большинстве окон должно показать онлайн-версию этого руководства в браузере. Стартовая страница руководства может быть показана выбором меню

Информация→Руководство пользователя. Естественно, оно будет работать только если вы подключены к сети и **Gwyddion** может найти подходящий веб-браузер. Если нужно настроить это поведение, смотрите изменение настроек, например, если вы хотите использовать оффлайн-версию этого руководства.

Подсказка Команда Информация→Совет дня показывает советы по обработке данных и рекомендует полезные возможности, которые вы могли упустить.

### ***Главное окно***

Главное окно, также называемое окном инструментов, является одним из двух окон, которые запускаются при старте **Gwyddion** (если не заданы файлы, которые нужно открыть), другое – это браузер данных. Закрытие этого окна приводит к выходу из **Gwyddion**

*Окно инструментов* содержит набор меню **Gwyddion** и несколько строк с кнопками, связанными с обычными функциями и инструментами. В меню функции сгруппированы следующим образом:

*Файл* объединяет команды, которые используются для загрузки и сохранения файлов. Некоторые из глобальных команд (например, *Выход*) также расположены здесь. История недавно

открытых файлов может просматриваться командой **Файл→Последние файлы→Предыдущие документы**.

*Правка* предоставляет команды управления историей редактирования (**Отменить**, **Повторить**) и редакторы для различных глобальных ресурсов, таких как градиенты и материалы для псевдоцветного и трёхмерного представления данных или цвета по умолчанию, используемого для масок.

*Обработка данных* автоматически строится из всех модулей обработки данных, расположенных в каталоге module **Gwyddion** (зависит от операционной системы). Это меню вместе с панелью кнопок **Инструменты** содержит большую часть команд, которые могут понадобиться для анализа данных СЗМ. Подмножество этих функций также доступно на панели кнопок **Обработка данных**. Эти кнопки служат кнопками быстрого запуска часто используемых функций из одноимённого меню. Все функции, доступные на этой панели, также можно найти и в меню.

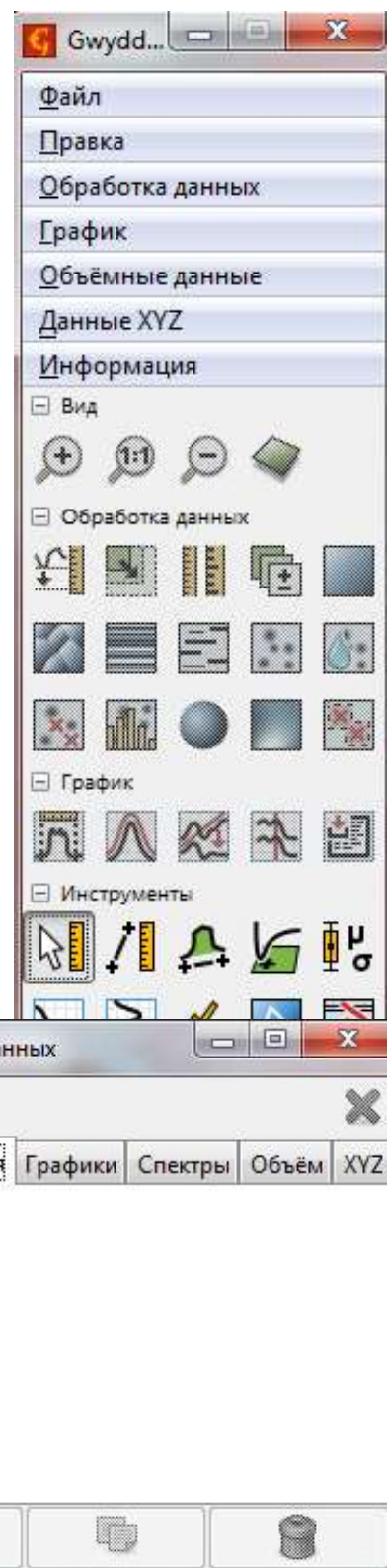
*График* подобен **Обработке данных**, за исключением того, что состоит из функций для работы с графиками.

**Обработка графиков** включает в себя аппроксимацию функциями, экспорт данных графиков и т.п. Панель кнопок **График** снова содержит подмножество наиболее часто используемых функций меню **График**.

**Объём** подобно функциям **Обработки данных**, за исключением того, что содержит функции работы с объёмными данными.

### ***Браузер данных***

**Браузер данных** — окно, которое показывает структуру текущего активного файла. Оно



показывает содержимое так, как оно представлено внутри **Gwyddion**, что может в чём-то отличаться от организации данных в оригинальном программном обеспечении.

**Gwyddion** поддерживает произвольное число двумерных полей данных в файле. В зависимости от контекста в этом руководстве они также называются каналами или полями высоты. Размеры каналов в одном файле могут различаться, физические размеры и значения также могут быть произвольными физическими величинами.

Дополнительно в том же файле могут присутствовать трёхмерные объёмные данные, одномерные данные, представленные в виде графиков, и данные спектроскопии в точках. Окно просмотра данных предназначено для просмотра и управления всеми доступными данными в файле.

#### Управление браузером

Поскольку окно браузера данных всегда показывает структуру текущего активного файла, его содержимое меняется при переходе между разными окнами, которые могут показывать данные различных файлов. Независимо, сохранён ли файл в родном формате **Gwyddion** (.gwy) или в другом, после того, как он загружен, его структура отображается как если бы он хранился в формате **Gwyddion**.

Окно браузера данных содержит четыре вкладки, по одной на каждый тип данных, который может быть в файле:

- Каналы
- Графики
- Спектры
- Объём

Каждый список показывает имена объектов данных и некоторые дополнительные свойства, которые зависят от типа данных. Имена можно править двойным кликом по ним.

Отдельные каналы, графики или спектры можно удалять, копировать или извлечь в новый файл используя кнопки внизу окна браузера данных. Также можно скопировать их в другой файл, перетащив строку из окна браузера данных в любое окно, относящееся к целевому файлу. Чтобы переименовать элемент данных, его нужно выбрать и нажать Enter или три раза кликнуть по нему мышкой.

Кнопка закрытия в правом верхнем углу браузера данных закрывает текущий файл, теряя все несохранённые изменения. Файл также закроется когда все окна, отображающие данные из этого файла, будут закрыты.

Если окно списка каналов закрыто, его можно вызвать используя команду *Информация→Показать браузерданных*.

#### Каналы

Браузер данных показывает миниатюры каналов, флажки, управляющие будет ли виден канал (т.е. будет ли он показан в отдельном окне), и названия каналов. Наличие презентации и/или маски отображается показом следующих символов справа от имени:

- М – маска
- Р – презентация
- С – калибровка

#### Графики

Список графиков показывает флажки видимости и имена графиков. Справа от имени отображается число кривых в графике.

#### Спектры

Список спектров показывает названия спектров и число точек в наборе. Поскольку данные спектроскопии в точке отображаются и обрабатываются только в связке с двумерными данными используя инструмент спектроскопии, у них нет флажков, управляющих видимостью.

#### Объём

Список объёмных данных показывает имя данных и число слоёв в направлении оси z, т.е. перпендикулярно сечению, показанному на экране.

### ***Выравнивание данных и вычитание фона***

Данные, полученные со сканирующих зондовых микроскопов нередко вообще никак не выровнены; микроскоп напрямую выводит сырые данные, пересчитанные из напряжения на пьезосканере, измерения деформации, данных интерферометра или значений другой системы обнаружения. Такой способ экспорта данных позволяет выбрать пользователю его/её собственный метод выравнивания данных.

Выбор метода выравнивания должен быть основан на конфигурации вашего СЗМ. В основном, для систем с независимыми сканерами для каждой оси, обычно достаточно



вычета плоскости. Для систем со сканером/сканерами, двигающимися по всем трём осям (сканер на пьезотрубке) нужно использовать выравнивание вычетом полинома второго порядка.

Разумеется, можно использовать выравнивание более высоких порядков для любых данных, однако, при этом могут быть утрачены реальные свойства поверхности (а именно её волнистость) и, следовательно, изменятся все статистические функции и величины, получаемые из анализа данной поверхности.

### **Исправить ноль и Обнулить среднее значение**



*Обработка данных → Выравнивание → Исправить ноль*

*Обработка данных → Выравнивание → Обнулить среднее значение*

Простейшие функции, связанные с выравниванием данных это Исправить ноль и Обнулить среднее значение, которые просто добавляют постоянное значение ко всем элементам поля данных чтобы соответственно переместить минимум или среднее значение в ноль.

### **Выравнивание плоскости**



*Обработка данных → Выравнивание → Выровнять плоскость*

Выравнивание плоскости обычно является одной из первых функций, применяемых к необработанным данным СЗМ. Плоскость рассчитывается по всем точкам изображения и затем вычитается из данных.

Если на изображение наложена маска, выравнивание плоскости предлагает или использовать данные под маской для аппроксимации плоскостью, исключить данные под маской или игнорировать маску и использовать полный набор данных.

### ***Наложение псевдоцвета***

Наложение псевдоцвета – основной метод визуализации двумерных данных. Используемый цветовой градиент (также называемый палитрой) может быть выбран с помощью щелчка правой кнопкой мыши на ту часть окна данных, где отображается карта псевдоцвета.

Выпадающее меню быстрого выбора предлагает список предпочитаемых градиентов псевдоцвета. Кроме того, оно позволяет вызвать полный список градиентов с помощью пункта меню Дополнительно. Предпочитаемые градиенты могут быть выбраны с помощью флажков в полном списке или в списке

редактора градиентов. Выбор строки в полном списке меняет градиент на выбранный, двойной щелчок (или нажатие Enter) также заканчивает выбор и закрывает окно списка. Градиенты с известными именами могут быть быстро выбраны набором начальных букв названия. Градиент по умолчанию (если он не указан в файле) также может быть задан в редакторе градиентов.

Дополнительный контроль над тем, как значения отображаются на шкалу цвета возможен с помощью инструмента Цветовой диапазон.

### Инструмент диапазон псевдоцвета



Инструмент диапазон цвета — специальный инструмент, назначением которого является не анализ или модификация данных, а контроль над тем, как значения данных отображаются на шкалу псевдоцвета. Он предлагает четыре основных способа отображения значений данных на шкалу цвета:

#### Полный

Значения данных линейно масштабируются на шкалу псевдоцвета, полный диапазон значений данных соответствует полному диапазону значений псевдоцвета. Этот тип включен по умолчанию (если вы не меняли настройки по умолчанию).

#### Фиксированный

Значения данных отображаются на значения цвета линейно, заданный пользователем диапазон данных (который может быть как меньше, так и больше полного) отображается на полный диапазон цвета. Данные вне диапазона отображаются цветом его краёв. Диапазон может быть задан несколькими способами:

- вводом нужных значений численно в окне инструмента,
- выбором диапазона на графике распределения высот в окне инструмента,
- выбором области в окне данных, диапазон затем устанавливается по минимальному и максимальному значениям только внутри этой области,
- нажатием кнопок *Под маской* или *Не под маской*, которые устанавливают диапазон равным диапазону значений под маской или не под маской соответственно или
- нажатием кнопки *Инвертировать шкалу*, которая меняет верхний и нижний предел диапазона отображения цвета

Если диапазон не задан вручную, фиксированный диапазон ведёт себя подобно полному.

Следует заметить, что операции обработки данных нередко изменяют диапазон значений — и поскольку фиксированный диапазон остаётся таким, каким он был задан, это может привести, например, к полностью чёрному окну данных. В таком случае можно обновить диапазон вручную или переключиться на другой тип отображения.

### Автоматический

Значения данных отображаются на цветовой диапазон линейно, эвристически определённый под-интервал полного диапазона данных отображается на полный диапазон цвета. Значения вне этого под-интервала отображаются цветом на границе диапазона.

### Адаптивный

Полный диапазон данных соответствует полному диапазону цвета, однако, значения данных отображаются на значения цвета нелинейно. Функция отображения основана на инверсной интегральной функции распределения высоты, следовательно плоские области в среднем получают большую часть цветового градиента, и меньшие изменения значения видны на них лучше, чем обычно.

## Базовые действия

Считывание значений и основные геометрические операции представляют собой основу любой программы обработки данных. **Gwyddion** предлагает большой набор функций для масштабирования данных, вращения, изменения количества точек и извлечения профилей. Основные из них:

### Инструмент считать значение

Инструмент Считать значение предлагает больше вариантов считывания данных: он показывает координаты и значения в последней точке окна данных, где была нажата кнопка мыши. Он может усреднить значение в круговой окрестности выбранной точки, это управляется опцией *Радиус усреднения*. Если этот радиус равен 1, показывается значение в единичной точке (как делает простейший метод). Кнопка *Установить ноль* сдвигает

поверхность таким образом, чтобы сделать текущее значение  $z$  новым нулевым уровнем.

*Считать значение* также может показывать наклон локальной грани или локальную кривизну поверхности. *Радиус усреднения* снова определяет радиус области, используемой для аппроксимации плоскостью. *Кривизна*, в частности, требует относительно больших площадей чтобы быть достоверной.

### Инструмент расстояний

Расстояния и разности значений можно измерять с помощью инструмента расстояния. Он показывает горизонтальное ( $\Delta x$ ), вертикальное ( $\Delta y$ ) расстояния и общее расстояние по плоскости ( $R$ ); азимут  $\varphi$  (измеряется так же, как наклон  $\varphi$ ) и разницу значений конечных точек  $\Delta z$  для набора линий выбранных для текущих данных.



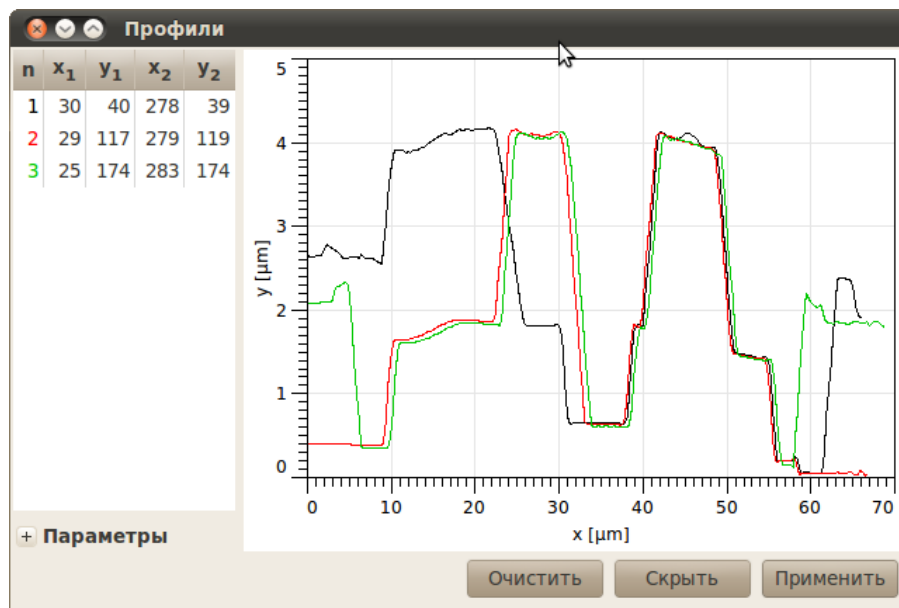
n	$\Delta x$ [ $\mu\text{m}$ ]	$\Delta y$ [ $\mu\text{m}$ ]	$\varphi$ [deg]	$R$ [ $\mu\text{m}$ ]	$\Delta z$ [ $\mu\text{m}$ ]
1	55,2	0,3	-0,3	55,2	-0,336
2	57,1	0,3	-0,3	57,1	-0,519
3	58,9	4,5	-4,4	59,1	0,063

Расстояния можно скопировать в буфер обмена или сохранить в текстовый файл с помощью кнопок под списком.

### Извлечение профиля

Инструмент извлечения профиля извлекает профили вдоль линий нарисованных и настроенных с помощью мыши на изображении и показываемые в виде живого предварительного результата в диалоговом окне. Профили могут быть различной «толщины», что означает что больше соседних точек данных в направлении поперёк направления профиля будет использоваться для расчёта одной точки профиля для получения более широких профилей. Это может быть весьма полезно для уменьшения шума при измерении объектов правильной формы.

После того, как профили выбраны, их можно извлечь в виде графиков (как отдельных, так и сгруппированных в одном окне графика), которые в свою очередь могут быть



проанализированы соответствующими функциями работы с графиками.

### Инструмент измерения зёрен

Инструмент измерения зёрен является интерактивным способом получения той же информации об отдельных зёрнах. После выбора зерна мышью в окне данных, все доступные величины показываются в окне инструмента.

Помимо физических характеристик, этот инструмент также показывает номер зерна.

#### Свойства зёрен

Инструменты распределений зёрен и измерения зёрен могут рассчитывать набор свойств зёрен. Некоторые из них:

#### Свойства значений

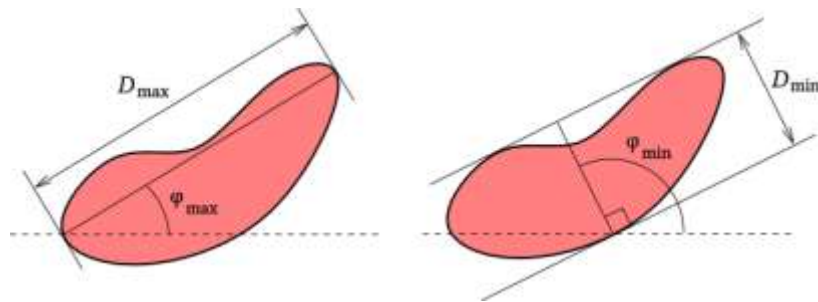
- *Минимум*, минимальное значение (высота), найденное внутри зерна.
- *Максимум*, максимальное значение (высота), найденное внутри зерна.
- *Среднее*, среднее всех значений внутри зерна, средняя высота зерна.

#### Свойства площадей

- *Площадь проекции*, площадь проекции зерна на плоскость.
- *Сторона эквивалентного квадрата*, сторона квадрата с той же площадью проекции, что и у зерна.
- *Радиус эквивалентного диска*, радиус диска с той же площадью проекции, что и у зерна.

### Свойства границы

- *Минимальный ограничивающий размер*, минимальный размер зерна в горизонтальной плоскости. Может быть показан как минимальная ширина зазора в горизонтальной плоскости, через который может пройти зерно.
- *Максимальный ограничивающий размер*, максимальный размер зерна в горизонтальной плоскости. Может быть показан как максимальная ширина зазора в горизонтальной плоскости, который можно заполнить зерном.
- *Максимальный радиус вписанного диска*, радиус максимального диска, который помещается внутри зерна.
- *Минимальный радиус описанной окружности*, радиус минимальной окружности, целиком содержащей зерно.



### Свойства, относящиеся к кривизне

- *Центр кривизны по x*, горизонтальное положение центра квадратичной поверхности, которой аппроксимируется поверхность зерна.
- *Центр кривизны по y*, вертикальное положение центра квадратичной поверхности, которой аппроксимируется поверхность зерна.
- *Центр кривизны по z*, значение в центре квадратичной поверхности, которой аппроксимируется поверхность зерна. Следует отметить, что это значение на аппроксимирующей поверхности, не на поверхности зерна.
- *Кривизна 1*, меньшая кривизна (т.е. значение, обратное радиусу кривизны) в центре.
- *Кривизна 2*, большая кривизна (т.е. значение, обратное радиусу кривизны) в центре.

## **Обработка графиков**

Многие из модулей обработки данных получают какие-либо графики в результате своей работы. Графики можно экспортировать в текстовый файл или продолжать анализировать их в **Gwyddion** с помощью нескольких модулей обработки графиков. Эти модули доступны в меню График основного окна **Gwyddion**. Следует заметить, что число модулей обработки графиков на данный момент невелико, и они состоят в основном из простейших модулей для операций, которые часто встречаются при анализе данных СЗМ. Для более сложных аналитических операций лучше использовать вашу любимую программу для обработки графиков.

В этой секции кратко описываются имеющиеся в **Gwyddion** модули обработки графиков.

### **Базовые операции**

Прежде всего, функции масштабирования и чтения данных доступны непосредственно в окне графика:

- Логарифмический масштаб осей – горизонтальные и вертикальные оси могут переключаться между линейным и логарифмическим режимом с использованием кнопок логарифмического масштаба. Переключение на логарифмический масштаб доступно только для положительных значений (либо по абсциссе, либо по ординате).
- Увеличение и уменьшение масштаба – после выбора режима приближения можно нарисовать мышью область, которую нужно увеличить. Уменьшение восстанавливает тот масштаб, при котором видны все данные.
- Измерение расстояний – позволяет выбрать несколько точек на графике и показать расстояния и углы между ними.

### **Экспорт графиков кривых**

Данные графиков кривых могут быть экспортированы в текстовые файлы с использованием меню Экспортировать текст. Диалоговое окно экспорта позволяет выбрать между несколькими вариантами стилей, которые потом проще импортировать в другие пакеты программного обеспечения. Опции Экспорт меток, Экспорт единиц измерения и Экспорт метаданных позволяют добавить строки с дополнительной информацией перед блоком числовых данных. Это может оказаться полезным как напоминание, какие

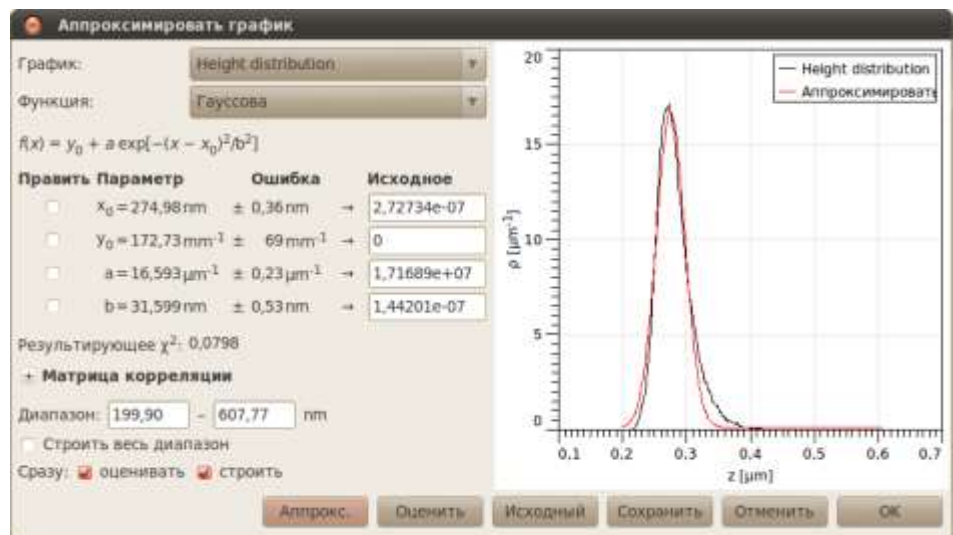
данные содержат исходный файл, но может вызывать проблемы при чтении файла другим программным обеспечением.

Также можно экспортировать векторное (EPS) или растровое (PNG) графическое представление графика используя пункты меню *Экспортировать PostScript* или *Экспортировать растр*.

## Аппроксимировать функцией

Аппроксимация функцией разрабатывалась для аппроксимации данных статистическими функциями, используемыми при оценке параметров шероховатости. Следовательно, большая часть доступных здесь функций является статистическими функциями поверхностей с гауссовой или экспоненциальной функцией автокорреляции. Тем не менее, здесь доступно несколько распространённых функций общего назначения.

В модуле аппроксимации и можно задать область, на которой будет производиться аппроксимация (с помощью мыши или численно), сначала надо



задать начальные параметры или дать модулю самому угадать их, и, затем, можно аппроксимировать данные используя алгоритм Левенберга – Маркардта.

В результате получается аппроксимированная кривая и набор её параметров. Отчёт об аппроксимации можно сохранить в файл используя кнопку *Сохранить*. Нажатие кнопки *ОК* добавляет аппроксимирующую кривую к графику, если это нежелательно, закройте диалоговое окно с помощью кнопки *Отмена*.

## Анализ зёрен

В **Gwyddion** реализовано несколько алгоритмов для работы с зёрнами. Прежде всего, можно использовать простые пороговые алгоритмы (порог по высоте, наклону и кривизне). Эти процедуры



могут быть очень эффективны для анализа частиц (чтобы пометить частицы лежащие на ровной поверхности).

### **По порогу**

*Обработка данных → Зёрна → Пометить по порогу*

Выделение по порогу это основной метод выделения зёрен. В этом модуле реализованы пороги по высоте, наклону и кривизне. Результаты отдельных пороговых методов можно объединять вместе используя логические операторы.

### **Метод Оцу**

*Обработка данных → Зёрна → Пометить по методу Оцу*

Автоматический метод Оцу выделения зёрен по порогу классифицирует значения данных в два класса, минимизируя дисперсию внутри класса для обоих. Метод хорошо подходит для изображений, которые содержат два относительно хорошо определённых диапазона значений.

### **Пометить по краю**

*Обработка данных → Зёрна → Пометить по краю*

Другой вариант функции выделения зёрен основан на обнаружении краёв (локальной кривизне функции высоты). Изображение обрабатывается фильтром «разница гауссианов».

### **Пометить по водоразделу**

*Обработка данных → Зёрна → Пометить по водоразделу*

Для более сложных структур данных эффективность пороговых алгоритмов может оказаться весьма плохой. Для этих данных выделение зёрен или частиц более эффективно можно проводить с помощью *алгоритма водораздела*.

Алгоритм водораздела обычно задействуется для определения локальных минимумов и разделения изображения на сегменты в обработке изображений. Поскольку проблема определения положения зёрен может сводиться к проблеме нахождения локальных экстремумов на поверхности, этот алгоритм также можно использовать для нужд разделения и выделения зёрен. Для удобства в дальнейшем мы будем считать данные инвертированными в направлении  $z$  при описании этого алгоритма (т.е. вершины зёрен формируют локальные минимумы в нижеследующем тексте). Мы применяем две стадии анализа зёрен:

1. Фаза обнаружения зёрен: В каждой точке инвертированной поверхности помещается виртуальная капля воды (количество воды регулируется параметром *Размер капли*). В случае, если капля не находится в локальном минимуме, она следует по пути наискорейшего спуска чтобы минимизировать свою потенциальную энергию. Как только капля достигает любого локального минимума, она остаётся там и покоится на поверхности. Подобным образом она частично заполняет объём локального минимума (см. рисунок ниже и его подпись). Этот процесс повторяется несколько раз (параметр *Число шагов*). В результате получается система водоёмов различного размера, заполняющих впадины в инвертированной поверхности. После этого рассчитывается площадь каждого водоёма и наименьшие из них отбрасываются в предположении, что они формируются в локальных минимумах, образованных шумом (все водоёмы размером меньше, чем задано параметром *Порог*, удаляются). Большие водоёмы используются для определения положения зёрен для разделения на следующем шаге. Подобным образом можно устранить шум на данных АСМ.
2. Фаза разделения: Зёрна, найденные на первом шаге, помечаются (каждое своим уникальным номером). Капли воды продолжают падать на поверхность и заполнять локальные минимумы (количество воды регулируется параметром *Размер капли*). Общее число шагов падения капли в каждую точку поверхности регулируется параметром *Число шагов*. Поскольку зёрна уже обнаружены и помечены после первого шага, следующие пять ситуаций возможны после достижения каплей локального минимума:
  - а. Капля достигла места, ранее отмеченного как определённое зерно. В этом случае капля объединяется с зерном, т.е. помечается как часть того же зерна.
  - б. Капля достигла места, где нет зерна, но отмеченное зерно найдено в ближайшей окрестности капли. В этом случае капля снова объединяется с зерном.
  - с. Капля достигла места, где нет зёрен, и при этом зёрен не было даже в ближайшей окрестности капли. В этом случае капля никак не помечается.

- d. Капля достигла места, где нет зерна, но более одного отмеченного зерна найдено в ближайшей окрестности. (т.е. в окрестности нашлось два разных зерна). В этом случае капля помечается как граница зерна.
- e. Капля достигла места, помеченного как граница зерна. В этом случае капля также помечается как граница зерна.

Таким путём мы определяем положения отдельных зёрен и затем определяем объём, занятый каждым зерном по отдельности. Если нас интересуют на поверхности не зёрна (выпуклости), а долины (впадины), можно использовать параметр *Инвертировать высоту*.

### **Статистика**

*Обработка данных → Зёрна → Статистика*

Эта функция рассчитывает общее число помеченных зёрен, их общую площадь (площадь проекции) как в виде абсолютного значения, так и в виде доли общей площади поля данных, общий объём зёрен, общую длину границ зёрен, среднюю площадь и сторону эквивалентного квадрата для отдельного зерна. Средний размер рассчитывается усреднением стороны эквивалентного квадрата, и его квадрат в общем случае не будет равным средней площади.

Общие характеристики области под маской могут быть получены с помощью инструмента «статистические величины» когда включена опция *Использовать маску*. Если инвертировать маску та же информация может быть получена об областях вне зёрен.

### **Распределения зёрен**



*Обработка данных → Зёрна → Распределения*

Распределения зёрен является наиболее мощным и сложным инструментом. Он работает в двух режимах: построение графиков и экспорт необработанных данных. В режиме построения графиков выбранные характеристики отдельных зёрен, рассчитываются, собираются, и строятся в виде суммарных графиков, показывающих их распределения.

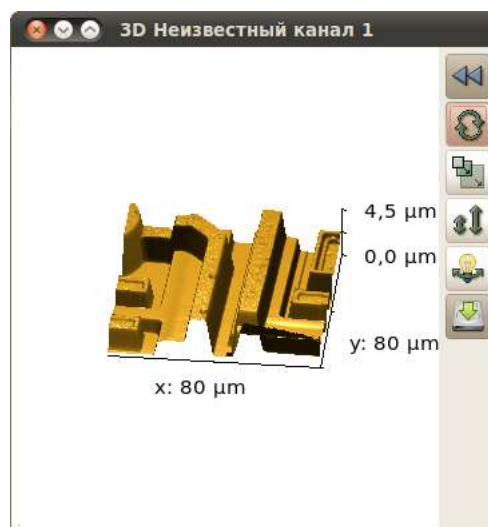
Экспорт необработанных данных полезен для экспертов, которым нужно, например, рассчитывать корреляцию характеристик отдельных зёрен. В этом режиме выбранные характеристики зёрен рассчитываются и записываются в виде

таблицы в текстовый файл, где каждая строка соответствует отдельному зерну, а столбцы соответствуют выбранным величинам. Порядок следования столбцов тот же, что и порядок следования величин в диалоговом окне; все величины записывается в единицах измерения СИ, что является общепринятым в **Gwyddion**.

### *Окно трёхмерного отображения данных с помощью OpenGL*

Трёхмерное отображение OpenGL для текущего окна данных можно вызвать кнопкой с изображением куба в ряду кнопок *Вид* главного окна.

Окно трёхмерного вида может существовать в двух вариантах: с основными и с расширенными элементами управления. Оно запускается только с основными элементами управления, этот вариант показан на следующем рисунке. Можно переключить окно к расширенному виду (и обратно) с помощью кнопки расширения в верхнем правом углу. Нажатие на вид с помощью правой кнопки мыши вызывает окно быстрого выбора градиента псевдоцвета/материала OpenGL.



### **Задания к лабораторной работе и порядок действий при их выполнении**

#### **Задание 1**

Постройте распределение по размерам люминесцирующих наночастиц на кремниевой подложке.

#### Порядок выполнения:

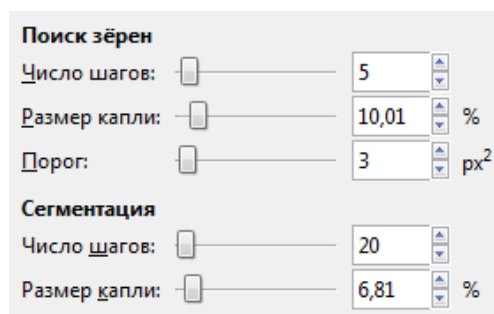
1) Запустите **Gwyddion**. Откройте файл АСМ изображения нанесенных на кремниевую подложку люминесцирующих наночастиц. Файлы такого расширения генерируются специальным программным обеспечением атомно-силового микроскопа.

2) Проведите первичную обработку изображения:

- Выравнивание данных: *Главное окно* → *Обработка данных* → *Выровнять данные вычитанием средней плоскости*. *Корректировать линии выравниванием срединного значения высоты*. *Убрать горизонтальные царапины*. *Исправить ноль*.

• Настройка цветов: *Главное окно* → *Диапазон псевдоцвета* - выбрать диапазон на графике распределения высот в окне инструмента, выбрать градиент в выпадающем меню *Окна данных* (щелчок правой кнопкой мыши по оси высот).

3) Проведите поиск и разметку зерен: *Обработка данных* → *Зёрна* → *Пометить по водоразделу*. Необходимо подобрать наиболее адекватные параметры алгоритма разметки зерен в открывшемся окне. Пример таких параметров представлен на рисунке справа.



4) Изучите статистику зерен: *Обработка данных* → *Зёрна* → *Статистика*. Отметьте общее количество зерен и их средний размер согласно данной функции.

5) Постройте распределение по максимальному ограничивающему размеру зерна: *Обработка данных* → *Зёрна* → *Распределения*. В окне распределения выберите свойство «Максимальный ограничивающий размер», выберите пункт «Строить графики».

6) Аппроксимируйте полученное распределение с помощью команды *Аппроксимировать функцией* (может быть вызвана щелчком правой кнопкой мыши на графике) наиболее подходящей функцией.

7) Результирующий график сохраните в формате JPG с помощью функции *Экспортировать в растр* (может быть вызвана щелчком правой кнопкой мыши на графике).

8) Полученные данные оформите в виде отчета, сделайте выводы о характере распределения по размерам люминесцирующих наночастиц, нанесенных на кремниевую подложку.

Дополнительно:

Распределение зерен можно построить с помощью средств программы LibreOffice Calc. Для этого необходимо в пункте 5) выбрать не «Строить графики», а «Экспортировать необработанные данные» и сохранить в формате RTF.

В текстовом файле заменить:

- «e» на «E»,
- точки на запятые.

Затем выполнить действия *Анализ данных* → *Гистограмма*, построить распределение зерен по полученному массиву данных.

## Задание 2

1) Определение размера частиц на СЭМ-изображении с помощью программного обеспечения Gwyddion.

Поместите СЭМ-изображение в программу: перетащите файл изображения в окно инструментов левой кнопкой мыши или откройте изображение через команды *Файл* → *Открыть*.

Определите длину масштабного отрезка автоматически выставляемую в Gwyddion, по пропорции рассчитайте реальный размер.

В поле *ширина* введите полученное в первой части значение (129,70), высоту программа заполнит автоматически в соответствии с количеством пикселей изображения (Рис. 3); кликните *ОК*.

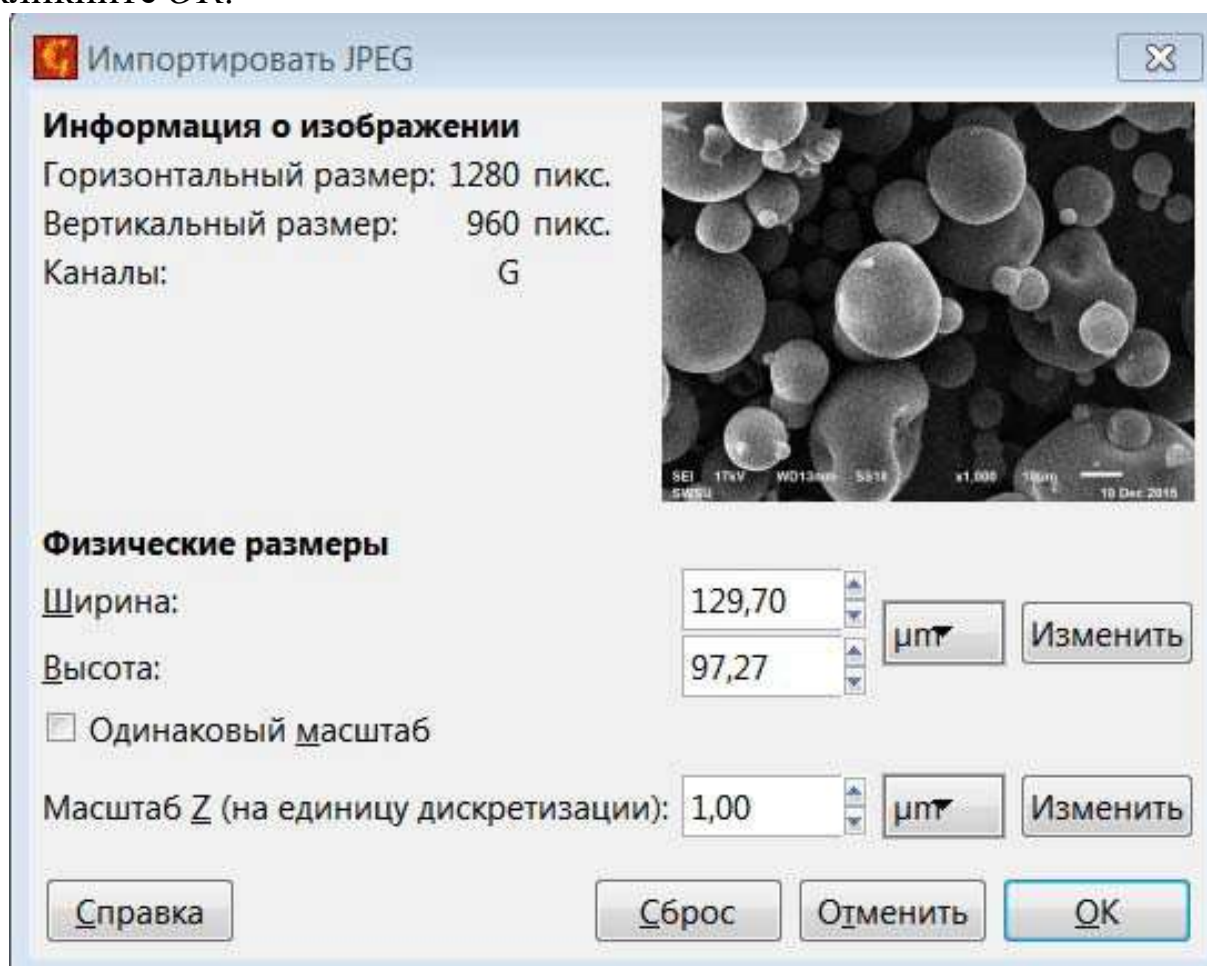



Рисунок 3

2) Установите размеры выбранных частиц:

в окне «браузер данных» задайте удобный градиент цветов,

кликнув правой кнопкой мыши градиентную шкалу в правой части рабочего поля; в окне «инструментов» выберите инструмент *расстояний* ; зажмите левую кнопку мыши с одного края частицы, протяните к другому краю и отпустите, появится отрезок с порядковым номером измерения; проделайте данную операцию три раза (Рис. 4); во вкладке «Расстояний» автоматически построенная таблица значений, столбец *R* соответствует модулям построенных отрезков, т.е. размеру частиц (Рис.5).

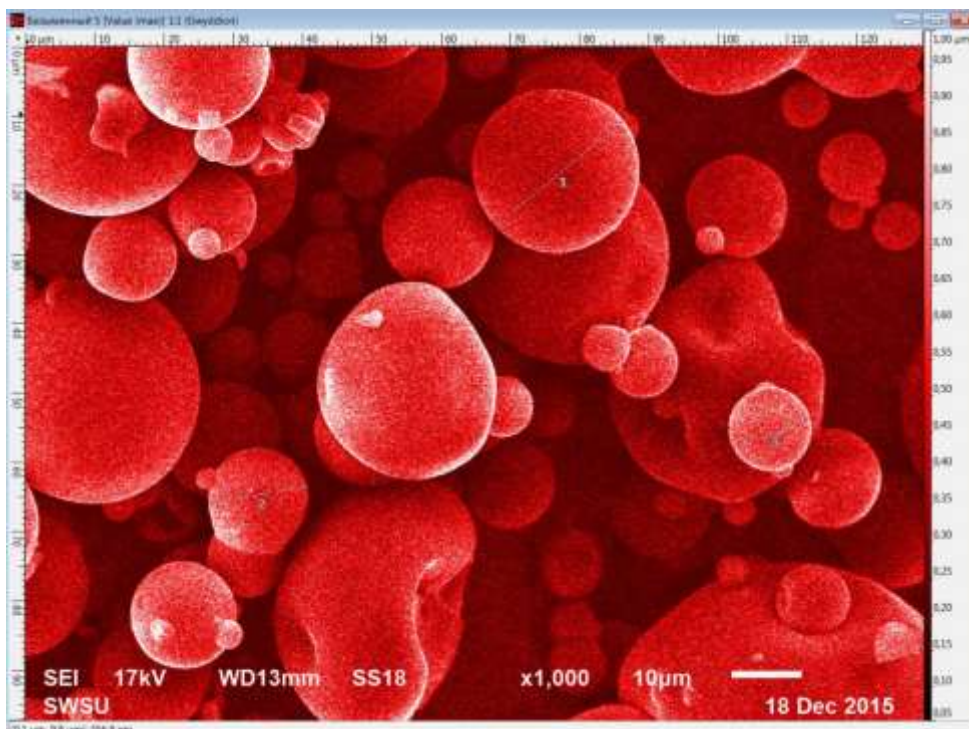


Рисунок 4

n	Δx [µm]	Δy [µm]	φ [°]	R [µm]	Δz [nm]
1	-18,3	15,1	-140,5	23,8	-419,6
2	14,2	-2,3	9,3	14,4	-176,5
3	11,0	-4,2	20,6	11,8	15,7

Нумеровать линии

Рисунок 5

## КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие функции выполняет модуль *Выравнивания данных и вычитания фона*?
2. Какую функцию выполняет инструмент *Извлечь профиль*?
3. В чем сущность *алгоритма водораздела*?
4. Каким инструментом необходимо воспользоваться для выделения зерен на АСМ изображении?
5. Маски нужны для выделения особых областей, таких как...
  - 1) все перечисленное;
  - 2) зёрна;
  - 3) дефекты;
  - 4) грани с определённой ориентацией.
6. Какой функцией анализа зерен необходимо воспользоваться для получения информации о среднем размере зерна?
7. Какой параметр необходимо использовать, чтобы алгоритмом водораздела выделить долины (впадины), а не зерна (выпуклости)?
8. Каким инструментом можно получить информацию о размере отрезка между двумя точками в плоскости изображения?
9. Каким инструментом можно получить рельеф поверхности вдоль линий нарисованных и настроенных с помощью мыши на изображении в виде графиков?
10. Модуль обработки графиков НЕ содержит функцию:
  - 1) экспорта в LibreOffice Calc;
  - 2) экспорта в растр;
  - 3) экспорта в текст;
  - 4) экспорта в PostScript.

## ЛИТЕРАТУРА

1. Petr Klapetek, David Nečas, Christopher Anderson. Русский перевод: Даниил Браташов. Руководство пользователя Gwyddion. 2004–2007, 2009–2017. <http://gwyddion.net/documentation/user-guide-ru/index.html>



## ЛАБОРАТОРНАЯ РАБОТА № 2.

### **Освоение программного обеспечения и документации по основным характеристикам и устройству твердотельного лазера. Планирование работ по использованию лазера для скрайбирования, резки и лазерной абляции**

**Цель работы:** познакомиться с устройством рабочей части иттербиевого волоконно-оптического лазера, его характеристиками и приемами работы с ним.

**Приборы и принадлежности:** лазерный гравёр ARGENT (ЦЛТ).

## **КРАТКАЯ ТЕОРИЯ**

### **Введение**

**ВНИМАНИЕ!!!** При работе с лазером необходимо соблюдать особые меры предосторожности! Самое главное при работе лазера на вас должны быть надеты специальные очки, и смотреть на разрезаемый образец вы должны через защитное стекло. При выполнении каждой лабораторной и практической работы необходимо соблюдать инструкцию по технике безопасности, которая размещении на сайте кафедры <https://swsu.ru/structura/up/ftd/kafedra-nt/instruktsii.php>

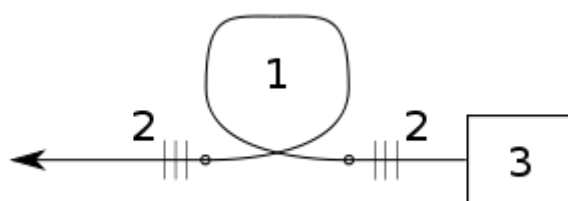
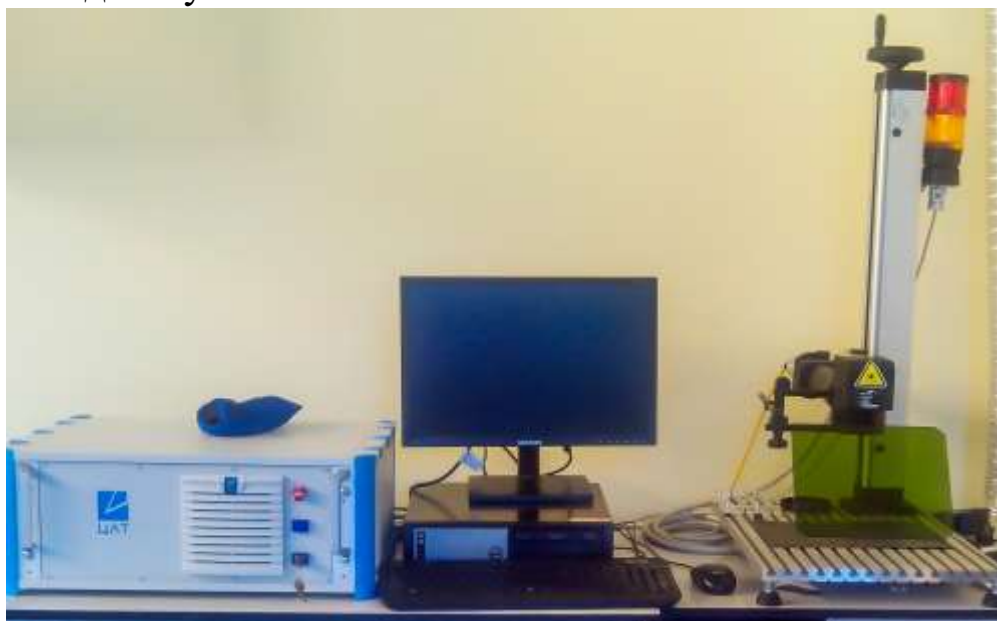
Лазерный маркирующий комплекс FMark-20RL позволяет решать задачи двойного назначения.

Технологические – маркировка и гравировка поверхностей изделий.

Исследовательские – прецизионная модификация поверхностей при лазерном воздействии, высокоскоростной испарение и абляция тугоплавких, твердых и сверхтвердых материалов.

- Тип лазера - иттербиевый волоконно-оптический лазер
  - Длина волны излучения – 1060 нм
  - Длительность импульсов излучения - 4,200 нс
  - Частота повторения импульсов -1.6 , 500 кГц
  - Средняя мощность - 50 Вт
- Диаметр пятна в зоне воздействия - 5 мкм

Волоконный лазер состоит из модуля накачки (как правило, широкополосные светодиоды или лазерные диоды), световода, в котором происходит генерация, и резонатора. Световод содержит активное вещество (легированное оптическое волокно – сердцевина без оболочки, в отличие от обычных оптических волноводов) и волноводы накачки. Конструкция резонатора обычно определяется техническим заданием, но можно выделить наиболее распространенные классы: резонаторы типа Фабри – Перо и кольцевые резонаторы. В промышленных установках для повышения выходной мощности иногда объединяют несколько лазеров в одной установке.



*Типичная схема  
волоконного лазера. 1 —  
активное волокно. 2 —  
брэгговские зеркала. 3 — блок  
накачки*

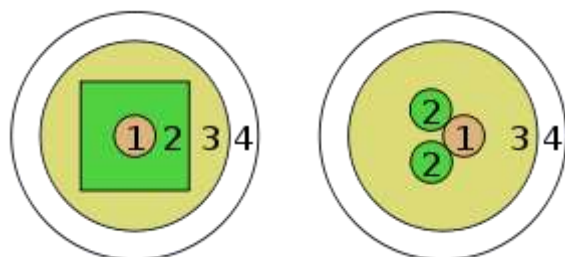
Волоконный лазер состоит из модуля накачки (как правило, широкополосные светодиоды или лазерные диоды), световода, в котором происходит генерация, и резонатора. Световод содержит активное вещество (легированное оптическое волокно – сердцевина без оболочки, в отличие от обычных оптических волноводов) и волноводы накачки. Конструкция резонатора обычно определяется техническим заданием, но можно выделить наиболее распространенные классы: резонаторы типа Фабри – Перо и кольцевые резонаторы.

### **Активное волокно**

Сверхчистый плавленный кварц, который является основным материалом оптических волокон, обладает высокой прозрачностью (оптические потери – несколько процентов на километре длины). Специальные примеси, вводимые в кварц легированием, превращают его в активную среду. Исходя из требований на частоту излучения (инфракрасный диапазон для телекоммуникаций) и малую пороговую мощность накачки, как правило, легирование выполняют редкоземельными элементами группы лантаноидов. Одним из распространённых типов волокон является эрбиевое, используемое в лазерных и усилительных системах, рабочий диапазон которых лежит в интервале длин волн 1530-1565 нм. Вследствие различной вероятности переходов на основной уровень с подуровней метастабильного уровня, эффективность генерации или усиления отличается для различных длин волн в рабочем диапазоне. Степень легирования редкоземельными ионами обычно зависит от длины изготавливаемого активного волокна.

Существуют различные конструкции накачки оптических волноводов, из которых наиболее употребительными являются чисто волоконные конструкции. Одним из вариантов является размещение активного волокна внутри нескольких оболочек, из которых внешняя является защитной (так называемое волокно с двойным покрытием).

Первая оболочка изготавливается из чистого кварца диаметром в несколько сотен микрометров, а вторая – из полимерного материала, показатель преломления которого подбирается существенно меньшим, чем у кварца. Таким образом, первая и вторая оболочки создают многомодовый волновод с большим поперечным сечением и числовой апертурой, в который запускается излучение накачки. Эффективное возбуждение ионов редкоземельных элементов достигается подбором диаметров активной сердцевины и волновода накачки. По такой технологии можно получить выходную мощность порядка 100 Вт.



Два различных типа оптических волокон для волоконных лазеров (не в масштабе). Слева: обычная схема с одним волноводом накачки, волокно с двойным покрытием.

Справа: схема технологии GTWave с двумя волноводами накачки (для примера). 1. Сердцевина, легированная редкоземельными ионами. 2. Волновод накачки. 3. Общая оболочка. 4. Защитная оболочка.

### Конструкция и принцип работы

Главная особенность этого лазера в том, что излучение здесь рождается в тонком, диаметром всего в 6-8 мкм, волокне 3 (сердцевине – например, активная среда иттербий), которое фактически находится внутри кварцевого волокна 2 диаметром 400-600 мкм. Схема конструкции волоконного лазера приведена на рис. 2.

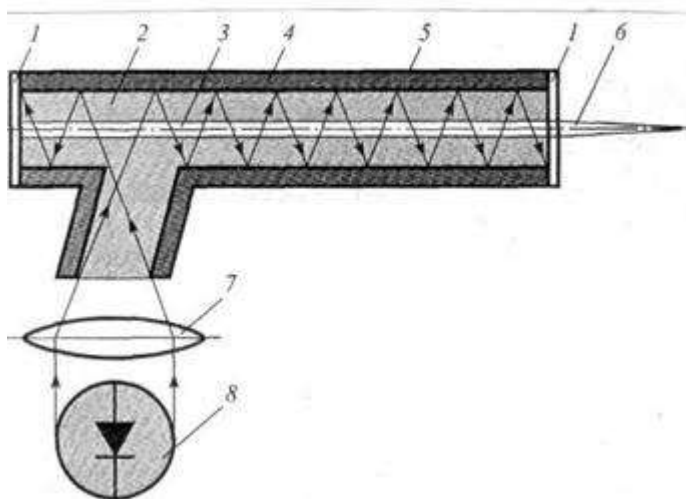


Рис. 2. Схема волоконного лазера

1 – торцевое зеркало; 2 – оптическое кварцевое волокно; 3 – активированное световедущее волокно; 4 – светоотражающее покрытие; 5 – защитная оболочка;

6 – лазерный луч; 7 – фокусирующая линза; 8 – светодиоды.

Излучение лазерных диодов 8 накачки вводится в кварцевое волокно 2 и распространяется вдоль всего сложного составного волокна, отражаясь от светоотражающего покрытия 4, имеющего длину несколько десятков метров, которое можно навивать на какую-либо поверхность. Упрощенно говоря, это излучение оптически накачивает сердцевину, именно в ней происходит эффект усиления спонтанного излучения ионов редкоземельных элементов. Вблизи концов волокна на сердцевине размещают два так называемых дифракционных зеркала, одно из которых сплошное 1, а другое полупрозрачное. Полупрозрачное зеркало выпускает только часть полученного после многократного

отражения светового излучения, в то время как сплошное зеркало не выпускает световой поток, отражая его. Тем самым часть излучения выходит в виде лазерного луча, а другая часть, испытывая переотражение от стенок волокна и двух зеркал, компенсирует ушедшее излучение. Система уравнивается. Таким образом, создается резонатор волоконного лазера. Через полупрозрачное зеркало выходит идеальный одномодовый лазерный пучок с весьма равномерным распределением мощности, что позволяет сфокусировать излучение в пятно малого размера с помощью фокусирующей линзы 7, и, наконец, через волокно выходит конечный сфокусированный лазерный луч 6.

Мощность волоконного лазера, применяемого для технологических целей, должна составлять от 100 Вт до 4...6кВт. Излучение такой мощности получают сочетанием многокаскадного усиления в волокнах с набором мощности излучения от нескольких лазеров с меньшей мощностью. Волоконный лазер со ступенчатым усилением показан на рис. 3.

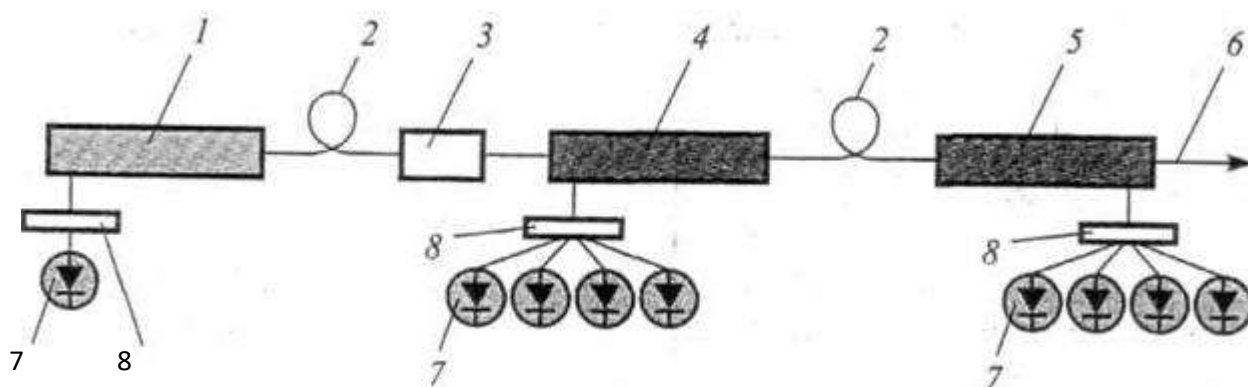


Рис. 3. Схема многокаскадного усиления в волокнах путем набора мощности излучения от нескольких лазеров с меньшей мощностью

1 – задающий волоконный лазер; 2 – соединяющее световое волокно; 3 – изолятор;

4 – мощный усиливающий волоконный лазер первой ступени; 5 – мощный усиливающий волоконный лазер второй ступени; 6 – лазерный луч; 7 – светодиоды;

8 – фокусирующие системы светодиодов.

Принцип работы такого лазера заключается в следующем. Сначала накачивается задающий волоконный лазер 1 с помощью излучения светодиодов 7, пропущенного через фокусирующие

системы светодиодов 8. Затем световое излучение передается по световому волокну 2 к изолятору. От изолятора излучение передается мощному волоконному лазеру первой ступени 4, накачивая его, а от лазера первой ступени соответственно к мощному волоконному лазеру второй ступени 5, который тоже накачивается. После фокусировки из лазера второй ступени выходит конечный лазерный луч 6.

С помощью такого лазера можно получить мощность излучения до 100Вт. При этом КПД лазера составляет до 23 %. В этом случае наблюдается весьма малое тепловыделение (около 8...10 Вт). Это дает возможность использовать воздушное охлаждение и исключить применение сложных систем водяного охлаждения, что присуще другим типам технологических лазеров.

Для получения мощностей 2, 4, 6 и 10 кВт создают блочные системы, в которых излучение отдельных одномодовых волоконных лазеров 1 мощностью 100 Вт собирается в одном волокне 3. И после прохождения через фокусирующую систему 4 лазерный луч попадает на обрабатываемую деталь 5. Схема такого волоконного лазера приведена на рис. 4.

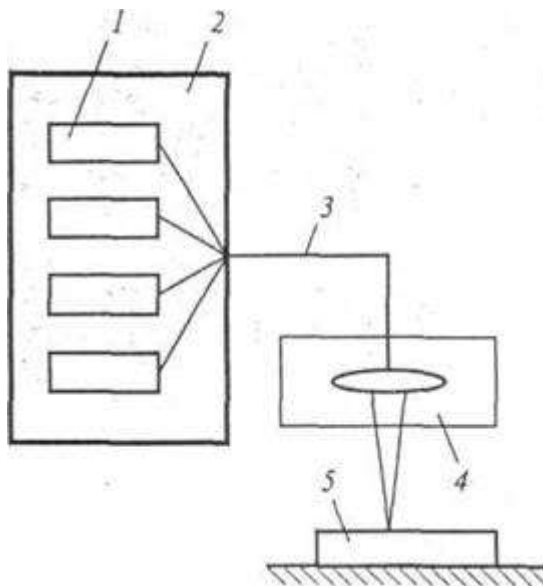


Рис. 4. Схема мощного волоконного лазера

1 – волоконные лазеры мощностью 100 Вт; 2 – мощный волоконный лазер;

3 – волокно, передающее суммированное излучение; 4 – фокусирующая система;

5 – обрабатываемая деталь

Набор одномодовых излучений обеспечивает суммированное излучение, близкое к одномодовому. Особенностью волоконных лазеров является то, что они работают только в непрерывном режиме, так как волокно не может выдерживать гигантские импульсы излучения. При длине волны 1,06 мкм такое излучение весьма эффективно при обработке различных материалов и в совокупности с простотой и надежностью волоконного лазера делает всю систему наиболее целесообразной для технологического применения.

Лазеры на оптических волокнах, легированных ионами редкоземельных элементов, с оптической накачкой обладают рядом преимуществ:

- благодаря волоконной структуре волоконные лазеры имеют низкие пороги генерации;
- их удобно использовать в качестве источников излучения ВОСП из-за простоты сопряжения с волокном линии;
- эффективно использование направленных разветвителей для расщепления пучка, так как при этом исключаются дифракционные потери на апертурах объемных элементов.

#### **4. Основные преимущества волоконных лазеров в сравнении с другими видами лазеров**

Волоконный лазер имеет ряд преимуществ перед другими видами лазеров. Разделим их на три группы: оптические, энергетические и технологические.

##### **Оптические.**

Длина излучения волны у волоконного лазера  $\lambda=1,09$  мкм. Такая длина волны дает волоконному лазеру ряд преимуществ:

1) Излучение с такой длиной волны будет прекрасно фокусироваться через стеклянные линзы, что позволяет сэкономить денежные средства при установке фокусирующей системы.

2) Излучение с такой длиной волны может передаваться по волокну на большие расстояния. Поэтому сама лазерная установка может находиться в удобном для работы месте, а волокно от лазерной установки уже непосредственно протягивается на место сварки.

3) Такое коротковолновое излучение очень интенсивно поглощается металлом:

- малый размер выходной апертуры луча (300 мкм) позволяет сфокусировать конечный лазерный луч в очень маленькую точку.

- у волоконного лазера малая расходимость луча, следовательно, увеличивается фокусное расстояние.

#### **Энергетические:**

- у волоконного лазера высокий КПД источника ( $\eta=35\%$ ), в то время как у других лазеров КПД достаточно мал. Например, у газовых лазеров он составляет  $\eta=5\%$ .

- возможность создания излучателей высокой мощности до 100кВт путем объединения излучений нескольких волоконных лазеров в одно.

- малая теплоотдача, не требует интенсивного охлаждения, а это значит, что снижается суммарное потребление энергии и лазер становится компактнее и проще в обслуживании и ремонте.

#### **Технологические:**

- для волоконных лазеров практически не требуется такое техническое обслуживание, как настройка, юстировка, чистка и др.

- допускает размещение в обычных рабочих помещениях цехов без учета специальных требований.

- компактность установок обусловлена тем, что лазер может занимать удобное для работы месторасположение, даже если оно находится на значительном расстоянии от места сварки и обработки деталей.

- возможность передачи излучения по световоду.

- срок работы до 100000 часов, так как большой нагрузки диоды и волокно не испытывают.

- отсутствие настроечных операций на лазере.

- стеклянная оптика (использование стеклянных фокусирующих линз) позволяет снизить затраты на фокусирующую систему.

- высокая эффективность проплавления.

### **5. Эффективность воздействия излучения волоконных лазеров на материалы**

Связь между основными источниками энергии и различными видами энергозатрат можно выразить с помощью следующих уравнений баланса:

$$Q_{\lambda} = Q_{\lambda} - (Q_{\phi} + Q_{отгр} + Q_{разр}) \quad (1)$$



$$Q_{\phi} = Q'_{\phi} + Q''_{\phi} \quad (2)$$

$$Q'_l + Q'_{\phi} = Q_s + Q_m \quad (3)$$

где  $Q'_l$  - энергия лазерного луча, поглощенная в процессе фотонных и электронных соударений на стенках канала;  $Q_s$  - энергия сфокусированного лазерного излучения;  $Q_{\phi}$  - энергия, поглощенная в объеме плазменно – парового факела внутри кратера и над его поверхностью;  $Q_{отр}$  - энергия лазерного луча, отраженная от поверхности основного металла и дна кратера;  $Q_{разр}$  - полная энергия продуктов разрушения, уносимая реактивной парогазовой струей;  $Q'_{\phi}$  - энергия, рассеиваемая в окружающую среду излучением из плазменного факела;  $Q''_{\phi}$  - часть энергии  $Q_{\phi}$  факела, поглощаемая стенками канала в результате конвективного и лучистого теплообмена;  $Q_s$  - полное теплосодержание жидкого металла в сварочной ванне;  $Q_m$  - энергия, отводимая вглубь основного и переплавленного металла вследствие теплопроводности.

Направления и расположения энергий, подведенных и выделившихся при проплавлении, показаны на рис. 5.

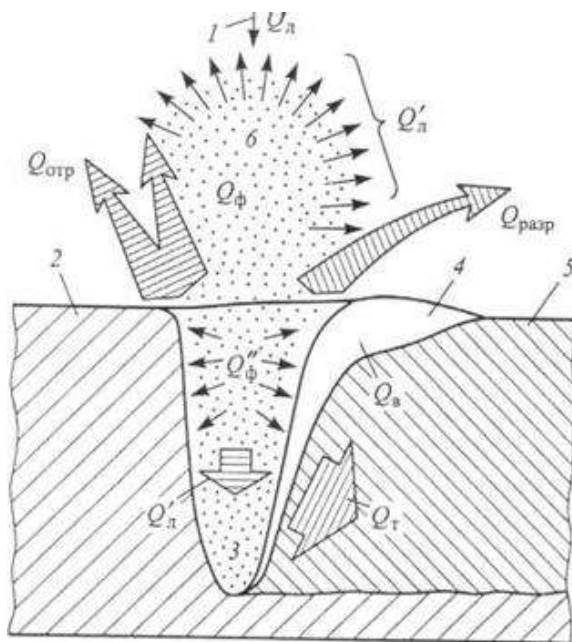


Рис. 5. Схема энергетических затрат в установившемся режиме проплавления металла сфокусированным лазерным излучением

Эффективность передачи энергии лазерного излучения обрабатываемому материалу определяется эффективным коэффициентом поглощения, который фактически характеризует эффективный КПД  $\eta_{эф}$  процесса лазерной обработки. Приближенное значение этого коэффициента можно получить из системы уравнений в виде

$$A_{эф} = \eta_{эф} = (Q_a' + Q_{\phi}'' ) / Q_a$$

(4)

На основе анализа многочисленных теоретических и экспериментальных данных об оценке тепловой эффективности различных процессов лазерной обработки можно установить соотношение между плотностью мощности и предельной температурой в зоне обработки при непрерывной генерации излучения.

### **Задания к лабораторной работе и порядок действий при их выполнении**

1. Составьте перечень возможных вариантов программы задания одной и той же средней скорости перемещения лазерного луча гравера. Скорость перемещения лазерного луча задается не в явном виде, а вычисляется системой на основе введенных пользователем параметров. Этими параметрами являются `step_size` и `step_period`. Они задаются в диалоговом окне `Advanced controls`. Перемещение луча по длине вектора производится с помощью так называемых микровекторов. По длине вектора располагаются несуществующие точки с шагом, равным параметру `step_size`. Луч переходит последовательно из одной к другой такой точке с максимально возможной скоростью через равные интервалы времени, равные `step_period`. Таким образом, средняя скорость движения луча вдоль вектора составляет `step_size/step_period`. Если бы механическая сканаторная система обладала нулевым моментом инерции, то мы бы наблюдали попадание импульсов лазера только в точки стояния, т.е. на границах микровекторов, т.к. импульсы модуляции не синхронны с движением зеркал. Но благодаря инерционности сканаторной системы, перемещение луча происходит более или менее равномерно, и импульсы лазера распределяются по длине вектора соответствующим образом. Артефакты, связанные с только что описанным принципом работы

системы управления, могут наблюдаться на скоростных сканирующих системах при задании чрезвычайно больших значений `step_period` и `step_size`, когда луч перепрыгивает относительно большие расстояния и находится относительно долго в одном положении, т.е. когда инерции отклоняющей системы недостаточно для сглаживания ступенчатого воздействия со стороны системы управления. При этом на маркируемой поверхности лазер оставляет отпечатки в виде неравномерно расположенных вдоль вектора точек, сгруппированных вокруг точек стояния луча. Так как одну и ту же среднюю скорость перемещения луча можно задать разными сочетаниями периода и размера микровекторов, производитель лазерной установки рекомендует задавать минимально возможный (100 мкс) период, и исходя из требуемой скорости, рассчитывать размер микровекторов. Для этого в меню программы предусмотрен соответствующий калькулятор скорости. Что касается числа проходов, данная величина задается в тексте программы при организации цикла `for`. Переменная-счетчик изменяется от 1 (один проход) до `n`, где `n` – требуемое число проходов. При выборе режимов для проведения эксперимента будем руководствоваться рекомендациями производителя, и установим `step_period` равным 100, изменение скорости будем производить за счет подбора значений `step_period`. Число проходов также будет изменяться в соответствии с изменением скорости: очевидно, что на высокой скорости при малом числе проходов деталь вырезана не будет, так как при данной мощности и скорости не будет обеспечена абляция материала по всей толщине заготовки.

2. Подберите режим резания, оптимально удовлетворяющий следующим требованиям (точность): допустимое отклонение реального размера от программного – 0,05 мм; эргономичность: готовая деталь должна легко извлекаться из заготовки без риска ее деформирования в процессе выемки; время обработки должно быть минимизировано; следует стремиться к уменьшению количества нагара, для этого варьируйте параметрами: скорость и число проходов лазерного луча по заданной траектории.

## **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Каков общий план технологического процесса резки на лазерном гравере ARGENT (ЦЛТ)?
2. Исходя из составленного плана резки, определите последовательность программного выставления параметров и настроек лазерного гравера.
3. Для чего при планировании необходимо изменять параметры: период и размер микровекторов. Каковы рекомендации производителя?
4. Каким информационным ресурсом нужно воспользоваться, чтобы найти инструкцию менеджмента безопасности лабораторий кафедры нанотехнологий, общей и прикладной физики?
5. На основе полученного опыта работы с лазерным гравером сделайте предположение о возможности его использования для резки кремниевых пластин.
6. Каков общий план технологического процесса скрайбирования на лазерном гравере ARGENT (ЦЛТ)?

## ПРАКТИЧЕСКАЯ РАБОТА № 1.

### Проектирование установок с использованием лазерного модуля КУ-008.

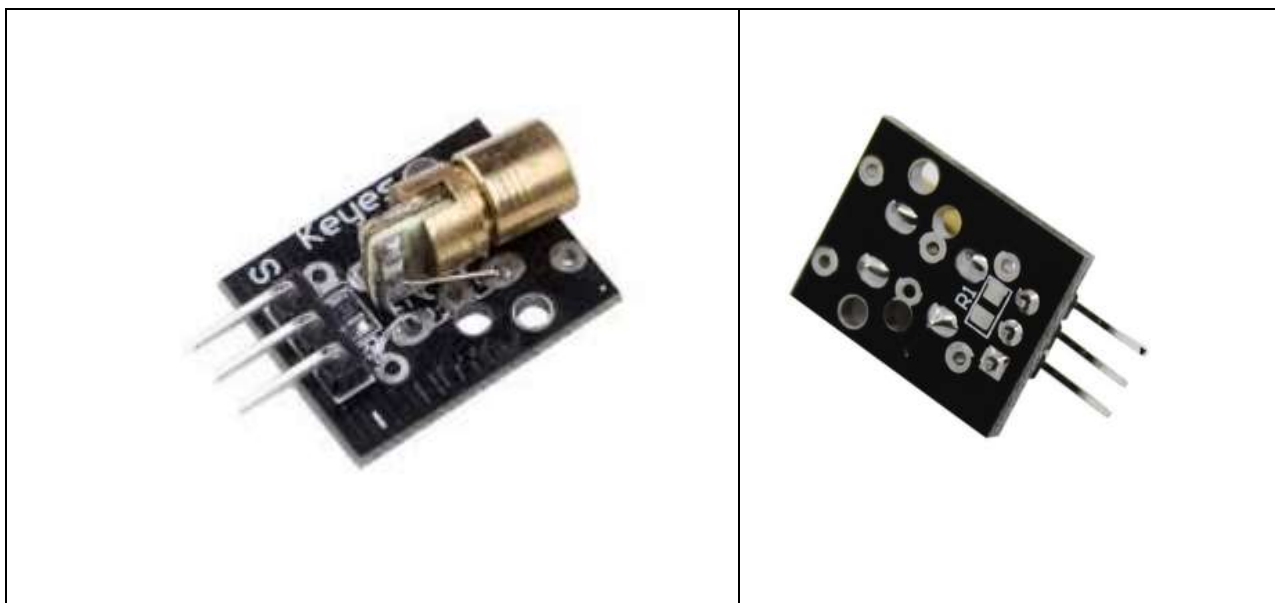
Спецификация лазерного модуля (КУ-008 datasheet):

<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)

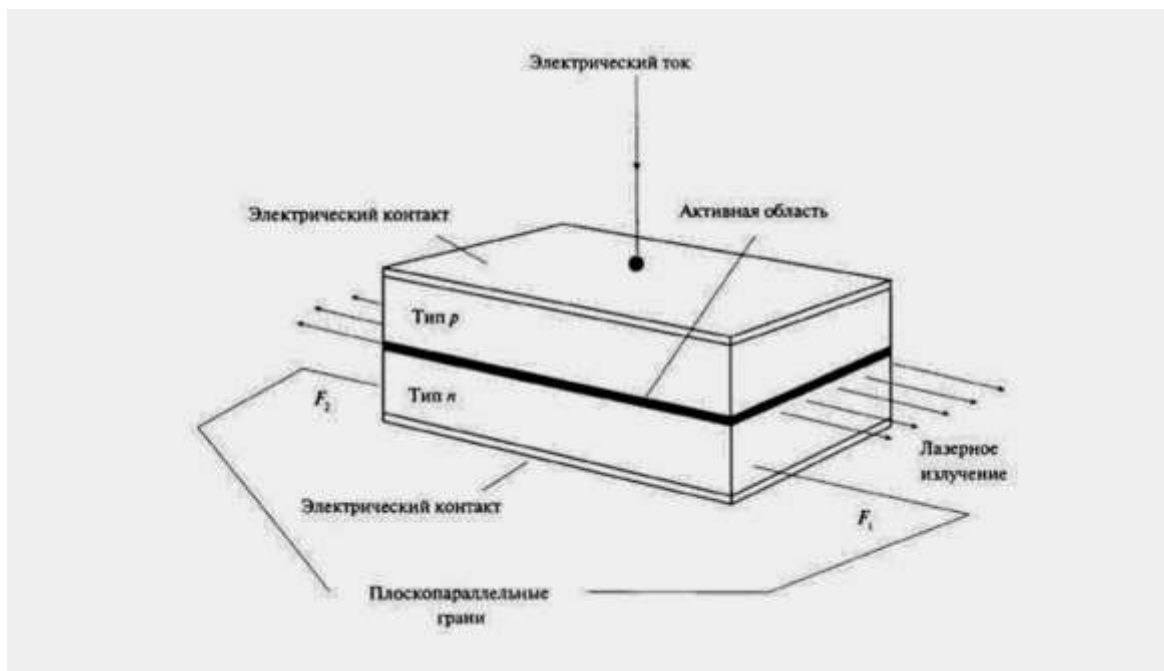


#### Принцип работы лазерного светодиода

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 КУ-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.

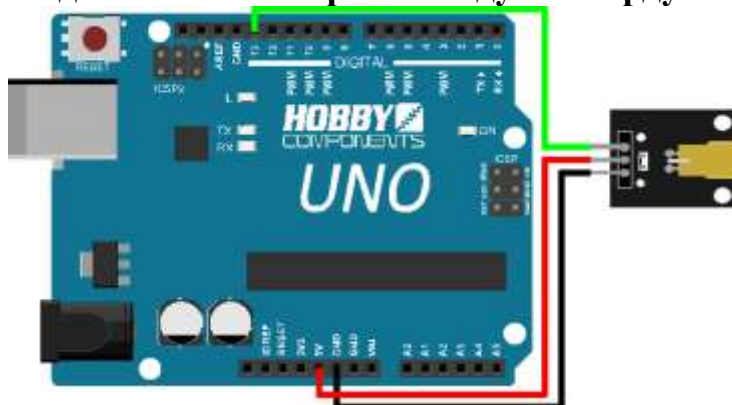


В основе модуля лазера "KY-008" лежит светодиод и два резистора на 1кОм и 91Ом, расположенных на плате с тремя выводами.

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.

### 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:

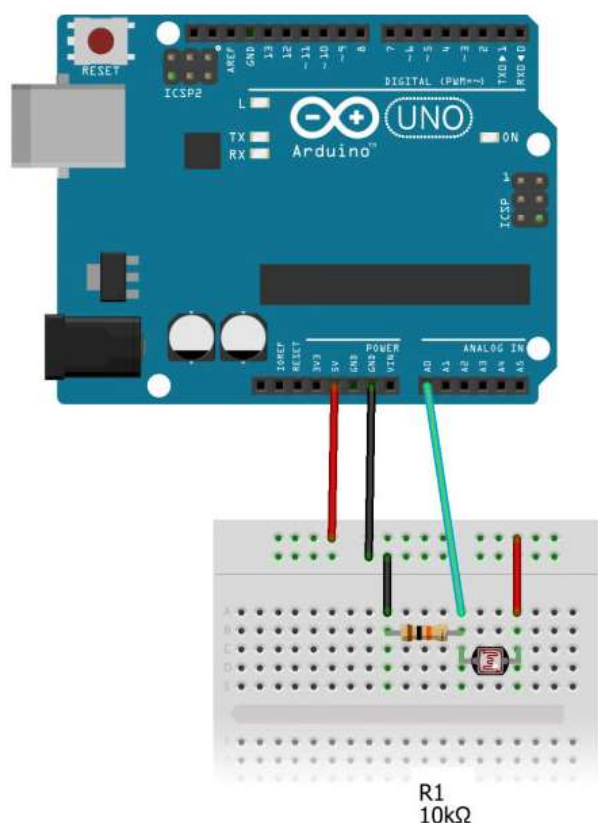
- [Arduino Uno](#) или Arduino-совместимая плата с аналоговыми входами.
- [Arduino IDE](#) (интегрированная среда разработки).
- Фоторезистор (фотоэлемент)
- Один резистор 10 кОм.
- Один потенциометр (переменный резистор).
- Макетная плата и соединительные провода.

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.

### **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](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.

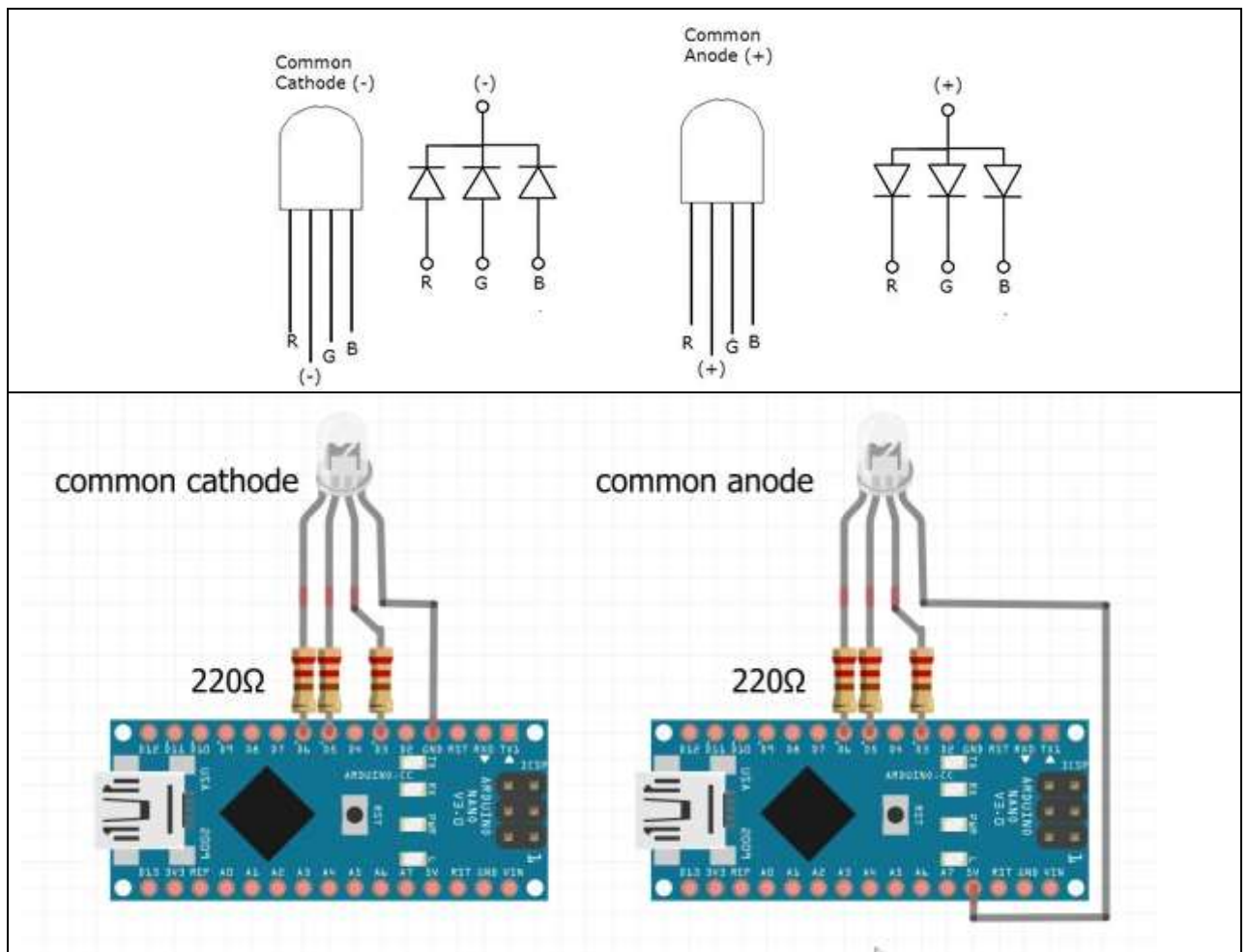


Рисунок 1 - Распиновка RGB светодиода

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».

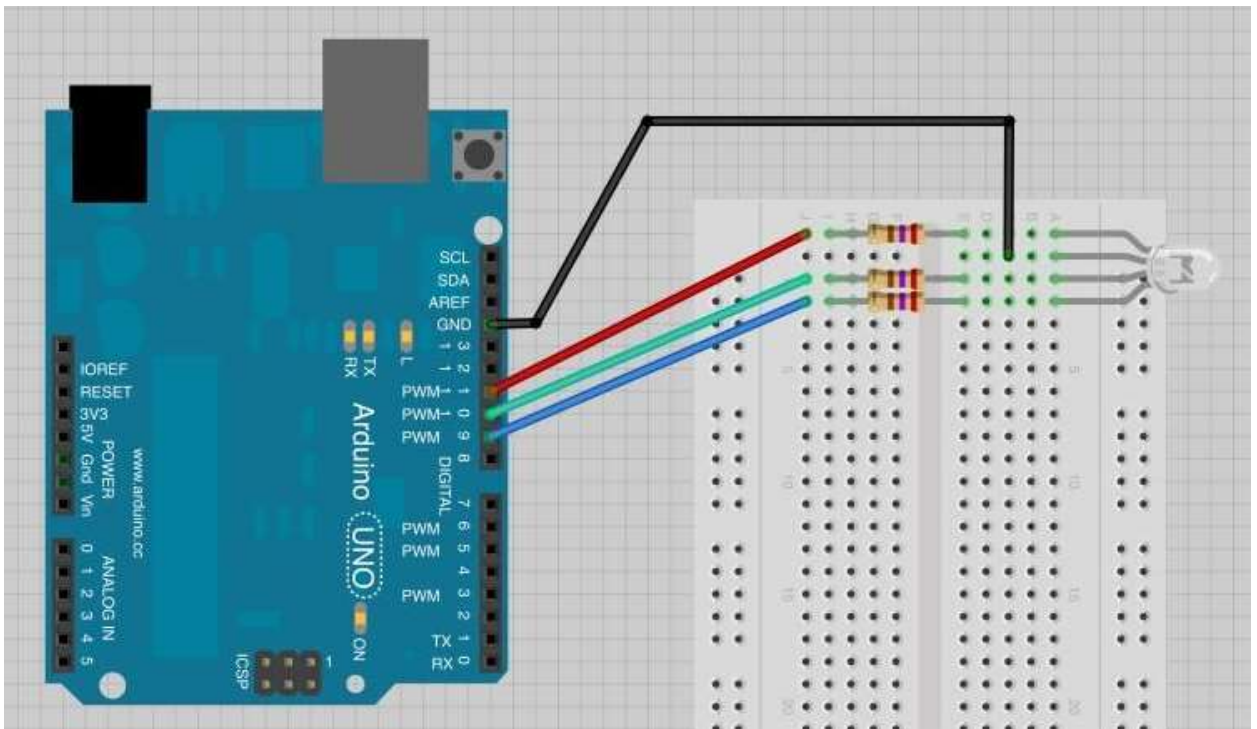


Рисунок 2 - Схема подключения RGB LED к Ардуино на макетной плате

Модуль «RGB светодиод» 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 «широтно 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

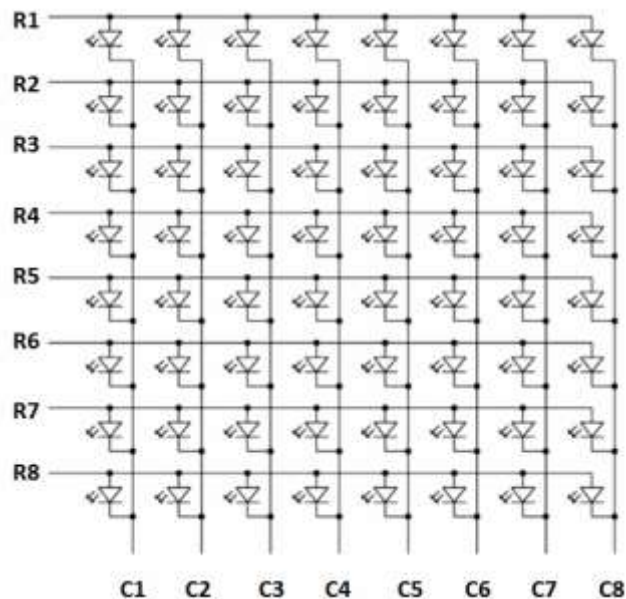
Сегментные индикаторы (шкалы или цифры) состоят из отдельных светодиодов, соединенных вместе. Например, у группы светодиодов могут быть соединены все катоды. Такой индикатор имеет приписку «с общим катодом», в противном случае — «с общим анодом».

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

Если требуется светодиодная матрица с большим разрешением, то её просто-напросто составляют из нескольких  $8 \times 8$  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								

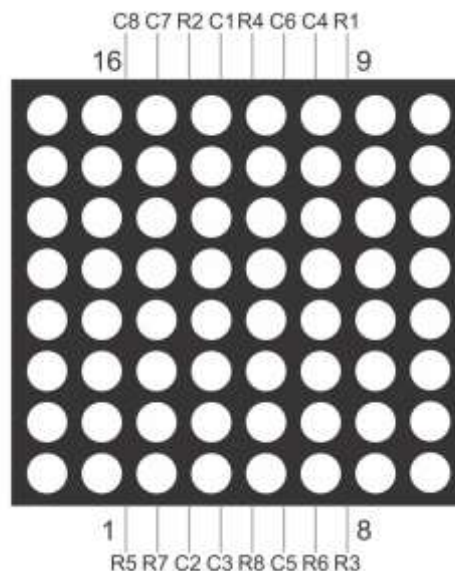
	C1	C2	C3	C4	C5	C6	C7	C8
R1								
R2								
R3								
R4								
R5								
R6								
R7								
R8								

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

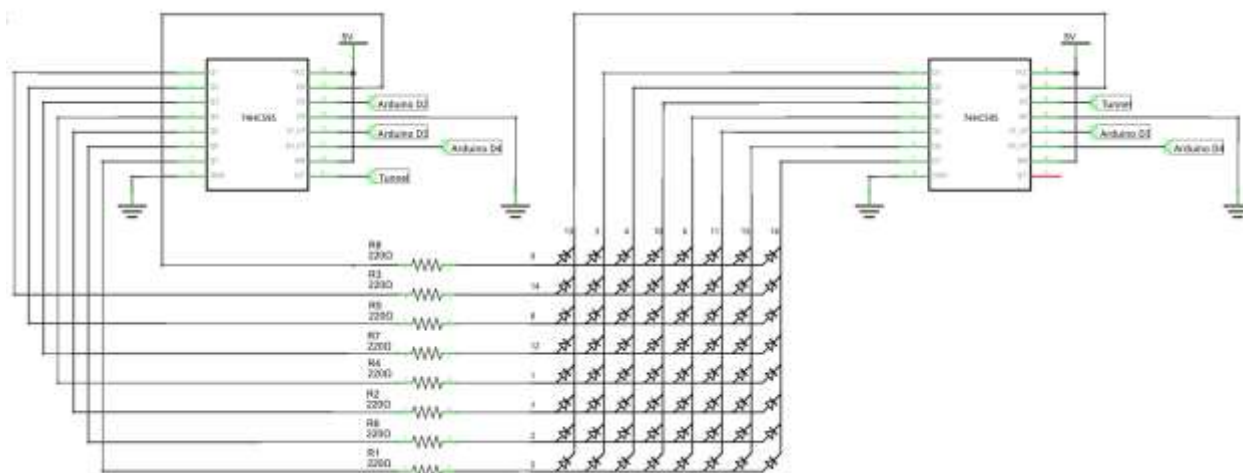
В светодиодной матрице 8×8 красного свечения нумерация выводов начинается с нижнего левого угла. При этом, нумерация ног 1-16 не связана никакой логикой с нумерацией колонок и строк C и R.



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.

### 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;

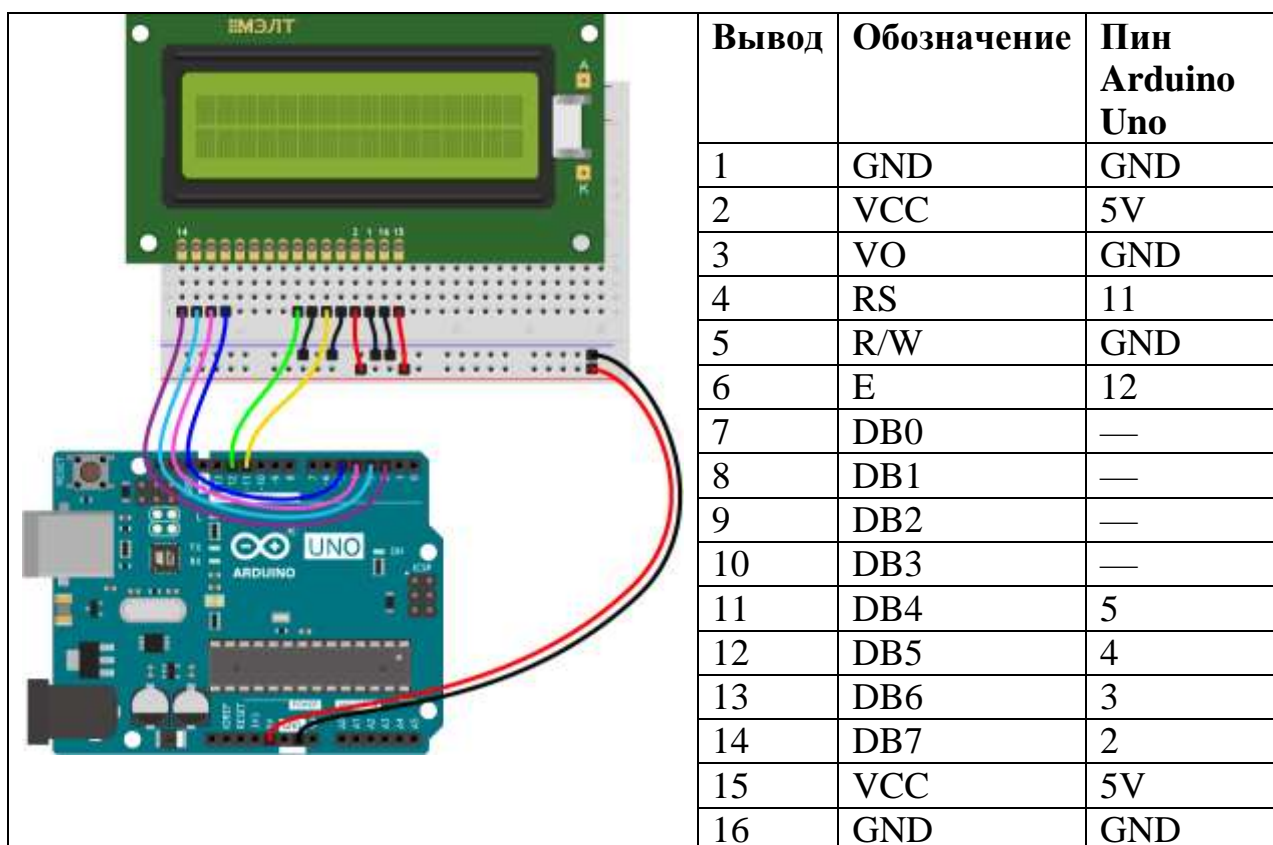
15. Плюс подсветки;

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



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
```



```

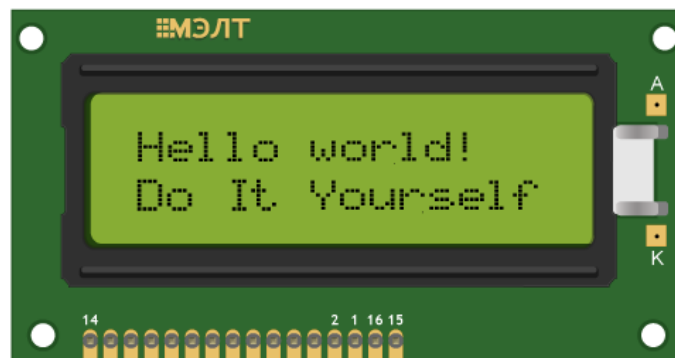
#include <LiquidCrystal.h>

// 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	У	x	½	!	1	A	Q	a	q	1	◀	±	Б	С	б	с					
2	x	÷	"	2	B	R	b	r		+	ё	б	ъ	ш	щ	x	¾	"	2	B	R	b	r	▶	±	Б	Т	б	т						
3	x	→	#	3	C	S	c	s		◊	Ж	В	ы	!!	д	x	¾	#	3	C	S	c	s	▶	±	Б	У	б	у						
4	x	←	\$	4	D	T	d	t	▶	✓	З	Г	ь	ъ	Ф	x	÷	\$	4	D	T	d	t	!!	...	Ю	«	Д	Ф	д	ф				
5	x	\	%	5	E	U	e	u	◀	i	И	ё	э	х	ц	x	≡	%	5	E	U	e	u	...	!	¥	"	Е	Х	е	х				
6	x	г	&	6	F	V	f	v	◀	1	И	ж	ю	ш	щ	x	г	&	6	F	V	f	v	▶	и	о	ч	ж	ц	ж	ц				
7	x	н	'	7	G	W	g	w	▲	²	Л	э	я	!	ё	x	✓	'	7	G	W	g	w	↓	ш	§	f	З	Ч	з	ч				
8	x	б	0	<	8	H	h	x	Р	³	П	и	«	И	"	ё	x	Р	▲	<	8	H	h	x	€	у	ё	И	Ш	и	ш				
9	x	µ	0	>	9	I	Y	i	y	т	°	У	й	»	↑	~	ё	x	°	т	у	>	9	I	Y	i	у	и	°	и	Ш	й	ш		
A	x	ú	≤	*	:	J	Z	j	z	-	€	Ф	к	«	↓	ё	x	▲	≤	*	:	J	Z	j	z	ø	æ	Е	е	К	ь	к	ъ		
B	x	0	≥	+	;	K	[	k	[	◀	■	Ч	л	"	и	ё	x	■	≥	+	;	K	[	k	[	F	f	«	»	Л	Ы	л	ы		
C	x	ï	↓	,	<	L	φ	l	φ	)	■	Ш	м	ъ	и	у	x	◊	◊	,	<	L	\	l		К	к	◊	М	ь	м	ь			
D	x	ï	¥	-	=	M	]	m	]	т	°	ь	н	¿	и	ё	x	◊	р	-	=	M	]	m	]	и	и	-	и	Э	и	э	и		
E	x	€	≠	.	>	N	^	n	^	◀	◊	Ы	п	ф	¿	и	ё	x	◊	≠	.	>	N	^	n	^	~	у	у	ø	»	О	Ю	о	ю
F	x	€	≠	.	>	N	^	n	^	◀	◊	Ы	п	ф	¿	и	ё	x	◊	≠	.	>	N	^	n	^	~	у	у	ø	»	О	Ю	о	ю
	x	€	≠	.	>	N	^	n	^	◀	◊	Ы	п	ф	¿	и	ё	x	◊	≠	.	>	N	^	n	^	~	у	у	ø	»	О	Ю	о	ю
	x	€	≠	.	>	N	^	n	^	◀	◊	Ы	п	ф	¿	и	ё	x	◊	≠	.	>	N	^	n	^	~	у	у	ø	»	О	Ю	о	ю

Для вывода символа на дисплей необходимо передать его номер в шестнадцатеричной системе из таблицы знакогенератора.

Так букве Я соответствует код B1 в шестнадцатеричной системе. Чтобы передать на экран строку «Яндекс», необходимо в явном виде с помощью последовательности \x## встроить в строку код символа:

```
lcd.print("\xB1ndex");
```

Вы можете смешивать в одной строке обычные символы и явные коды как угодно. Единственный нюанс в том, что после того, как компилятор в строке видит последовательность \x, он считывает за ним все символы, которые могут являться разрядами шестнадцатеричной системы даже если их больше двух. Из-за этого нельзя использовать символы из диапазона 0-9 и A-F следом за двузначным кодом символа, иначе на дисплее отобразится неправильная информация. Чтобы обойти этот момент, можно использовать тот факт, что две записанные рядом строки склеиваются.

Сравните две строки кода для вывода надписи «Яее»:

```
lcd.print("\xB1eee"); // ошибка
lcd.print("\xB1" "eee"); // правильно
```

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");
}

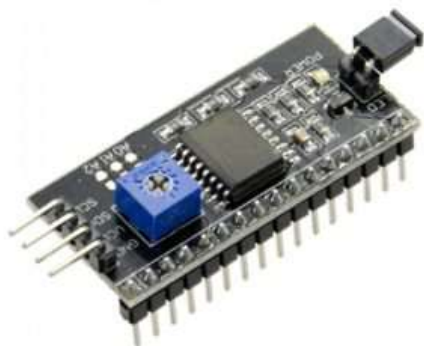
void loop() {
}
```

### **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.

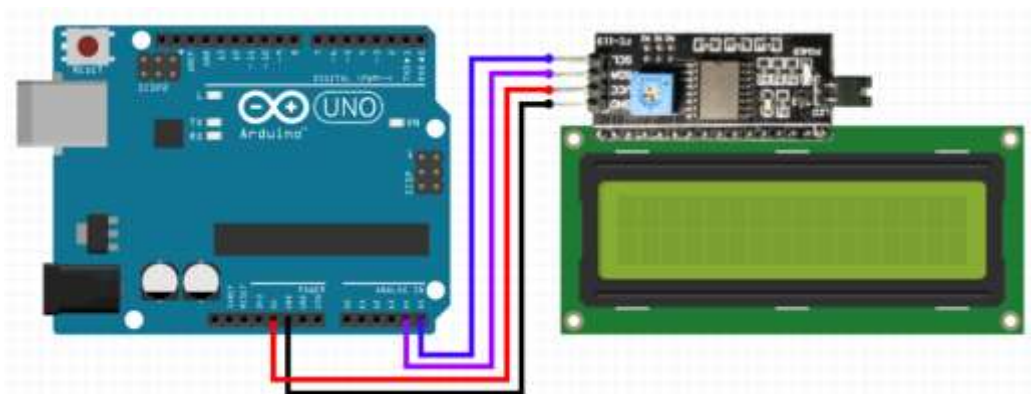
Жидкокристаллический монитор с поддержкой i2c подключается к плате при помощи четырех проводов – два провода для данных (к аналоговым входам), два провода для питания.

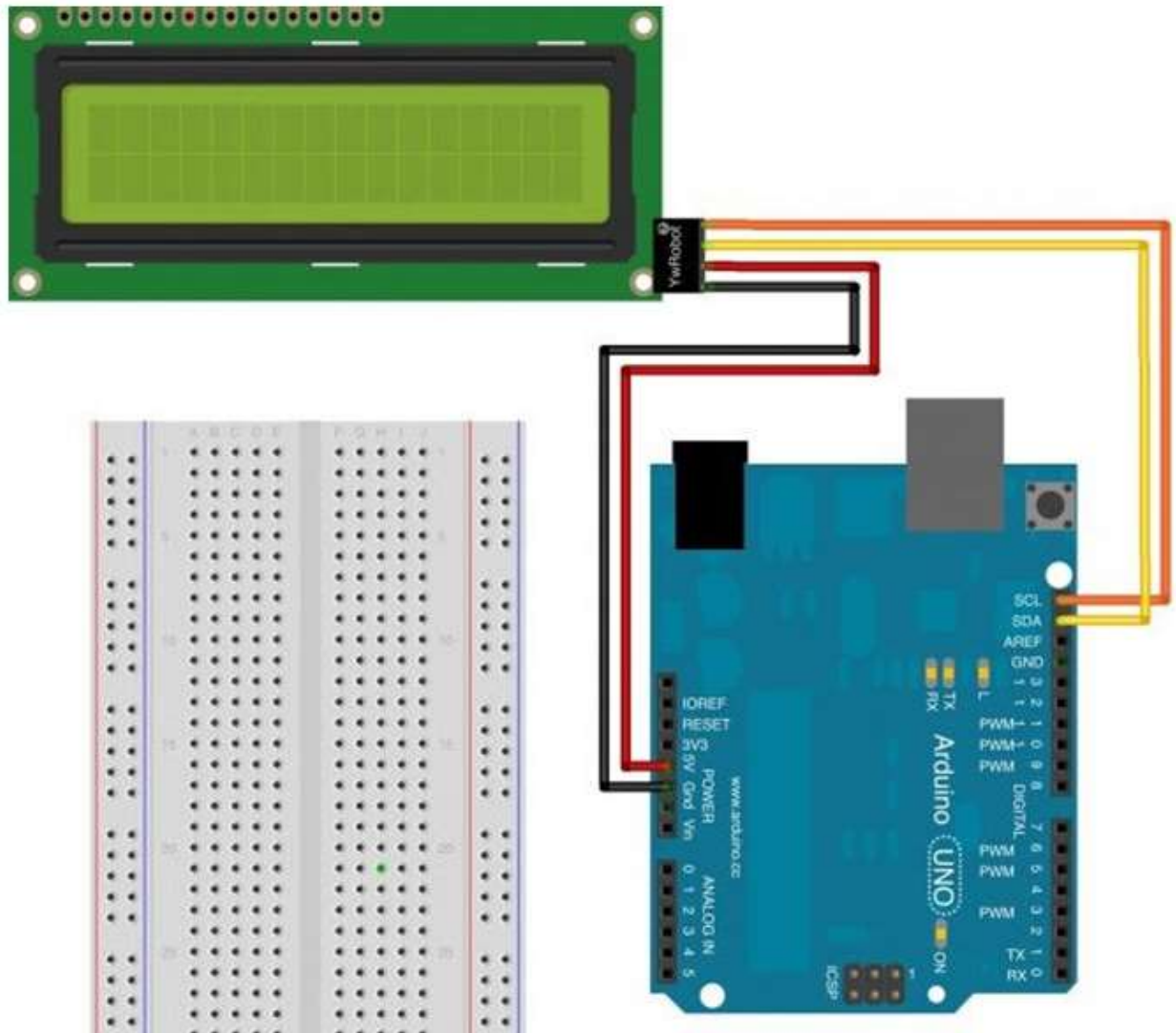
Вывод GND подключается к GND на плате.

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/>