МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Радиофизический факультет
Кафедра полупроводниковой электроники

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК
Руководитель ООП
к.ф.-м.н., доцент
_____ В.А. Мещеряков
«25» января 2022 г.

ДИПЛОМНАЯ РАБОТА

АВТОМАТИЗАЦИЯ СИСТЕМЫ ИЗМЕРЕНИЯ СПЕКТРОВ ЛЮМИНЕСЦЕНЦИИ ПОЛУПРОВОДНИКОВЫХ СТРУКТУР НА БАЗЕ МОНОХРОМАТОРА МДР-23 И МИКРОКОНТРОЛЛЕРА ATmega328P

по основной образовательной программе подготовки специалиста
по специальности 11.05.01 «Радиоэлектронные системы и комплексы»

Григорьева Алексея Андреевича

Руководитель ВКР:
к.ф.-м.н., доцент
_____ В.В. Копьев
«25» января 2022 г.

Автор работы:
студент группы №7610
_____ А. А. Григорьев

Томск – 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Радиофизический факультет
Кафедра полупроводниковой электроники

УТВЕРЖДАЮ
Руководитель ООП
к.ф.-м.н., доцент
_____ В.А. Мещеряков
«21» декабря 2021 г.

ЗАДАНИЕ

на подготовку ВКР специалиста

студенту Григорьеву Алексею Андреевичу группы №7610

1. Тема задания: Автоматизация системы измерения спектров люминесценции полупроводниковых структур на базе монохроматора МДР-23 и микроконтроллера ATmega328P

2. Срок сдачи студентом выполненной ВКР:

а) на кафедре 25.01.2022,

б) в ГАК 02.02.2022.

3. Краткое содержание работы:

Разработка программы управления системой измерения спектров электролюминесценции с использованием среды разработки LabVIEW. Программа управления должна сочетать в себе следующие функции: управление измерителем фотосигнала, управление шаговым двигателем монохроматора с переменной частотой и визуализация результатов измерений.

4. Календарный график выполнения ВКР:

а)	изучение литературы	22.12.2021 – 28.12.2021
б)	разработка программного обеспечения автоматизации спектров электролюминесценции светодиодов при помощи монохроматора МДР-23	24.12.2021 – 28.12.2021
в)	разработка и сборка принципиальной схемы	24.12.2021 – 12.01.2022
г)	Написание ВКР и сдача в ГАК, размещение текста ВКР в электронной библиотеке ТГУ	12.01.2022 – 02.02.2022

5. Дата выдачи задания «21» декабря 2021 г.

Руководитель ВКР кандидат физ.-мат. наук,
доцент кафедры полупроводниковой электроники _____ В. В. Копьев

Задание принял к исполнению _____ А. А. Григорьев

РЕФЕРАТ

МОНОХРОМАТОР, СПЕКТР, LABVIEW, МИКРОКОНТРОЛЛЕР Arduino Uno

Объектом исследования является монохроматор МДР-23.

Цель работы – создание устройства для автоматизации работы монохроматора МДР-23.

В процессе работы в программе LabView был создан виртуальный прибор, обеспечивающий функционирование источника-измерителя и монохроматора в полуавтоматическом режиме. Разработана печатная плата на основе микроконтроллера Arduino Uno в программном комплексе KiCad для автоматизации процессам управления.

В результате работы:

а) проведен литературный обзор по теме научно-исследовательской работы;

б) изучен принцип работы экспериментальной установки;

в) создан виртуальный прибор в среде разработки LabView;

г) изучен программный комплекс KiCad;

д) модернизирована печатная плата для автоматизации процессами;

е) выполнены практические измерения;

ж) выполнен анализ результатов и написан отчет.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 6
1 Спектры люминесценции и способы их измерения 8
1.1 Монохроматор МДР-23 8
1.1.1 Применение монохроматора 8
1.1.2 Устройство и работа монохроматора 9
1.1.3 Блок управления шаговым двигателем монохроматора 10
1.1.4 Конструкция блока 11
1.1.5 Кинематическая схема монохроматора 12
1.1.6 Конструкция монохроматора МДР-23 14
1.2 Излучение света в полупроводниках 18
1.2.1 Люминесценция полупроводников 18
1.2.2 Характеристики люминесцирующих веществ 20
1.2.3 Электролюминесценция 20
1.2 Микроконтроллер Arduino Uno и ATmega328 22
1.3.1 Микроконтроллер Arduino Uno 22
1.3.2 Микроконтроллер ATMega328 24
1.4 Графическая среда разработки приложений LabVIEW 26
1.5 Программный комплекс KiCad 28
1.6 Источник-измеритель Keithley 2611B 30
2 Методика измерений 33

2.1 Принцип автоматизации монохроматора МДР-23 33
2.2 Запуск LabVIEW..... 37
2.2.1 Создание виртуального прибора 38
3 Экспериментальная часть..... 43
3.1 Лицевая панель виртуального прибора 43
3.2 Блок-диаграмма виртуального прибора..... 44
3.3 Прошивка микроконтроллера Atmega328p с помощью ArduinoUNO... 51
3.4 Код программы для загрузки в микроконтроллер. 52
3.5 Схема автоматизации работы монохроматора МДР-23..... 57
4 Результаты измерений и обсуждения 57
ЗАКЛЮЧЕНИЕ 60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 61

ВВЕДЕНИЕ

Спектральные приборы во всем их многообразии являются мощным средством исследования структуры и состава различных веществ, в каком бы агрегатном состоянии оно не находилось. Как известно, роль спектрального прибора заключается в том, чтобы сложное, полихроматическое излучение оптического диапазона, падающее на его входное отверстие, разложить и зафиксировать положение отдельных спектральных линий, измерить интенсивность того или иного участка спектра, в той или иной спектральной линии.

Стандартные решения для автоматизации спектральных измерений, используемые подавляющим большинством производителей, представляют собой моноблочные приборы одноцелевого назначения (спектрофотометры, спектрофлуориметры и пр.). Связь таких приборов с внешним компьютером осуществляется по стандартным интерфейсам.

Такая аппаратура не допускает не только дооснащения дополнительными устройствами, работающими в едином программном поле с основными, но и никаких методических усовершенствований режимов измерения, даже если оптические узлы аппаратуры это позволяют.

В последнее время появилась возможность на базе компьютерных технологий, в том числе на базе решений фирмы National Instruments (которое включает в себя среду разработки приложений LabVIEW), значительно упростить решение такого рода задач. Можно уверенно сказать, что построение систем отображения и управления различными физическими и технологическими процессами является основной функцией программно-аппаратного комплекса фирмы National Instruments. При этом управление происходящими процессами может быть либо прямым, либо осуществляться через специальные программируемые логические контроллеры (programm able

logical controllers – PLC), что принято называть диспетчерским управлением и сбором данных (super visory control data acquisition – SCADA).

Помимо этого, имеется возможность в рамках одного прибора получать набор выходных аналоговых или цифровых сигналов, которые могут быть использованы для управления реальными объектами. Все это в совокупности легло в основу идеи создания так называемых виртуальных приборов. Аппаратная часть приборов значительно упрощается, включая в себя только унифицированные узлы по сбору данных и платформу, отвечающую требованиям того или иного промышленного назначения.

В связи с этим виртуальная среда разработки LabVIEW представляет в совокупности с устройствами полный аппаратно-программный комплекс для построения систем измерения и автоматизации.

С минимальными усилиями можно строить и программировать собственные платформы, которые будут решать те или иные практические задачи.

Работа была выполнена в коллективе.

1 Спектры люминесценции и способы их измерения

1.1 Монохроматор МДР-23

1.1.1 Применение монохроматора

В работе [1] монохроматор МДР-23 применяется для измерения спектров люминесценции светодиодов, видимого и инфракрасного диапазонов спектра. Под спектром люминесценции понимается зависимость интенсивности излучения от длины волны или энергии фотонов. Принцип действия прибора основан на явлении дисперсии света. Дисперсия света — это совокупность явлений, обусловленных зависимостью абсолютного показателя преломления вещества от частоты (или длины волны) света (частотная дисперсия), или, что то же самое, зависимостью фазовой скорости света в веществе от частоты (или длины волны).

Диспергирующим элементом в монохроматоре может быть дисперсионная призма или дифракционная решётка. В оптике дисперсионная призма — оптическая призма, обычно имеющую форму геометрической треугольной призмы, используемую в качестве спектроскопического компонента. Спектральная дисперсия — наиболее известное свойство оптических призм, хотя и не самая частая цель использования оптических призм на практике. Треугольные призмы используются для спектрального разложения света, то есть для разделения его на спектральные составляющие (цвета радуги для видимой области электромагнитного спектра)

Дифракционная решётка — оптический прибор, действие которого основано на использовании явления дифракции света. Представляет собой совокупность большого числа регулярно расположенных штрихов (щелей, выступов), нанесённых на некоторую поверхность.

Монохроматор МДР-23 (рисунок 1) предназначен для выделения монохроматического излучения в широком спектральном диапазоне. Монохроматическое излучение — электромагнитное излучение, обладающее

очень малым разбросом частот, в идеале — одной частотой (длиной волны). Монохроматор используется в научно-исследовательских и промышленных лабораториях в различных спектроскопических установках при проведении исследовании спектральных характеристик источников и приемников излучении.

Рисунок 1 – Монохроматор МДР-23

1.1.2 Устройство и работа монохроматора

Монохроматор МДР-23 построен по ассиметричной схеме Фасти со сферическими зеркальными объективами.

Оптическая схема монохроматора, показана на рисунке 2. Источник света 1 через конденсор 2 освещает входную щель 3 монохроматора. Поворотное зеркало 4 и зеркальный сферический объектив 5, в фокальной плоскости которого расположена входная щель 3, направляют параллельный пучок на дифракционную решетку 6. После дифракции параллельный пучок фокусируется зеркальным сферическим объективом 7 на выходную щель 10.

В зависимости от положения поворотного зеркала 8 пучок света попадает либо на одну щель 9, либо на другую щель 10. В случае

необходимости использования одновременно обеих выходных щелей в пучок вводится зеркало, срезанное по высоте.

Для проектирования источника света на щель монохроматора служит конденсор поскольку конденсор не ахроматичен, то фокусировка его в различных областях спектра требует изменения расстояния между входной щелью и конденсором и между конденсором и источником света.

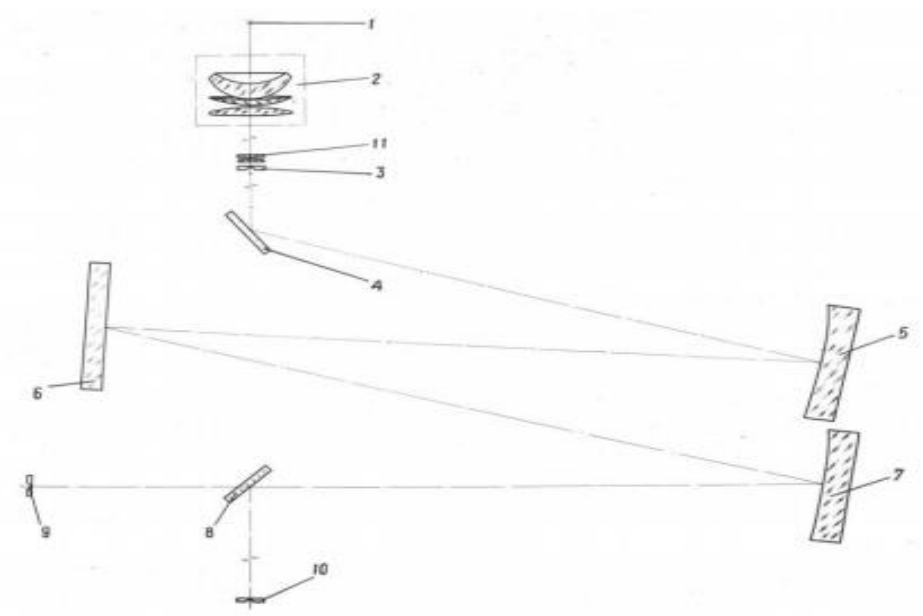

Рисунок 2 – Оптическая схема монохроматора [1]

Данный монохроматор используется при изучении оптических свойств кристаллов и других веществ и материалов в ультрафиолетовой, видимой и ближней инфракрасной областях спектра.

В состав прибора входят монохроматор МДР-23, блок управления, конденсор, рельсы, набор светофильтров и комплект ЗИП.

1.1.3 Блок управления шаговым двигателем монохроматора

Данный блок предназначен для управления шаговыми двигателями типа ШДР-711 в устройствах сканирования и управления другими шаговыми

двигателями точных приводов в различных спектральных приборах. Блок обеспечивает работу в двух режимах: в ручном и автоматическом

В ручном режиме управление шаговым двигателем осуществляется по командам с передней панели блока, скорость вращения шагового двигателя и направление выбирает оператор. В этом режиме работы клавиша РУЧН на передней панели блока должна быть нажата.

В автоматическом режиме управление шаговым двигателем осуществляется по командам, поступающим от внешней электронно-вычислительной машины. Скорость вращения шагового двигателя в этом режиме максимальная для данного шагового двигателя, а моменты пуска и направление вращения задаются программой электронно-вычислительной машины. Клавиша РУЧН на передней панели блока в этом режиме работы должна быть отпущена.

1.1.4 Конструкция блока

Общий вид управления шаговым двигателем показан на рисунке 3. Монохроматор управляется с помощью блока 1. На передней панели блока над сетевым выключателем 2 расположен индикатор 3 включения сети, а под сетевым выключателем – сетевой предохранитель 4.

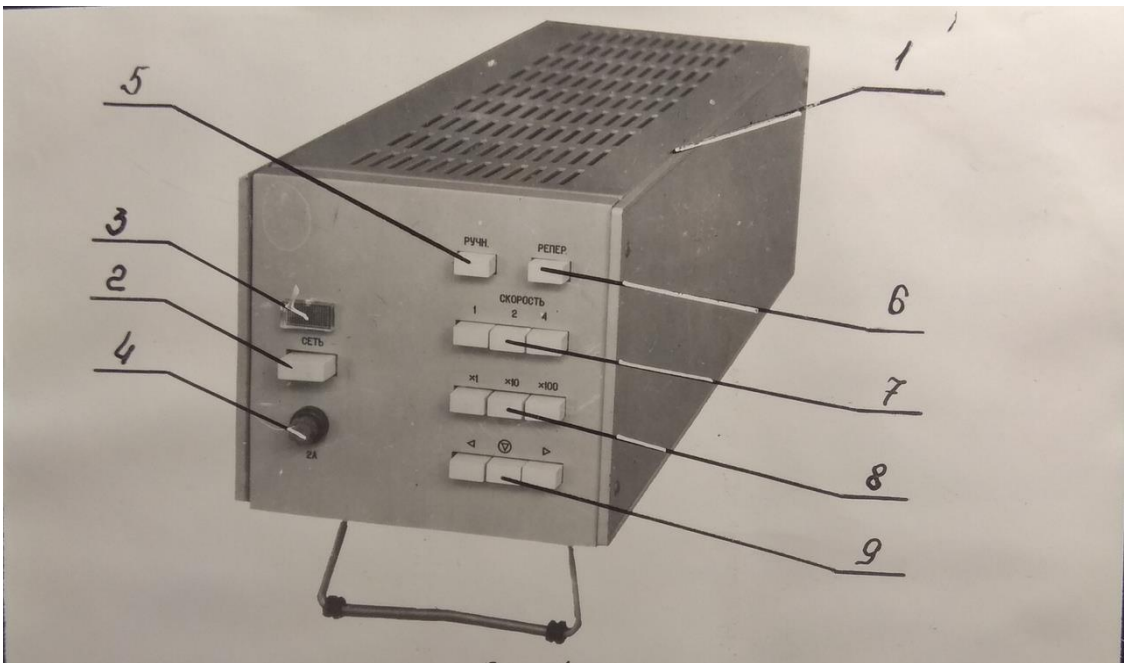

Рисунок 3 – Блок управления шаговым двигателем [1]

При нажатой клавише РУЧН 5 обеспечивается ручной режим работы блока, при отпущенной клавише – автоматический режим работы. Нажатием клавиши 6 РЕПЕР подключается устройство управления реперным отметчиком самописца ЛКС4-003, если самописец работает совместно с блоком. Переключателями 7 и 8 может быть установлена одна из девяти рабочих скоростей, а переключателями 9 направление вращения шагового двигателя в ручном режиме.

1.1.5 Кинематическая схема монохроматора

Кинематическая схема (рисунок 4) обеспечивает поворот дифракционной решетки при развертке спектра и отсчета длины волны излучения, попадающего на выходную щель. Развертка спектра осуществляется автоматически от шагового двигателя.

Поворот решетки производится синусным механизмом состоящим из винта 12, гайки 13 и рычага 14, ось вращения которого совмещена с плоскостью дифракционной решетки.

Показания цифрового счетчика 15 соответствуют длинам волн спектра для решетки 1200 штрихов на миллиметр.

При развертке спектра движение от шагового двигателя 16 передаются через шестерни и червячную пару на винт 12 синусного механизма и на счетчик дли волн.

Для нанесения на запись реперных меток служит реперное устройство. При вращении винта 12 через шестерни 17 движение передается диску 18, помещенному между двумя парами магнитов и герметизированных магнитоуправляемых контактов (герконов). Диск имеет вырезы, обеспечивающие срабатывание геркона и нанесение на запись реперных меток через один нанометр, и одно отверстие, несколько смещенное от центра, обеспечивающее нанесение на запись реперных меток через десять нанометров.

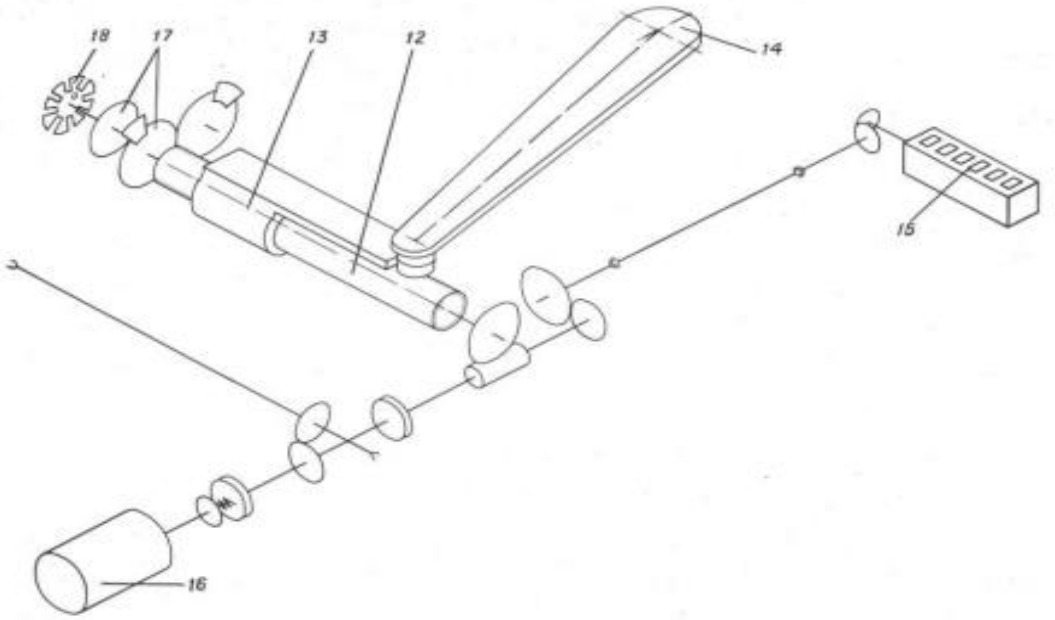

Рисунок 4 – Кинематическая схема монохроматора [1]

1.1.6 Конструкция монохроматора МДР-23

На основании 19 (рисунок 5) монохроматора размещены основные элементы оптической и кинематической схем.

Сменные дифракционные решетки в оправе 20 устанавливают на держатель 21 так, чтобы шаровые опоры 22 вошли в лунки 23, а упор 24 был прижат к площадке держателя.

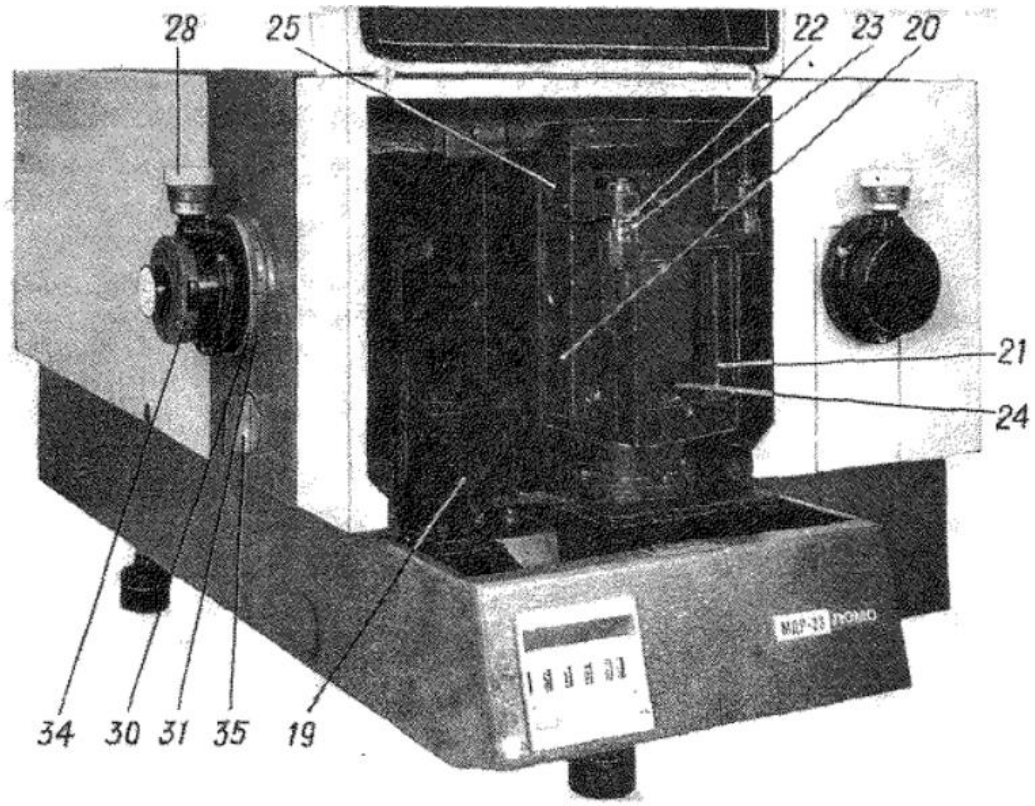

Рисунок 5 – Основание монохроматора МДР-23 [1]

Рукоятка 25 служит для удобства установки решетки. Установку дифракционных решеток осуществляют через окно в корпусе монохроматора, которое закрывается крышкой 26 (рисунок 6). На оправе каждой решетки награвировано соответствующее ей число штрихов на миллиметр, а на оправах решеток 1200 штрихов на миллиметр – дополнительное обозначение «I» или «II», определяющее рабочую область решетки: соответственно от 200 до 500 мм или от 350 до 1000 мм.

Для уменьшения действующего относительного отверстия монохроматора до 1:8 на решетку может быть помещена диафрагма 27 (рисунок 7). В нерабочем состоянии решетки закрываются выдвижными крышками.

Входная и две выходные щели монохроматора – комбинированные с двойным раскрытием; точность отсчета при раскрытии от 0,2 мм – 0,001 мм, при раскрытии от 0,2 мм до 2,2 мм – 0,01 мм. Отсчеты ширины раскрытия входной щели осуществляется по шкале барабанчика 28 (рисунок 5), выходных щелей – по шкалам барабанчиков 29 (рисунок 6).

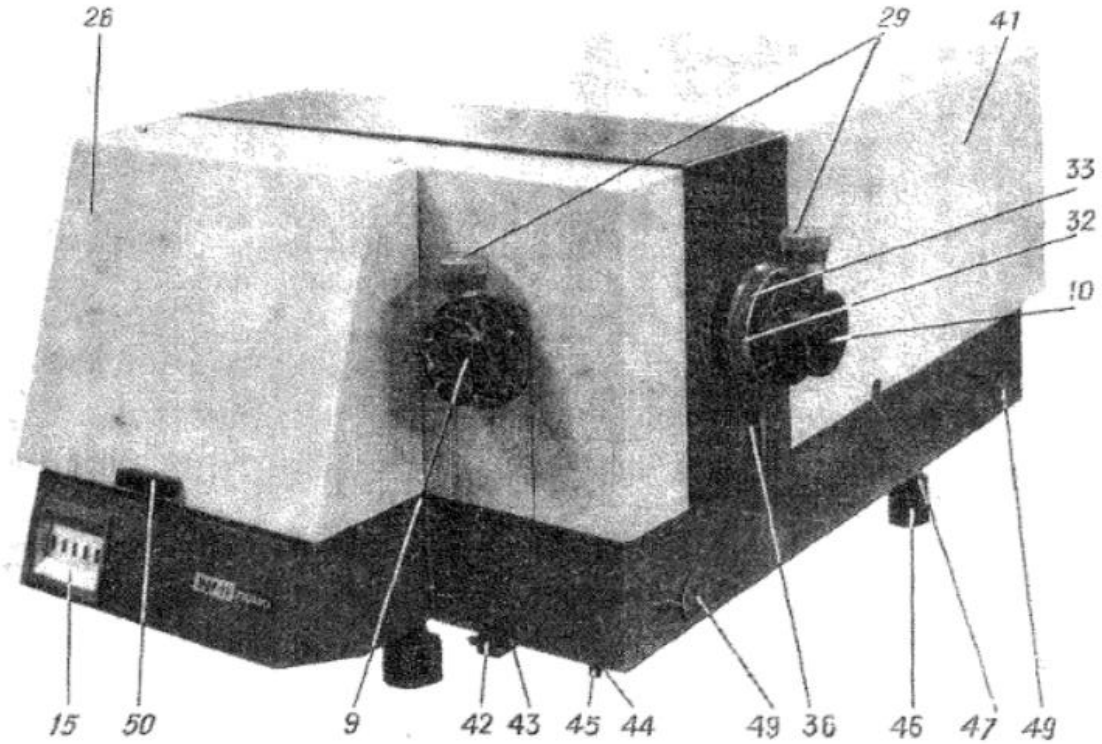

Рисунок 6 – Монохроматор МДР-23 [1]

Ножки входной и выходных щелей изготовлены с радиусом кривизны 130,3 и 140,6 мм соответственно.

Для фокусировки входную щель и выходную щель 10 можно перемещать вдоль оптической оси. Перемещение входной щели осуществляется вращением кольца 30 (рисунок 5), при этом отсчет снимается

по шкале 31, а перемещение выходной щели – вращением кольца 32 (рисунок 6), при этом отсчет снимается по шкале 33.

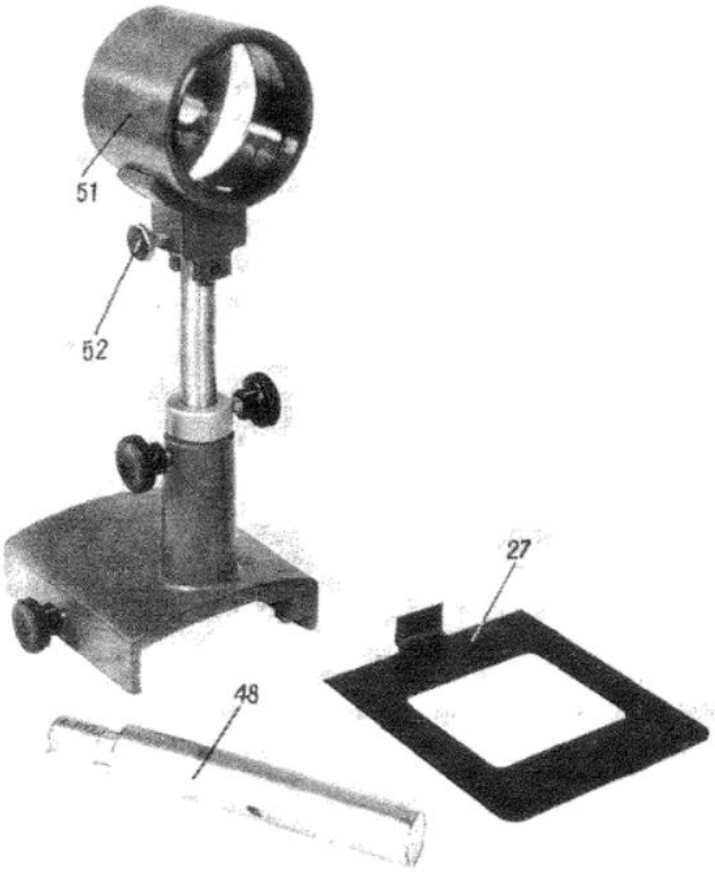

Рисунок 7 – Кварцевый конденсор с диафрагмой [1]

В насадке 34, надеваемой на входную щель, могут быть установлены диафрагмы с фигурными вырезами, ограничивающая высоту щели, и сменные светофильтры. За входной щелью находится затвор, включение которого осуществляется рукояткой 35. Включение зеркала, направляющего свет на выходную щель 10 (рисунок 6), или зеркала, срезанного по высоте, осуществляется рукояткой 36.

В монохроматоре, который используется в установке СДЛ-2, введение выходного поворотного зеркала осуществляется автоматически с помощью электродвигателя; зеркало, срезанное по высоте отсутствует.

При работе только с одной из выходных щелей вторая должна быть закрыта крышкой.

Длина волны излучения, проходящего через выходную щель монохроматора, определяется по счетчику 15, одно оцифрованное деление которого соответствует 0,1 нм для решетки 1200 штрихов на миллиметр, а одно не оцифрованное деление шкалы соответствует 0,025 нм.

При работе с решеткой 600 штрихов на миллиметр показания счетчика следует умножать на два, а при работе с решеткой 300 штрихов на миллиметр – на четыре.

На кронштейне синусного механизма расположен узел реперного устройства 37 (рисунок 8).

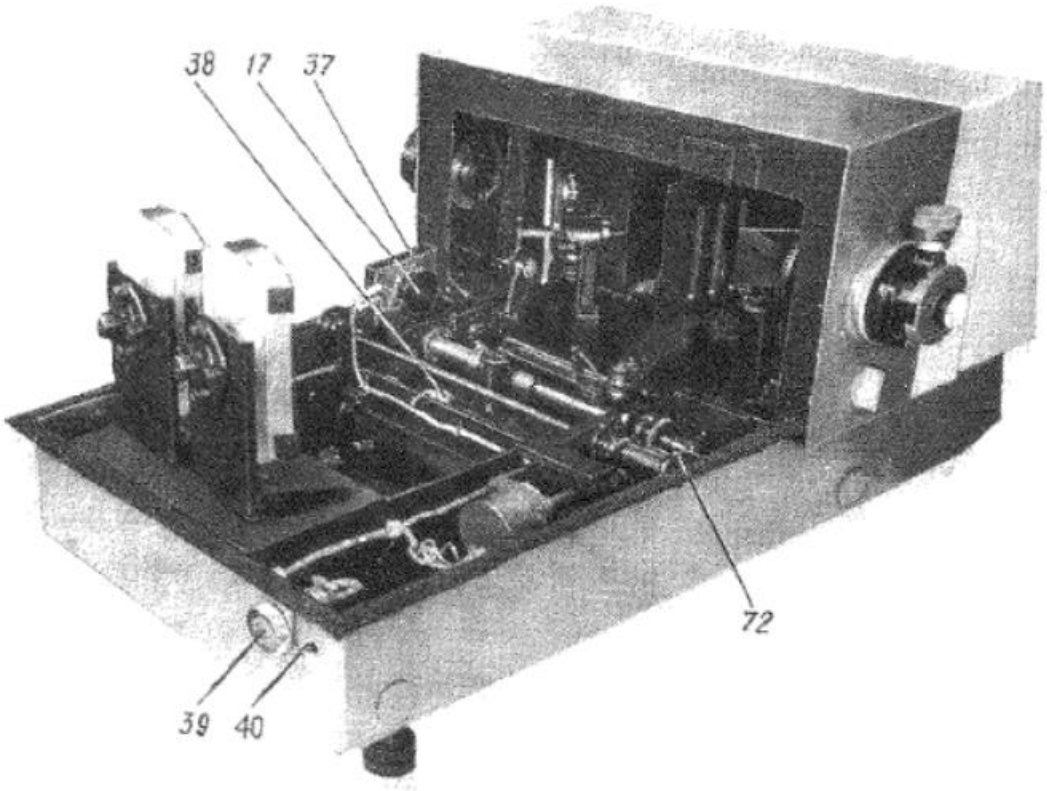

Рисунок 8 – Механизм работы монохроматора [1]

Движение от винта синусного механизма через шестерни 17 передается диску с вырезами, которые обеспечивают нанесение запись на реперных меток через 1 или 10 нм.

Автоматическая развертки спектра осуществляется от шагового двигателя. Максимальная скорость развертки спектра предназначена для быстрого поворота дифракционной решетки при возврате решетки в исходное положение.

Для отключения шагового двигателя на концах рабочего диапазона каждой дифракционной решетки в монохроматоре предусмотрены концевые выключатели 38. Монохроматор подключается к блоку управления через разъем 39, клемма 40 служит для заземления монохроматора.

Все узлы монохроматора закрыты кожухом 41. Под входной щелью монохроматора закрепляются рельсы длиной 1000 мм для установки осветительной системы, под выходными щелями – рельсы длиной 400 мм для установки приемников излучения.

Рельсы вводятся в направляющие 42, прижимаются к упорному винту 43 и закрепляются винтом 44 с гайкой 45. Регулировка положения рельсов по высоте осуществляется с помощью болтов, расположенных на свободных концах рельсов. Монохроматор установлен на три опоры 46, регулируемые по высоте с помощью винтов 47.

1.2 Излучение света в полупроводниках

1.2.1 Люминесценция полупроводников

При рекомбинации избыточных носителей заряда в полупроводнике может появляться оптическое излучение – люминесценция.

Образование свободных электронов и дырок – генерация носителей заряда – происходит при воздействии теплового хаотического движения атомов кристаллической решетки (тепловая генерация), при воздействии поглощенных полупроводником квантов света (световая генерация) и других энергетических факторов. Так как полупроводник всегда находится под действием всех этих факторов или хотя бы одного, генерация носителей происходит непрерывно.

Одновременно с генерацией в полупроводнике непрерывно происходит и обратный процесс – рекомбинация носителей заряда, т. е. возвращение электронов из зоны проводимости в валентную зону, в результате чего исчезает пара носителей заряда. В состоянии термодинамического равновесия процессы генерации и рекомбинации носителей заряда взаимно уравновешены. При этом в полупроводнике существуют равновесные концентрации электронов n_0 и дырок p_0 [2].

Возбуждение люминесценции можно осуществлять различными способами. Фотолюминесценция – излучение, появляющееся за счет поглощаемой энергии возбуждающего света. Энергия фотонов возбуждающего света обычно превышает ширину запрещенной зоны полупроводника. Свечение, возникающее при бомбардировке образца электронами, называется катодолюминесценцией, излучение, возникающее под действием электрического поля, – электролюминесценцией.

Созданные при возбуждении избыточные носители заряда за время релаксации теряют избыточную энергию и занимают энергетические уровни вблизи краев разрешенных зон. Затем происходит их рекомбинация, в результате которой энергия исчезнувшей электронно-дырочной пары выделяется в виде кванта оптического излучения либо идет на нагрев кристалла.

В полупроводниках в зависимости от характера энергетического спектра, типа и концентрации легирующих примесей, наличия дефектов кристаллической структуры, температуры и условий возбуждения могут наблюдаться различные механизмы рекомбинации избыточных электронов и дырок. При некоторых из них испускаются кванты света, т.е. наблюдается люминесценция.

1.2.2 Характеристики люминесцирующих веществ

В работе [3] спектром излучения называется функция распределения излучаемой веществом энергии (или числа квантов) по частотам *J(ν)*, длинам волн *J(λ)* или энергиям излучаемых квантов *J(hν)*. Спектры люминесценции центров свечения одного сорта состоят из нескольких полос. В простейшем случае это может быть и одна полоса гауссовой формы. Одновременное присутствие многих центров свечения существенно усложняет спектр люминесценции.

Интенсивность люминесценции есть поток люминесцентного излучения с единицы поверхности люминесцирующего тела в единицу времени:

$$J = \frac{1}{S}\int_0^{\infty} j(\mathrm{v})\, d\mathrm{v}, \qquad (1)$$

Понятие квантового выхода используется только в случае оптического возбуждения. Квантовый выход есть отношение полного числа испущенных квантов люминесценции к числу поглощенных квантов возбуждающего излучения:

$$\eta_{\text{кв}} = \frac{N_{\text{л}}}{N_{\text{пг}}}, \qquad (2)$$

где $N_{\text{л}}$ – число квантов люминесценции, $N_{\text{пг}}$ – число квантов возбуждающего света, поглощенных в веществе.

Исследование зависимости энергетического выхода от длины волны возбуждающего света представляет большой интерес. При этом для возбуждения люминесценции удобно использовать монохроматический свет.

1.2.3 Электролюминесценция

Электролюминесценцией называется вид люминесценции, при котором светящееся тело черпает энергию непосредственно из электрического поля.

При электролюминесценции в кристаллах происходит преобразование энергии электрического поля в излучение. Весь этот процесс делится на три качественно различных этапа.

1. Возбуждение. Электрическое поле переводит электроны кристалла в возбужденное состояние, энергия которого отличается от основного приблизительно на ширину запрещенной зоны. Возбуждение кристалла можно осуществить увеличением числа свободных носителей заряда по сравнению с равновесной концентрацией или увеличение кинетической энергии носителей в зоне проводимости и в валентной зоне, но и возбуждением примесных центров. Однако способы возбуждения примесей электрическим полем ничем не отличаются от способов создания неравновесных носителей, хотя возбуждение примесей может и не сопровождаться увеличением проводимости. В связи с этим в дальнейшем будут рассматриваться только процессы создания неравновесных носителей.

2. Миграция энергии возбуждения. При электролюминесценции передача энергии по кристаллу до места, где происходит рекомбинация, может играть существенную роль. Это объясняется тем, что во многих случаях перенос энергии осуществляется свободными носителями, которые испытывают действие электрического поля. В наиболее общем случае надо учитывать не только дрейф свободных носителей, но и процессы их захвата, а также движение экситонов и резонансную передачу энергии.

3. Излучение. Процесс излучения – перехода в невозбужденное состояние обычно происходит с участием дефектов кристалла. Наряду с такими переходами возможны также межзонные и внутри зонные переходы.

1.2 Микроконтроллер Arduino Uno и ATmega328

1.3.1 Микроконтроллер Arduino Uno

Arduino Uno (рисунок 10) – это устройство на основе микроконтроллера ATMega328. В его состав входит все необходимое для удобной работы с микроконтроллером: 14 цифровых входов/выходов (из них 6 могут использоваться в качестве ШИМ-выходов), 6 аналоговых входов, кварцевый резонатор на 16 МГц, разъем USB, разъем питания, разъем для внутрисхемного программирования (ICSP) и кнопка сброса. Для начала работы с устройством достаточно просто подать питание от AC/DC-адаптера или батарейки, либо подключить его к компьютеру посредством USB-кабеля.

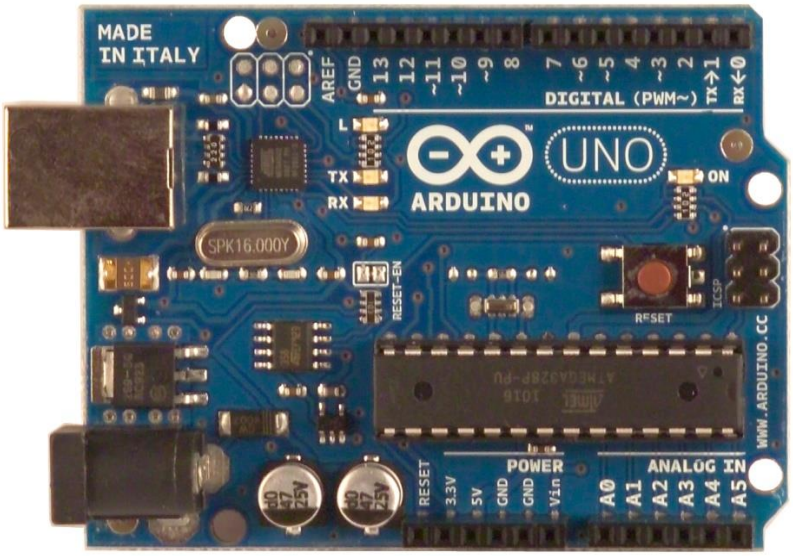

Рисунок 10 – Микроконтроллер Arduino Uno [4]

Arduino Uno может быть запитан от USB либо от внешнего источника питания – тип источника выбирается автоматически

В качестве внешнего источника питания (не USB) может использоваться сетевой AC/DC-адаптер или аккумулятор/батарея. Штекер адаптера

(диаметр – 2.1 мм, центральный контакт – положительный) необходимо вставить в соответствующий разъем питания на плате. В случае питания от аккумулятора/батареи, ее провода необходимо подсоединить к выводам Gnd и Vin разъема POWER.

Напряжение внешнего источника питания может быть в пределах от 6 до 20 В. Однако, уменьшение напряжения питания ниже 7 В приводит к уменьшению напряжения на выводе 5 В, что может стать причиной нестабильной работы устройства. Использование напряжения больше 12 В может приводить к перегреву стабилизатора напряжения и выходу платы из строя. С учетом этого рекомендуется использовать источник питания с напряжением в диапазоне от 7 В до 12В.

Объем Flash-памяти ATMega328 составляет 32 КБ (из которых 0.5 КБ используются загрузчиком). Микроконтроллер также имеет 2 КБ памяти SRAM и 1 КБ EEPROM

Каждый из 14 цифровых выводов может работать в качестве входа или выхода. Уровень напряжения на выводах ограничен 5 В. Максимальный ток, который может отдавать или потреблять один вывод, составляет 40 мА. Все выводы сопряжены с внутренними подтягивающими резисторами (по умолчанию отключенными) номиналом 20-50 кОм.

Arduino Uno предоставляет ряд возможностей для осуществления связи с компьютером, еще одним Arduino или другими микроконтроллерами. В ATmega328 имеется приемопередатчик UART, позволяющий осуществлять последовательную связь посредством цифровых выводов 0 (RX) и 1 (TX). Микроконтроллер ATMega16U2 на плате обеспечивает связь этого приемопередатчика с USB-портом компьютера, и при подключении к ПК позволяет Ардуино определяться как виртуальный COM-порт. Прошивка микросхемы 16U2 использует стандартные драйвера USB-COM, поэтому установка внешних драйверов не требуется [4].

1.3.2 Микроконтроллер ATMega328

Микроконтроллер ATMega328 (рисунок 11) является 8-ми разрядным CMOS микроконтроллером с низким энергопотреблением, основанным на усовершенствованной AVR RISC архитектуре.

ATMega328/P — микроконтроллер семейства AVR, как и все остальные имеет 8-битный процессор и позволяет выполнять большинство команд за один такт.[10]

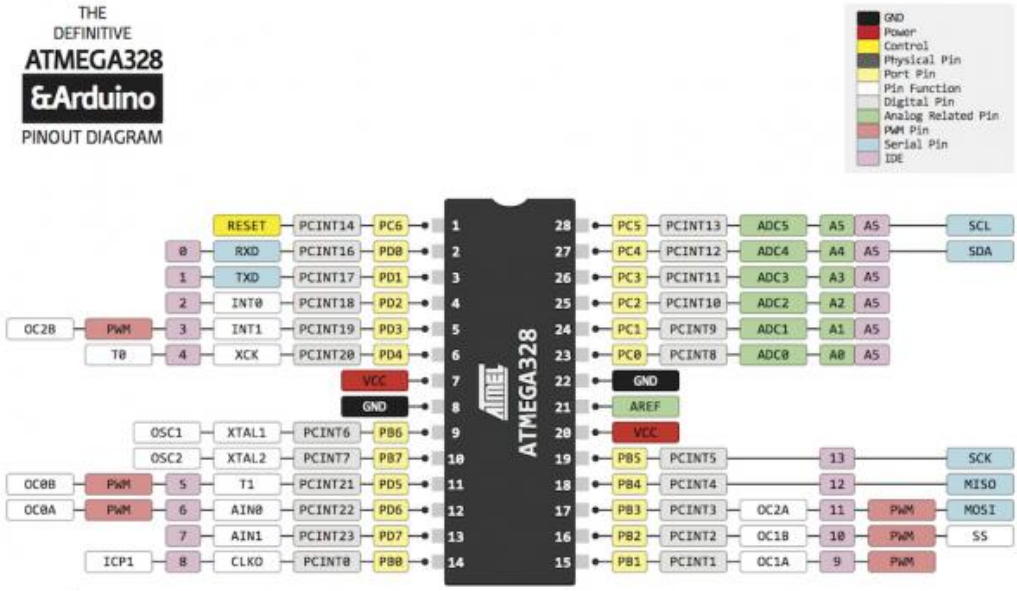

Рисунок 11 – Схема микроконтроллера ATMega328 [10]

Характеристики ATMega328/P:

Память:

- 32 kB Flash (память программ, имеющая возможность самопрограммирования)
- 2 kB ОЗУ
- 1 kB EEPROM (постоянная память данных)

Периферийные устройства:

- Два 8-битных таймера/счетчика с модулям сравнения и делителями частоты
- 16-битный таймер/счетчик с модулем сравнения и делителем частоты, а также с режимом записи
- Счетчик реального времени с отдельным генератором
- Шесть каналов PWM (аналог ЦАП)
- 6-канальный ЦАП со встроенным датчиком температуры
- Программируемый последовательный порт USART
- Последовательный интерфейс SPI
- Интерфейс I2C
- Программируемый сторожевой таймер с отдельным внутренним генератором
- Внутренняя схема сравнения напряжений
- Блок обработки прерываний и пробуждения при изменении напряжений на выводах микроконтроллера

Специальные функции микроконтроллера ATmega328:

- Сброс при включении питания и программное распознавание снижения напряжения питания
- Внутренний калибруемый генератор тактовых импульсов
- Обработка внутренних и внешних прерываний
- 6 режимов сна (пониженное энергопотребление и снижение шумов для более точного преобразования АЦП)

Напряжения питания и скорость процессора**:**

- 1.8 — 5.5 В при частоте до 4 МГц
- 2.7 — 5.5 В при частоте до 10 МГц
- 4.5 — 5.5 В при частоте до 20 МГц

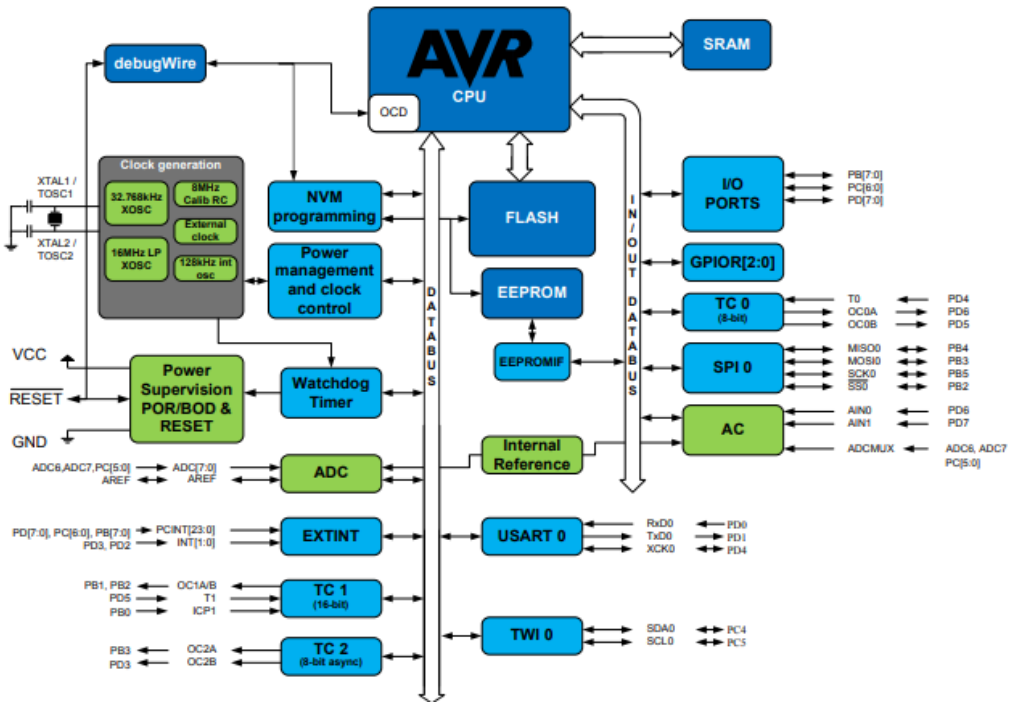

Рисунок 12 – Блок-диаграмма ATMega328 [10]

1.4 Графическая среда разработки приложений LabVIEW

LabView – это среда разработки и платформа для выполнения программ, созданных на графическом языке программирования «G» фирмы National Instruments.

Говоря простым языком, LabView — Это среда создания приложений для задач сбора, обработки, визуализации информации от различных приборов, лабораторных установок, а также для управления технологическими процессами и устройствами.

Разработка виртуальных приборов управления приборами в среде LabVIEW, отличается от разработки аналогичных компьютерных систем управления, основанных на языках программирования C++ или Basic. В алгоритмических языках программирования, при создание программных модулей, создается последовательность текстовых команд, образующих алгоритм обработки данных. В среде LabVIEW используется графический язык программирования «Джей» (G). Алгоритм обработки данных создается в графической иконной форме, и образующий так называемую блок-диаграмму.

Каждая иконка представляет собой узел обработки данных, который выполняется только при достаточности входных данных на его подключенных (обязательных) входах.

Любой разрабатываемый виртуальный прибор всегда состоит из двух составных частей: лицевой панели управления и функциональной панели. Лицевая панель виртуального разрабатываемого прибора представляет собой интерактивный интерфейс для пользователя этого виртуального прибора, которая эмитирует лицевую панель традиционного лабораторного прибора. На этой панели располагают элементы управления, которые являются средствами ввода данных в программу, и элементы индикации - вывода данных из программы.

Второй частью любого виртуального прибора является функциональная панель, которая является исходным программным кодом виртуального прибора, созданным на языке графического программирования. Компонентами блок-диаграммы являются: виртуальные приборы более низкого уровня, встроенные функции LabVIEW, константы и структуры управления выполнением алгоритмом программы. Чтобы задать потоки данных между определенными объектами функциональной панели, они соединены посредством проводов (wires). Вводимые и выводимые данные с панели управления передаются в программу и обратно посредством своих выводов на функциональной панели. Система программирования в LabVIEW отвечает концепции модульного программирования.

Каждый виртуальный прибор можно использовать как самостоятельное приложение, так и в качестве виртуального подприбора (ВПП). Согласно этой концепции, исходная прикладная задача сначала разбивается на ряд подзадач, для выполнения которых создаются соответствующие подприборы. После этого на блок-диаграмме виртуальный прибора из созданных виртуальных приборов создается виртуальный прибор более высокого уровня, который выполняет поставленную задачу в целом [2].

1.5 Программный комплекс KiCad

KiCad — распространяемый по лицензии GNU GPL программный комплекс класса EDA с открытым исходным кодом, предназначенный для разработки электрических схем и печатных плат.

Кроссплатформенность компонентов KiCad обеспечивается использованием библиотеки wxWidgets. Поддерживаются операционные системы Linux, Windows NT 5.x, FreeBSD и Solaris.

Программы, входящие в KiCad:

- kicad — менеджер проектов;
- eeschema — редактор электрических схем (Рисунок 13);
- встроенный редактор символов схем (библиотечных компонентов);
- pcbnew — редактор печатных плат (Рисунок 14);
- встроенный редактор образов посадочных мест (библиотечных компонентов);
- 3D Viewer — 3D-просмотрщик печатных плат (Рисунок 15) на базе OpenGL (часть pcbnew);
- gerbview — просмотрщик файлов Gerber (фотошаблонов);
- cvpcb — программа для выбора посадочных мест, соответствующих компонентам на схеме;
- wyoeditor — текстовый редактор для просмотра отчётов.

Функции компонентов KiCad:

eeschema обеспечивает:

- создание однолистовых и иерархических схем,
- проверку их корректности ERC (контроль электрических правил),
- создание списка электрических цепей netlist для редактора топологии платы pcbnew или для Spice-моделирования схемы,

- доступ к документации на используемые в схеме электронные компоненты (datasheet).

pcbnew обеспечивает:

- разработку плат, содержащих до 16 слоёв меди и до 12 технических слоёв (шелкография, паяльная маска и т. п.),
- выход на внешние трассировщики соединений посредством генерации описания платы на Specctra Design Language (on-line FreeRoute и др.),
- генерацию технологических файлов для изготовления печатных плат (Gerber-файлы для фотоплоттеров, файлы сверловок и файлы размещения компонентов),
- послойная печать схем и чертежей печатных плат на принтере или плоттере (в форматах PostScript, HPGL, SVG и DXF), с рамкой формата или без неё.
- gerbview позволяет просматривать Gerber-файлы[12].

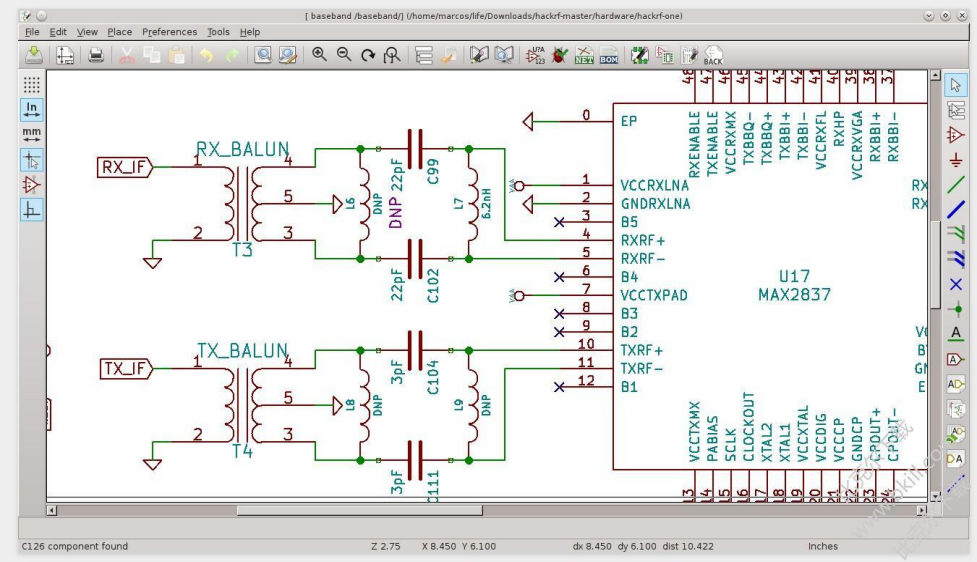

Рисунок 13 – Окно редактора электрических схем [12]

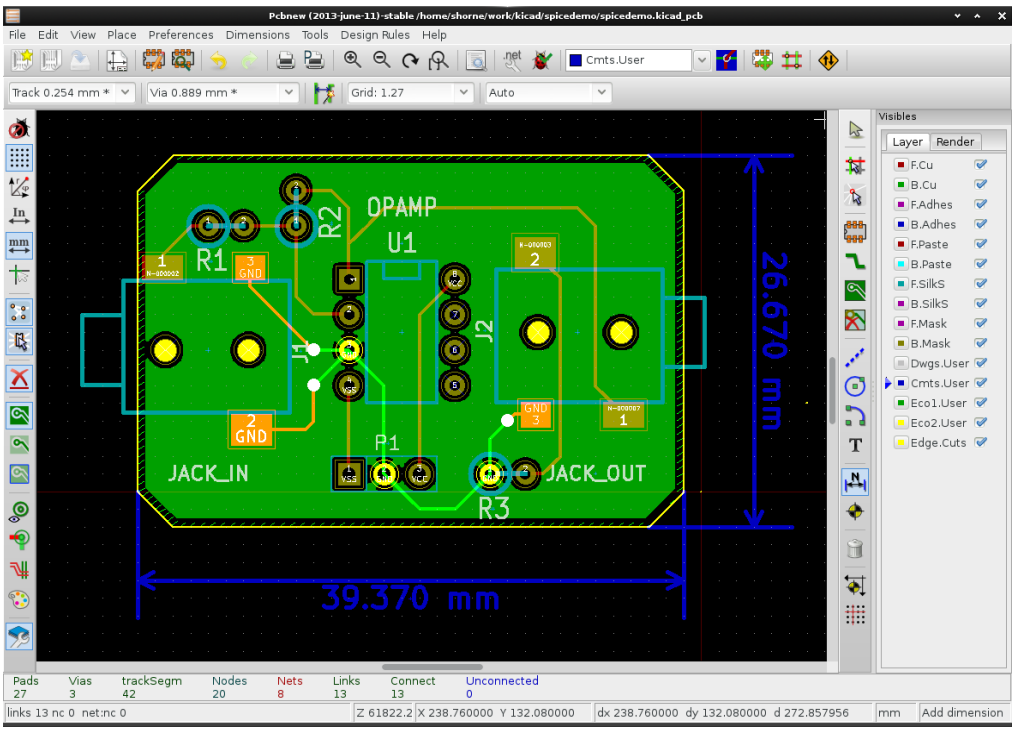

Рисунок 14 – Окно редактора печатных плат [12]

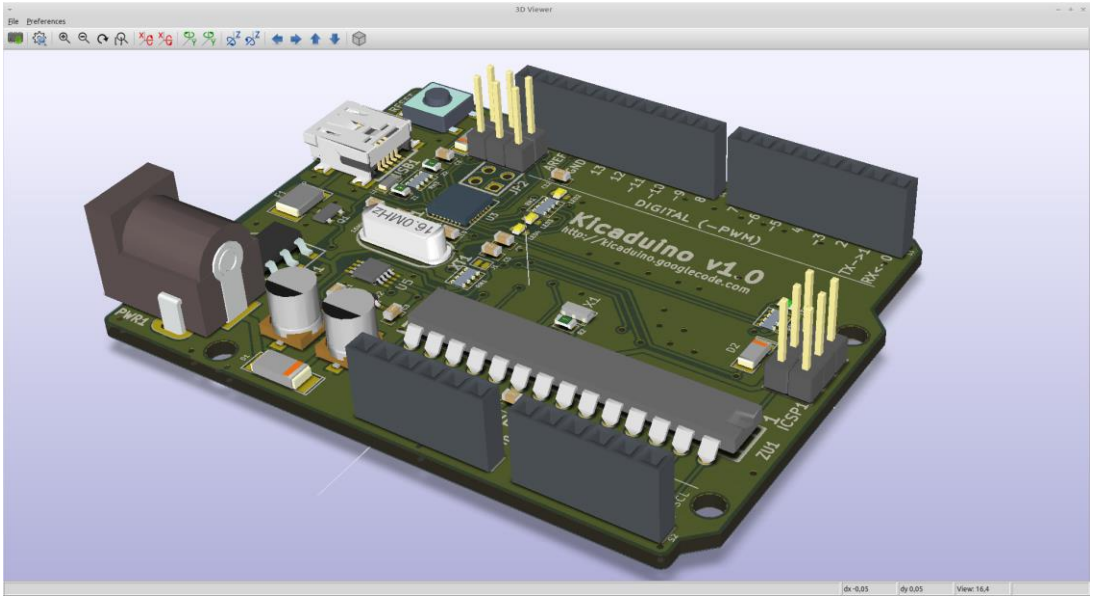

Рисунок 15 – Трехмерный вид печатной платы в 3D [12]

1.6 Источник-измеритель Keithley 2611B

Источник/Измеритель KEITHLEY (рисунок 16) SourceMeter® 2611B представляет собой комбинацию 6 1/2 разрядного прецизионного, малошумящего, высокостабильного источника питания постоянного

тока с малошумящим, высокоимпедансным мультиметром, имеющим высокую стабильность и высокий класс точности. Любой источник-измеритель KEITHLEY серии 2600B имеет встроенный TSP (Test Script Processor) процессор, что позволяет существенно увеличить скорость и пропускную способность, а наличие встроенного TSP Express Software, позволяет быстро снимать вольтамперные характеристики (ВАХ) без дополнительного программирования. Кроме того, источники-измерители KEITHLEY SourceMeter® 2611B поддерживают программное обеспечение предыдущей серии KEITHLEY SourceMeter® [14] 2611/2612: LabTracer II и TestScript Builder.

Рисунок 16 – Источник-измеритель Keithley 2611B [14]

Источник/измеритель KEITHLEY SourceMeter® 2611B может быть сконфигурирован, как:

- прецизионный источник питания
- источник тока (TRUE RMS)
- мультиметр: напряжение постоянного тока, сила постоянного тока, сопротивление с 6½ –разрядным разрешением

- генератор импульсов (длительность импульса >100 мкс)
- генератор сигнала (генератор формы тока: 12500 сэмпл./сек и генератор формы напряжения: 20000 сэмпл./сек)
- электронная нагрузка

Особенности измерителя Keithley 2611B:

- количество каналов: 1 канал
- 4-квадрантная рабочая область
- возможность последовательного и параллельного подключения до 32 источников-измерителей
- 2 АЦП на канал (ток и напряжение) с высокой скоростью считывания (до 20000 изм./сек)
- цифровой интерфейс ввода/вывода
- возможность сохранения данных на USB-носитель
- возможность управления через Ethernet (LXI Class C)
- встроенная функция проверки контактов
- высокоемкостной режим (только для измерений по постоянному току)
- встроенное программное обеспечение TSP Express Software для быстрого снятия ВАХ
- программное обеспечение LabTracer 2.0 (Freeware) с графическим интерфейсом для снятия основных параметров
- программное обеспечение Test Script Builder для создания, изменения и загрузки тестовых сценариев с программным редактором, аналогичным Visual Basic
- возможность интегрирования в измерительные системы

2 Методика измерений

2.1 Принцип автоматизации монохроматора МДР-23

На рисунке 16 показана функциональная схема с помощью, которой происходит управление монохроматором МДР-23.

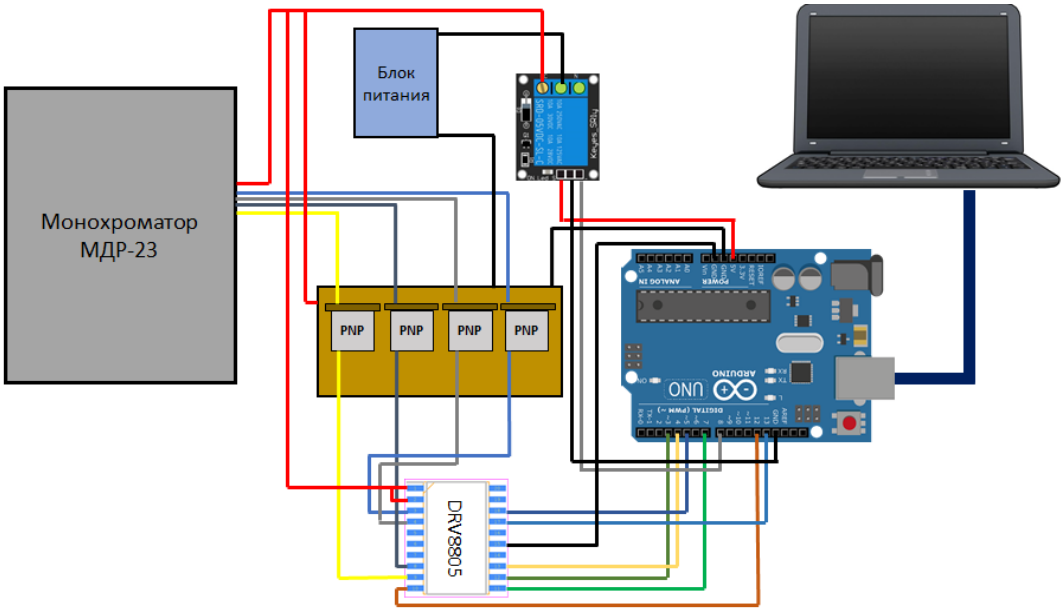

Рисунок 16 – Функциональная схема автоматизации монохроматора МДР-23

Монохроматор МДР-23 снабжен шаговым двигателем типа ШДР-711, который обеспечивает установку длины волны в плоскости выходной щели путем поворота дифракционной решетки. Главным элементом схемы является микроконтроллер ATMega328, который обрабатывает все команды от персонального компьютера. Команды принимаются и обрабатываются, после чего через порты контроллера передается последовательность импульсов на драйвер, который управляет непосредственно шаговым двигателем (рисунок 17). Скорость вращения шагового двигателя, моменты пуска и направление вращения задаются управляющей программой.

Шаговый двигатель монохроматора приводится в движение П-образными импульсами, последовательно поступающими на его обмотки. Прямоугольные импульсы сдвинуты относительно друг друга на период следования импульсов генератора. Скорость вращения вала шагового

двигателя определяется частотой следования импульсов генератора, а направление вращения – фазовыми соотношениями между импульсами, поступающими на обмотки двигателя [1].

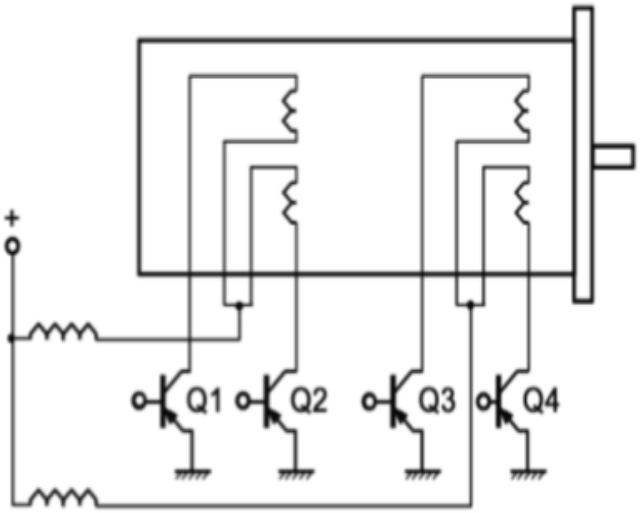

Рисунок 17 – Схема управления шаговым двигателем [1]

На рисунке 18 представлены П-образные импульсы, последовательно поступающие на обмотки шагового двигателя.

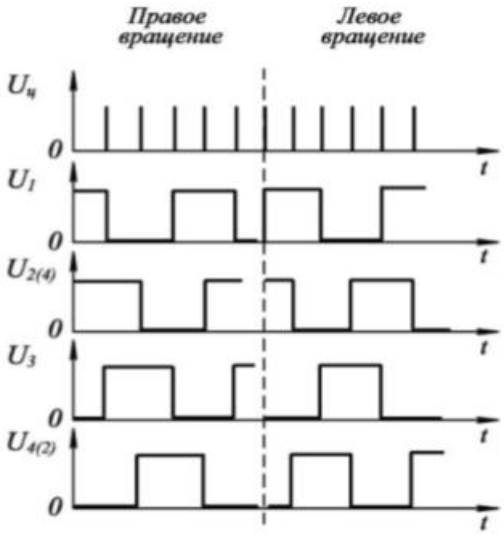

Рисунок 18 – Последовательность импульсов шагового двигателя[1]

В таблице 1 представлена режимы работы шагового двигателя ШДР-711. Управление осуществляется с помощью драйвера DRV8805 ножками SM0 и SM1. Возможны три режима работы шагового двигателя: полношаговый, полушаговый и микрошаговый режимы. Также есть режим резерва или простоя.

Таблица 1 Режимы работы шагового двигателя

SM1	SM0	MODE
0	0	2-phase drive (full step)
0	1	1-2 phase drive (half step)
1	0	1-phase exitation (wive drive)
1	1	Reserved
SM1	SM0	MODE

На рисунке 20 представлена схема драйвера DRV8805, с помощью которого и осуществляется управление шаговым двигателем. Драйвер может иметь как 20 ножек, так и 16, разница лишь в количестве GND. Мы использовали драйвер на 20 ножек. Питание происходит от блока питания 10-15В. Основными пинами, которыми мы пользуемся для запуска двигателя, являются: SM0, SM1, DIR, STEP, nEBL, OUT1, OUT2, OUT3 и OUT4. Пины SM0 и SM1 задают режимы работы двигателя (рисунок 15). DIR задает направление вращения, левое или правое. STEP задает частоту вращения двигателя. nEBL является своего рода выключателем, активирует все остальные пины. И оставшиеся четыре пина OUT1, OUT2, OUT3 и OUT4 подключаются к биполярным транзисторам pnp типа, которые в свою очередь подключены к обмоткам шагового двигателя.

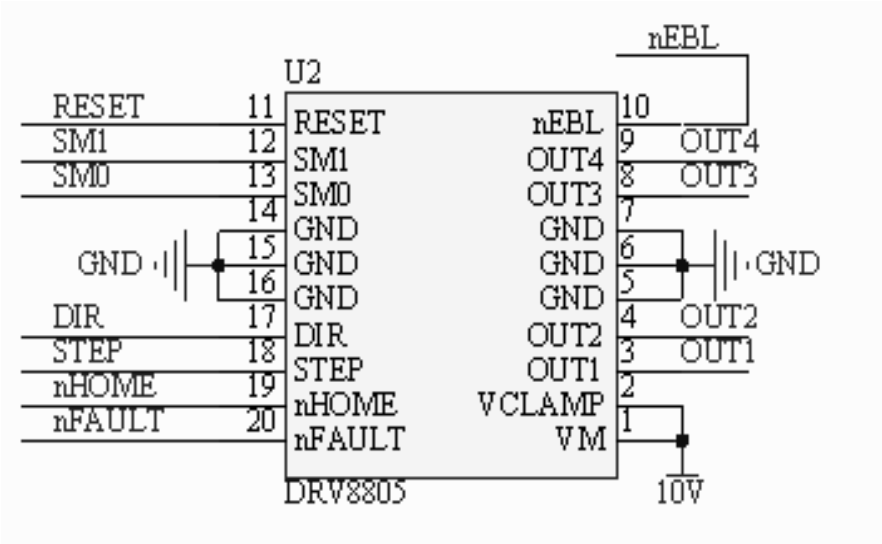

Рисунок 19 – Схема драйвера шагового двигателя [9]

За основу для разработки печатной платы был взять микроконтроллер Arduino Uno (рисунок 20).

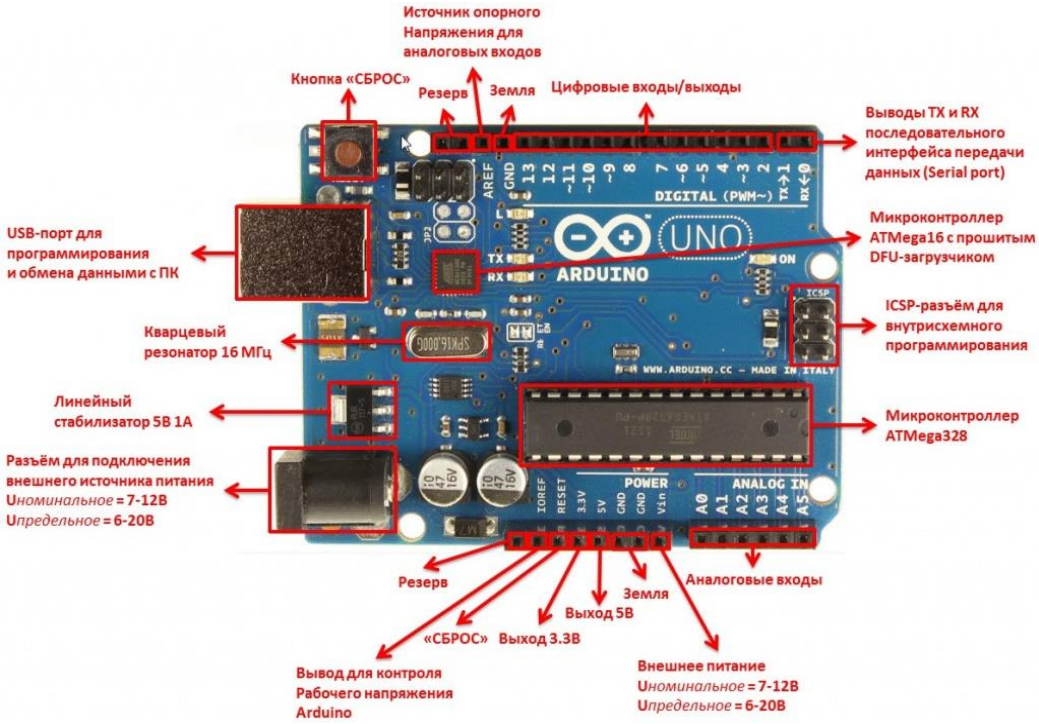

Рисунок 20 – Схема микроконтроллера Arduino Uno [4]

Наша задача состоит в том, чтобы на одной плате разместить все необходимые нам компоненты для автоматизации монохроматора. Для работы необходим микроконтроллер ATMega328, на основе которого и построена плата Arduino Uno. Типовая схема включения любого микроконтроллера содержит ряд компонентов и цепей, обеспечивающих его нормальное функционирование. Так что необходима минимальная обвязка самого микроконтроллера для правильной работы. USB-порт для подключения к ПК и питания платы. И несколько цифровых входов/выходов.

На рисунке 21 справа представлена таблица контактов коннектора ШДР. Общий контакт ДШ-5, а ДШ-1, ДШ-2, ДШ-3, ДШ-4 идут на обмотки шагового двигателя.

Конт.	Цепь
7,8	Корпус
3	ДШ-5
1	ДШ-1
6	ДШ-2
9	ДШ-3
5	ДШ-4

Рисунок 21 – Коннектор ШДР [1]

2.2 Запуск LabVIEW

Программы LabVIEW называются виртуальными приборами (ВП), так как они функционально и внешне подобны реальным (традиционным) приборам. Однако они столь же подобны программам и функциям на популярных языках программирования, таких как C или Basic.

После запуска LabVIEW появляется стартовое диалоговое окно (рисунок 22), предлагающее выбрать дальнейшие действия. Окно разделено на две части, которые предлагают две группы вариантов действий. Левая часть позволяет создать новый ВП, пустой проект или создать ВП по шаблону.

Правая часть диалогового окна позволяет получить доступ к справочным материалам, найти и запустить готовые примеры ВП [5].

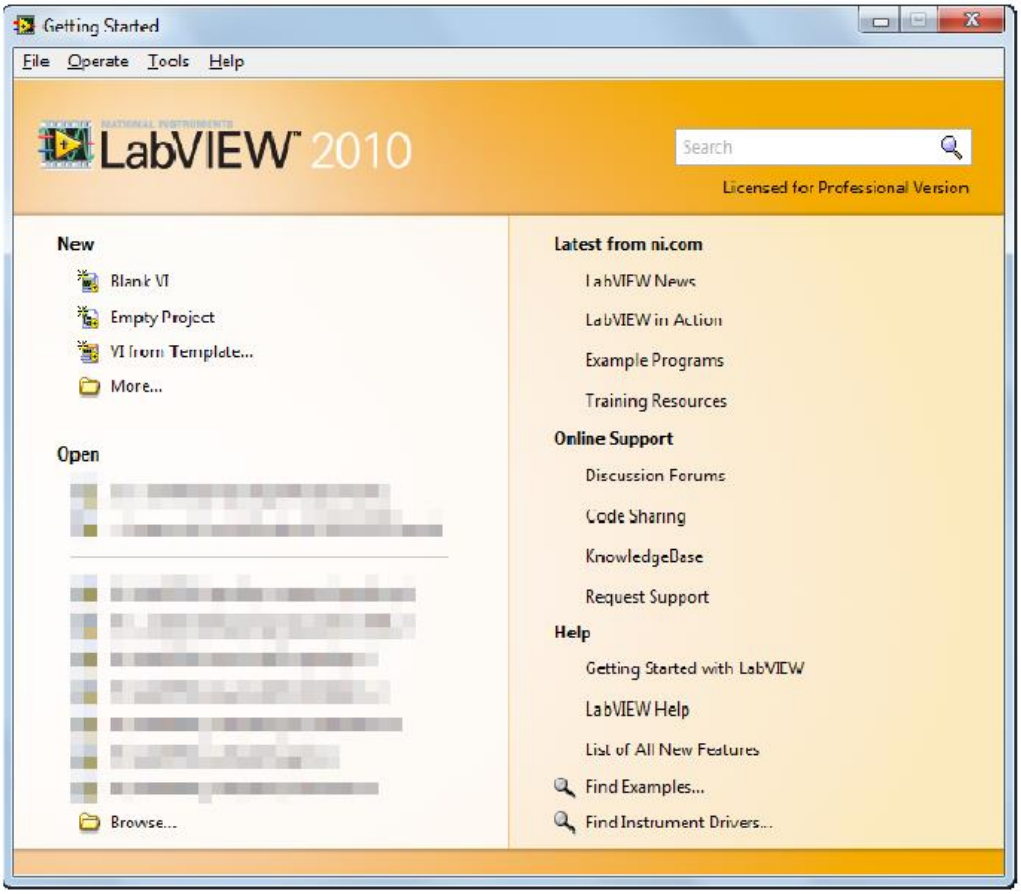

Рисунок 22 – Стартовое окно LabVIEW [5]

2.2.1 Создание виртуального прибора

При создании пустого виртуального прибора открываются два окна, содержащие лицевую панель (Рисунок 24) и блок-диаграмму (Рисунок 25) виртуального прибора.

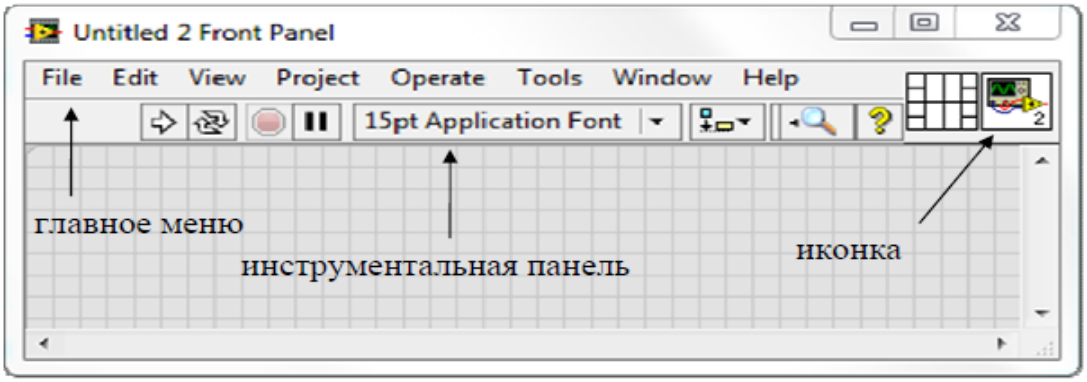

Рисунок 23 – Лицевая панель [5]

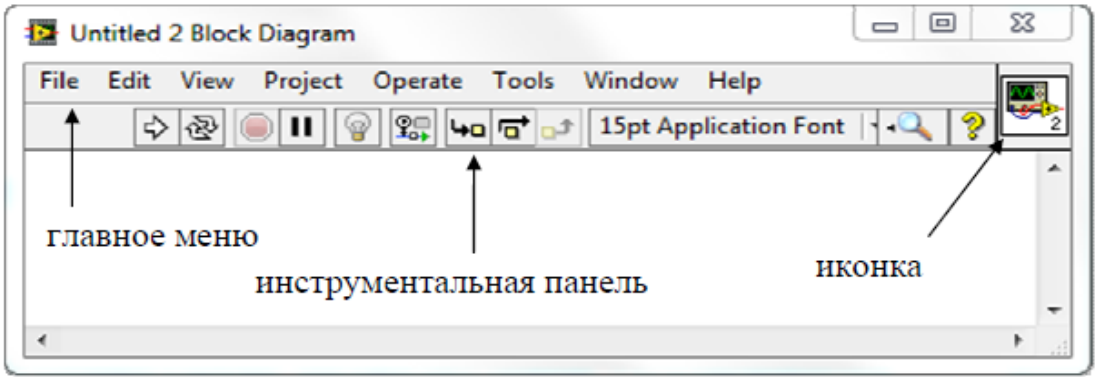

Рисунок 24– Блок-диаграмма [5]

На лицевой панели (рисунок 24) разрабатывается внешний вид будущего виртуального прибора. На ней создаются элементы управления и отображения, которые являются интерактивными средствами ввода и вывода данных этого виртуального прибора.

На блок-диаграмме разрабатывается исходный код будущего виртуального прибора. В отличии от классических языков программирования, исходный код LabVIEW представляет собой блок-диаграмму, где все команды, операторы циклов и сравнения изображаются графическими обозначениями. Блок-диаграмма состоит из узлов, терминалов и проводников данных.

В верхней части каждого окна располагается главное меню с пунктами File, Edit, View, Project, Operate, Tools, Windows и Help. Под главным меню расположена инструментальная панель, служащая для запуска и

редактирования ВП. Инструментальная панель окна блок-диаграммы отличается дополнительными инструментами для отладки ВП. В правом верхнем углу каждого окна находится иконка, наложенная на соединительную панель ВП.

Окна блок-диаграммы и лицевой панели содержат инструментальную панель (рисунки 23, 24).

Свободное пространство каждого окна образует рабочую область, снабженную горизонтальной и вертикальной полосами прокрутки. При построении ВП в рабочей области лицевой панели визуально размещаются элементы управления и индикации, на панели блок-диаграммы составляется блок-диаграмма.

Построение ВП осуществляется с помощью трех вспомогательных палитр: палитры Элементов управления и отображения, палитры Функции и палитры Инструменты. Все перечисленные палитры можно вывести для постоянного или временного отображения и разместить в любом месте экрана.

Палитру инструментов можно вывести с помощью щелчка правой кнопки мыши при зажатой клавише «Shift». Все операции по созданию, редактированию и отладке ВП выполняются с помощью палитры Инструменты (рисунок 25).

Рисунок 25 – Палитра инструментов [5]

Для размещения элементов управления и отображения данных на лицевую панель используется палитра элементов управления и отображения

(рисунок 25). Палитра элементов управления и отображения доступна только с лицевой панели. Для вывода палитры на экран следует щелкнуть правой кнопкой мыши в рабочем пространстве лицевой панели.

Все элементы управления и отображения на палитре сгруппированы по разделам.

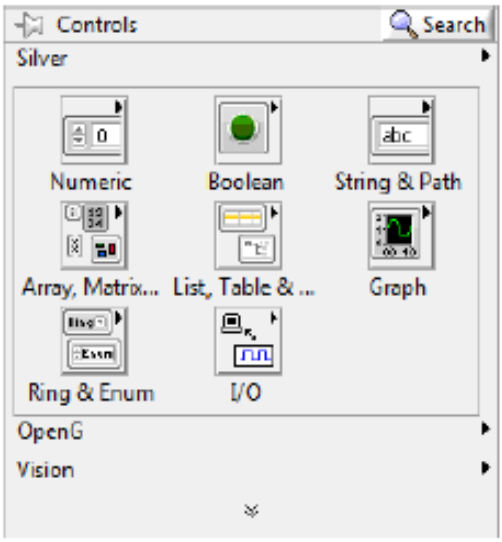

Рисунок 26 – Палитра элементов управления и отображения [55]

Для вывода палитры на экран следует щелкнуть правой кнопкой мыши в рабочем пространстве блок-диаграммы (Рисунок 27).

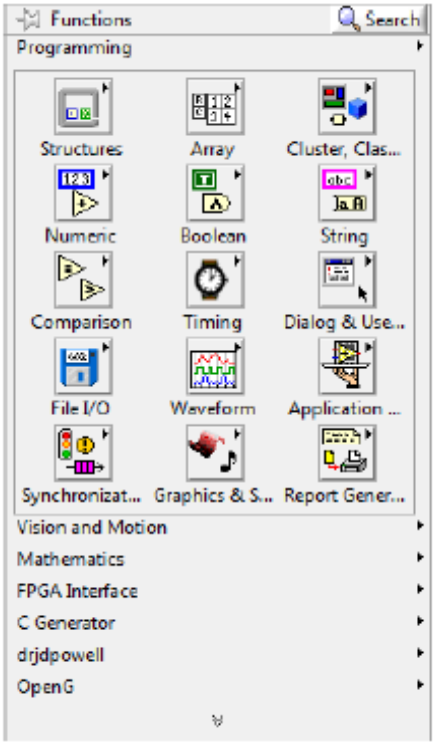

Рисунок 27 – Палитра функций [5]

Для проектирования ВП необходимо сформировать его лицевую панель и разработать блок-диаграмму. При формировании лицевой панели производится выбор и установка на ней элементов управления и индикаторов из палитры элементов данной панели. Аналогично при разработке блок-диаграммы производится выбор и установка на ней функциональных элементов и подприборов из палитры функций данной панели.

Установка каждого элемента на лицевой панели сопровождается появлением соответствующего терминала данных на панели блок-диаграммы. Терминалы элементов управления представляют порты ввода информации в блок-диаграмму, а терминалы индикаторов – порты вывода информации из блок-диаграммы на лицевую панель. Для обработки введенной информации и программного управления параметрами и режимами работы элементов лицевой панели на панели блок-диаграммы размещаются необходимые константы, функции, подприборы и структуры, которые также имеют терминалы для ввода и вывода информации. Все перечисленные элементы

представляют узлы блок-диаграммы, которые соединяются с терминалами элементов управления и индикации и между собой линиями, называемыми проводниками. В такой схеме через узлы в процессе обработки проходят данные, идущие по проводникам от входных терминалов к выходным. Узлы – это объекты на блок-диаграмме, которые имеют одно или более полей ввода/вывода данных и выполняют алгоритмические операции ВП. Они аналогичны операторам, функциям и подпрограммам текстовых языков программирования [11].

3 Экспериментальная часть

3.1 Лицевая панель виртуального прибора

Для начала работы в среде разработки LabView был создан виртуальный прибор, позволивший частично автоматизировать работу монохроматора МДР-23. На рисунке 31 изображена лицевая панель виртуального прибора в LabView. На лицевой панели разрабатывается внешний вид будущего виртуального прибора. На ней создаются элементы управления и отображения, которые являются интерактивными средствами ввода и вывода данных этого виртуального прибора [7].

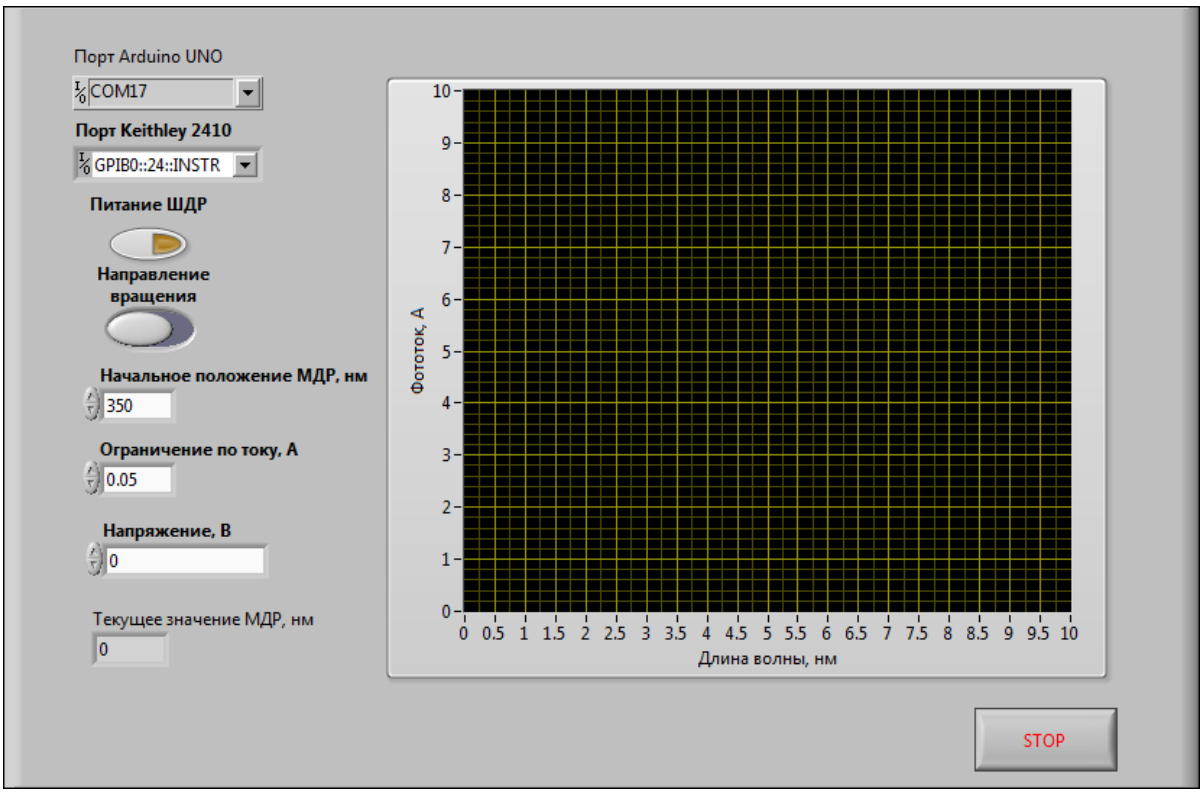

Рисунок 28 – Лицевая панель виртуального прибора

Как видно из рисунка панель создавалась с максимально удобным для пользователя графическим интерфейсом. В левой верхней части находится окно COM-порта, которое позволяет выбрать нужный нам порт для работы с программой. Ниже расположен порт, который позволяет задействовать источник-измеритель Keithley 2410. Далее кнопка включения (выключения) питания ШДР и кнопка направления вращения (левое вращение и правое вращение). Также мы можем задать начальное положение МДР, ограничение по току, напряжение и посмотреть текущее значение МДР. В правой нижней части присутствует кнопка «STOP» для прекращения выполнения программы. На всю ширину панели установлен дисплей для построения графиков зависимости величины фототока от длины волны.

3.2 Блок-диаграмма виртуального прибора

На блок-диаграмме разрабатывается исходный код будущего виртуального прибора. В отличии от классических языков программирования,

исходный код LabVIEW представляет собой блок-диаграмму, где все команды, операторы циклов и сравнения изображаются графическими обозначениями. Блок-диаграмма состоит из узлов, терминалов и проводников данных.

Данная среда разработки очень удобна для создания виртуальных приборов, т.к. она наглядно показывает, в каком месте схемы присутствует ошибка. На рисунке 29 представлена блок-схема виртуального прибора.

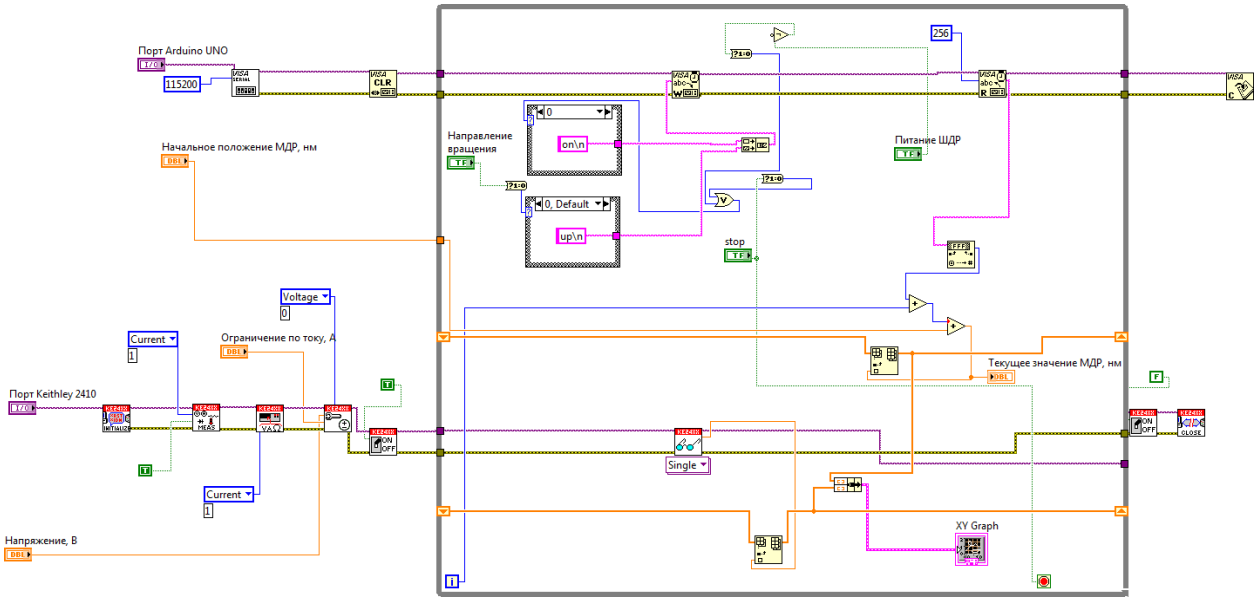

Рисунок 29 – Блок-диаграмма виртуального прибора

Для того, чтобы виртуальная среда разработки LabView смогла распознать микроконтроллер и источник измеритель, необходимы подВП т.е. подприборы. В нашем случае эти подприборы уже написаны и находятся в свободном доступе. На блок-диаграмме эти подВП показаны небольшими блоками с определенными рисунками, собственно, которые и отражают выполняемую ими функцию. В остальном блок-диаграмма виртуального прибора разрабатывалась самостоятельно.

Программирование потоков передачи данных осуществляется при помощи механизма графического связывания терминалов. Терминалы данных имеют прямоугольную форму и содержат буквенно-графическое обозначение,

характеризующее тип и форму представления воспринимаемых ими данных. Таким образом, по виду терминала можно определить, является ли он источником или приемником данных, какие типы данных он воспринимает – числовые, логические или строковые, а для числовых – является ли число целым или вещественным.

Линии связи между терминалами отображают потоки данных в разрабатываемом приложении. Данные могут передаваться только в одном направлении: от источника к приемникам сигнала. Причем цвет и толщина линий связи характеризуют типы передаваемых данных. Так линия одинарной толщины соответствует скалярной величине, утолщенная линия соответствует одномерному массиву, двойная линия – двумерному массиву, утолщенная двойная линия – массиву с размерностью более 2 [9].

Тип данных может быть задан в меню свойств созданного терминала, либо в контекстном меню (рисунок 30).

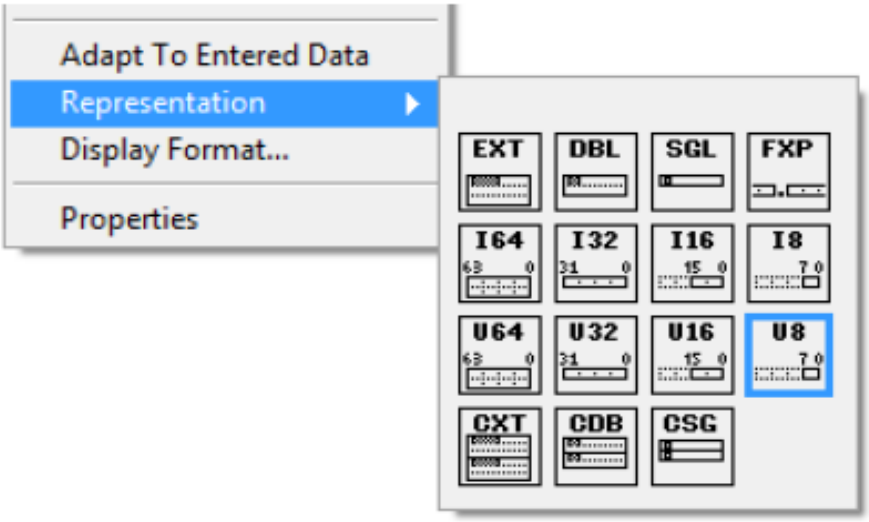

Рисунок 30 – Контекстное меню представления типа данных [12]

Для выполнения арифметических, тригонометрических, логарифмических и комплексных операций с числовыми данными и для преобразования типов числовых данных используются математические функции, расположенные в палитре Mathematics (рисунок 31).

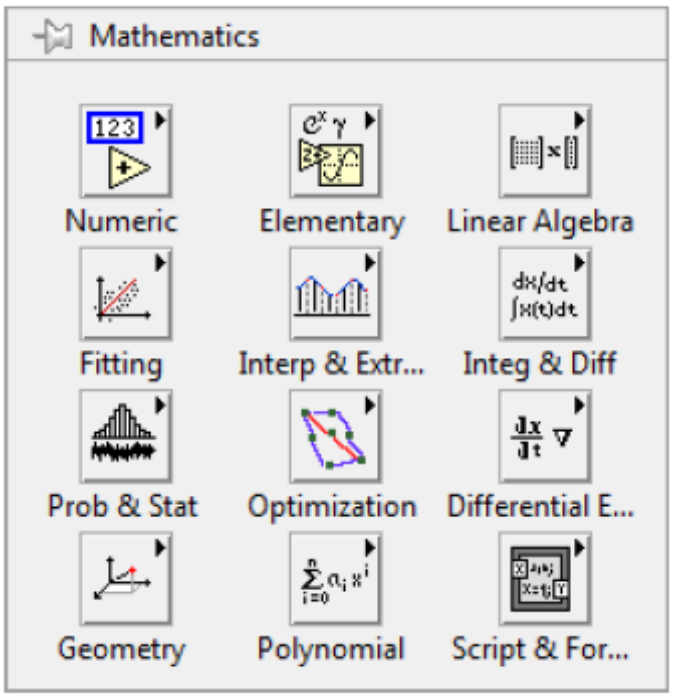

Рисунок 31 – Палитра математических функций [13]

Логические функции используются для выполнения логических операций над значениями как простых логических величин, так и массивов этих величин. Функции И (And), ИЛИ (Or), Исключающее ИЛИ (Exclusive Or), НЕ (Not), И-НЕ (Not And), ИЛИ-НЕ (Not Or), Исключающее ИЛИ-НЕ (Not Exclusive Or), Исключение (Implies) являются полиморфными. Оба входа этих функция должны иметь логические или числовые значения, причем эти значения могут быть скалярами, массивами или кластерами. Логические функции расположены в палитре Programming-Boolean (рисунок 32).

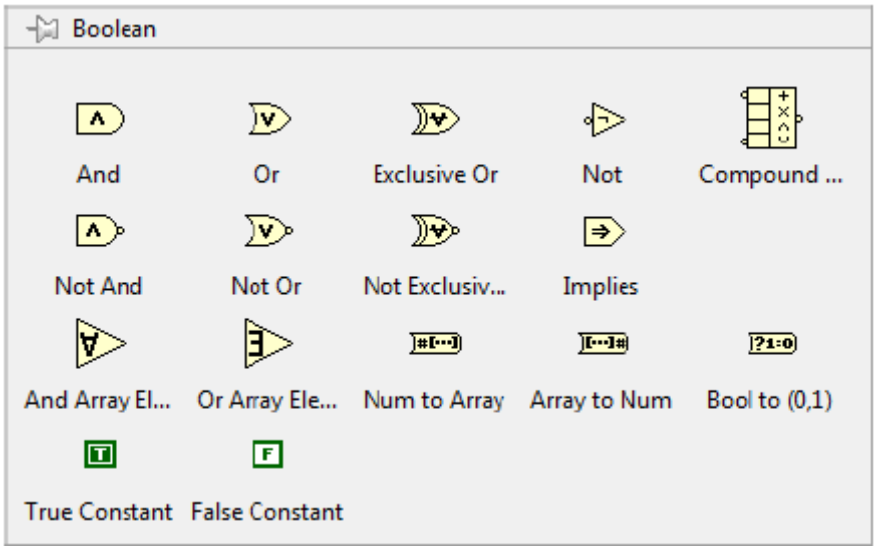

Рисунок 32 – Палитра логических операций [13]

Строки представляют собой последовательности отображаемых и неотображаемых символов в стандарте ASCII. Часто строки используются в качестве простых текстовых сообщений. В процессе управления приборами цифровые данные передаются в виде символьных строк, которые преобразуются затем в цифры. Строковые функции располагаются в палитре Programming – String (рисунок 33).

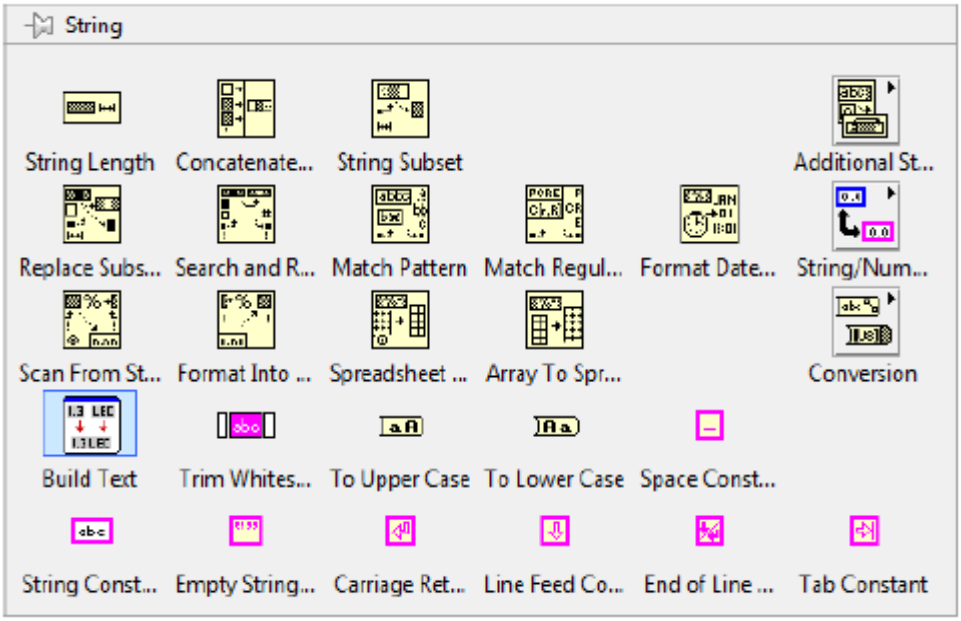

Рисунок 33 – Палитра строковых функций [13]

Под ветвлением понимается конструкция языка программирования, обеспечивающая выполнение определенной команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (набора команд) в зависимости от значения некоторого выражения.

В LabVIEW оператор ветвления реализуется на основе структуры Вариант (Case). Структура варианта, показанная на рисунке 37, имеет две или более поддиаграмм и аналогична операторам варианта (switch…case) или логическим оператором (if…else) в текстовых языках программирования. Лишь одна из поддиаграмм выполняется в зависимости от логического, числового или строкового значения, которое подается на терминал селектора. Структура варианта располагается в палитре Programming – Structure.

В теле структуры варианта можно выделить несколько основных элементов: Тело (2) – описывает блок-диаграммы для различных случаев; селектор (1) – отображает значение, с которым ассоциирована данная блок-диаграмма. Можно задавать в поле единичное значение, диапазон или список значений; терминал (3) – определяет, какая из описанных блок-диаграмм будет выполнена [8].

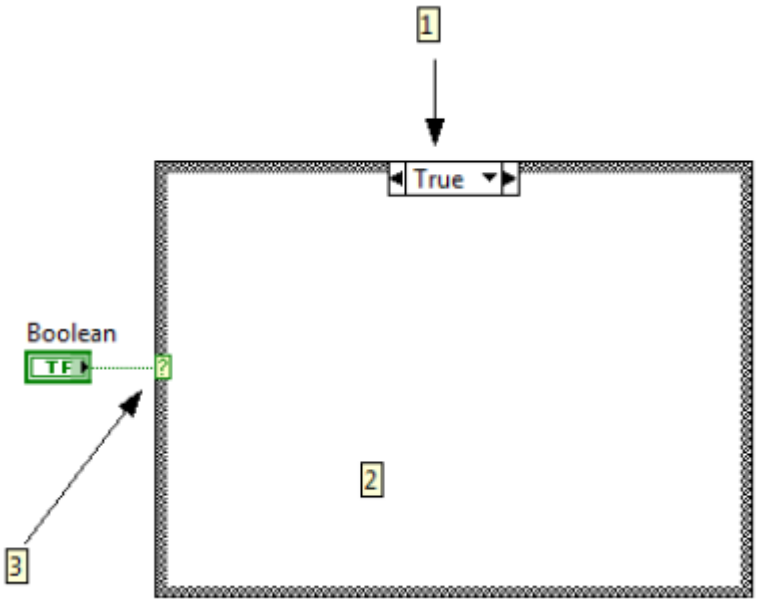

Рисунок 34 – Структура Case [13]

Если к терминалу структуры вариат подключено логическое значение, то структура будет иметь два варианта: ЛОЖЬ и ИСТИНА. Если к терминалу селектора подключены числовые или строковые данные, то структура может иметь почти неограниченное количество вариантов. При размещении структуры варианта на блок-диаграмме в первый раз она будет представлена в логической форме. Для того чтобы использовать в структуре числовые значения, необходимо подать числовой тип данных на терминал селектора. Терминал структуры вариант всегда должен быть подключен.

Цикл – это разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций. Последовательность инструкций, предназначенная для многократного исполнения, называется телом цикла, а единичное выполнение тела цикла называется итерацией. В LabVIEW различают два типа циклов: с фиксированным числом итераций (For Loop) и цикл по условию (While Loop). Структуры циклов расположены в палитре Programming – Structures.

Циклы могут иметь неограниченное количество входных и выходных терминалов для получения и выдачи данных. В терминалы могут подаваться как скалярные, так и векторные значения. Для создания нового терминала достаточно подключить провод к одной из внешних границ рамки цикла. При подключении таким образом массива к структуре For Loop автоматически создается перечисляемый вход. Каждая итерация цикла будет считывать новое значение из массива.

Цикл по условию (While Loop) выполняет многократное повторение операции над потоком данных, пока не выполнится логическое условие выхода. На рисунке 35 приведен общий вид структуры цикла по условию.

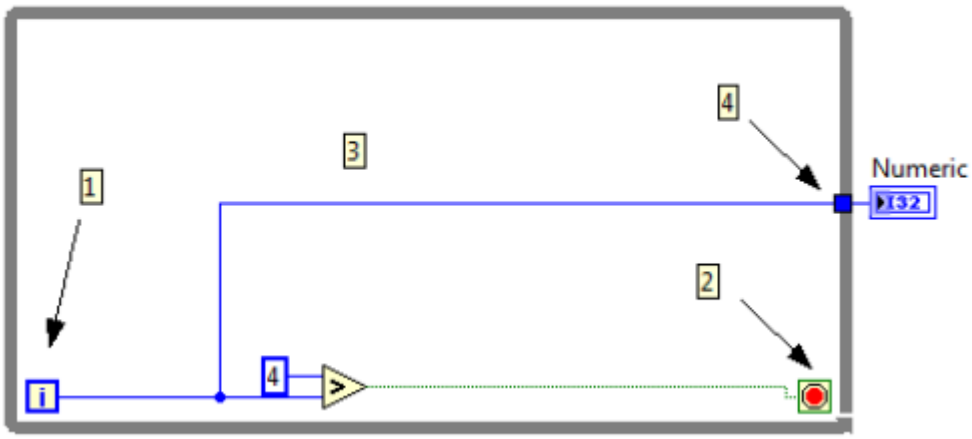

Рисунок 35 – Структура цикла по условию [13]

В его структуре можно выделить несколько основных компонентов. Тело цикла (3), содержащее инструкции, которые необходимо повторять; счетчик итераций (1), отсчитывающий номер текущей итерации; терминал условия завершения (2), имеющий два варианта работы – остановка, если на входе терминала ИСТИНА и остановка, если на входе терминала ЛОЖЬ (переключение между режимами осуществляется при помощи щелчка правой кнопки мыши); выходные терминалы (4), на которых появляются значения только после завершения работы цикла. По умолчанию на выходе структуры цикла по условию создается терминал скалярного вида [13].

3.3 Прошивка микроконтроллера Atmega328p с помощью ArduinoUNO.

При помощи ПО ArduinoIDE, используя стандартный скетч ArduinoISP, была запрограммирована плата ArduinoUNO как программатор.

Программатор подключался к разработанному чипу, по 4-х проводному интерфейсу данных ISP.

3.4 Код программы для загрузки в микроконтроллер.

Код программы для загрузки в микроконтроллер:

```
int t = 1;
boolean rele;
boolean dir;
char command;
const int LINE_BUFFER_SIZE = 80;
void setup () {
pinMode(8, OUTPUT); //реле
pinMode(5, OUTPUT); //STEP
pinMode(13, OUTPUT); //DIR
pinMode(3, OUTPUT); //SM0
pinMode(4, OUTPUT); //SM1
pinMode(7, OUTPUT); //RESET
pinMode(12, OUTPUT); //nEBL
Serial.begin(115200);
rele = HIGH;
dir = HIGH;
digitalWrite(8, rele);
attachInterrupt(0, nm, RISING);
}
int read_line(char* buffer, int bufsize)
{
for (int index = 0; index < bufsize; index++) {
// Wait until characters are available
while (Serial.available() == 0) {
}
char ch = Serial.read();
Serial.print(ch);
```

```
if (ch == '\n') {
buffer[index] = 0;
return index;
}
buffer[index] = ch;
}
char ch;
do {
while (Serial.available() == 0) {
}
ch = Serial.read();
Serial.print(ch);
} while (ch != '\n');
buffer[0] = 0;
return -1;
}
void loop() {
char line[LINE_BUFFER_SIZE];
if (read_line(line, sizeof(line)) < 0) {
Serial.println("Error: line too long");
return;
}
if (strcmp(line, "on") == 0) {
rele = LOW;
digitalWrite(8, rele);
} else if (strcmp(line, "off") == 0) {
rele = HIGH;
digitalWrite(8, rele);
}
else if (strcmp(line, "up") == 0) {
```

```
dir = HIGH;
digitalWrite(13, dir);
}
else if (strcmp(line, "down") == 0) {
dir = LOW;
digitalWrite(13, dir);
}
digitalWrite(3, LOW);
digitalWrite(4, HIGH);
one_step();
}
void one_step()
{
digitalWrite(5, HIGH);
delay(t);
digitalWrite(5, LOW);
delay(t);
}
void nm()
{
Serial.print("1");
}
```

Оболочка для программирования Arduino представляет из себя язык C++ с поддержкой большого количества библиотек для облегчения процесса написания программ. C++ предлагает широкий выбор различных типов данных. Написание кода программы начинается с объявления типа данных и далее идут две обязательные функции setup() и loop().

Типы данных (переменная) в Arduino — это ячейка оперативной памяти с именем, хранящая в себе числовое или буквенное значение. С числовыми

типами данных (переменными) можно выполнять любые математические операции: умножение, деление, сложение, вычитание, преобразование в другой тип данных и т.д.

Переменные типа boolean могут принимать одно из двух значений: true или false. (Каждая переменная типа boolean занимает в памяти один байт.)

Тип данных char - это знаковый тип, т.е. переменные данного типа могут хранить числовые значения в диапазоне от -128 до 127. В качестве беззнакового однобайтового (8 бит) типа данных используйте тип данных byte.

Целочисленный тип int - это основной тип данных для хранения чисел. В Arduino Uno (и других платах на базе микроконтроллеров ATmega) переменные типа int хранят 16-битные (2-байтовые) значения. Такая размерность дает диапазон от -32768 до 32767 (минимальное значение -2^15 и максимальное значение (2^15)-1). В переменных типа int отрицательные числа представляются с помощью техники дополнительного кода. Старший бит, который иногда называют "знаковым битом", указывает на то, является ли данное число отрицательным. Остальные биты инвертируются, после чего к результату добавляется 1.

AttachInterrupt() – задает функцию, которую необходимо вызвать при возникновении внешнего прерывания. Заменяет предыдущую функцию, если таковая была ранее ассоциирована с прерыванием. В большинстве плат Ардуино существует два внешних прерывания: номер 0 (цифровой вывод 2) и 1 (цифровой вывод 3).

AttachInterrupt() определяет условие, при котором должно срабатывать прерывание:

- LOW – прерывание будет срабатывать всякий раз, когда на выводе присутствует низкий уровень сигнала;
- CHANGE – прерывание будет срабатывать всякий раз, когда меняется состояние вывода;

- RISING – прерывание сработает, когда состояние вывода изменится с низкого уровня на высокий;
- FALLING – прерывание сработает, когда состояние выводов изменится с высокого на низкий.

PinMode() конфигурирует режим работы указанного вывода: как вход либо как выход.

Delay() приостанавливает выполнение программы на указанный промежуток времени (в миллисекундах).

DigitalRead() считывает уровень сигнала HIGH или LOW с указанного цифрового вывода.

DigitalWrite() отправляет на цифровой вывод значение HIGH или LOW.

Оператор do ... while. Цикл do работает так же, как и цикл while, за исключением того, что условие проверяется в конце цикла, таким образом, цикл do будет всегда выполняться хотя бы раз.

Оператор For. Конструкция for используется для повторения блока операторов, заключенных в фигурные скобки. Счетчик приращений обычно используется для приращения и завершения цикла. Оператор for подходит для любых повторяющихся действий и часто используется в сочетании с массивами коллекций данных/выводов.

Функция If..else. Конструкция if..else предоставляет больший контроль над процессом выполнения кода, чем базовый оператор if, позволяя осуществлять несколько проверок, объединенных вместе. Например, аналоговый вход может быть проверен и выполнено одно действие, если на входе меньше 500, или другой действие, если на входе 500 или больше.

Функция setup() вызывается, когда стартует скетч. Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т.д. Функция setup() запускает только один раз, после каждой подачи питания или сброса платы Arduino.

После вызова функции setup(), которая инициализирует и устанавливает первоначальные значения, функция loop() делает точь-в-точь то, что означает её название, и крутится в цикле, позволяя вашей программе совершать вычисления и реагировать на них [14].

3.5 Схема автоматизации работы монохроматора МДР-23

На рисунке 26 представлена схема управления монохроматором:

1 – плата с PNP транзисторами, радиаторами охлаждения, микроконтроллером ATMega 328p, драйвером шагового двигателя DRV8805;

2 – реле-регулятор;

3 – конектор ШДР;

4 – блок питания;

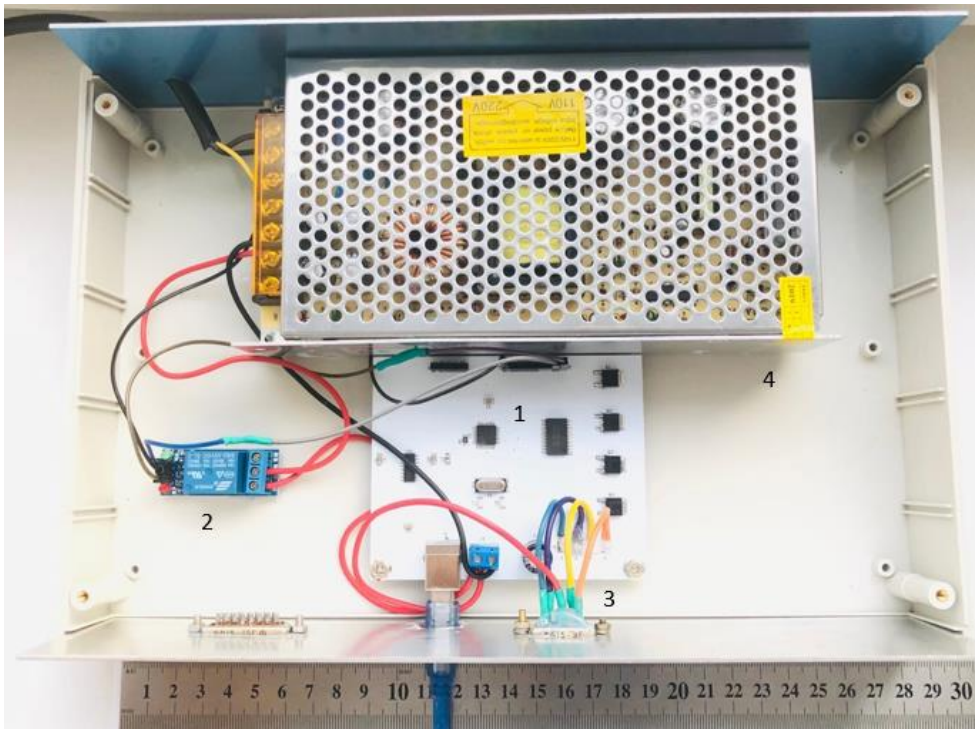

Рисунок 36 – Схема управления монохроматором МДР-23

4 Результаты измерений и обсуждения

В качестве источника излучения был использован светодиод красного свечения, представленный на рисунке 38.

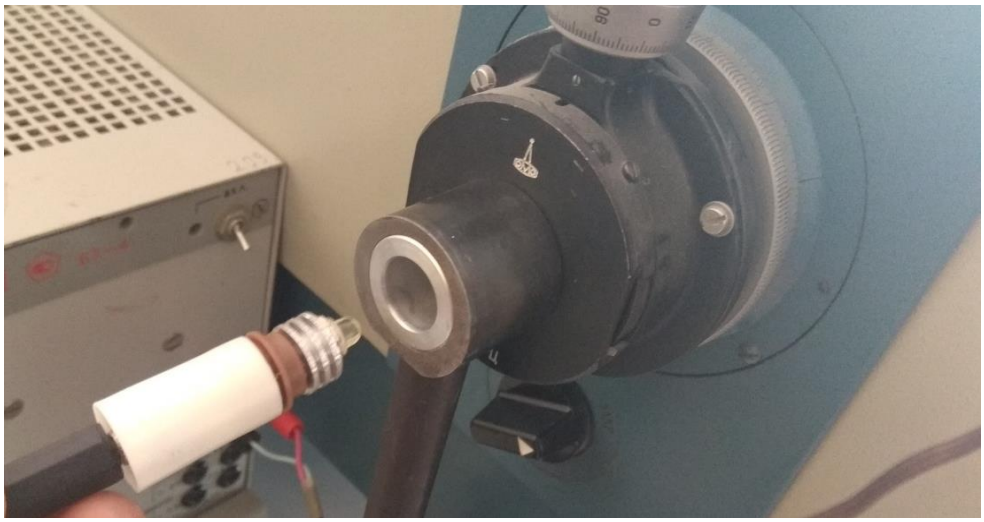

Рисунок 37 – Светодиод и входная щель монохроматора

На виртуальном приборе, также, как и на блоке управления шаговым двигателем, была установлена максимальная скорость измерения. Длины волн на виртуальном приборе были выбраны исходя из того, что светодиод излучает красный свет, а диапазон длин волн красного свечения варьируется от 625 до 740 нм.

На рисунке 39 видно, что значение величины фототока изменяется плавно образно, без резких скачков и спадов величин. При длине волны равной 658 нм величина фототока максимальна.

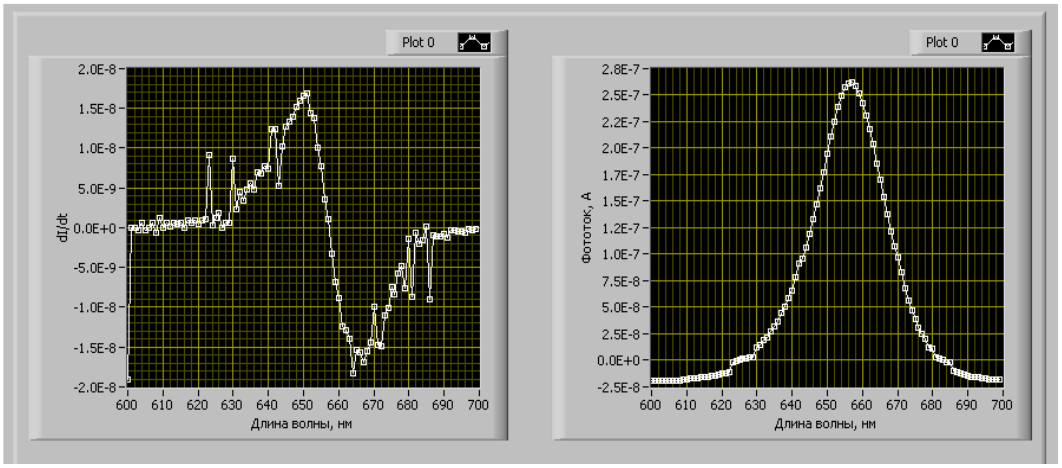

Рисунок 38 – Графики, полученные на виртуальном приборе

Было проведено сравнение спектров электролюминесценции «красного» светодиода, измеренного при помощи собранной в настоящей работе автоматизированной установки и коммерческого спектрометра OceanOptics USB4000+ (рисунок 22). Наблюдается совпадение измеренных спектров.

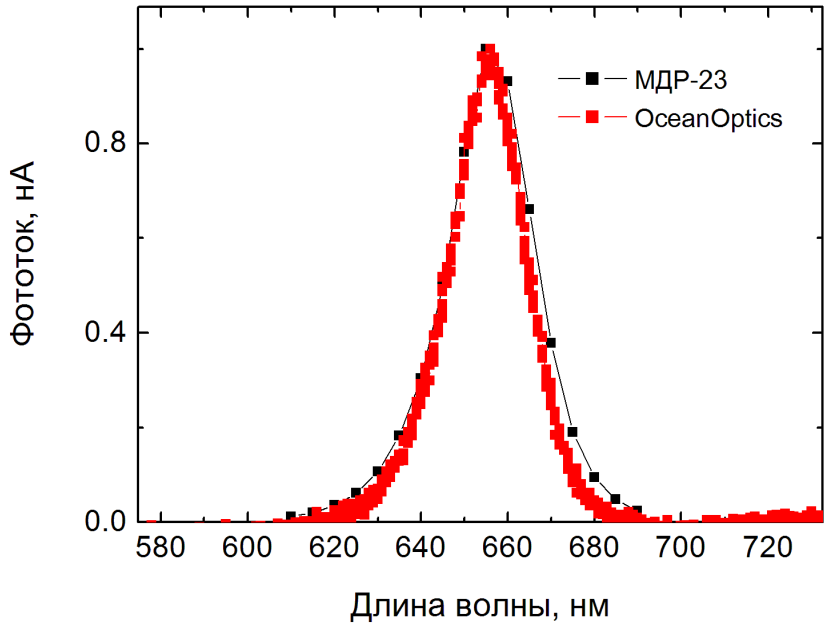

Рисунок 39– Спектры, полученные при помощи собранной в настоящей работе автоматизированной установки и коммерческого спектрометра OceanOptics USB4000+

ЗАКЛЮЧЕНИЕ

В процессе выполнения работы были проведены комплексные испытания устройства, исследовано устройство монохроматора и его периферийное оборудование, изучена среда разработки LabView, создан полуавтоматизированный виртуальный прибор, исследованы графики зависимости величины фототока от длины волны излучения светодиода.

Использование виртуального прибора совместно с монохроматором и источником-измерителем позволяет грамотно подойти к решению поставленной задачи энергосбережения. Благодаря такому комплексу устройств можно проводить проверку качества изготавливаемых светодиодных структур и сравнивать светодиоды различных производителей.

В результате работы сделано следующее:

1) проведен литературный обзор;
2) освоена среда разработки LabView;
3) разработано программное обеспечение для автоматизации процесса измерения спектров электролюминесценции при помощи монохроматора МДР-23;
4) проведены комплексные испытания устройства;
5) проанализированы полученные данные.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Топорцев А. С. Новые приборы и методы измерений. Монохроматоры – издательство Москва, 1950г. – 46с.
2. Знакомимся с LabView [Электронный ресурс] – URL: http://easyelectronics.ru/znakomimsya-s-labview.html (Дата обращения 23.09.2021).
3. Гермогенов В. П. Материалы, структуры и приборы полупроводниковой оптоэлектроники. – Томск. Издательский Дом Томского государственного университета 2015. – 271с.
4. Микроконтроллер Arduino Uno [Электронный ресурс] – URL: https://doc.arduino.ua/ru/hardware/Uno (Дата обращения 04.10.2021).
5. Васильев А.С., Лашманов О.Ю. Основы программирования в среде LabVIEW. – Санкт-Петербург. Университет ИТМО 2015. – 80с.
6. Микроконтроллер Arduino Uno [Электронный ресурс] – URL: https://otransformatore.ru/raznoe/arduino-nano-nano-raspinovka-razlichnyh-plat-i-shemy-podklyucheniya/ (Дата обращения 06.10.2021).
7. Магда Ю. С. LabVIEW: практический курс для инженеров и разработчиков. – Москва: ДМК. 2012 – 207 с.
8. Трэвис Дж., Кринг Дж. LabVIEW для всех: графическое программирование стало простым и увлекательным. – М.: Нью-Джерси Прентис Холл, 2009. – 1–32 с.
9. Евдокимов Ю. К., Линдваль В. Р., Щербаков Г. И. LabVIEW в научных исследованиях. – Москва: ДМК Пресс, 2012. – 399 с.
10. Микроконтроллер ATMega328p [Электронный ресурс] – URL: https://www.microchip.com/en-us/product/ATmega328p (Дата обращения 12.10.2021).
11. Микроконтроллер ATMega328p [Электронный ресурс] – URL: https://www.ruselectronic.com/atmega328p/ (Дата обращения 21.10.2021).

12. Программный комплекс KiCad [Электронный ресурс] – URL: http://myowndevice.ru/index.php/theory/item/13-kicad-programma-trassirovki-plat (Дата обращения 16.09.2021).

13. Источник-измеритель 2611B Keithley [Электронный ресурс] – URL: http://www.eliks.ru/kio/index.php?SECTION_ID=1203&ELEMENT_ID=216352 (Дата обращения 16.10.2021).

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

ТАРИФ NEW
Free
ИЗМЕНИТЬ

МОДУЛИ И КОЛЛЕКЦИИ
Подключено: 1 смотреть
ПОДКЛЮЧИТЬ ЕЩЕ

БАЛЛЫ
0
ПОПОЛНИТЬ

ПОЛЬЗОВАТЕЛЬ
fwaa@bk.ru
ПРОВЕРИТЬ ДОКУМЕНТ

МЕНЮ
ru

ГЛАВНАЯ / КАБИНЕТ / РЕЗУЛЬТАТЫ ПРОВЕРКИ

Оригинальность	Заимствования	Цитирования	Самоцитирования
80,85%	19,15%	0%	0%

ПОЛНЫЙ ОТЧЕТ | КРАТКИЙ ОТЧЕТ | ИСТОРИЯ ОТЧЕТОВ | РАСПЕЧАТАТЬ | ВЫГРУЗИТЬ | СОЗДАТЬ ССЫЛКУ

Свойства документа

Структура документа NEW

Параметры проверки

Текстовые метрики

Имя исходного файла	Diplomnaya_rabota.pdf
Авторы документа	Григорьев Алексей Андреевич
Название документа	Diplomnaya_rabota_07610
Тип документа	Дипломная работа

РЕДАКТИРОВАТЬ СВОЙСТВА