

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
по выполнению лабораторных работ  
по дисциплине  
**«ОРГАНИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ»**  
**(«ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА»)**  
для студентов направления подготовки  
44.03.05 Педагогическое образование (с двумя профилями подготовки)  
направленность (профиль):  
«Информатика и математика»

Ставрополь, 2026

## СОДЕРЖАНИЕ

### Оглавление

|                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------|-----|
| ВВЕДЕНИЕ                                                                                                     | 3   |
| Лабораторная работа 1 Дополнительное образование как компонент педагогической системы                        | 5   |
| Лабораторная работа 2 Особенности организации системы дополнительного образования по информатике и ИКТ       | 12  |
| Лабораторная работа 3 Изучение возможностей среды визуального программирования LEGO Mindstorms Education EV3 | 18  |
| Лабораторная работа 4 Блоки действий (зеленая палитра) LEGO Mindstorms Education EV3                         | 30  |
| Лабораторная работа 5 Работа с датчиками в среде LEGO Mindstorms Education EV3                               | 43  |
| Лабораторная работа 6 Операции с данными в среде LEGO Mindstorms Education EV3                               | 52  |
| Лабораторная работа 7 Знакомство с программно-аппаратной платформой Arduino                                  | 65  |
| Лабораторная работа 8 Цифровые выходы Arduino. Широтно-импульсная модуляция (ШИМ)                            | 74  |
| Лабораторная работа 9 Аналоговые входы платы Arduino и резистивные датчики                                   | 86  |
| Лабораторная работа 10 Проектирование устройств с датчиком температуры на базе Arduino                       | 98  |
| Лабораторная работа 11 Программирование пьезодинамика и воспроизведение мелодий в проектах Arduino           | 105 |
| Лабораторная работа 12 Подключение и программирование семисегментного индикатора и ЖК-дисплея                | 119 |
| Лабораторная работа 13 Применение ультразвукового датчика в проектах Arduino                                 | 130 |
| Лабораторная работа 14 Виды и назначение двигателей в устройствах на базе Arduino                            | 139 |
| СПИСОК ЛИТЕРАТУРЫ                                                                                            | 153 |

## **ВВЕДЕНИЕ**

### **Цель и задачи освоения дисциплины**

Цель изучения дисциплины «Организация дополнительного образования по информатике» - формирование набора профессиональных компетенций будущего бакалавра по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность (профиль) «Информатика и математика».

Основными задачами изучения дисциплины являются:

- сформировать знания, умения и навыки для разработки дополнительных образовательных программ и их компонентов, в том числе с использованием информационно-коммуникационных технологий;
- развивать умения и навыки для осуществления педагогической деятельности по направлению подготовки, а также знания для разработки методического обеспечения данной деятельности;
- сформировать знания о способах организации деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ;
- сформировать навыки организации педагогической деятельности по направлению подготовки с использованием дистанционных технологий и электронных сервисов;
- сформировать навыки сборки схем устройств и программирования работы собранных устройств в виртуальном эмуляторе, а также в среде образовательной робототехнической платформы;
- развивать навыки и умения представления результатов научно-исследовательской деятельности в информационно-образовательном пространстве с помощью современных цифровых инструментов;
- развивать творческие способности, логическое мышление, а также образное и техническое мышление обучающихся;
- развивать умение выстраивать гипотезу, сопоставлять с полученным результатом, умение выразить свой замысел.

### **В результате освоения дисциплины студенты должны:**

#### **Знать:**

- способы организации образовательной деятельности обучающихся при обучении информатике и ИКТ; приемы мотивации школьников к учебной и учебно-исследовательской работе по информатике и ИКТ;
- компоненты информационной образовательной среды и их дидактические возможности; принципы и подходы к организации информационной образовательной среды для обучения информатике и ИКТ;
- современные тенденции развития цифровых технологий и возможности данных технологий для решения задач профессиональной деятельности.

#### **Уметь:**

- организовать различные виды деятельности обучающихся в образовательном процессе по информатике и ИКТ; применять приемы, направленные на поддержание познавательного интереса;
- обосновывать и включать образовательные ресурсы в информационную образовательную среду и процесс обучения информатике и ИКТ;
- использовать цифровые технологии для решения задач профессиональной деятельности.

**Владеть:**

- умениями по организации разных видов деятельности обучающихся при обучении информатике и ИКТ и приемами развития познавательного интереса;
- умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в том числе, для реализации дистанционных образовательных технологий и электронного обучения;
- навыками выбора наиболее оптимальной технологии для решения задач профессиональной деятельности.

**Наименование лабораторных работ**

| № Темы дисциплины       | Наименование тем дисциплины, их краткое содержание                                                             |
|-------------------------|----------------------------------------------------------------------------------------------------------------|
| 1.                      | <b>Лабораторная работа 1.</b> Дополнительное образование как компонент педагогической системы                  |
| 2.                      | <b>Лабораторная работа 2.</b> Особенности организации системы дополнительного образования по информатике и ИКТ |
| 3.                      | <b>Лабораторная работа 3.</b> Изучение возможностей среды визуального программирования LEGO Mindstorms EV3     |
| 4.                      | <b>Лабораторная работа 4.</b> Блоки действий (зеленая палитра) LEGO Mindstorms EV3                             |
| 5.                      | <b>Лабораторная работа 5.</b> Работа с датчиками в среде LEGO Mindstorms Education EV3                         |
| 6.                      | <b>Лабораторная работа 6.</b> Операции с данными в среде LEGO Mindstorms Education EV3                         |
| 7.                      | <b>Лабораторная работа 7.</b> Знакомство с программно-аппаратной платформой Arduino                            |
| 8.                      | <b>Лабораторная работа 8.</b> Цифровые выходы Arduino. Широтно-импульсная модуляция (ШИМ)                      |
| 9.                      | <b>Лабораторная работа 9.</b> Аналоговые входы платы Arduino и резистивные датчики                             |
| 10.                     | <b>Лабораторная работа 10.</b> Проектирование устройств с датчиком температуры на базе Arduino                 |
| 11.                     | <b>Лабораторная работа 11.</b> Программирование пьезодинамика и воспроизведение мелодий в проектах Arduino     |
| 12.                     | <b>Лабораторная работа 12.</b> Подключение и программирование семисегментного индикатора и ЖК-дисплея          |
| 13.                     | <b>Лабораторная работа 13.</b> Применение ультразвукового датчика в проектах Arduino                           |
| 14.                     | <b>Лабораторная работа 14.</b> Виды и назначение двигателей в устройствах на базе Arduino                      |
| <b>Итого за семестр</b> |                                                                                                                |
| <b>Итого</b>            |                                                                                                                |

## СТРУКТУРА И СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ РАБОТ

### Лабораторная работа 1

#### Дополнительное образование как компонент педагогической системы

**Цель работы:** сформировать у студентов знания о системе дополнительного образования как о компоненте педагогической системы в целом, изучить основные понятия системы дополнительного образования, выявить цели и задачи дополнительного образования, рассмотреть направления, содержание системы дополнительного образования, дополнительные общеобразовательные программы общеразвивающего характера, проанализировать особенности системы дополнительного образования для различных категорий детей.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

#### Содержание:

1. Нормативно-правовая база организации дополнительного образования. Цели дополнительного образования.
2. Особенности разработки, утверждения и реализации программ в системе дополнительного образования детей.
3. Особенности системы дополнительного образования для различных категорий детей

#### **1. Нормативно-правовая база организации дополнительного образования.**

Нормативно-правовую базу организации дополнительного образования в Российской Федерации составляют следующие документы:

1. Федеральный Закон от 29.12.2012г. № 273-ФЗ «Об образовании в Российской Федерации»;
2. Стратегия развития воспитания в РФ на период до 2025 года (распоряжение Правительства РФ от 29 мая 2015 г. № 996-р);
3. Постановление Главного государственного санитарного врача РФ от 28.09.2020 N 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
4. Постановление Главного государственного санитарного врача РФ от 28.01.2021 № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические

нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» (разд. VI. Гигиенические нормативы по устройству, содержанию и режиму работы организаций воспитания и обучения, отдыха и оздоровления детей и молодежи»);

5. Паспорт федерального проекта «Успех каждого ребенка» (утвержден на заседании проектного комитета по национальному проекту «Образование» 07 декабря 2018 г., протокол № 3);

6. Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

7. Приказ Министерства просвещения Российской Федерации от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;

8. Приказ Министерства просвещения Российской Федерации от 13.03.2019 № 114 «Об утверждении показателей, характеризующих общие критерии оценки качества условий осуществления образовательной деятельности организациями, осуществляющими образовательную деятельность по основным общеобразовательным программам, образовательным программам среднего профессионального образования, основным программам профессионального обучения, дополнительным общеобразовательным программам».

Целями развития дополнительного образования детей являются создание условий для самореализации и развития талантов детей, а также воспитание высоконравственной, гармонично развитой и социально ответственной личности.

## **2. Особенности разработки, утверждения и реализации программ в системе дополнительного образования детей.**

Дополнительное образование – вид образования, который направлен на всестороннее удовлетворение образовательных потребностей человека в интеллектуальном, духовно-нравственном, физическом и (или) профессиональном совершенствовании и не сопровождается повышением уровня образования (ФЗ №273, гл.1 ст.2 п.14).



Рис. 1. Направления дополнительных образовательных программ и целевые группы

Образовательная программа – комплекс основных характеристик образования (объем, содержание, планируемые результаты) и организационно-педагогических условий, который представлен в виде учебного плана, календарного учебного графика, рабочих программ учебных предметов, курсов, дисциплин (модулей), иных компонентов, оценочных и методических материалов, а также в предусмотренных настоящим Федеральным законом случаях в виде рабочей программы воспитания, календарного плана воспитательной работы, форм аттестации (ФЗ №273, гл.1 ст.2 п. 9).

Содержание дополнительных общеразвивающих программ и сроки обучения по ним определяются образовательной программой, разработанной и утвержденной организацией, осуществляющей образовательную деятельность. Дополнительные общеразвивающие программы формируются с учетом пункта 9 статьи 2 Федерального закона «Об образовании в РФ» (Порядок, п.5) Содержание дополнительных предпрофессиональных программ определяется образовательной программой, разработанной и утвержденной организацией, осуществляющей образовательную деятельность, в соответствии с федеральными государственными требованиями (ФЗ №273, гл.10 ст.75 п.4).

Структурно в образовательной программе можно выделить следующие элементы: 1) титульный лист; 2) раздел 1. Комплекс основных характеристик программы; 3) раздел 2. Комплекс организационно-педагогических условий; 4) список литературы; 5) приложения.

### 3. Особенности системы дополнительного образования для различных категорий детей.

Доступность дополнительного образования предполагается для различных категорий детей:

- равнодоступность для детей из городов и детей из сел;
- доступность для детей из семей с различными социально-экономическими характеристиками;

- доступность образовательных программ в новых секторах дополнительного (неформального) образования;
- доступность для детей вне зависимости от гендерных различий;
- доступность для детей вне зависимости от возрастных особенностей;
- доступность для детей с ОВЗ.

Процесс получения дополнительного образования расширяет возможность для увеличения количества детей с неординарными способностями, проявляющимися за рамками учебной деятельности.

Для реализации этой возможности целесообразны такие педагогические технологии как:

- формирование навыков публичного обсуждения и отстаивания своих идей и результатов в творчестве, исследовательской работе;
- индивидуализированная психологическая поддержка и помощь с учетом индивидуального своеобразия личности каждого одаренного ребенка;
- дифференцированное обучение одаренных детей по специально разработанным программам;
- применение поощрительной системы в этой работе.

Работа по выявлению и поддержке одаренных и талантливых детей обычно осуществляется поэтапно:



Рис. 2. Этапы работы по выявлению и поддержке одаренных детей

Инклюзивность еще одно критически важное требование к современному дополнительному образованию.

Основными проблемами по привлечению детей с ОВЗ многие педагоги называют необходимость, сложность и трудоемкость адаптации программ для различных категорий детей, а также отсутствие специальных приспособлений (подъемники и др.). Другим важным аспектом при работе с детьми с ОВЗ является необходимость специальной подготовки педагога к взаимодействию с обучающимися.

Для учащихся с ограниченными возможностями здоровья, детей-инвалидов, инвалидов организации, осуществляющие образовательную деятельность, организуют образовательный процесс по дополнительным общеобразовательным программам с учетом особенностей психофизического развития указанных категорий учащихся.

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет». Также

необходима возможность подключения к электронной информационно-образовательной среде университета.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет.

#### Задания

1. Подготовьте доклад на одну из тем: «Организация системы дополнительного образования в различные периоды общественного развития», «Известные отечественные педагоги, организовавшие систему дополнительного образования», «Организация системы дополнительного образования: зарубежный опыт», «Направления и содержание системы дополнительного образования», «Адаптивные образовательные программы дополнительного образования детей, имеющих ограничения здоровья (ОВЗ)», «Системы дополнительного образования для одаренных детей». Разработайте презентацию на основе вашего доклада.

2. Изучите документы, составляющие нормативно-правовую базу реализации дополнительных образовательных программ, и занесите полученную информацию в таблицу:

| № п/п | Документ                                                                                                                                                          | Дата принятия | Сроки реализации (если имеются) | Цели и задачи | Ответственные в реализации | Основные пункты, регламентирующие деятельность в сфере дополнительного образования |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------------------------|---------------|----------------------------|------------------------------------------------------------------------------------|
| 1.    | Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 N 273-ФЗ                                                                                  |               |                                 |               |                            |                                                                                    |
| 2.    | Национальный проект «Образование» (24 декабря 2018 г.)<br>Правительства РФ от 31.03.2022 N 678-р)                                                                 |               |                                 |               |                            |                                                                                    |
| 3.    | Концепция развития дополнительного образования до 2030 г.                                                                                                         |               |                                 |               |                            |                                                                                    |
| 4.    | ПРИКАЗ от 9 ноября 2018 г. N 196 об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам |               |                                 |               |                            |                                                                                    |

|    |                                                        |  |  |  |  |  |
|----|--------------------------------------------------------|--|--|--|--|--|
| 5. | Госпрограмма<br>«Развитие<br>образования» (2018<br>г.) |  |  |  |  |  |
|----|--------------------------------------------------------|--|--|--|--|--|

3. Заполните таблицу, отражающую виды и характеристики дополнительных образовательных программ:

| Характеристики образовательных программ                                    | Дополнительные общеобразовательные программы |                                               | Дополнительные профессиональные программы |
|----------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------|-------------------------------------------|
|                                                                            | Дополнительные общеразвивающие программы     | Дополнительные предпрофессиональные программы |                                           |
| Нормативно-правовая база формирования содержания образовательной программы |                                              |                                               |                                           |
| Цели и задачи обучения                                                     |                                              |                                               |                                           |
| Контингент обучающихся                                                     |                                              |                                               |                                           |
| Типы организаций, имеющих право реализации образовательной программы       |                                              |                                               |                                           |

4. На основе анализа сайта <https://p26.навигатор.дети/> (Навигатор дополнительного образования Ставропольского края) приведите направленность и количество дополнительных образовательных программ для детей, реализуемых в Северо-Кавказском федеральном университете, г. Ставрополь. Данные занесите в таблицу:

| № п/п | Направленность программ | Направления программ в рамках данной направленности | Количество реализуемых программ по данной направленности | Примеры реализуемых программ (2-3 примера по каждой направленности) |
|-------|-------------------------|-----------------------------------------------------|----------------------------------------------------------|---------------------------------------------------------------------|
| 1.    | Социально-гуманитарная  |                                                     |                                                          |                                                                     |
| 2.    | ...                     |                                                     |                                                          |                                                                     |
| ...   |                         |                                                     |                                                          |                                                                     |

5. Проведите анализ статистических данных, приведенных на сайте <http://www.dop.edu.ru/> (Информационный портал системы дополнительного образования детей) для сравнительной характеристики состояния дополнительного образования детей в России и Ставропольском крае. Заполните таблицу:

|  |                     |                                  |
|--|---------------------|----------------------------------|
|  | Количество в стране | Количество в Ставропольском крае |
|--|---------------------|----------------------------------|

| Направленность образовательных программ | Обучающихся по данной направленности | Организаций, реализующих программы направленности | Педагогов, задействованных в реализации программ направленности | Обучающихся по данной направленности | Организаций, реализующих программы направленности | Педагогов, задействованных в реализации программ направленности |
|-----------------------------------------|--------------------------------------|---------------------------------------------------|-----------------------------------------------------------------|--------------------------------------|---------------------------------------------------|-----------------------------------------------------------------|
| Художественная                          |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| Физкультурно-спортивная                 |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| Естественнонаучная                      |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| Социально-гуманитарная                  |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| Техническая                             |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| Туристско-краеведческая                 |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| Интеллектуальные игры                   |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |
| <b>Выводы:</b>                          |                                      |                                                   |                                                                 |                                      |                                                   |                                                                 |

6. Проведите анализ источников с целью заполнения данных следующей таблицы:

| Виды организаций              | Представленность детей с ОВЗ                                                                                    |                           |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------------------|
|                               | Доля организаций, в которых обучаются дети с ОВЗ (иными словами, присутствует возможность обучения детей с ОВЗ) | Процент обучающихся с ОВЗ |
| Государственные организации   |                                                                                                                 |                           |
| Муниципальные организации     |                                                                                                                 |                           |
| Негосударственные организации |                                                                                                                 |                           |
| Все организации               |                                                                                                                 |                           |

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и выполнить в тетради представленные практические задания.
4. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку и содержание выполненных заданий.

#### **Контрольные вопросы**

1. Приведите классификацию видов дополнительного образования.
2. Представьте цели и задачи дополнительного образования.
3. Назовите направленности системы дополнительного образования детей.
4. Опишите особенности одного из видов адаптированных дополнительных общеразвивающих программ:

- 1) для слепых и слабовидящих детей
- 2) для детей с нарушениями опорно-двигательного аппарата
- 3) для обучающихся с тяжелыми нарушениями речи
- 4) для обучающихся с нарушениями слуха
- 5) для обучающихся с расстройствами аутистического спектра (РАС)
- 6) для обучающихся с задержкой психического развития (ЗПР)

5. Приведите характеристики следующих педагогических подходов к проектированию образовательных программ для одарённых детей: обогащение, углубление, проблематизация, ускорение.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [1, 2];

дополнительная литература – [1, 2, 4, 11, 13, 24, 29].

### **Лабораторная работа 2**

#### **Особенности организации системы дополнительного образования по информатике и ИКТ**

**Цель работы:** рассмотреть цели, задачи, содержание системы дополнительного образования по информатике и ИКТ, а также особенности организации дополнительного образования в области робототехники.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием

приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### **Теоретическая часть**

#### *Содержание:*

1. Общая характеристика дополнительного образования в области информатики и ИКТ.

2. Введение в образовательную робототехнику.

#### **1. Общая характеристика дополнительного образования в области информатики и ИКТ.**

В настоящее время некоторые направления дополнительного образования как составной части непрерывного образования предназначены не только для обеспечения досуга, но и направлены на предпрофессиональную и профессиональную подготовку детей, на подготовку к жизни в современном мире, на формирование готовности личности к непрерывному образованию. Дополнительное образование в области информатики и ИКТ также нацелено на подготовку детей к жизни в современном информационном обществе. На уровне государства формируются различные программы, создающие возможности получения образования по информатике для каждого ребенка. Например, создание детских технопарков «Кванториум» осуществляется на основе паспорта приоритетного проекта «Доступное дополнительное образование для детей» и других нормативно-правовых документов.

Основная причина обращения к дополнительному образованию заключается в несоответствии возможностей современной школы в области обучения информатике и потребностей общества и личности в освоении ИКТ. Это связано:

- 1) с высокими темпами развития области ИКТ и инертностью школьных программ по информатике;
- 2) с недостаточным обеспечением школ компьютерной техникой и квалифицированными кадрами;
- 3) с невозможностью учесть потребности и способности каждого обучающегося в основной школе, т.к. школьные программы ориентированы на среднего ученика;
- 4) с необходимостью развивать творческие способности одаренных в области ИКТ детей.

В современных условиях информатика и информационные технологии – это одна из фундаментальных отраслей научного знания, формирующая системно-информационный, мировоззренческий, аналитический подход к анализу окружающего мира, стремительно развивающаяся и постоянно расширяющаяся область практической деятельности человека, связанная с использованием ИКТ. Анализ практики обучения информатике в учреждениях ДПО свидетельствует о том, что ДПО позволяет наиболее полно осуществить идеи личностно-ориентированного образования, индивидуальный подход в обучении, дифференциацию обучения.

#### **2. Введение в образовательную робототехнику.**

Образовательная робототехника – новое для отечественного образования направление, в настоящее время она интенсивно развивается как в нашей стране, так и в

мировой образовательной практике. В настоящее время происходит быстрое обновление робототехнических конструкторов (за последние десятилетия компания LEGO несколько раз обновила состав робототехнического конструктора и программное обеспечение к нему), в сети Интернет неуклонно растет число информационных и учебных материалов по образовательной робототехнике, в то же время большая часть этих материалов представлена на английском языке и требует адаптации к российской образовательной системе.

В основе образовательной робототехники лежат игровые технологии, этим в значительной степени обусловлена ее популярность у школьников и молодежи. В то же время образовательная робототехника – это интегративная предметная область, как и для любого интегративного курса для нее характерно соединение нескольких учебных предметов в единое целое и выделение ведущего предмета.

Во всех известных нам разработках ведущим предметом для изучения образовательной робототехники является информатика. Это может быть связано с тем, что знакомство учащихся с робототехникой начинается именно с создания простейших программ-алгоритмов для робота на основе какой-либо базовой модели (например, двухмоторной тележки).

На сегодняшний день образовательная робототехника в российском образовании осваивается учащимися в различных кружках, на элективных курсах посредством образовательных конструкторов:

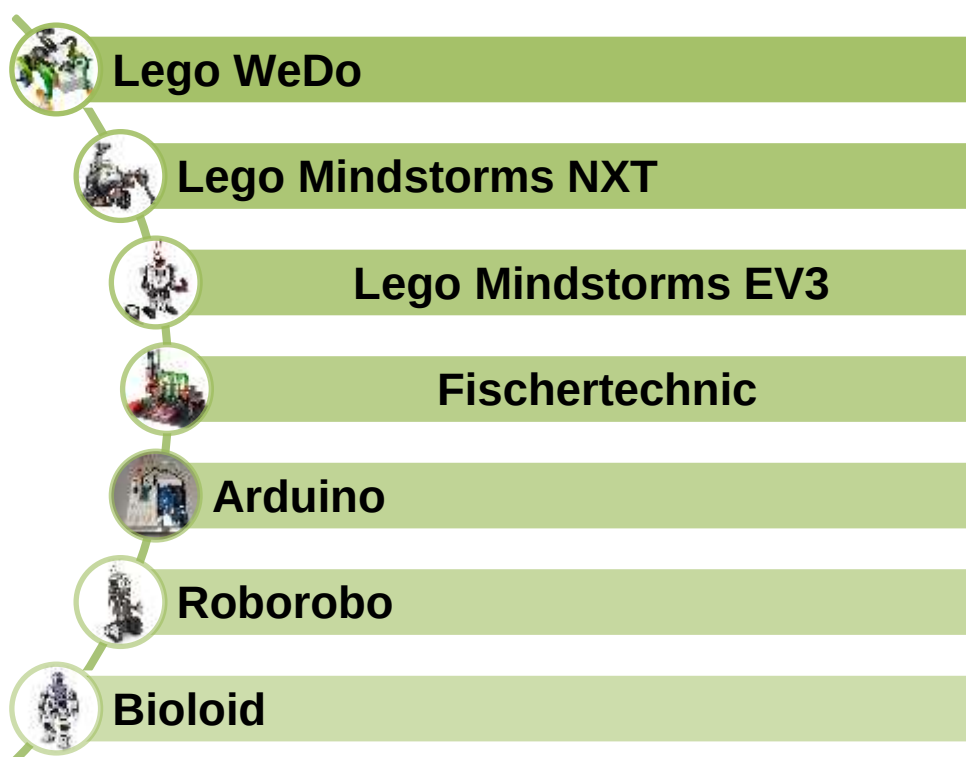


Рис. 3. Популярные робототехнические конструкторы

**Характеризуя образовательную робототехнику как интегративный курс, можно выделить несколько аспектов ее преподавания:**



Рис. 4. Аспекты преподавания робототехники как интегративного курса

Работа с конструкторами LEGO основывается на принципе «learning by making», т.е. «обучение через действие». Данный принцип реализует циклическую модель обучения, которая включает 4 этапа: 1) взаимосвязь, 2) конструирование, 3) рефлексия, 4) развитие.

«Взаимосвязь» предполагает, что пополнение багажа знаний происходит, когда вновь приобретенные опыт и знания удастся соединить с уже имеющимися или сделать их стимулом, отправной точкой для нового этапа обучения.

«Конструирование» подразумевает создание моделей и генерацию идей. Предлагаются три вида «Конструирования»:

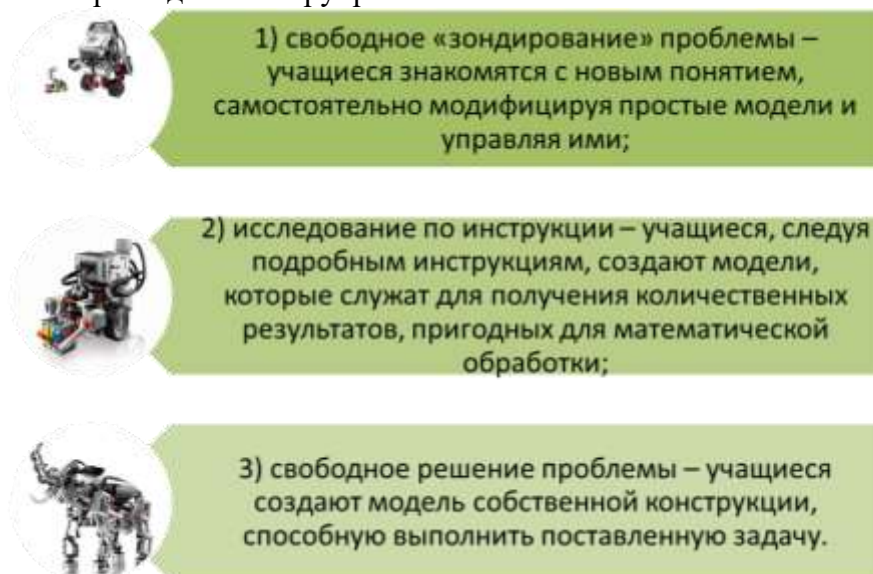


Рис. 5. Виды конструирования в системе циклической модели обучения робототехнике

«Рефлексия» предполагает осмысление того, что сделано, создано, модифицировано, поиск словесной формулировки полученного знания, способов представления результатов опыта, путей его применения в комплексе с другими идеями и решениями.

Поддержка творческой атмосферы, эмоциональной и физической радости от успешно выполненной работы реализуется на этапе «Развитие» при выполнении более сложных заданий, способствующих углублению полученного опыта, развитию креативных и исследовательских навыков.

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет». Также необходима возможность подключения к электронной информационно-образовательной среде университета.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет.

#### Задания

1. На основе анализа сайта <https://p26.навигатор.дети/> (Навигатор дополнительного образования Ставропольского края) заполните таблицу, отражающую характеристики дополнительных образовательных программ для детей, реализуемых в Северо-Кавказском федеральном университете, г. Ставрополь по технической направленности (приведите описание 5 и более программ):

| № п/п | Название дополнительной образовательной программы | Общее описание дополнительной образовательной программы | Цели дополнительной образовательной программы | Содержание дополнительной образовательной программы | Результат освоения дополнительной образовательной программы |
|-------|---------------------------------------------------|---------------------------------------------------------|-----------------------------------------------|-----------------------------------------------------|-------------------------------------------------------------|
| 1.    |                                                   |                                                         |                                               |                                                     |                                                             |
| 2.    |                                                   |                                                         |                                               |                                                     |                                                             |
| ...   |                                                   |                                                         |                                               |                                                     |                                                             |

2. Рассмотрите наиболее распространенные робототехнические наборы и заполните таблицу:

| Название робототехнического набора | Компания-производитель, страна | Состав набора | Программное обеспечение | Возможность работы в виртуальной среде при отсутствии набора (да/нет) | Возраст обучающихся |
|------------------------------------|--------------------------------|---------------|-------------------------|-----------------------------------------------------------------------|---------------------|
| 1. Arduino                         |                                |               |                         |                                                                       |                     |
| 2. Bioloid                         |                                |               |                         |                                                                       |                     |
| 3. Набор «Матрёшка Z»              |                                |               |                         |                                                                       |                     |
| 4. Fischertechnik                  |                                |               |                         |                                                                       |                     |
| 5. Робот ROBOTIS MINI.             |                                |               |                         |                                                                       |                     |
| 7. LEGO Education WeDo             |                                |               |                         |                                                                       |                     |
| 8. LEGO Mindstorms Education EV3   |                                |               |                         |                                                                       |                     |
| 9. LEGO Mindstorms Robot Inventor  |                                |               |                         |                                                                       |                     |

3. Используя литературные источники, рассмотрите наиболее известные соревнования и олимпиады по образовательной робототехнике. Результаты рассмотрения

зафиксируйте в таблице (рассмотреть не менее 5-6 примеров международных, всероссийских и региональных соревнований):

| № п/п | Название соревнования, его статус | Время и место проведения | Регламент соревнований (основные положения) | Возраст участников, состав команд | Описание особенностей соревнования (цели и задачи соревнования, используемое оборудование и др.) | Победители соревнования (указать за последние несколько лет) |
|-------|-----------------------------------|--------------------------|---------------------------------------------|-----------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| 1.    |                                   |                          |                                             |                                   |                                                                                                  |                                                              |
| 2.    |                                   |                          |                                             |                                   |                                                                                                  |                                                              |
| ...   |                                   |                          |                                             |                                   |                                                                                                  |                                                              |

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и выполнить в тетради представленные практические задания.
4. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку и содержание выполненных заданий.

#### **Контрольные вопросы**

1. Приведите цели и задачи системы дополнительного образования по информатике.
2. Что понимают под образовательной робототехникой?
3. Опишите особенности циклической модели обучения на основе образовательного оборудования LEGO: развитие – взаимосвязь – конструирование – рефлексия.
4. Опишите 1) задачи педагога при подготовке учащихся к соревнованиям по образовательной робототехнике; 2) возможные формы организации работы и методики подготовки к робототехническим соревнованиям.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 5];

дополнительная литература – [6, 9, 14, 15, 16, 17, 21, 23, 25].

### **Лабораторная работа 3**

#### **Изучение возможностей среды визуального программирования LEGO Mindstorms Education EV3**

**Цель работы:** изучение возможностей среды программирования роботов LEGO Mindstorms Education EV3 на примере программирования мобильной платформы.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

#### **Теоретическая часть**

*Содержание:*

3. Введение.
4. Набор Lego Mindstorms Education EV3.
5. Обзор среды программирования.
6. Программирование роботов. Основы алгоритмизации.
7. Палитры программирования и программные блоки.
8. Зеленая палитра «Действие». Моторы.
9. Программа 1. Перемещение по прямой линии

#### **1. Введение.**

Обучающие конструкторы LEGO Mindstorms Education и контроллер EV3 предназначены для разработки роботизированных систем. Программное обеспечение с возможностью визуального программирования LEGO Mindstorms Education EV3 позволяет составлять программы, проверять их функциональность и тестировать с помощью собранных робототехнических устройств.

В состав конструктора входят различные компоненты. **Контроллер** – это устройство управления и контроля процессами системы, в которой он установлен. Контроллер преобразует код в управляющие сигналы и выдает на внешние устройства. С внешних устройств он получает данные о рабочих процессах и условиях окружающей среды, с помощью чего способен самостоятельно контролировать некоторые действия системы.

**Модуль EV3** – это программируемый интеллектуальный контроллер, который контролирует и управляет датчиками и моторами.

**Большой мотор** – это мощный мотор, который имеет встроенный датчик вращения с разрешением 1 градус для точного контроля.

**Датчик цвета** – это цифровой датчик, который может определять цвет или яркость света, поступающие в небольшое окошко на лицевой части датчика.

**Гироскопический датчик** – это цифровой датчик, который обнаруживает вращательное движение вокруг одной оси.

**Датчик касания** – это аналоговый датчик, который может определять, когда красная кнопка нажата, а когда отпущена.

**Ультразвуковой датчик** – это цифровой датчик, который определяет расстояние до находящегося перед ним объекта.

## **2. Набор LEGO Mindstorms Education EV3.**

В набор входят:

- Три электросервомотора;
- Встроенные в моторы датчики вращения и ультразвуковой датчик;
- Датчик цвета, гироскопический датчик и два датчика касаний;
- Перезаряжаемая аккумуляторная батарея;
- Колеса;
- Соединительные кабели;
- Инструкции по сборке;
- Элементы LEGO® Technic для создания множества моделей.

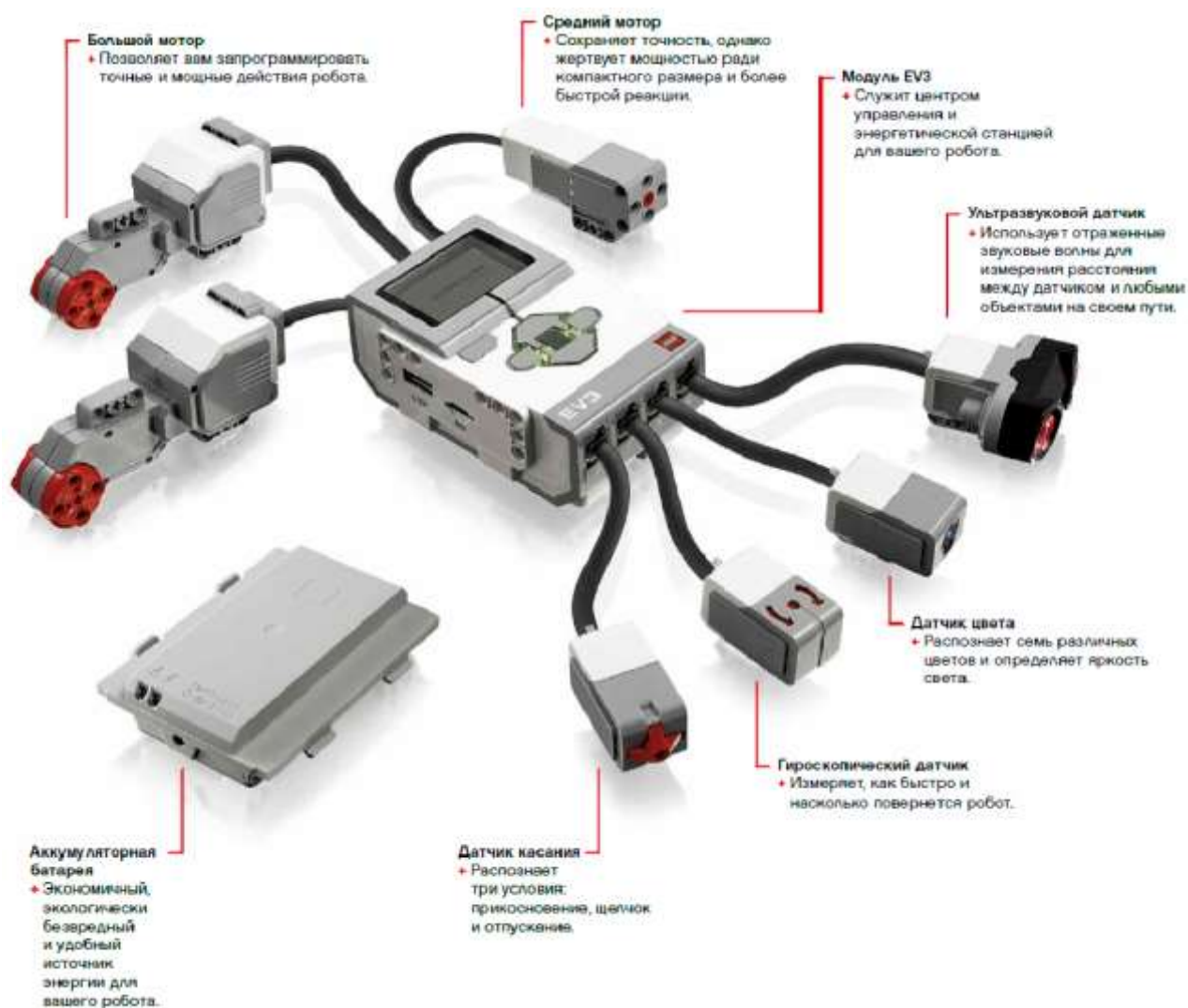


Рис. 6. Состав набора LEGO Mindstorms Education EV3

### 3. Обзор среды программирования.

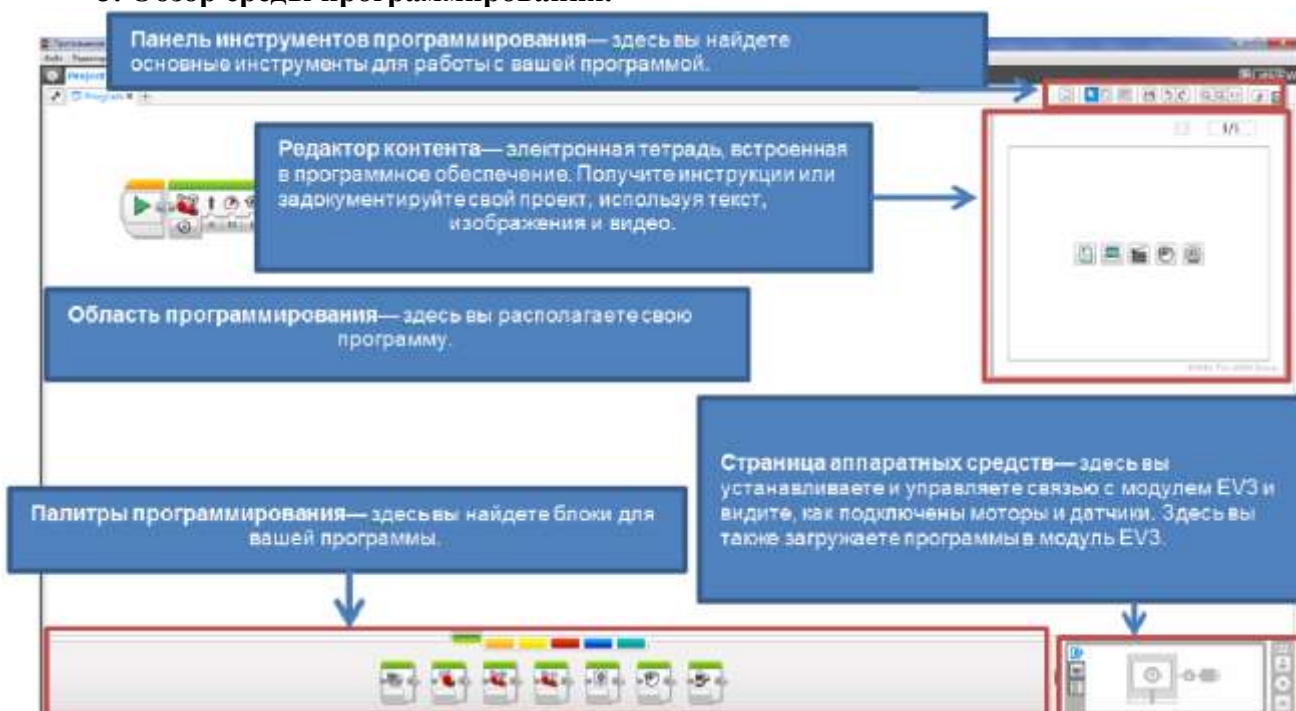


Рис. 7. Общий вид среды программирования LEGO Mindstorms Education EV3

#### 4. Программирование роботов. Основы алгоритмизации.

Решение задач на компьютере основано на понятии алгоритма. Алгоритм – это точное предписание, определяющее вычислительный процесс, ведущий от начальных данных к исходному результату. Алгоритм означает точное описание некоторого процесса, инструкцию по его выполнению. Алгоритмизация – это процесс построения алгоритма для решения задач на компьютере.

В программном обеспечении LEGO Mindstorms Education EV3 используется графический способ описания алгоритма, который получил самое широкое распространение. Для описания используются блоки, которые соединяются между собой линиями связи.

По структуре выполнения алгоритмы делятся на три вида:

- линейные;
- ветвления;
- циклические.

Линейный алгоритм (линейная структура) – это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз. Схема представляет собой последовательность блоков, которые располагаются сверху вниз или слева направо в порядке их выполнения.



Рис. 8. Пример записи линейного алгоритма на языке программирования EV3

Но на практике часто встречаются задачи, в которых необходимо при различных условиях действовать по-разному. Такие задачи можно описать с помощью алгоритмов разветвляющейся структуры. Выбор направления продвижения по схеме алгоритма осуществляется по итогам проверки заданного условия. Ветвящиеся процессы в EV3 описываются оператором Переключатель.

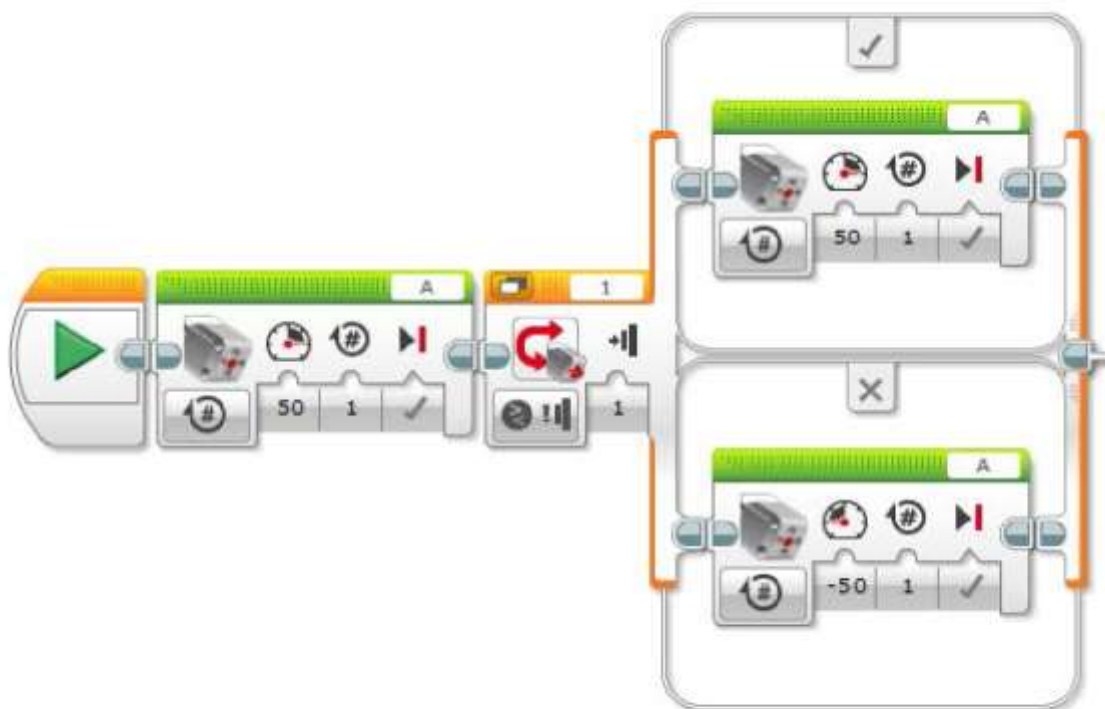


Рис. 9. Пример записи алгоритма с ветвлением

Для решения некоторых задач нужно повторение отдельных участков вычислений. В таких задачах применяются алгоритмы циклической структуры (циклические алгоритмы). Цикл – последовательность команд, которая повторяется до тех пор, пока не будет выполнено заданное условие.

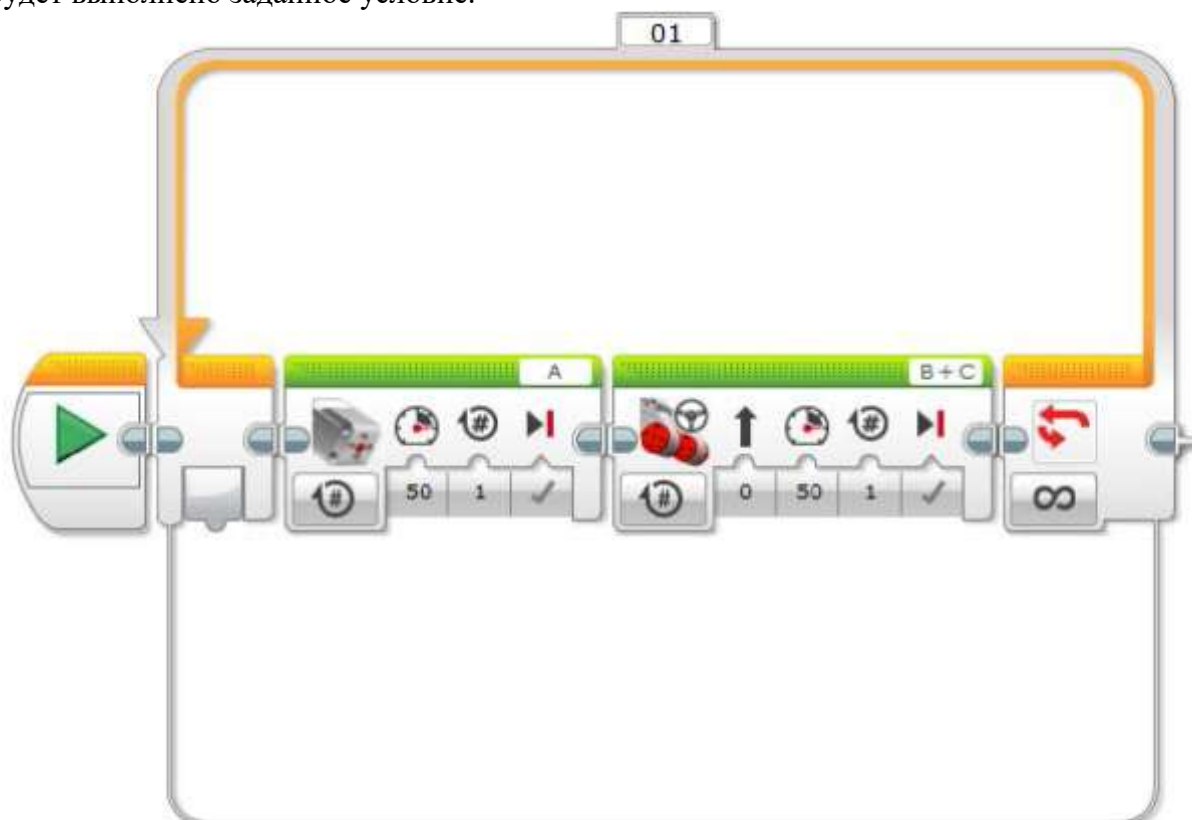


Рис. 10. Пример записи циклического алгоритма в среде EV3

Существуют циклы с известным и с неизвестным числом повторений. В цикле с неизвестным числом повторений выход из тела цикла, как правило, происходит при выполнении записанного условия.

#### **5. Палитры программирования и программные блоки.**

В нижнем разделе среды программирования находятся команды для программирования робота. Разработчики применили оригинальный прием и, сгруппировав программные блоки, присвоили каждой группе свой цвет, назвав группы палитрами.

Все программы в среде Lego Mindstorms Education EV3 состоят из блоков. У каждого блока есть один или несколько регулируемых параметров.

##### **Блоки действий (Зеленый):**

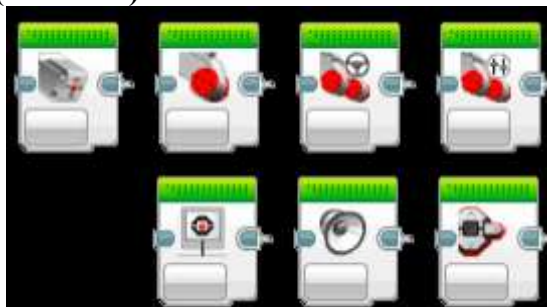


Рис. 11. Блоки действий среды EV3

Блоки действий управляют действиями в рамках программы. Они контролируют вращение моторов, а также изображения, звук и подсветку модуля EV3.

##### **Блок выполнения программ (Оранжевый):**



Рис. 12. Блоки выполнения программ среды EV3

Блок выполнения программы управляют процессом выполнения программ. Все создаваемые вами программы будут начинаться со стартового блока.

##### **Блоки датчиков (Желтый):**

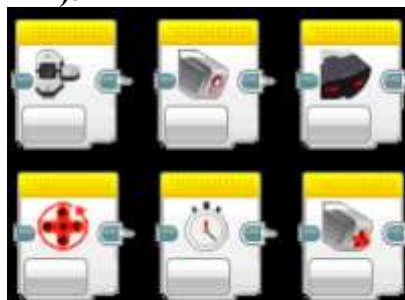


Рис. 13. Блоки датчиков среды EV3

Блоки датчиков позволяют программе считывать входящие данные с датчика цвета, ИК-датчика, датчика касания и многое другое.

##### **Блоки операции над данными (Красный):**

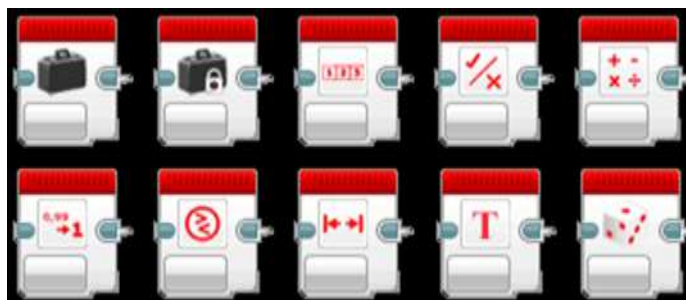


Рис. 14. Блоки операций над данными среды EV3

Блоки операций над данными позволяют вводить и считывать переменные величины, сравнивать характеристики и многое другое.

#### **6. Зеленая палитра «Действие». Моторы.**

Здесь расположены **программные блоки управления моторами:**

- Средний мотор – предназначен для управления средним мотором.
- Большой мотор – для управления большим мотором.
- Рулевое управление
- Независимое управление моторами



Рис. 15. Общий вид зеленой палитры среды EV3

**А также блоки:**

- Вывод на экран
- Вывод звука
- Индикатор состояния модуля

Рассмотрим настройку моторов на примере большого мотора:

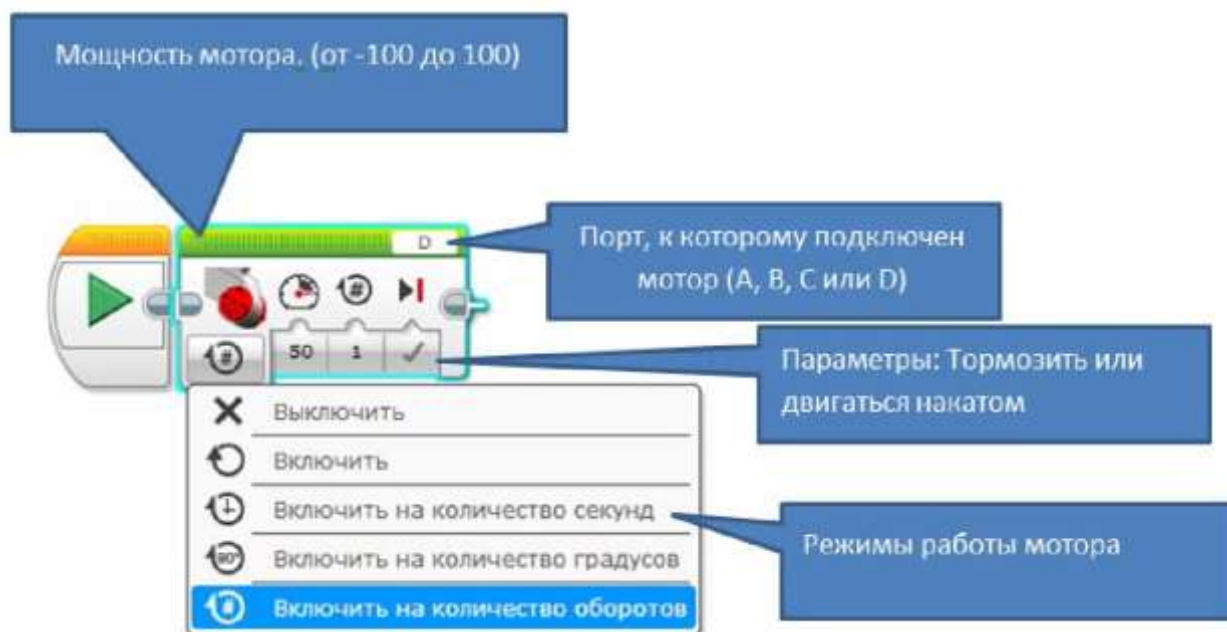


Рис. 16. Параметры настройки большого мотора в среде EV3

Режим: «Включить» включает мотор с заданным параметром «Мощность» и после этого управление передается следующему программному блоку программы. Мотор будет продолжать вращаться, пока не будет остановлен следующим блоком «Большой мотор» с режимом «Выключить» или следующий блок «Большой мотор» не будет содержать другие параметры выполнения. Режим «Включить на количество секунд» включает большой мотор с установленной мощностью на указанное количество секунд, и только по завершению времени мотор остановится, а управление в программе перейдет к следующему программному блоку. Аналогично поведет мотор себя в режимах «Включить на количество градусов» и «Включить на количество оборотов»: только после выполнения установленного вращения мотора он остановится, и управление в программе перейдет к следующему блоку.

Параметр мощность может принимать значения от -100 до 100. Положительные значения мощности задают вращение мотора по часовой стрелке, отрицательные – против часовой. При значении мощности равном 0 мотор вращаться не будет, чем «выше» значение мощности, тем быстрее вращается мотор.

Параметр мощность задается только целыми значениями, параметры: секунды, градусы, обороты могут принимать значения с десятичной дробью. Но следует помнить, что минимальный шаг вращения мотора равен одному градусу.

Отдельно следует сказать о параметре «Тормозить в конце». Данный параметр, если установлен в значение «Тормозить» заставляет мотор тормозить после выполнения команды, а если установлен в значение «Двигаться накатом», то мотор будет вращаться по инерции, пока сам не остановится.

Следующие два программных блока «Рулевое управление» и «Независимое управление моторами» реализуют управление парой больших моторов. По умолчанию левый большой мотор подключается к порту «В», а правый – к порту «С». Но вы можете в настройках блока поменять порты подключения в соответствии с требованиями вашей конструкции.



Рис. 17. Блок «Рулевое управление»

Программный блок «Независимое управление моторами» похож на программный блок «Рулевое управление». Он также управляет двумя большими моторами, только вместо параметра «Рулевое управление» появляется возможность независимого управления мощностью каждого мотора. При равном значении параметра «Мощность» для левого и правого мотора робот будет двигаться прямолинейно. Если на один мотор подать отрицательное значение мощности (например -50), а на второй – положительное значение (например, 50), то робот будет разворачиваться на месте.



Рис. 18. Блок «Независимое управление моторами»

## 7. Программа 1. Перемещение по прямой линии.

Этой программой робот будет двигаться по прямой линии вперед и назад. Работа моторов будет измеряться тремя единицами: количеством оборотов, количеством секунд и количеством градусов.

### Выполнение.

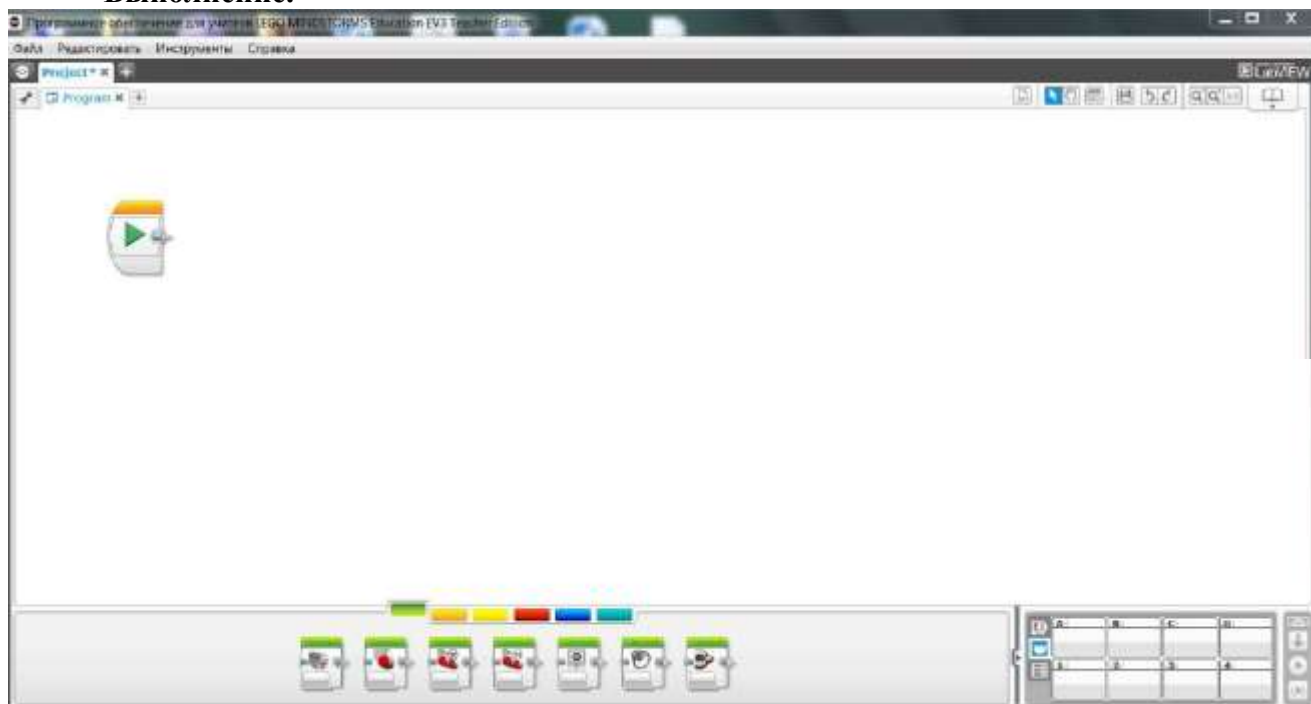


Рис. 18. Стартовое окно новой программы

Во вкладке «Действие» выбираем блок «Рулевое управление» и переносим к блоку «Старт», соединяем два блока.



Рис. 19. Добавление блока «Рулевое управление»

В конфигурациях блока «Рулевое управление» количество оборотов ставим равное 2.



Рис. 20. Редактирование блока «Рулевое управление»

Открываем вкладку «Управление операторами» и выбираем блок «Ожидание», соединяем с блоком «Рулевое управление». Блок «Ожидание» по умолчанию имеет значение 1 сек.



Рис. 21. Добавление блока «Ожидание»

Во вкладке «Действие» выбираем блок «Рулевое управление» и переносим к блоку «Ожидание».



Рис. 22. Добавление второго блока «Рулевое управление»

Редактируем последний добавленный блок «Рулевое управление». Нажимаем на блок, выбираем значок оборотов и выбираем «Включить на количество градусов».

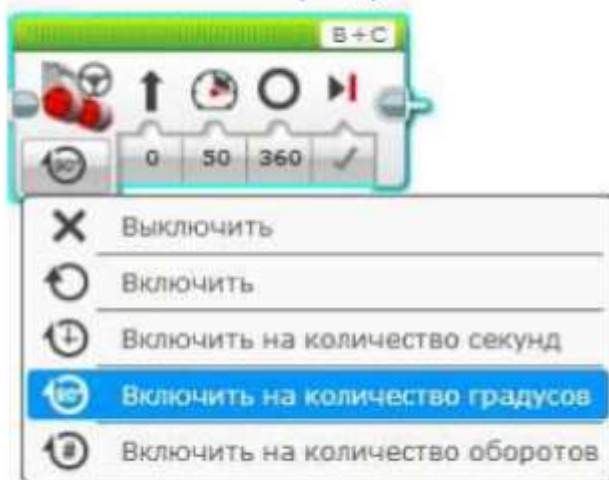


Рис. 23. Редактирование второго блока «Рулевое управление»

Далее нажмем на значок «Мощность» и выставим мощность работы моторов -50. Это значит, что робот будет двигаться назад на средней мощности.

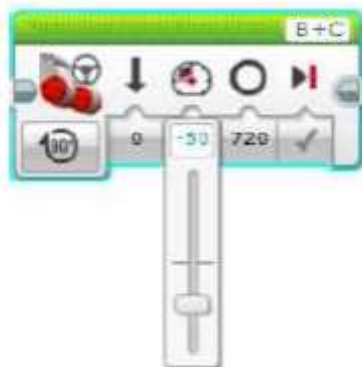


Рис. 24. Установка значения мощности

После этого нажимаем на значок «Градусы» и ставим количество градусов 720. Это означает, что моторы выполнят два оборота.



Рис. 25. Установка количества оборотов

Открываем вкладку «Управление операторами» и выбираем блок «Ожидание», соединяем с блоком «Рулевое управление».



Рис. 26. Добавление второго блока «Ожидание»

Во вкладке «Действие» выбираем блок «Рулевое управление» и переносим к блоку «Ожидание». В настройках блока нажимаем на значок оборотов и выбираем «Включить на количество секунд». Это значит, что работа моторов будет измеряться в секундах.

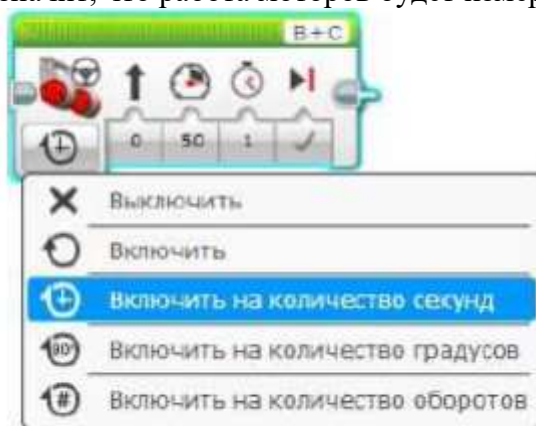


Рис. 27. Добавление третьего блока «Рулевое управление»

Готовая программа:



Рис. 28. Программа «Перемещение по прямой линии»

Далее проверяем подключение к контроллеру. В правом нижнем углу располагается окно управления подключением.

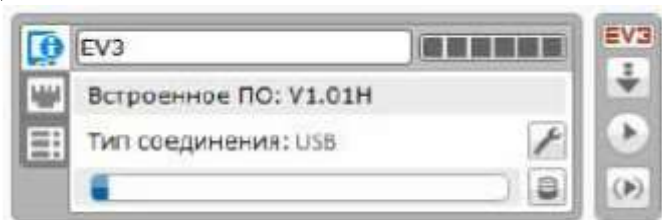


Рис. 29. Проверка подключения к контроллеру

При активном подключении значок  будет красным. Заходим в «Просмотр портов» и проверяем соответствие номеров портов с моторами. Моторы подключены к портам В и С.

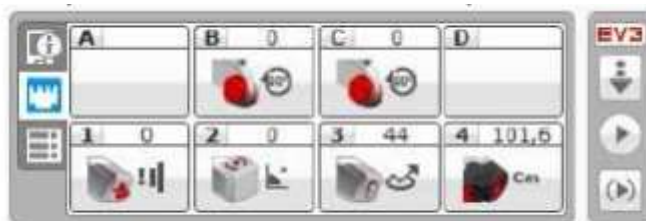


Рис. 30. Просмотр подключения портов

Если все подключено правильно, нажимаем на кнопку «Загрузить и запустить». Программа готова.

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет» и необходимым программным обеспечением, а также возможностью подключения к электронной информационно-образовательной среде университета, наборы конструкторов LEGO Mindstorms Education EV3 Core Set 45544 (базовый набор), наборы LEGO Mindstorms Education EV3 Expansion Set 45560 (ресурсный набор), зарядные устройства.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, среда LEGO Mindstorms Education EV3.

#### Задания

1. Согласно инструкции, приведенной в среде LEGO Mindstorms Education EV3, собрать одну из базовых моделей робота – мобильную платформу.
2. В среде LEGO Mindstorms Education EV3 составить программу, которая задает следующее движение мобильной платформе: вперед 3 сек. – назад 3 сек.
3. В среде LEGO Mindstorms Education EV3 составить программу, которая задает следующее движение мобильной платформе: вперед 4 оборота – назад 4 оборота – вперед 2 оборота.
4. В среде LEGO Mindstorms Education EV3 составить программу, которая задает следующее движение мобильной платформе: вперед 360° – назад 360°.

#### Указания по порядку выполнения работы

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.

3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.

4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить робота и провести тестирование работы программы.

5. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Что такое робот?
2. Какие функции выполняет контроллер набора Lego Mindstorms Education EV3?
3. Для чего предназначен модуль EV3?
4. Перечислите элементы информационно-управляющей системы среды Lego Mindstorms Education EV3.
5. Перечислите элементы исполнительной системы среды Lego Mindstorms Education EV3.
6. Перечислите элементы сенсорной системы среды Lego Mindstorms Education EV3.
7. Назовите элементы окна программирования среды Lego Mindstorms Education EV3.
8. В какие порты модуля EV3 подключаются моторы?
9. К какому типу алгоритма по структуре выполнения относится алгоритм Программы 1?

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 5];

дополнительная литература – [5, 7, 8, 10, 12, 19, 22, 26, 27, 28].

### **Лабораторная работа 4**

#### **Блоки действий (зеленая палитра) LEGO Mindstorms Education EV3**

**Цель работы:** исследование команд зеленой палитры «Действие» среды LEGO Mindstorms Education EV3 и их применение в программировании роботов.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

*Содержание:*

1. Введение.
2. Программа «Работа с объектом».
3. Экран, звук, индикатор состояния модуля.

#### 1. Введение.

Зеленая палитра «Действие» содержит программные блоки управления моторами:

- средний мотор – предназначен для управления средним мотором,
- большой мотор – для управления большим мотором,
- рулевое управление,
- независимое управление моторами,
- а также блоки:
- вывод на экран,
- вывод звука,
- индикатор состояния модуля.



Рис. 31. Блоки палитры «Действие»

#### 2. Программа «Работа с объектом»

В данной программе роботу будет задана задача в виде сближения с предметом и удаления от него. В основе действия лежит работа ультразвукового датчика.

*Выполнение.* Создаем новый проект. Во вкладке «Действие» выбираем блок



«Рулевое управление» и добавляем в программу. Нажимаем на значок оборотов и выбираем «Включить». Это означает, что движение робота неограниченно и будет продолжаться до тех пор, пока не будет добавлено новое ограничение.

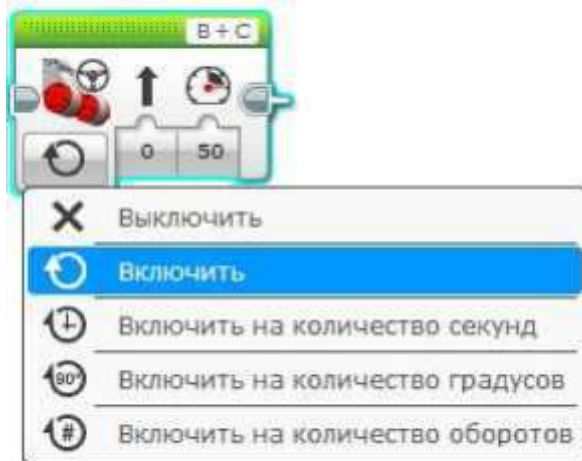


Рис. 32. Редактирование блока «Рулевое управление»

Открываем вкладку «Управление операторами» и выбираем блок «Ожидание»



, добавляем в программу. Нажимаем на блок и начинаем его редактирование. Выбираем значок «Время». Во всплывающем списке выбираем «Ультразвуковой датчик» – «Изменить» – «Расстояние в сантиметрах».

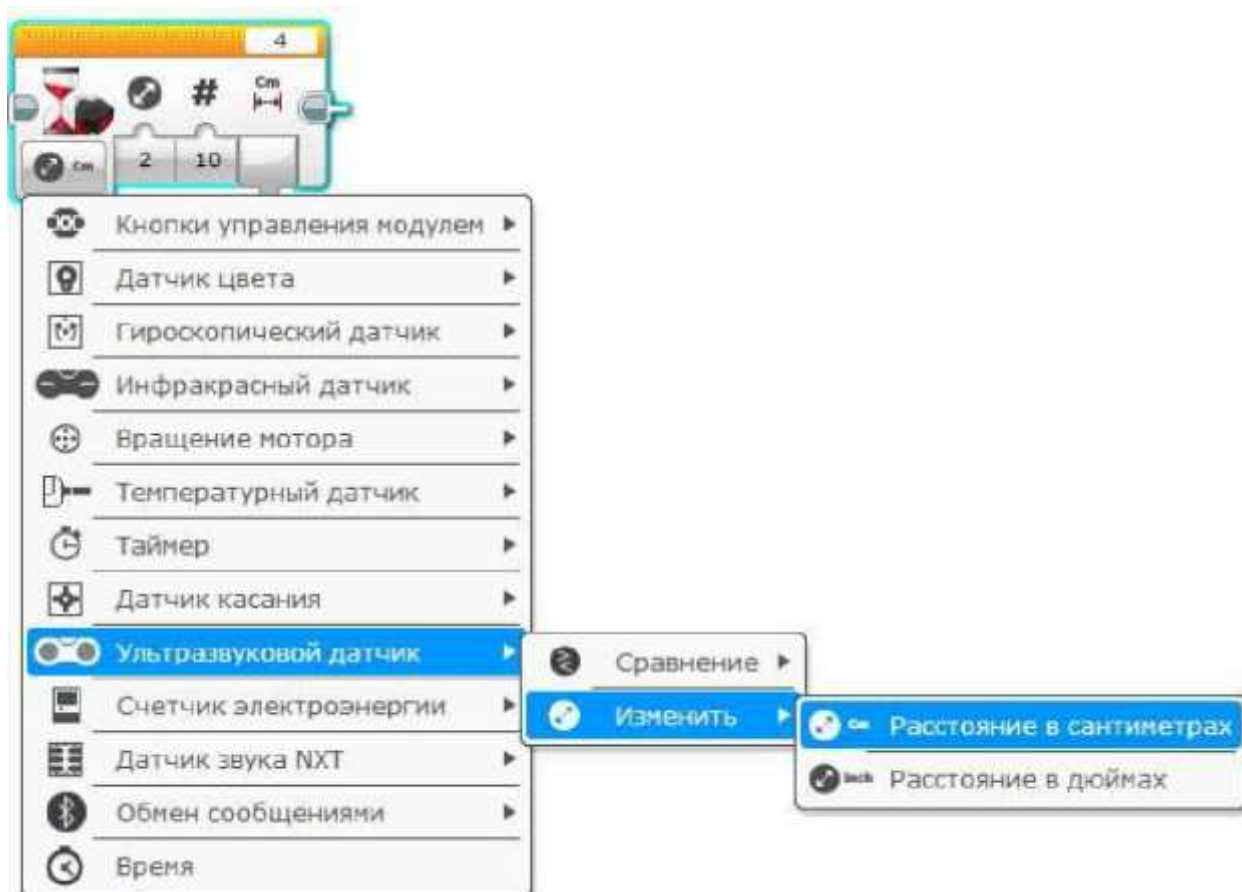


Рис. 33. Редактирование блока «Ожидание»

Далее нажимаем на цифру 2 и выбираем «1», что означает «Уменьшить». Это означает, что направление движения робота будет стремиться к уменьшению расстояния между ним и предметом.

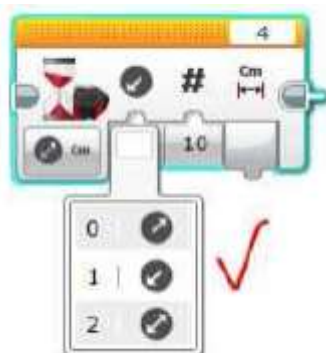


Рис. 34. Установка направления движения робота

Далее в строке «Сумма» ставим цифру 11. Это значит, что робот уменьшит свое расстояние до предмета на 11 см.

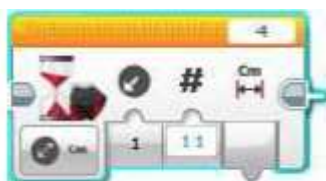


Рис. 35. Установка значения строки «Сумма»



Во вкладке «Действие» выбираем блок «Рулевое управление» и добавляем в программу к последнему блоку.

Нажимаем на значок оборотов и выбираем «Выключить». Это значит, что робот остановится после совершения предыдущего действия.

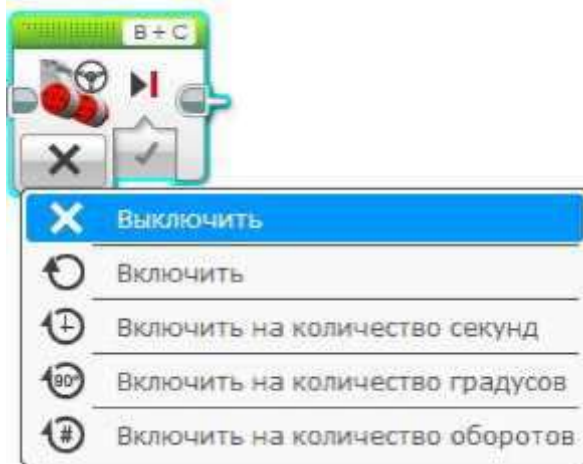


Рис. 36. Установка количества оборотов

Открываем вкладку «Управление операторами» и выбираем блок «Ожидание»



, добавляем в программу к последнему блоку. Во вкладке «Действие» выбираем



блок «Рулевое управление» и добавляем в программу к последнему блоку. Редактируем блок. Нажимаем на значок оборотов и выбираем «Включить».

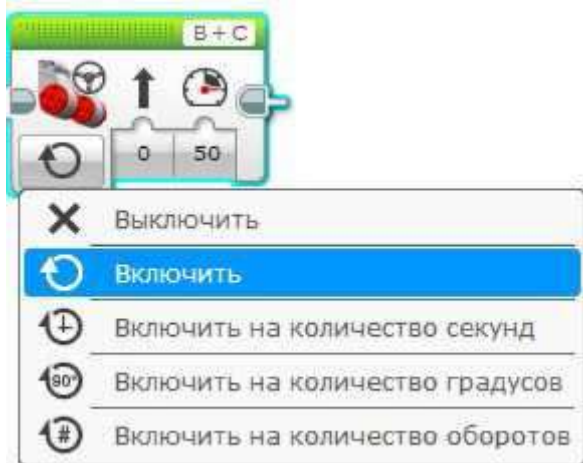


Рис. 37. Редактирование блока «Рулевое управление»

Далее устанавливаем мощность «-50».

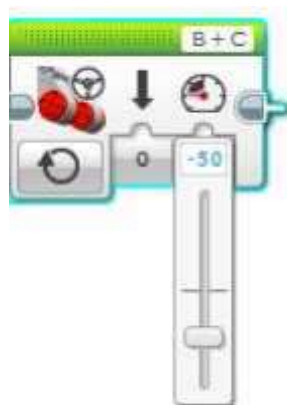


Рис. 38. Установка мощности моторов

Открываем вкладку «Управление операторами» и выбираем блок «Ожидание»



, добавляем в программу. Нажимаем на блок и начинаем его редактирование. Выбираем значок «Время». Во всплывающем списке выбираем «Ультразвуковой датчик» – «Изменить» – «Расстояние в сантиметрах».

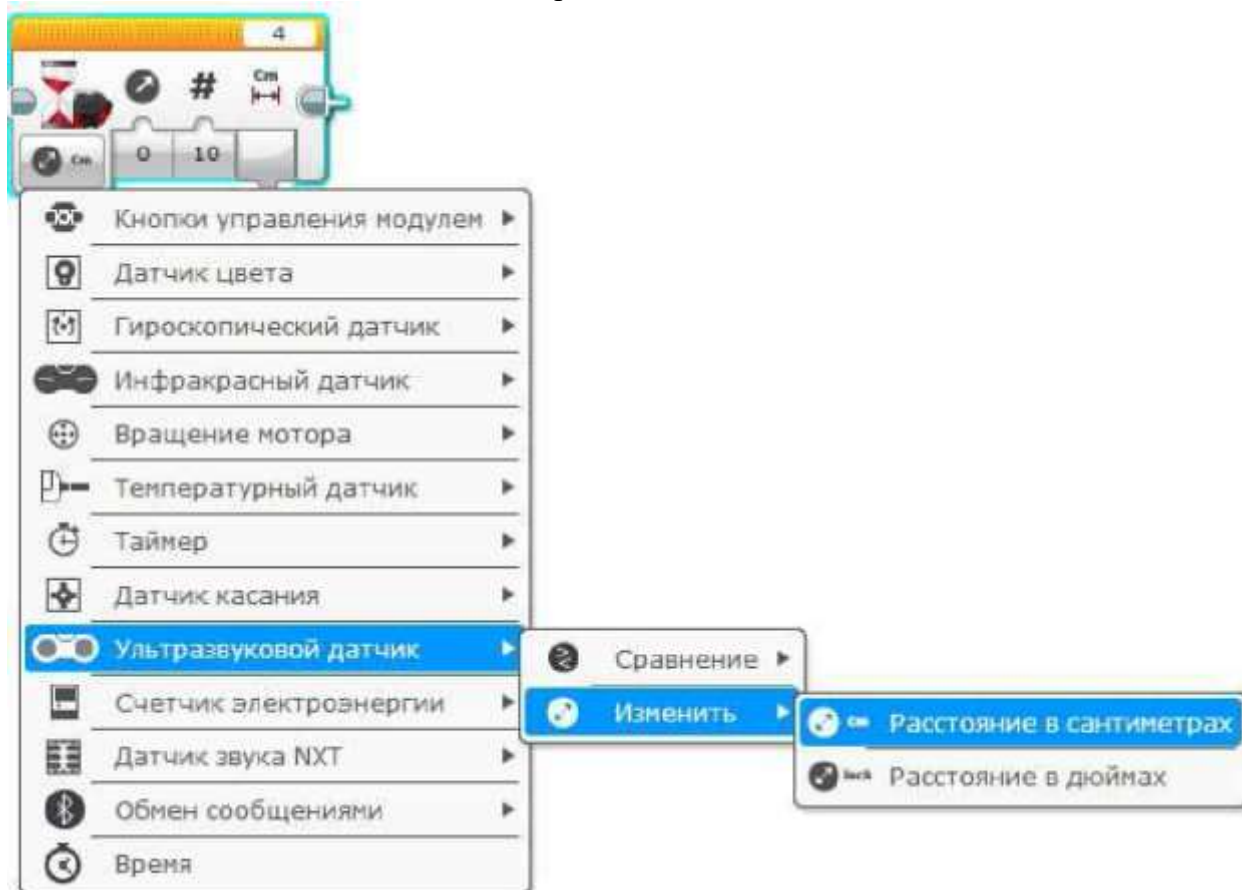


Рис. 39. Редактирование блока «Ожидание»

Далее нажимаем на цифру 2 и выбираем «0», что означает «Увеличить». Это означает, что направление движения робота будет стремиться к увеличению расстояния между ним и предметом.

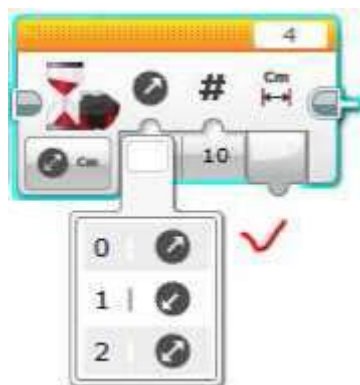


Рис. 40. Установка направления движения

Далее в строке «Сумма» ставим цифру 6. Это значит, что робот увеличит свое расстояние от предмета на 6 см.



Рис. 41. Заполнение строки «Сумма»

Во вкладке «Действие» выбираем блок «Рулевое управление» и добавляем в программу к последнему блоку. Нажимаем на значок оборотов и выбираем «Выключить». Это значит, что робот остановится после совершения предыдущего действия.

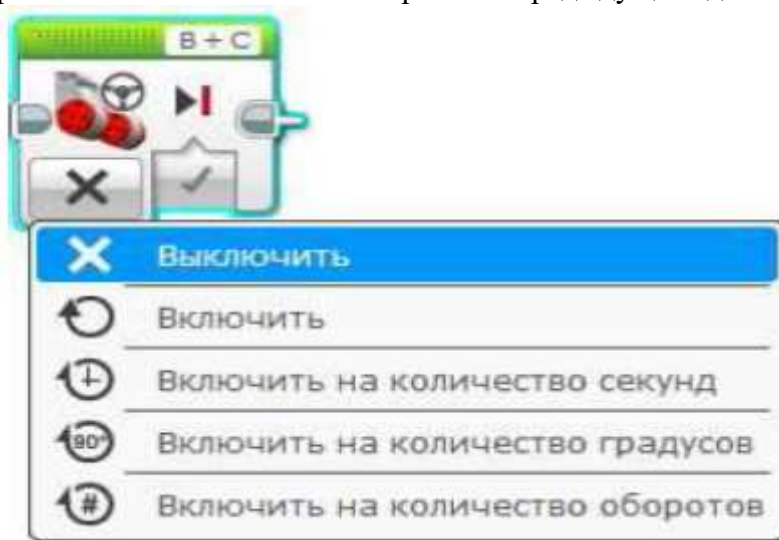


Рис. 42. Остановка движения

После этого программа завершена.



Рис. 43. Общий вид программы

Запускаем программу.

### 3. Экран, звук, индикатор состояния модуля

Для работы с дисплеем в EV3 предусмотрен отдельный блок – «Экран».

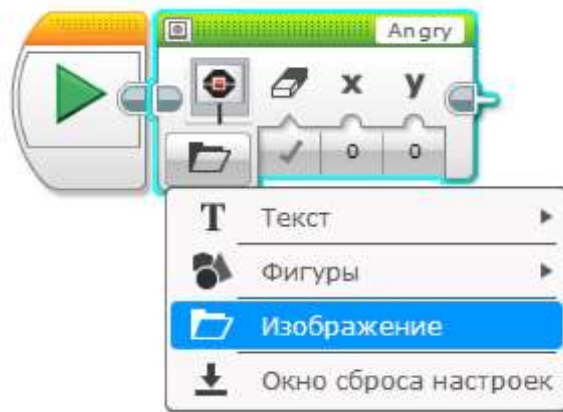


Рис. 44. Блок для работы с дисплеем

Этот блок имеет несколько режимов:

- вывод текста;
- вывод фигур
- вывод готового изображения;
- сброс настроек и вывод стандартного изображения;
- вывод пользовательского изображения.

Вы можете создавать более сложные изображения, поместив в программе подряд несколько блоков «Экран». Изображение может быть динамическим, т.е. изменяться во время выполнения программы, если подключить шины к концентратору данных и передавать по ним новые значения координат. Запомните, что при выводе готового изображения в параметре «расположение» указываются координаты его левого нижнего угла.

*Например,* напомним программу вывода на дисплей прямой линии, которая разделит его вертикально на две равные части. Для этого необходимо задать в параметрах команды координаты начальной и конечной точек прямой.

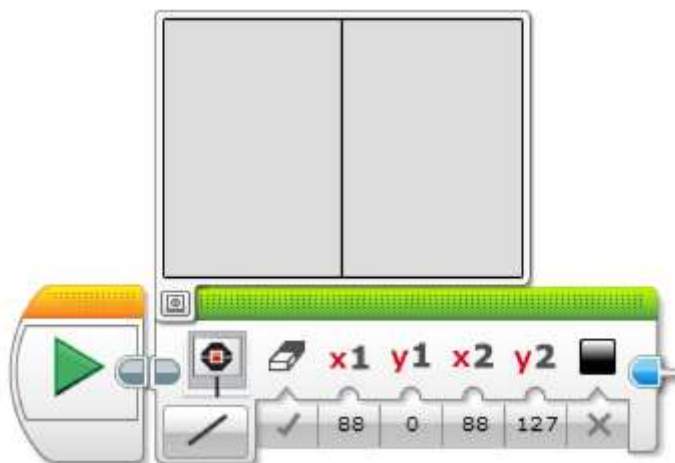


Рис. 45. Вывод прямой линии

Выполняя данную программу, состоящую всего из одной команды, микроконтроллер очистит экран дисплея, а затем выведет на него изображение вертикальной прямой линии.

Существует еще один способ создания собственного изображения на экране дисплея – для этого нужно включить редактор изображений. В этом режиме можно добавлять на экран готовые изображения. Однако среда EV3 предоставляет возможность самостоятельно создать изображение при помощи редактора изображений (Редактор изображений). Открыть редактор можно в меню «Инструменты» среды EV3.

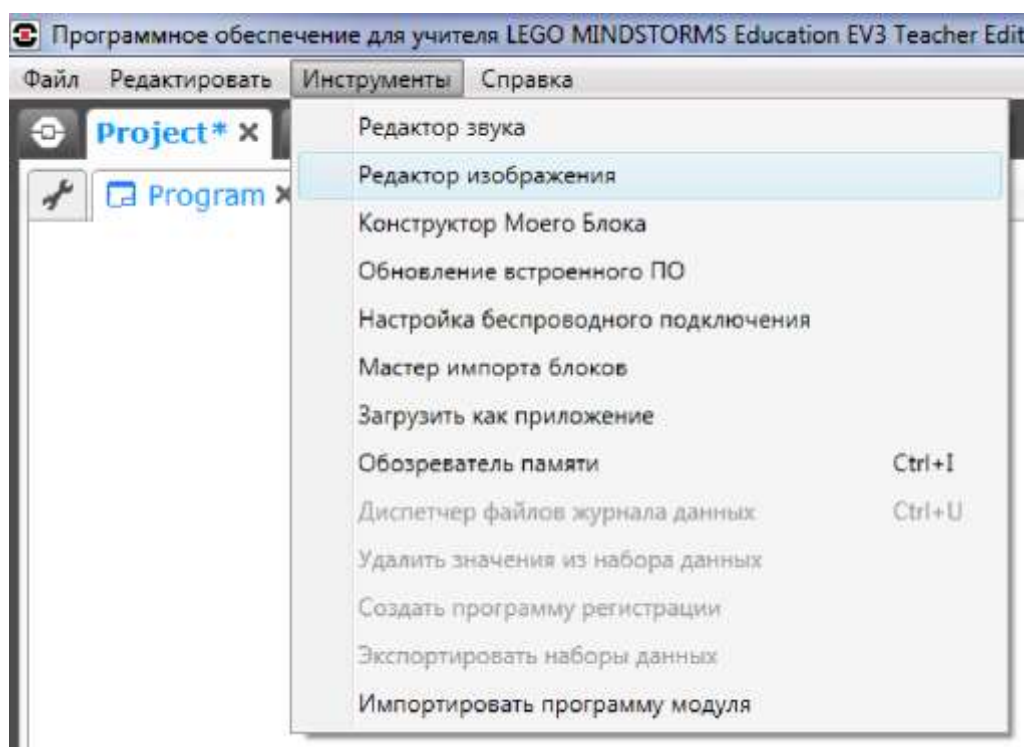


Рис. 46. Редактор изображений

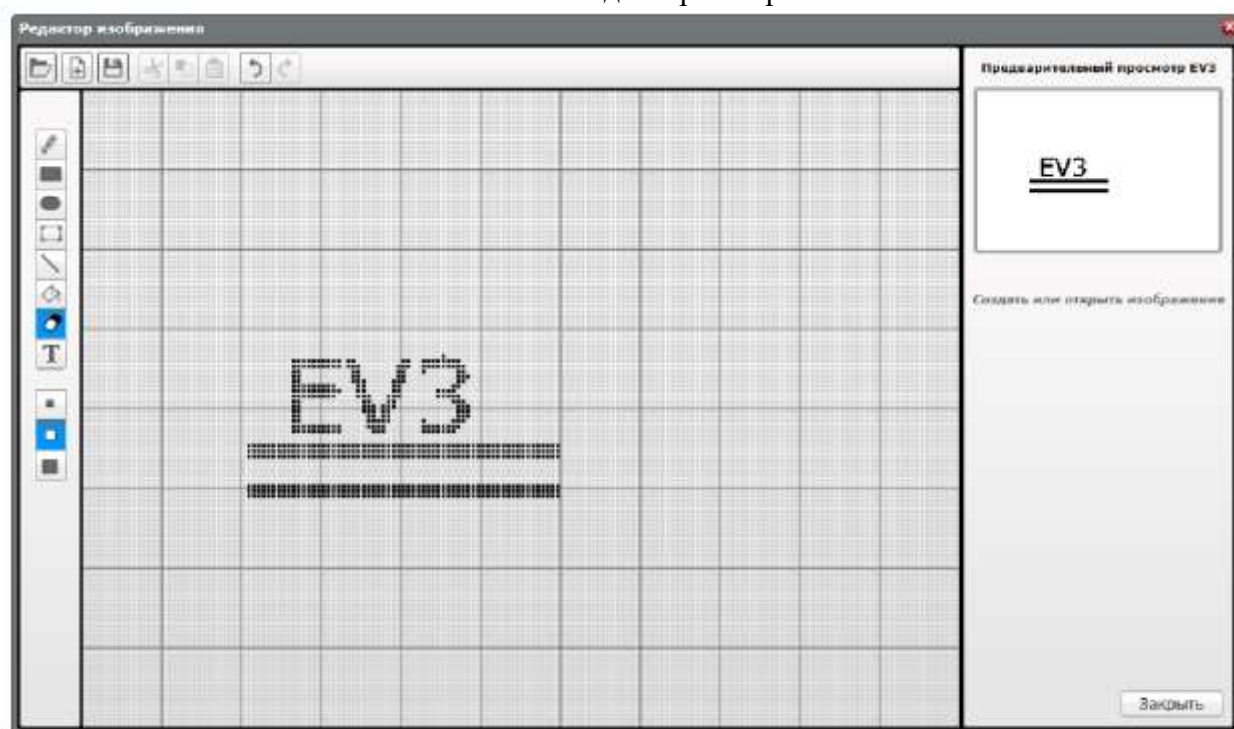


Рис. 47. Работа с редактором изображений

Поскольку дисплей блока EV3 черно-белый, редактор не поддерживает цветные палитры.

Помимо ручной отрисовки, редактор поддерживает импорт готовых изображений. Для этой цели служит кнопка «Открыть», кнопка «Сохранить» позволяет сохранить готовое изображение и добавить его к списку изображений проекта. Редактор поддерживает форматы gif, jpg, jpeg, bmp, png.

Создадим собственное изображение в «Редакторе изображений» и заставим робота показывать его на дисплее. Пусть этим изображением будет бьющееся сердце, поскольку микроконтроллер, по сути, и есть сердце робота.

Нам необходимо два изображения сердца – одно большое, другое – маленькое. Когда Вы запустите в программе их циклическую смену, будет казаться, что у робота действительно бьется сердце.

**Задание:** Откройте «Редактор изображений». Нарисуйте в рабочей области сердце так, как вам больше нравится. Чтобы сделать его заметнее, можно воспользоваться заливкой. У Вас может получиться что-то похожее на рисунок.

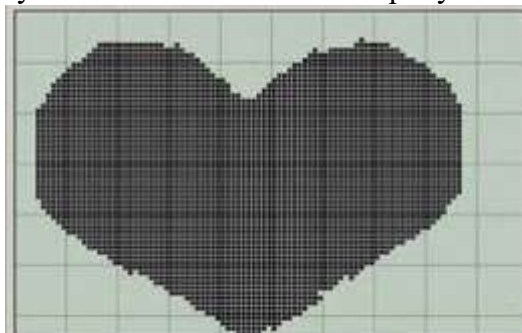


Рис. 48. Вывод рисунка

Сохраните полученное изображение с именем Heart1, нажав на кнопку сохранить.

Далее необходимо нарисовать еще одно сердце, но меньшего размера. Постарайтесь при этом оставить без изменений его форму. Сохраните второе изображение с именем Heart2 и закройте редактор. Теперь составим программу. Для этого поместите в рабочую область подряд два блока «Экран», заключив их в рамку команды «Цикл». Цикл необходим для того, чтобы изображения сменяли друг друга все время, пока работает программа. У Вас должна получиться структура, показанная на рисунке.

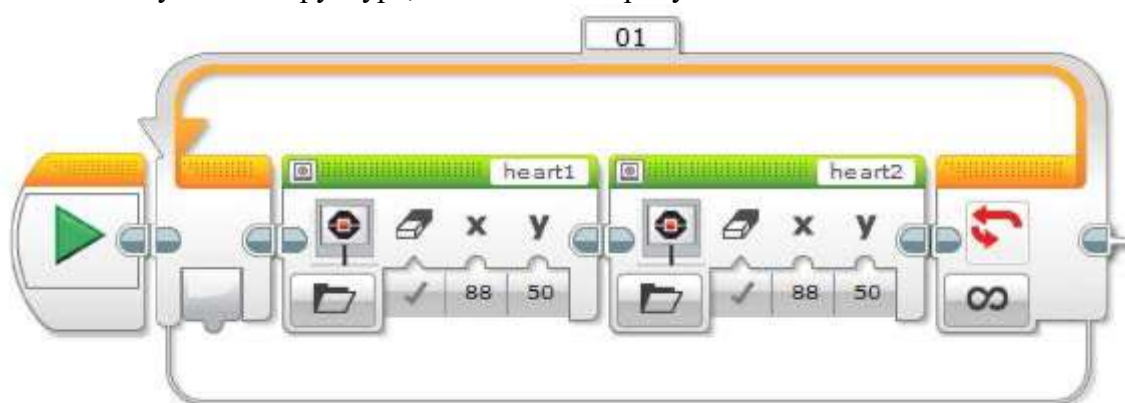


Рис. 49. Использование цикла

Но при запуске данная программа не работает. На экране Вы видите статическое изображение сердца, которое не изменяется. В чем же дело? Все дело в том, что программа работает правильно, но Вы не видите этого, потому что смена изображений происходит слишком быстро. Добавим в программный код два блока ожидания «Ожидание», установив в параметрах ожидание истечения времени в 1 секунду. Окончательный вариант программы представлен на рисунке.

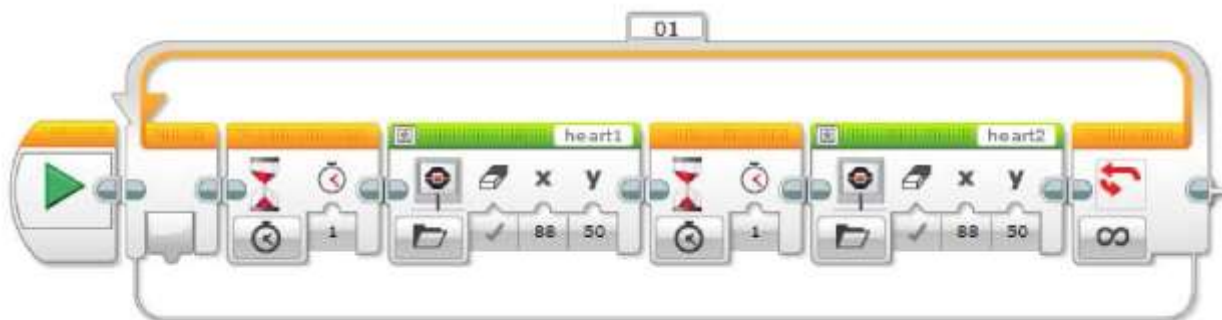


Рис. 50. Окончательный вариант программы

### **Звук. Работа с динамиком.**

Еще одним важным средством коммуникации робота является воспроизведение звуковых сигналов. Робот может также воспринимать звуки из окружающей среды при помощи звукового датчика. Для работы со звуком в языке EV3 предназначен блок «Звук».



Рис. 51. Добавление блока «Звук»

Звуки можно выбирать из стандартной библиотеки звуков. Если в блоке «Звук» установлен флажок «Ожидать завершения», то звуковой файл будет воспроизведен полностью, и только потом осуществлён переход к следующей команде.



Рис. 52. Редактирование блока «Звук»

Так же можно создать собственную звуковую дорожку. Этот способ аналогичен подобному для блока «Экран»

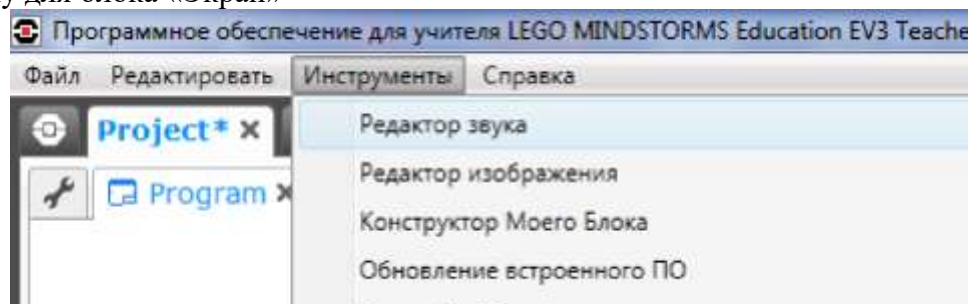


Рис. 53. Редактор звука

Редактор звука позволяет открыть для редактирования звуковой файл формата rso, rsf, mp3 или wav. Нажав на кнопку «Открыть», Вы можете выбрать файл произвольной длительности. В отличие от микрофонной записи, длина импортированного из операционной системы файла не ограничена. Но, как и в случае записи, сохранить для загрузки в команду «Звук» можно только 10 секунд из любого места файла.

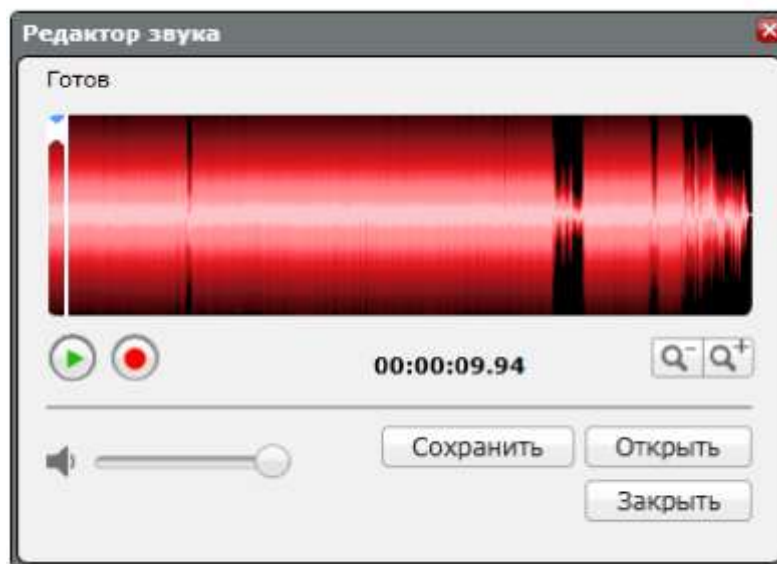


Рис. 54. Работа с редактором звука

**Блок «Индикатор состояния модуля».**

Вокруг кнопок управления модулем EV3 смонтирована цветовая индикация, которая может светиться одним из трех цветов: зеленым, оранжевым или красным. За включение – выключение цветовой индикации отвечает соответствующий режим. Параметр «Цвет» задает цветовое оформление индикации. Параметр «Импульсный» отвечает за включение – отключение режима мерцания цветовой индикации. Как можно использовать цветовую индикацию? Например, можно во время различных режимов работы робота использовать различные цветовые сигналы. Это поможет понять: так ли выполняется программа, как мы запланировали.

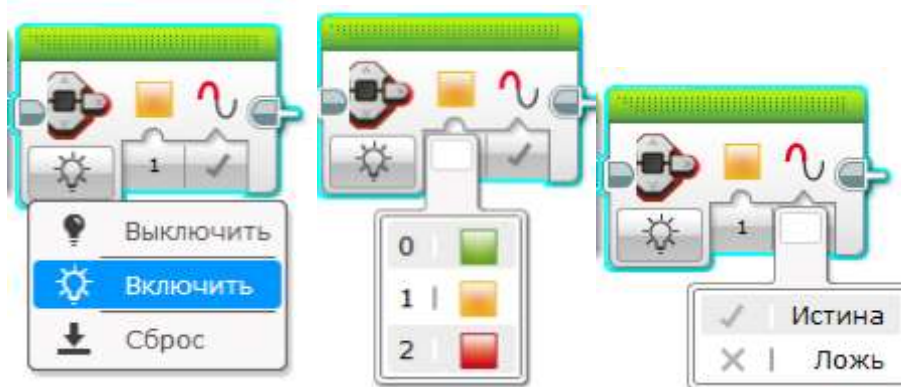


Рис. 55. Блок «Индикатор состояния модуля»

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет» и необходимым программным обеспечением, а также возможностью подключения к электронной информационно-образовательной среде университета, наборы конструкторов LEGO

Mindstorms Education EV3 Core Set 45544 (базовый набор), наборы LEGO Mindstorms Education EV3 Expansion Set 45560 (ресурсный набор), зарядные устройства.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, среда LEGO Mindstorms Education EV3.

#### Задания

1. Составить программу (рисунок представлен ниже), сделать выводы о ее работе.



Рис. 56. Программа к заданию 1

2. Создать программу, в соответствии с которой робот уменьшает расстояние от себя до предмета на 10 см, затем останавливается.
3. Создать программу для робота: уменьшить расстояние до предмета на 12 см – вернуться назад на 5 см.
4. Разработать программу, в соответствии с которой робот увеличивает расстояние от себя до предмета на 10 см, затем останавливается.
5. Создайте программу, которая делит экран модуля по горизонтали пополам.

#### Указания по порядку выполнения работы

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить робота и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

#### Содержание отчета

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### Контрольные вопросы

1. Назовите виды блоков (палитры) среды Lego Mindstorms Education EV3.
2. Для каких задач используется блок «Индикатор состояния модуля»?
3. Как вывести готовое изображение на экран модуля? Как создать свое изображение?
4. Как задавать звуки в программах в среде Lego Mindstorms Education EV3?

5. Как настраивать цветовую индикацию в программах?
6. Соотнесите блоки (палитры) среды Lego Mindstorms Education EV3 и основные структурные системы роботов (информационно-управляющая система, исполнительная система, сенсорная система).

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

**Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 5];

дополнительная литература – [5, 7, 8, 10, 12, 19, 22, 26, 27, 28].

### **Лабораторная работа 5**

#### **Работа с датчиками в среде LEGO Mindstorms Education EV3**

**Цель работы:** рассмотрение видов датчиков, входящих в состав комплекта LEGO Mindstorms Education EV3, изучение их назначения, а также разработка программ для робота с применением датчиков.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

#### **Теоретическая часть**

*Содержание:*

1. Введение.
2. Датчик касания.
3. Датчик цвета.
4. Гироскопический датчик.
5. Ультразвуковой датчик.
6. Инфракрасный датчик и маяк.

7. Датчик Вращение мотора.
8. Кнопки управления модулем.

### 1. Введение.

Датчик или сенсор (англ. Sensor) – это первичный преобразователь, элемент измерительного, сигнального, регулирующего или управляющего устройства системы, который преобразует контролируемую величину в сигнал, удобный для измерения, передачи, преобразования хранения и регистрации информации о состоянии объекта измерений. Датчики широко используются при построении систем автоматизированного управления.

Датчики в среде программирования EV3 представлены жёлтой палитрой.



Рис. 57. Датчики

### 2. Датчик касания.

Аналоговый датчик касания позволяет определить, нажата, отпущена или нажата и отпущена кнопка датчика. Датчик работает в трёх режимах: измерение, сравнение и ожидание.

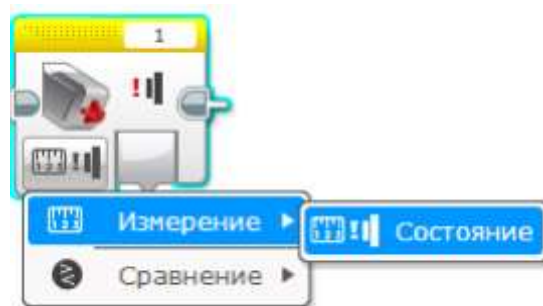


Рис. 58. Датчик касания

**Пример 1:** Создадим программу, при нажатии на датчик касания наш робот начинает двигаться, при отпускании кнопки робот останавливается.

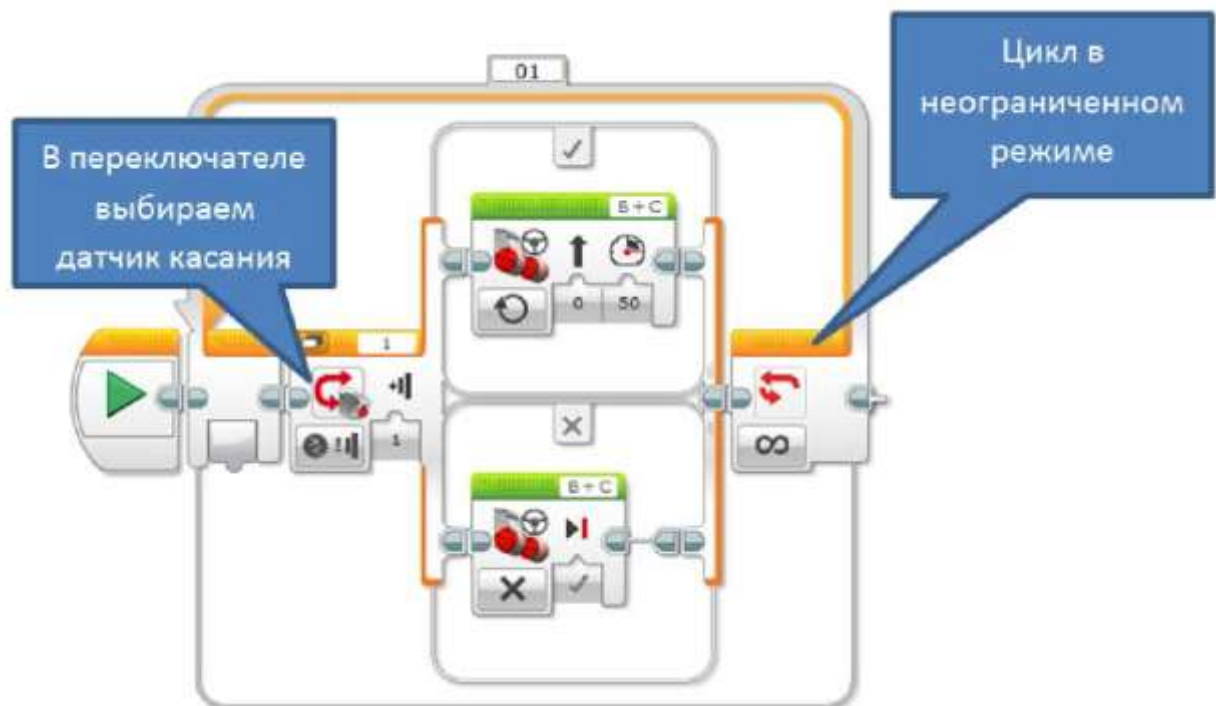


Рис. 59. Программа с использованием датчика касания

### 3. Датчик цвета.

Цифровой датчик цвета EV3 способен:

Различить семь различных цветов, обозначенных цифрами (1 – чёрный, 2 – синий, 3 – зелёный, 4 – желтый, 5 – красный, 6 – белый, 7 – коричневый). А также:

- определить отсутствие цвета перед датчиком (код – 0);
- определить яркость отраженного от поверхности света по 100-бальной шкале;
- определить яркость внешнего освещения по 100-бальной шкале.

Этот датчик помогает в решении задачи движения по черной линии, сортировки и т.д.



Рис. 60. Датчик цвета

**Пример 2:** создадим программу для движения по черной линии с одним датчиком цвета.

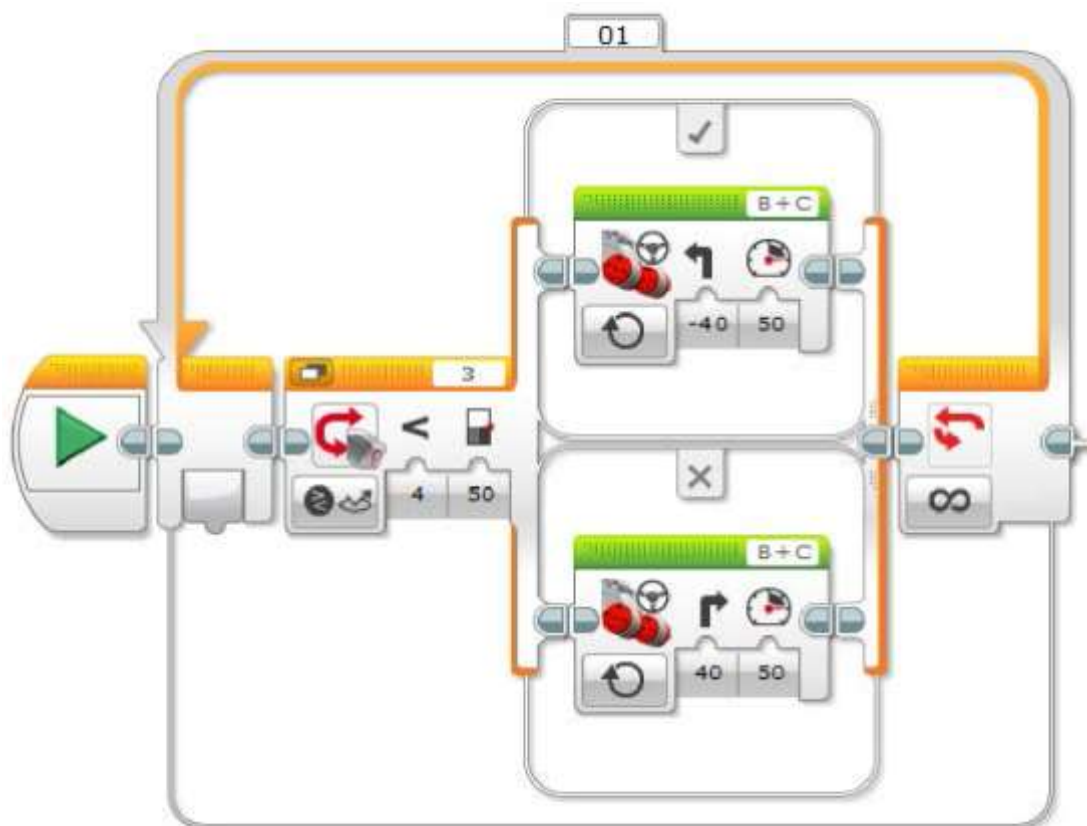


Рис. 61. Программа с использованием датчика цвета

### 4. Гироскопический датчик.

Цифровой гироскопический датчик предназначен для измерения угла и направления вращения робота, а также скорости его вращения. Точность измерения составляет +/-30, максимальная скорость проведения измерений 4400/сек., частота опроса датчика 1кГц.

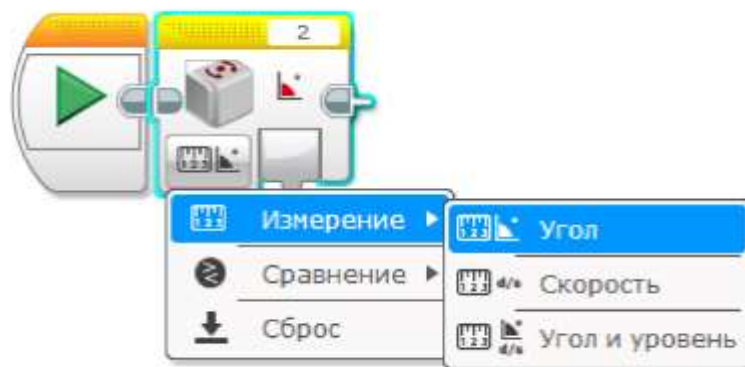


Рис. 62. Настройка гироскопического датчика

Гироскопический датчик определяет движение вокруг одной оси вращения. Это направление указано стрелкой на корпусе датчика. Угол и направление вращения может быть положительным или отрицательным. Вращение по часовой стрелке считается положительным, против часовой стрелки – отрицательным. Единица измерения скорости – градусы в секунду.

Будучи включенным, при длительном нахождении робота в неподвижном состоянии, определяемое значение угла не остаётся постоянным, а ошибочно меняется или дрейфует. Поэтому, чем больше времени проходит от начала первого обращения к гироскопическому датчику до чтения показаний, тем менее точным становятся результаты за счёт систематического накопления ошибки. Поэтому перед началом каждого измерения всегда необходимо производить обнуление угла при помощи режима Сброс.

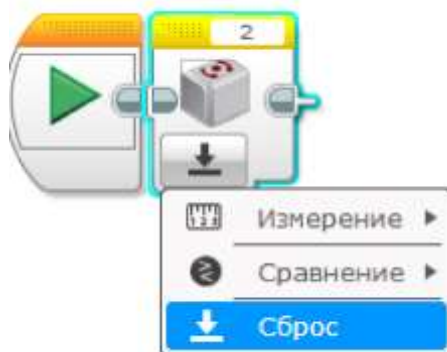


Рис. 63. Активация режима «Сброс» гироскопического датчика

**Пример 3:** создадим программу, что бы робот двигался вперед, поворачивался на 45 градусов и двигался дальше.



Рис. 64. Программа с использованием гироскопического датчика

## 5. Ультразвуковой датчик.

Цифровой ультразвуковой датчик генерирует ультразвуковые волны и считывает их отражение для обнаружения и измерения расстояния до объектов. Он также может излучать или принимать волны от других ультразвуковых датчиков, например, для инициирования начала работы программы.

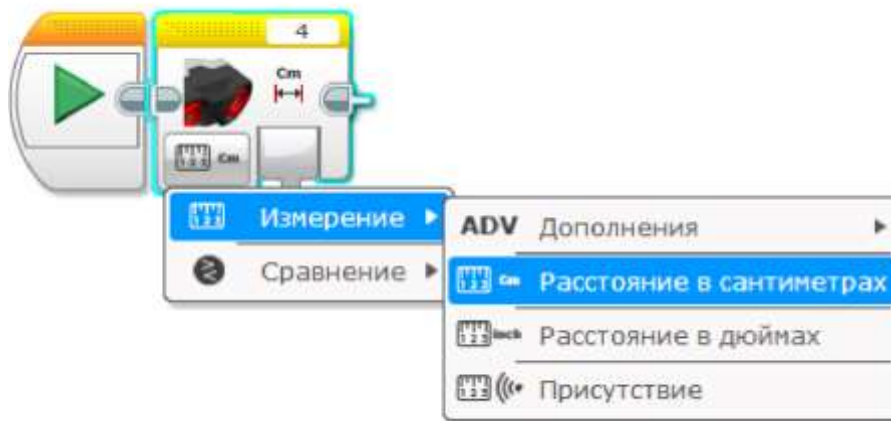


Рис. 65. Ультразвуковой датчик

Минимальное определяемое расстояние – 3 см, максимальное – 250 см. Точность определения расстояния +/- 1 см. Если смотреть на датчик спереди, то левая сторона содержит передатчик сигнала, правая – приемник сигнала.

Угол излучения пучка лучей составляет примерно 200. Если мы поместим какой-либо объект на расстоянии 1 м перед датчиком, то ширина пучка лучей у объекта составит 65 см.

Ультразвуковой датчик работает в режиме измерения, сравнения и ожидания.

**Пример 4:** создадим программу, чтобы робот мог объезжать препятствия.

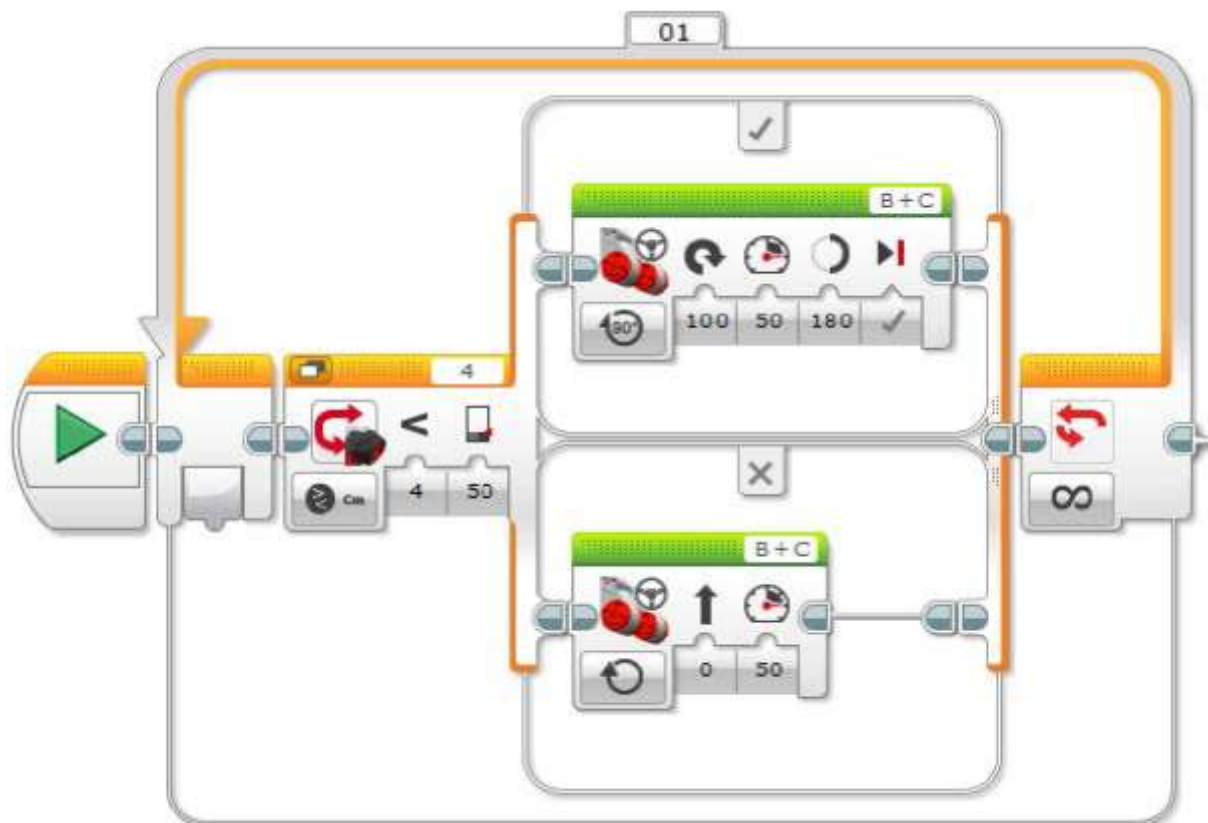


Рис. 66. Программа с использованием ультразвукового датчика

На рисунке, если расстояние до препятствия меньше 50 см, то робот поворачивает. Если расстояние от 50 и больше, то робот едет прямо.

## 6. Инфракрасный датчик и маяк.

Цифровой инфракрасный датчик (ИК-датчик) определяет расстояние до любого объекта, расстояние и расположение инфракрасного маяка, считывает и распознает сигналы

от инфракрасного маяка. Маяк (ИК-маяк) позволяет осуществлять дистанционное управление роботом. Радиус действия маяка – до 2 метров.

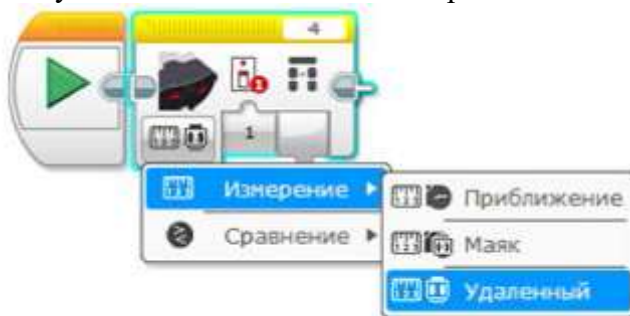


Рис. 67. Инфракрасный датчик

Режим определения относительного расстояния до объекта (Приближение).

В этом режиме датчик посылает инфракрасные сигналы и оценивает мощность отражённого сигнала. Если мы измеряем расстояние до объекта в том случае, когда он находится очень далеко, результат будет иметь значение 100, если очень близко (минимум 1 см) – 0. Мощность отражённого излучения сильно зависит от материала и цвета. Для определения точного расстояния используйте ультразвуковой датчик.

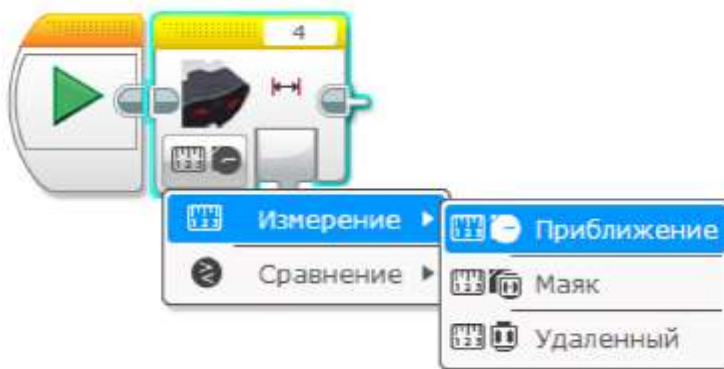


Рис. 68. Режим определения относительного расстояния до объекта инфракрасного датчика

Режим определения расстояния и углового положения маяка

В режиме совместной работы с маяком инфракрасный датчик может обнаружить приблизительное относительное расстояние и угол отклонения маяка.

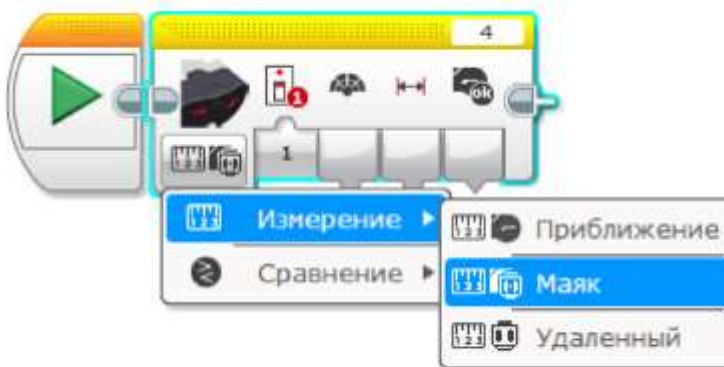


Рис. 69. Режим совместной работы с маяком

В случае, когда маяк расположен прямо перед датчиком, относительный результат измерения угла будет равен 0, максимальное расположение маяка слева, против часовой стрелки -25, справа по часовой стрелке 25.

Режим дистанционного управления.

Инфракрасный датчик способен определять состояние кнопок на маяке. Маяк имеет четыре канала, следовательно, можно организовать дистанционное управление одновременно четырьмя роботами.

Допускается одновременное нажатие двух управляющих кнопок. Каждый из этих вариантов пронумерован.

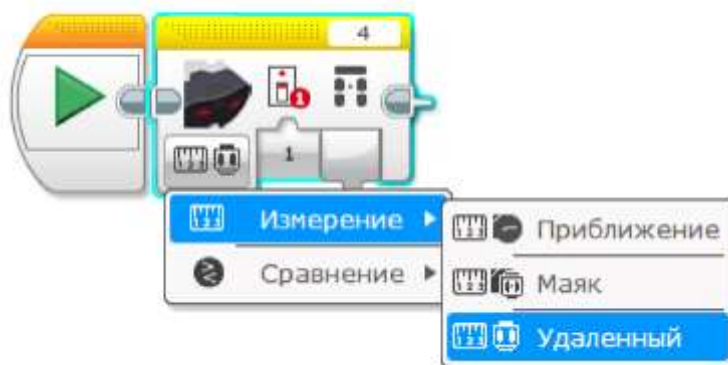


Рис. 70. Режим дистанционного управления

### Пример 5:

Инфракрасные лучи – это электромагнитное излучение, подчиняющееся законам оптики. Следовательно, чем светлее объект, тем большее количество лучей будет от него отражаться, чем темнее – тем меньшее количество лучей будет от него отражаться. Используя этот закон физики, мы можем проградуировать наш датчик так, чтобы он показывал не расстояние до объекта (будем полагать, что расстояние до всех объектов равны), а цвет объекта.

Установим робота с ИК-датчиком перед объектом, например, белого цвета, и зафиксируем показание, допустим 70 единиц. Затем направим датчик в сторону чёрного объекта (расстояние должно быть таким же) и зафиксируем результат, допустим 40 единиц. Повторим действия с объектами других цветов. Объекты должны отличаться только цветом, форма, размер и материал должны быть одинаковыми.

В программе пропишем: если измеренное расстояние попадает в диапазон от X до Y единиц, значит, цвет белый и т.д. Практика показывает, что разница между черными и белыми объектами на расстоянии около 30 см может достигать 20 единиц. В случае, когда необходимо определять цвета, например, красный, зелёный, синий, разброс значений будет не таким большим, но его можно обнаружить. Значения также будут различаться в зависимости от материала и температуры объекта.

### 7. Датчик вращения мотора.

Большой и средний моторы содержат встроенные датчики вращения, которые могут определять количество совершённых оборотов или градусов, а также мгновенную мощность мотора.

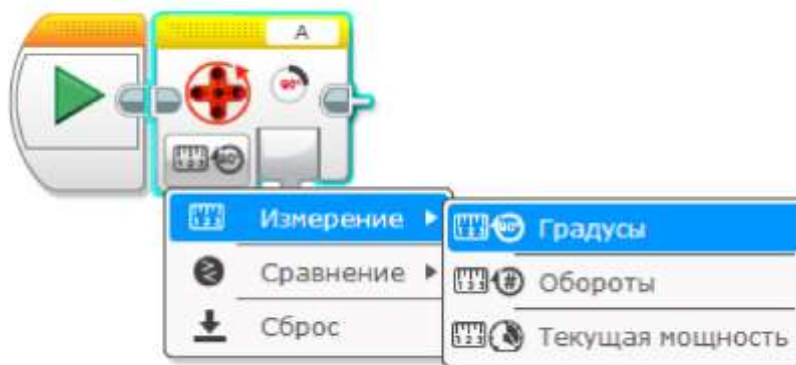


Рис. 71. Датчик вращения мотора

Подобно любому другому датчику датчик вращения используется в режимах измерения, ожидания и сравнения. Отсчёт сделанных оборотов (или градусов) начинается с момента запуска программы. Поэтому если необходим точный отсчёт оборотов или градусов, начиная с определенного момента, необходимо произвести сброс значений датчика. Если отсчёт начинается с начала выполнения программы, сброс не обязателен.

При повороте оси мотора по часовой стрелке происходит суммирование всех оборотов, при повороте оси против часовой стрелки – вычитание.

#### 8. Кнопки управления модулем.

Модуль EV3 анализирует данные от пяти кнопок управления модулем (влево, центр, вправо, вверх и вниз), расположенных на передней панели. Существует возможность узнать, какая кнопка нажата, проверить режим нажатия (Нажатие, освобождение или щелчок) и получить логический вывод о нажатии (Истина или Ложь).

На следующем рисунке изображен программный блок «Кнопки управления модулем EV3».

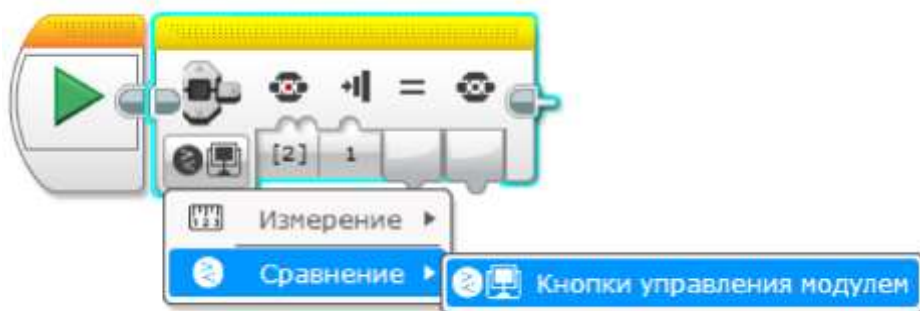


Рис. 72. Программный блок «Кнопки управления модулем EV3»

Кнопка «Назад» на модуле EV3 не относится к кнопкам управления модулем, а прерывает выполнение программы. Одновременное нажатие двух кнопок недопустимо.

У каждой кнопки есть свой номер.

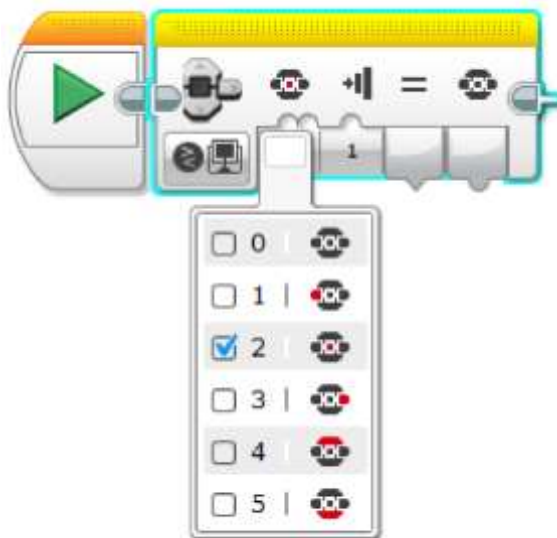


Рис. 73. Настройка программного блока «Кнопки управления модулем EV3»

Как и в других датчиках кнопки имеют режимы Сравнение, Измерение и Изменение.

**Пример 6:** создадим программу, которая активируется нажатием на среднюю кнопку EV3.



Рис. 74. Программа с использованием программного блока «Кнопки управления модулем EV3»

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет» и необходимым программным обеспечением, а также возможностью подключения к электронной информационно-образовательной среде университета, наборы конструкторов LEGO Mindstorms Education EV3 Core Set 45544 (базовый набор), наборы LEGO Mindstorms Education EV3 Expansion Set 45560 (ресурсный набор), зарядные устройства.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, среда LEGO Mindstorms Education EV3.

#### Задания

1. Создайте программу для робота, который едет по прямой, останавливается перед предметом на расстоянии 10 см и воспроизводит любой звук.
2. Составить программу для робота, который при показаниях УЗ-датчика менее 20 см движется назад от предмета, а при показаниях датчика более 20 см движется вперед к предмету, т.е. так робот сохраняет дистанцию от препятствия в 20 см.
3. Разработать программу, с помощью которой робот будет функционировать в качестве радара, т.е. робот будет измерять скорость движущихся автомобилей и, если скорость выше заданного значения, то робот будет издавать предупреждающий сигнал. Алгоритм работы следующий: 1) радар ожидает движущийся объект; 2) как только в зоне контроля появляется объект (автомобиль), то производится измерение расстояния до него; 3) ожидает некоторое время, например, 1 секунду; 4) далее производится еще одно измерение расстояния до объекта; 5) подсчитывается пройденное расстояние и сравнивается с заданным значением; 6) заданное значение выводится на экран и в случае превышения скорости подается сигнал тревоги.
4. Напишите программу, чтобы робот двигался вперед 3 секунды, поворачивался на 90 градусов и двигался еще 3 секунды.
5. Разработайте программу для движения по черной линии с использованием датчика цвета.

#### Указания по порядку выполнения работы

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить робота и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

#### Содержание отчета

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного

задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Что такое датчик?
2. В какие порты модуля EV3 подключаются датчики?
3. Что представляет собой датчик цвета?
4. Что такое ультразвуковой датчик?
5. Как работает гироскопический датчик?
6. Назовите виды и назначение остальных датчиков, входящих в состав комплекта LEGO Mindstorms Education EV3.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 5];

дополнительная литература – [5, 7, 8, 10, 12, 19, 22, 26, 27, 28].

### **Лабораторная работа 6**

#### **Операции с данными в среде LEGO Mindstorms Education EV3**

**Цель работы:** исследование возможностей и применение для программирования робота красной палитры среды программирования Lego Mindstorms Education EV3.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

#### Содержание:

1. Введение.
2. Блок «Математика».
3. Блок «Округление».
4. Блок «Сравнение».
5. Блок «Интервал».
6. Блок «Случайное значение».
7. Блок «Операции над массивом».

#### 1. Введение.

Красная палитра среды программирования LEGO Mindstorms Education EV3 содержит блоки, предназначенные для осуществления операций над данными следующих типов: числовыми и логическими данными, а также текстом.

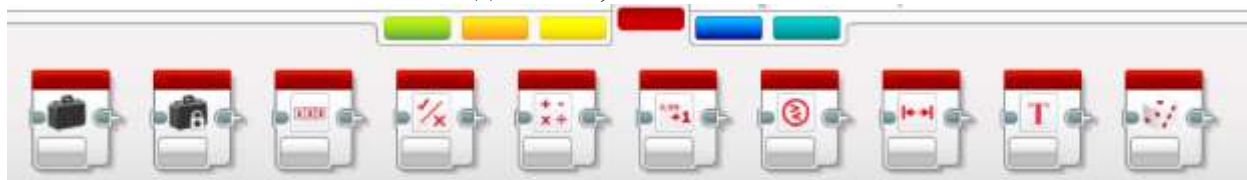


Рис. 75. Красная палитра

Для работы с числовыми данными используются типы «Числовое значение» и «Числовой массив». Для преобразования логических данных применяются типы «Логическое значение» и «Логический массив». Для операций над текстовыми данными используется тип «Текст».

Тип данных «Числовое значение» позволяет выполнять различные математические операции над числами. Числа в программе могут быть как положительными, так и отрицательными, быть целыми значениями или содержать десятичную дробь. Например, 3,14, -2,5, 20.

Перед тем, как начать обрабатывать различные типы данных в программах, необходимо научиться их создавать и хранить. Для этих целей среда программирования предоставляет два вида программных блоков: «Переменная» и «Константа». Эти блоки позволяют создать в памяти робота специальные ячейки, позволяющие записывать, извлекать и редактировать различные типы данных.

Программный блок «Константа» позволяет создавать ячейку памяти для хранения одного из пяти типов данных. Требуемое значение записывается в ячейку на этапе создания программы и остается неизменным во время выполнения всей программы. Для получения значения, записанного в блок «Константа» используется «Вывод».

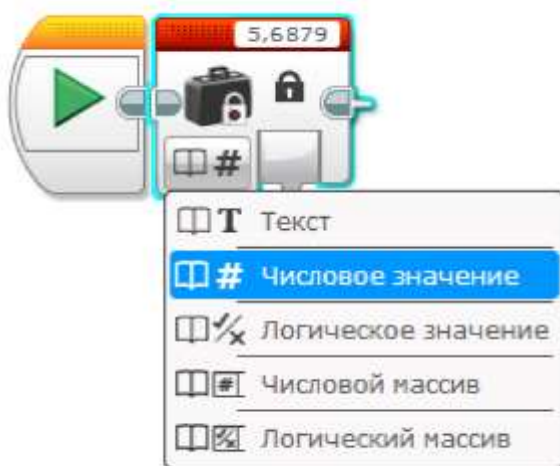


Рис. 76. Блок «Константа»

В программном блоке «Переменная» существует два режима «Считывание» и «Записать». Прежде чем использовать переменную, ей необходимо задать имя, выбрав параметр блока «Добавить переменную». Имя переменной может содержать только заглавные и строчные буквы латинского алфавита, цифры, а также символы \_ и -. Задать значение переменной можно, записав или передав число в параметр «Значение».

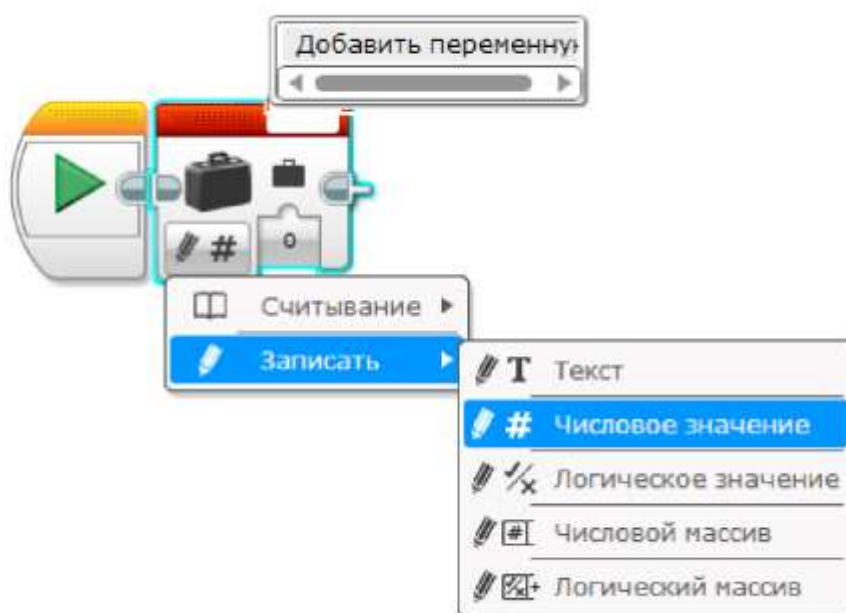


Рис. 77. Блок «Переменная»

## 2. Блок «Математика».

Блок «Математика» служит для выполнения математических вычислений. Он позволяет выполнить выбранную математическую операцию над двумя числами, заданными параметрами «a» и «b».

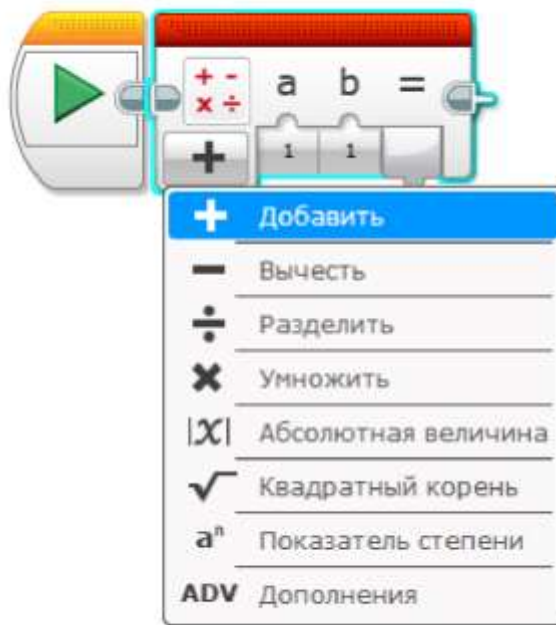


Рис. 78. Блок «Математика»

В режиме «Дополнения» количество параметров для расчета увеличивается до четырех. В параметр «Уравнение» можно вписать любую произвольную формулу.

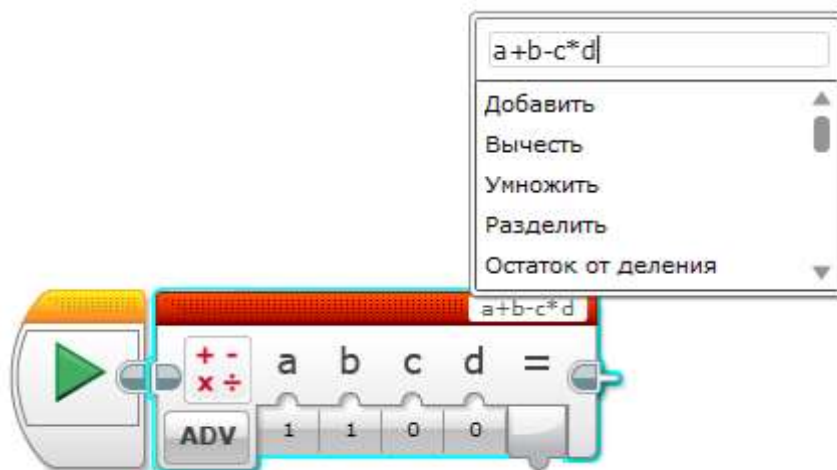


Рис. 79. Настройка блока «Математика»

### 3. Блок «Округление».

Режимы «До ближайшего», «округлить к большему» и «округлить к меньшему» производят округление до целого значения. В режиме «Отбросить дробную часть» можно задать количество остающихся знаков дробной части после запятой.

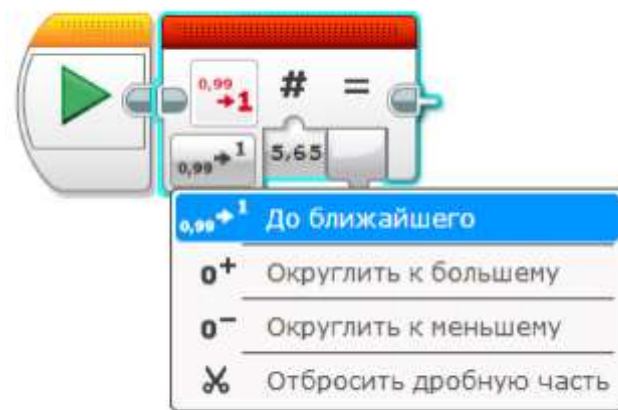


Рис. 80. Блок «Округление»

#### 4. Блок «Сравнение».

Блок предназначен для сравнения двух числовых значений, которые могут вписываться в окна блока или приходить по проводникам. На выходе появляется логическое значение «Истина» или «Ложь», в зависимости от результата сравнения.

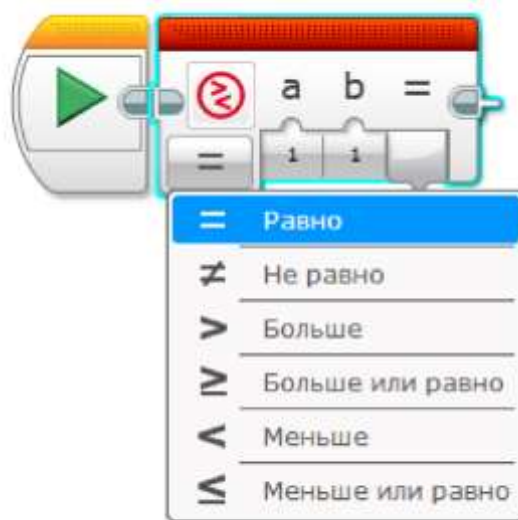


Рис. 81. Блок «Сравнение»

#### 5. Блок «Интервал».

Блок предназначен для определения нахождения числа относительно диапазона: внутри или снаружи. Значения могут, как непосредственно вписываться в окна блока, так и приходить по проводника. На выходе блока появляется логическое значение «Истина» или «Ложь», в зависимости от полученного результата.

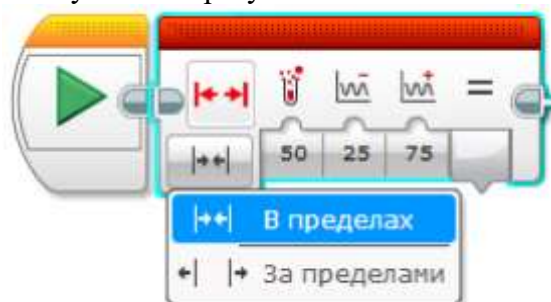


Рис. 82. Блок «Интервал»

#### 6. Блок «Случайное значение».

С помощью данного блока можно при необходимости получить случайные числовые или логические значения в предварительно заданном диапазоне.

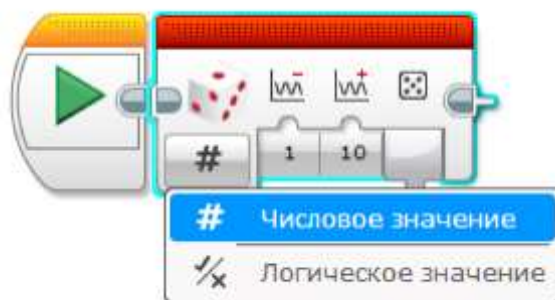


Рис. 83. Блок «Случайное значение»

## 7. Блок «Операции над массивом».

Массив – это набор однотипных элементов, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу. Индекс первого элемента равен 0. В среде EV3 можно работать с одномерными числовыми и логическими массивами.

Блок Операции над массивом позволяет определять длину массива, читать, записывать и удалять требуемые элементы массива.

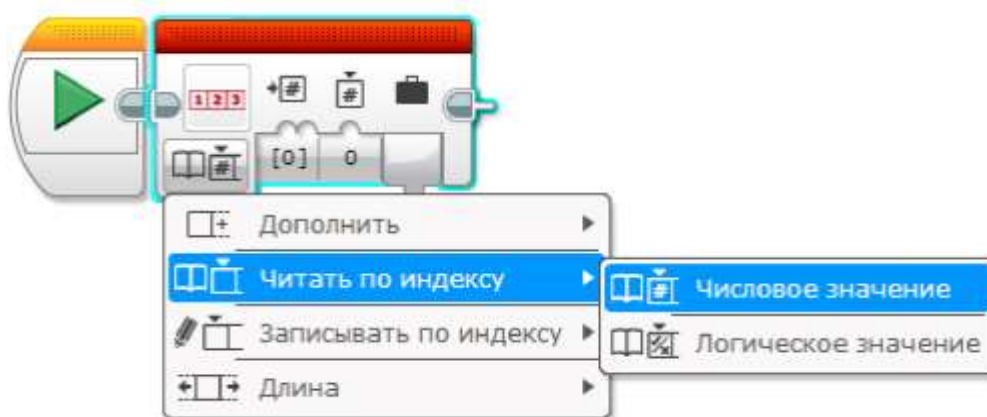


Рис. 84. Блок «Операции над массивом»

Прежде чем начать работу с массивами, необходимо их инициировать, т.е. указать тип (числовой или логический) и присвоить имя. Данные в массив можно вносить в ручном или автоматическом режиме (считывая показания с датчиков). Для создания массива необходимо использовать блок «Переменная».

**Пример 1. Программа прямолинейного движения для проезда роботом расстояния в 0,5 метра.**

**Решение:**

За один полный оборот мотора робот проезжает расстояние, равное длине окружности колеса. Это расстояние можно найти, умножив число Пи ( $\pi = 3,14159$ ) на диаметр колеса. Диаметр колеса из образовательного набора Lego Mindstorms Education EV3 равен 56 мм. Если переведем расстояние в 0,5 метра в миллиметры (500 мм) и разделим на расстояние, которое робот проходит за один оборот мотора, то узнаем: сколько оборотов мотора необходимо для проезда всего заданного расстояния.

1. Используя программный блок «Константа», заведем в программу постоянное число Пи, равное примерно 3,14159.



Рис. 85. Использование блока «Константа»

2. Используя программный блок «Переменная», создадим в программе переменную «d» и занесем в нее значение диаметра колеса.



Рис. 86. Использование блока «Переменная»

3. Используя программный блок «Математика», умножим значение блока «Константа» на значение переменной d. Для передачи значения из переменной d в программный блок «Математика» используем второй программный блок «Переменная» в режиме «Считывание». Для передачи значений между программными блоками используются шины данных. Чтобы установить шину данных, необходимо «потянуть» выходной параметр одного программного блока и «присоединить» его к входному параметру другого программного блока.

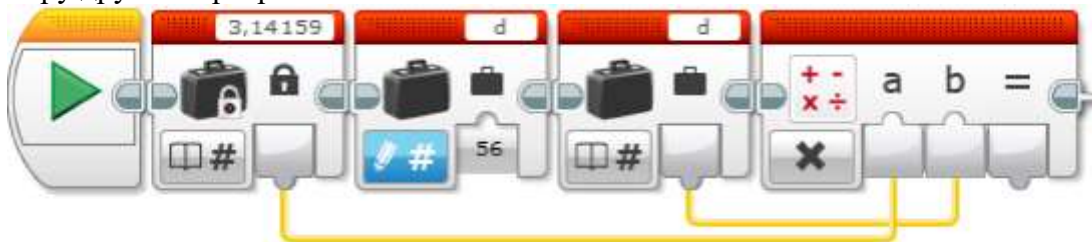


Рис. 87. Добавление блока «Математика»

4. Используя программный блок «Математика», разделим значение пути (500 мм) на значение, полученное в шаге 3.

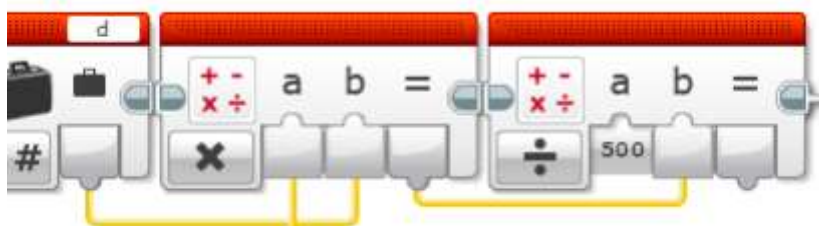


Рис. 88. Умножение с помощью блока «Математика»

5. Полученное в шаге 4 значение подадим в параметр «Обороты» блока «Рулевое управление».

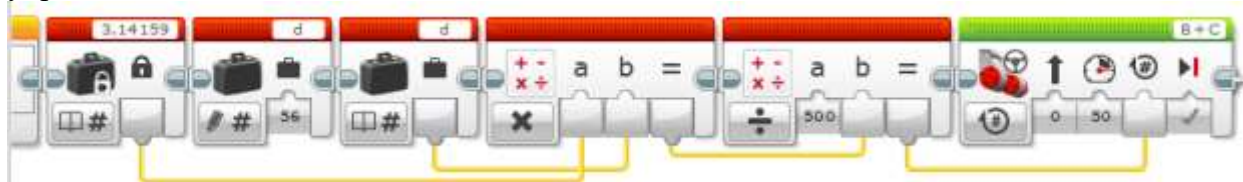


Рис. 89. Добавление блока «Рулевое управление»

Загрузим полученную программу в нашего робота. Поставим робота на ровную свободную площадку и запустим программу. Измерив расстояние, пройденное роботом, убедимся в правильности нашей программы.

### Пример 2. Измерение расстояния до предмета.

Создадим программу для робота с использованием ультразвукового датчика, который будет измерять расстояние до объекта, пока оно не станет равным или меньше 8 см. Если расстояние будет равно 8 см, то робот на экране должен выдать стандартное изображение «Ur» в течение 2 секунд. В процессе работы данные о расстоянии необходимо отображать на экране контроллера.

#### Решение.

Создаем проект. Открываем вкладку «Управление операторами» и выбираем блок



«Цикл». Соединяем с блоком «Старт».

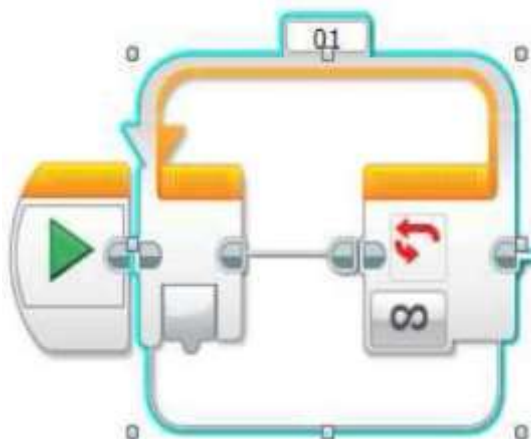


Рис. 90. Добавление блока «Цикл»

Во вкладке «Датчик» выбираем блок «Ультразвуковой датчик» и добавляем в цикл. Теперь редактируем блок. Нажимаем на кнопку «Измерение». Далее во всплывающем списке выбираем «Сравнение» – «Расстояние в сантиметрах».

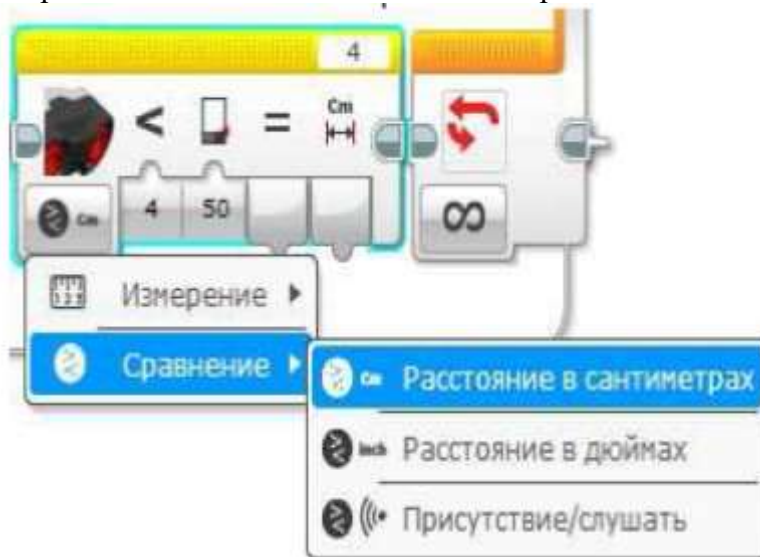


Рис. 91. Настройка ультразвукового датчика  
В «Тип сравнения» выбираем значок «Меньше».

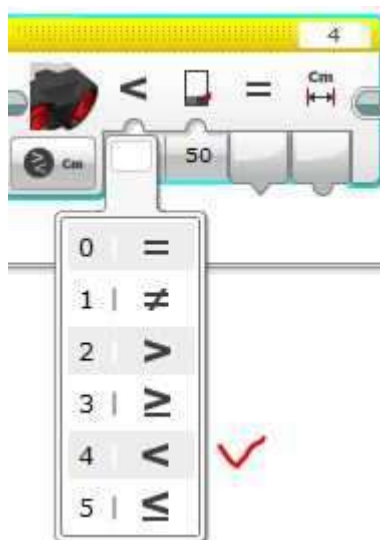


Рис. 92. Настройка типа сравнения ультразвукового датчика  
Пороговое значение указываем 8.

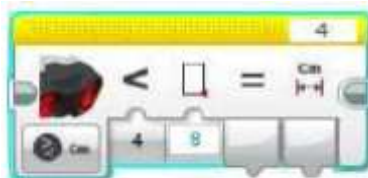


Рис. 93. Установка порогового значения

Во вкладке «Операции с данными» выбираем блок «Текст»  и добавляем к блоку «Ультразвуковой датчик». Начинаем редактирование блока «Текст». Для этого возвращаемся к блоку «Ультразвуковой датчик» и соединяем точку «Расстояние в сантиметрах» ультразвукового датчика с точкой соединения «А» блока «Текст». На рисунке линия (желтая).



Рис. 94. Настройка блока «Текст»

Далее в блоке «Текст» в позиции «В» ставим единицу измерения английскими буквами «cm».



Рис. 95. Настройка единиц измерения в блоке «Текст»



После этого во вкладке «Действие» выбираем блок «Экран» и добавляем к блоку «Текст». Далее в блоке «Экран» нажимаем на кнопку «Изображение» и выбираем позицию «Текст» – «Сетка».

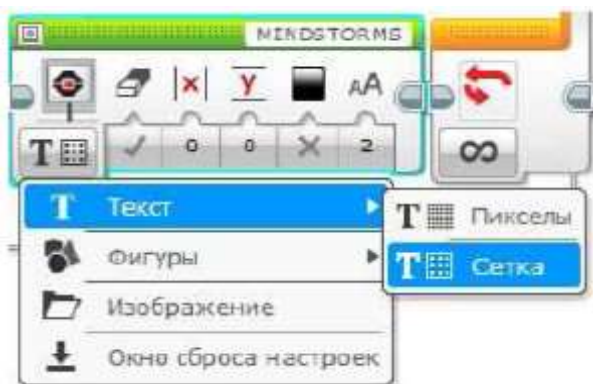


Рис. 96. Настройка блока «Экран»

Далее нажимаем на строку в правом верхнем углу блока, где написано «MINDSTORMS» и выбираем «Проводной».



Рис. 97. Настройка блока «Экран»

Возвращаемся к блоку «Текст» и соединяем точку «Результат» с точкой «Текст» блока «Экран». На рисунке оранжевая линия.



Рис. 98. Соединение блока «Экран» с блоком «Текст»

Возвращаемся к блоку «Экран». Столбец «X» будет иметь значение 5, а строка «Y» значение 6.



Рис. 99. Задание параметров блока «Экран»

Теперь все блоки цикла созданы. Осталось отредактировать сам цикл. Для этого в конце цикла в блоке нажимаем на значок бесконечности и выбираем в списке «Логическое значение».



Рис. 100. Редактирование блока «Цикл»

После этого возвращаемся к блоку «Ультразвуковой датчик» и тянем



соединительную линию от точки «Результат сравнения» с точкой «Пока не будет



истина: Истина» в конце цикла (на рисунке зеленая линия).

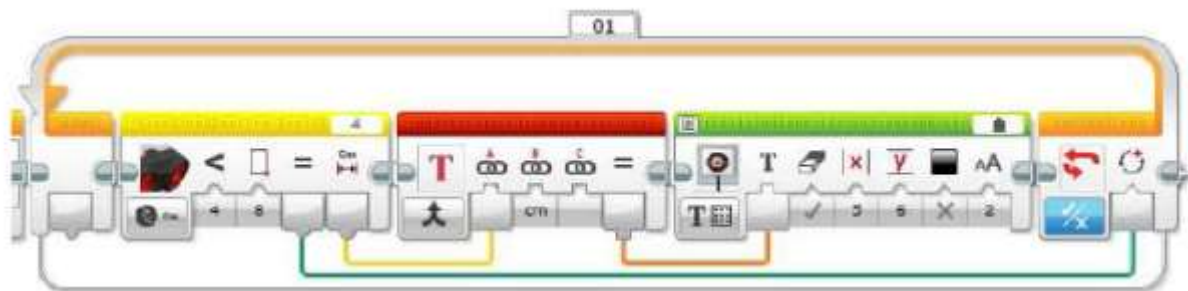


Рис. 101. Соединение блоков

После этого во вкладке «Действие» выбираем блок «Экран» и соединяем с циклом. Редактируем блок. Нажимаем на строку «Имя файла» в верхнем правом углу и выбираем из списка «Изображение проекта» – «Up».



Рис. 102. Редактирование блока «Экран»

Далее необходимо использовать блок «Ожидание» из вкладки «Управление операторами». Присоединяем данный блок к блоку «Экран» и включаем его на 2 секунды.



Рис. 103. Добавление блока «Ожидание»

Программа готова.

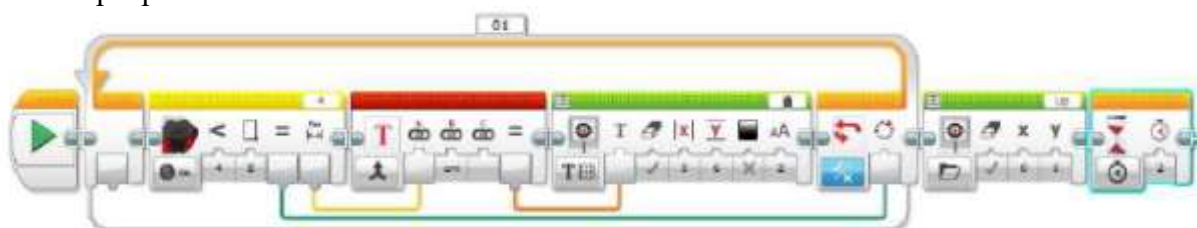


Рис. 104. Программа прямолинейного движения робота

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет» и необходимым программным обеспечением, а также возможностью подключения к электронной информационно-образовательной среде университета, наборы конструкторов LEGO Mindstorms Education EV3 Core Set 45544 (базовый набор), наборы LEGO Mindstorms Education EV3 Expansion Set 45560 (ресурсный набор), зарядные устройства.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, среда LEGO Mindstorms Education EV3.

#### Задания

1. Создавайте программу для робота с использованием ультразвукового датчика, который будет измерять расстояние до объекта, пока оно не станет равным или меньше 15 см. Если расстояние будет равно 8 см, то робот на экране должен выдать стандартное изображение «Ur» в течение 5 секунд. В процессе работы данные о расстоянии необходимо отображать на экране контроллера.

2. Создайте программу, чтобы робот проехал по траектории в виде квадрата.

3. Напишите программу по предложенной ниже схеме. Опишите ее работу. Подключите робота и необходимые компоненты для тестирования программы.

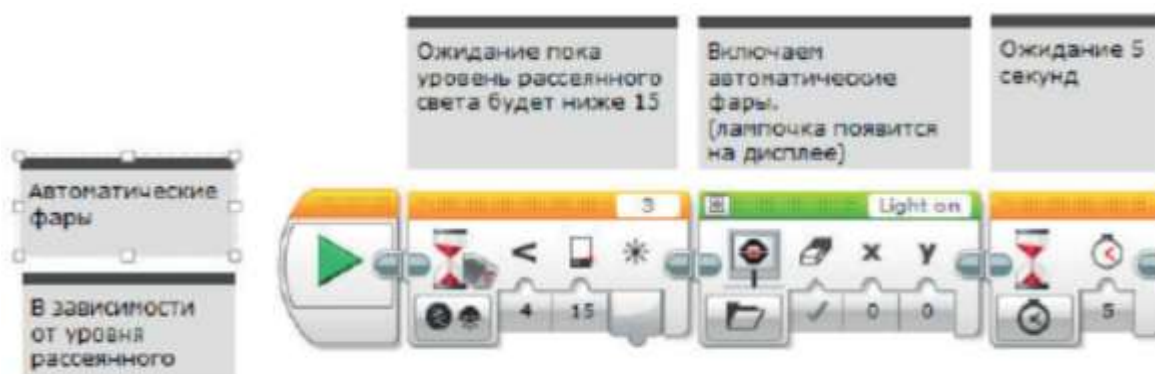


Рис. 105. Программа к заданию 3

4. Составьте программу по предложенной ниже схеме. Исследуйте и опишите ее работу. Подключите робота и необходимые компоненты для тестирования программы.

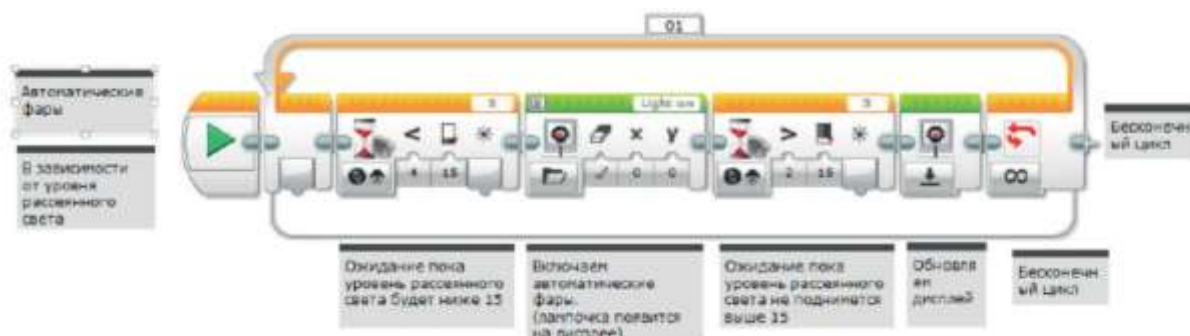


Рис. 106. Программа к заданию 4

### Указания по порядку выполнения работы

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить робота и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

### Содержание отчета

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Какие типы данных можно обрабатывать в программах среды Lego Mindstorms Education EV3?
2. Для чего применяется красная палитра среды LEGO Mindstorms Education EV3?
3. Назовите блоки красной палитры и их назначение в среде LEGO Mindstorms Education EV3.
4. Как задать диапазон для блока «Случайное значение»?
5. Перечислите режимы блока «Округление».
6. Какое максимальное количество параметров можно задать в блоке «Математика»?

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 5];

дополнительная литература – [5, 7, 8, 10, 12, 19, 22, 26, 27, 28].

### **Лабораторная работа 7**

#### **Знакомство с программно-аппаратной платформой Arduino**

**Цель работы:** изучение возможностей онлайн-среды Tinkercad от Autodesk для разработки проектов на базе Arduino.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

*Содержание:*

1. Программно-аппаратная платформа Arduino.
2. Мигание встроенного светодиода.
3. Эффект бегущей волны из огоньков светодиодов.

#### 1. Программно-аппаратная платформа Arduino.

Arduino (<http://www.arduino.cc/>) – это программно-аппаратная платформа для разработки электронных устройств, поддерживаемая разработчиками по всему миру и пользующаяся огромной популярностью в России и за рубежом. Платформа позволяет общаться и взаимодействовать с окружающим миром с помощью всевозможных датчиков, сенсоров, моторов и других узлов. Платформа состоит из двух частей: программной и аппаратной.

В качестве программной части выступает кроссплатформенная среда разработки Arduino IDE, которая может запускаться на операционных системах Windows, Linux, Mac OS. С помощью Arduino IDE можно программировать платы. Также существует онлайн-среда моделирования работы платы Arduino, которая разработана компанией Autodesk – Tinkercad (<https://www.tinkercad.com/>). Именно в данной среде будут разработаны наши проекты. Для работы с платформой Arduino необходимо выбрать вкладку Цепи (Circuits).

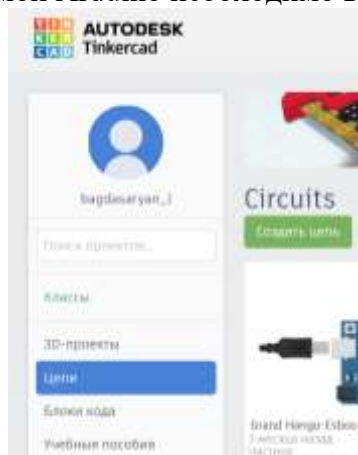


Рис. 107. Начало работы с Tinkercad

В качестве аппаратной части Arduino выступают различные платы. Мы будем работать с платой Arduino UNO – одной из основных моделей Arduino, предназначенной для начинающих разработчиков.

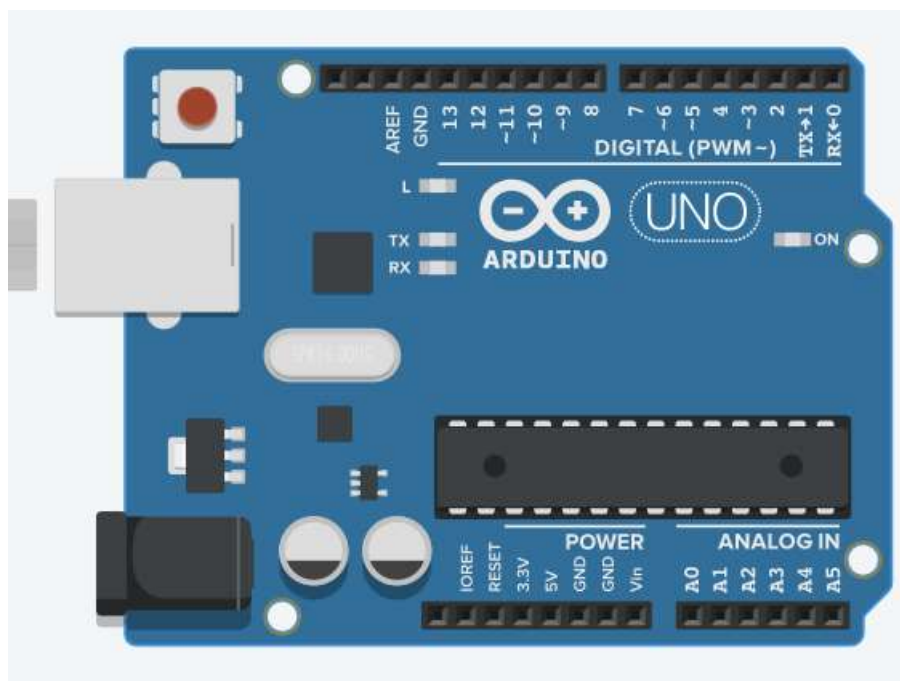


Рис. 108. Плата Arduino UNO

Алгоритм разработки устройства на базе Arduino включает:

1. Определение цели, отражающей конечный результат работы над проектом.
2. Разработка алгоритма. Алгоритм – это последовательность шагов, описывающих работу устройства. Алгоритм может быть представлен в любой форме (словесный, графический и др.). Представленная последовательность шагов должна обладать всеми свойствами алгоритма (свойство результативности, детерминированности, дискретности и др.).
3. Выбор компонентов аппаратной части. На данном шаге необходимо выяснить, какое дополнительное аппаратное обеспечение (физическое или в среде моделирования) будет подключаться к плате Arduino.
4. Программирование работы устройства. Программы Arduino обычно называют скетчами (англ. sketch – набросок). На данном этапе нужно написать скетч или код программы для функционирования платы Arduino.
5. Создание аппаратной части проекта. На этом шаге к плате подключаются все заданные аппаратные компоненты.
6. Тестирование и отладка. Здесь определяются и устраняются ошибки программной или аппаратной составляющей проекта.

## **2. Мигание встроенного светодиода.**

/\* C++ code

\* Мигание встроенным в плату светодиодом (LED)

\* Встроенный LED подсоединен к выходу 13 платы и мигает

\* с интервалом в 1 секунду (1 секунду включен, 1 секунда паузы)

\*/

int ledPin = 13; // подключение LED к выводу 13

void setup()

{

pinMode(ledPin, OUTPUT); // определение порта 13 как выхода

}

void loop()

{

digitalWrite(ledPin, HIGH); // включение LED, задание высокого уровня (1)

```

    delay(1000); // задание паузы в 1 секунду (1000 мс = 1 секунда), сохранение высокого
уровня (1) на выходе в течение этого времени
    digitalWrite(ledPin, LOW); // выключение LED, задание низкого уровня (0)
    delay(1000); // задание паузы в 1 секунду (1000 мс = 1 секунда), сохранение низкого
уровня (0) на выходе в течение этого времени
}
/* команды, заданные в функции loop (), будут повторяться в цикле бесконечно до
остановки программы */

```

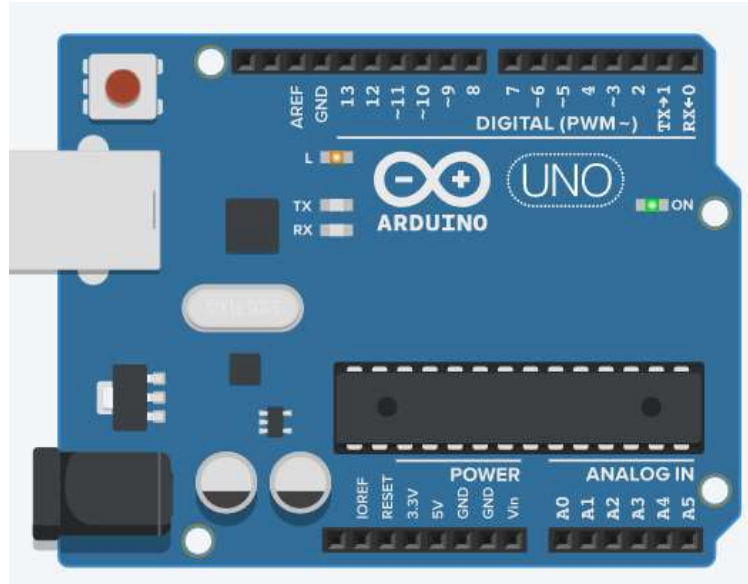


Рис. 109. Мигание встроенного светодиода

В данном примере функция с названием `setup()` нужна для начальной конфигурации программы. Данная функция вызывается один раз при первом старте программы и на протяжении всего периода работы основной программы больше не вызывается. Зачем так сделано? Дело в том, что чаще всего при первом старте программы, необходимо проинициализировать периферию. Если это делать в основном цикле, то процессор будет выполнять не нужную работу. Поэтому нужно пока запомнить, что код, написанный в теле функции `setup()`, выполнится один раз. А вот основная программа, которая будет выполняться на протяжении всей работы микроконтроллера, записывается в тело функции `loop()`.

Т.е. для начальной настройки и выполнения разово кода программы заполняем тело функции `setup()`. Для работы основной программы заполняем тело функции `loop()`.

Для того чтобы помигать светодиодом, нам необходимо посмотреть где он подключен. На плате Arduino UNO в примере он подключен к 13 ножке выводов DIGITAL. Сначала в функции `setup()` мы настроим ножку, а затем в функции `loop()` будем включать и выключать светодиод. Цифровая ножка микроконтроллера может выполнять всего два действия, либо выводить значение 1 или 0, либо читать значение, переданное на ножку тоже 1 или 0. В данный момент, так как нам нужно зажигать светодиод, то вывод должен быть настроен как выход.

Задача функции `pinMode()` является настройка пина на вход или на выход. Для того чтобы данная функция поняла, что ей делать, она принимает два аргумента. Первый аргумент отвечает за номер пина. В нашем случае это 13, а второй аргумент (все аргументы, передаваемые функциям, пишутся через запятую) это направление. В нашем случае это выход OUTPUT. в конце функции как можно заметить стоит точка с запятой. Данная точка с запятой говорит о том, что текущая команда закончилась. Следует запомнить, как таблицу

умножения, что каждая команда обязательно должна заканчиваться точкой с запятой. Вот таким образом мы настроили вывод.

Теперь переходим к телу функции `loop()`. Первая строчка `digitalWrite(13, HIGH);` – это функция, которая выводит в вывод микроконтроллера некое значение. Для того чтобы она понимала какое значение и куда ей выводить мы снова передаем два аргумента. Первый это номер вывода, а второй `HIGH`, то есть 1. Как только эта функция отработает, на выводе 13 появится логическая единица (логическая единица всегда равна напряжению питания МК. Чаще всего это 5 вольт или 3,3 вольта). Следующая строка `delay(1000);` – это пауза. Данной функции в качестве аргумента передается значение времени в миллисекундах. В нашем случае это 1000 мс или 1 секунда. Теперь рассмотрим, что произойдет. Сначала функция `digitalWrite(13, HIGH);` выводит на ножку логическую единицу. Затем функция `delay(1000);` останавливает работу программы на 1 секунду, а затем выполняется функция `digitalWrite(13, LOW);`. Как не трудно догадаться она выводит на ножку логический ноль. Затем снова останавливается на 1 секунду. Так как функция `loop()` является циклической, то после паузы программа возвращается к началу тела функции `loop()` и все начинается с начала.

### **3. Эффект бегущей волны из огоньков светодиодов.**

В этом проекте используется пять светодиодов для имитации эффекта бегущей световой волны.

*Алгоритм для этого проекта следующий:*

1. Включить первый светодиод.
2. Ждать 500 мс.
3. Выключить первый светодиод.
4. Включить второй светодиод.
5. Ждать 500 мс.
6. Выключить второй светодиод.
7. Аналогичным образом дойти до пятого светодиода, а затем повторить данные действия в обратном порядке – от пятого до первого светодиода.
8. Задать бесконечный цикл для повторения всех пунктов.

*Компоненты, необходимые для реализации проекта:*

- 5 светодиодов.
- 5 резисторов номиналом 560 Ом.
- Макетка.
- Провода.
- Плата Arduino.

Для подключения светодиодов используем цифровые выходы с номерами со 2-го по 6-ой. Для соединения с каждым светодиодом используем резисторы номиналом 560 Ом.

*Введите следующий код в среде разработки:*

```
// C++ code
// Проект Эффект бегущей волны из светодиодов
void setup()
{
  pinMode(2, OUTPUT); // определение порта 2 как выхода
  pinMode(3, OUTPUT); // определение порта 3 как выхода
  pinMode(4, OUTPUT); // определение порта 4 как выхода
  pinMode(5, OUTPUT); // определение порта 5 как выхода
  pinMode(6, OUTPUT); // определение порта 6 как выхода
}
void loop()
{
  digitalWrite(2, HIGH); // включение первого светодиода, задание высокого уровня (1)
```

```

delay(500); // ожидание в 500мс или в 0,5 секунды
digitalWrite(2, LOW); // выключение первого светодиода, задание низкого уровня (0)
digitalWrite(3, HIGH); // включение второго светодиода
delay(500); // ожидание
digitalWrite(3, LOW); // выключение второго светодиода
digitalWrite(4, HIGH); // включение третьего светодиода
delay(500); // ожидание
digitalWrite(4, LOW); // выключение третьего светодиода
digitalWrite(5, HIGH); // включение четвертого светодиода
delay(500); // ожидание
digitalWrite(5, LOW); // выключение четвертого светодиода
digitalWrite(6, HIGH); // включение пятого светодиода
delay(500); // ожидание
digitalWrite(6, LOW); // выключение пятого светодиода
digitalWrite(5, HIGH); // включение четвертого светодиода
delay(500); // ожидание
digitalWrite(5, LOW); // выключение четвертого светодиода
digitalWrite(4, HIGH); // включение третьего светодиода
delay(500); // ожидание
digitalWrite(4, LOW); // выключение третьего светодиода
digitalWrite(3, HIGH); // включение второго светодиода
delay(500); // ожидание
digitalWrite(3, LOW); // выключение второго светодиода
// далее функция loop() завершится
// и будет вызвана снова
}

```

В функции **void setup()** выполняется настройка контактов на работу в режиме цифровых выходов, потому что с их помощью мы будем управлять подачей электрического тока на светодиоды. В функции **void loop()** мы определяем, когда включить каждый светодиод, вызывая функцию `digitalWrite()`.

Схему можно описать несколькими способами. При сопоставлении монтажной схемы с вызовами функций в скетче становится понятно, как она работает. Например, если производится вызов `digitalWrite(2, HIGH)`, на цифровой выход 2 подается ток с напряжением 5 В, который течет через ограничительный резистор, через анод светодиода, через катод и завершает свой путь в разъеме GND («земля»), замыкая цепь. Затем, когда производится вызов `digitalWrite(2, LOW)`, течение тока прекращается и светодиод гаснет.

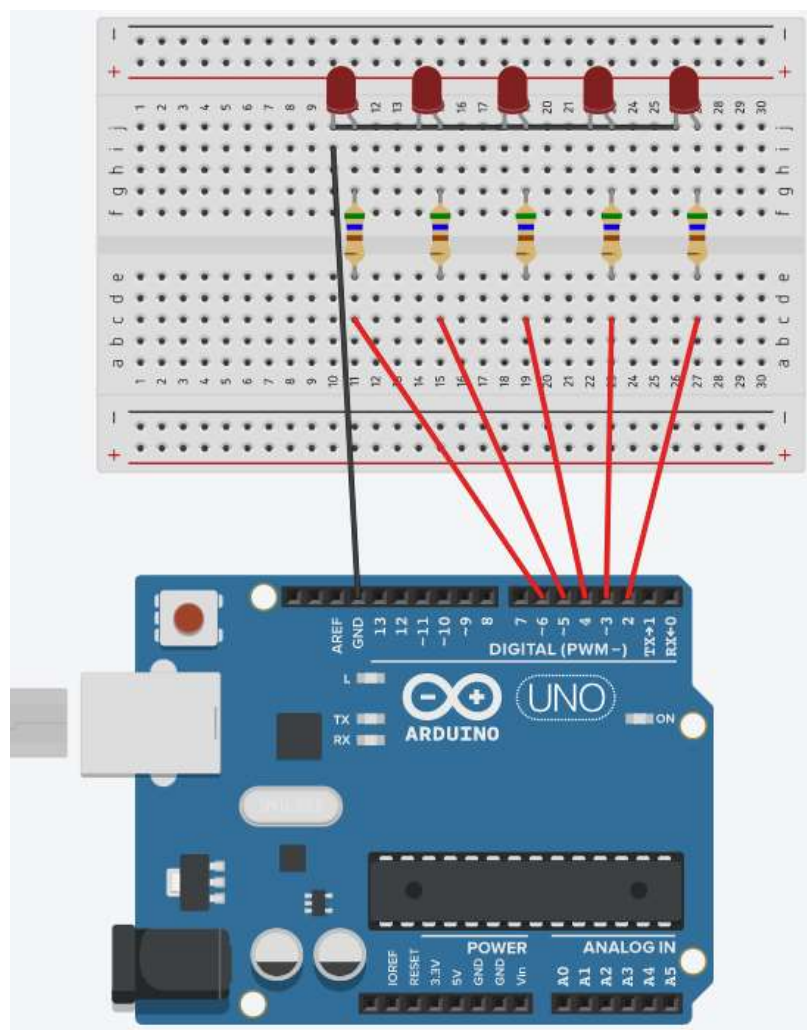


Рис. 110. Схема программы «Эффект бегущей волны из огоньков светодиодов»  
Использование переменных.

Переменные используются в компьютерных программах для хранения данных. Например, в скетче для проекта 1 мы вызывали функцию `delay(500)`, чтобы выполнить задержку перед выключением светодиодов. Слабое место этого скетча – недостаточная гибкость. Если появится желание изменить время задержки, придется вручную изменять каждое его вхождение в скетче. Чтобы исправить эту проблему, добавим в скетч переменную, представляющую значение задержки для функции `delay()`.

Введите следующую строку в скетч проекта 1 перед функцией `void setup()`, сразу за вступительным комментарием:

```
int d = 250;
```

Она присваивает число 250 переменной с именем `d`. Затем замените все числа 500 в скетче именем переменной `d`. Теперь, выполняя скетч, Arduino будет передавать в вызовы функции `delay()` значение переменной `d`. Когда вы загрузите измененный скетч, волна будет бежать существенно быстрее, потому что значение задержки уменьшилось до 250. `int` указывает, что переменная предназначена для хранения целого числа со знаком (integer) в диапазоне от –32 768 до 32 767. Такой переменной можно присвоить любое число, не имеющее дробной части. Теперь, чтобы изменить задержку, достаточно лишь изменить объявление переменной в начале скетча. Например, если присвоить переменной число 100, скорость бегущей волны увеличится еще больше:

```
int d = 100;
```

Далее можно поэкспериментировать с программой, меняя значение задержки и подачу сигналов высокого и низкого уровня.

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

### Задания

1. Сгенерировать сигнал (по номеру варианта, заданного преподавателем) с помощью светодиода, пользуясь азбукой Морзе (в азбуке Морзе буквы обозначаются точками и тире. Точка – короткий сигнал. Тире – длинный, эквивалентный трем точкам.):



Рис. 111. Пример ускоренного радиообмена с использованием азбуки Морзе

Варианты:

1. БЛГ
2. ЗДР
3. SOS
4. QRR
5. QRQ
6. ДСВ
7. СПБ
8. 88
9. QRO
10. QRL
11. QRC

2. Сгенерировать сигнал Вашего имени с помощью светодиода, пользуясь алфавитом азбуки Морзе:

|   |         |   |         |   |         |   |             |
|---|---------|---|---------|---|---------|---|-------------|
| А | ·—      | И | ··      | Р | ·—·     | Ш | — — — —     |
| Б | — ...   | Й | · — — — | С | ...     | Щ | — — · —     |
| В | · — —   | К | — · —   | Т | —       | Ъ | · — — · — · |
| Г | — — ·   | Л | · — ... | У | ·· —    | Ы | — · — —     |
| Д | — ...   | М | — —     | Ф | ·· — ·  | Ь | — ... —     |
| Е | ·       | Н | — ·     | Х | ...     | Э | ·· — ...    |
| Ж | ·· — —  | О | — — —   | Ц | — · — · | Ю | ·· — —      |
| З | — — ... | П | · — — · | Ч | — — — · | Я | · — · —     |

Рис. 112. Алфавит азбуки Морзе

3. С помощью трёх светодиодов симитировать работу светофора, предварительно ознакомившись с помощью сети Интернет с требованиями к применению дорожных знаков и светофоров, касающихся длительности активности каждого сигнала светофора (ГОСТ).

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Что такое Arduino?
2. Приведите алгоритм планирования проекта для Arduino.
3. Какие основные типы данных используются в скетчах Arduino?
4. Охарактеризуйте особенности рисованных и принципиальных схем?
5. Что представляет собой электрический ток?
6. Охарактеризуйте понятия силы тока, напряжения и мощности.
7. Что такое диод? Светодиод?
8. Приведите описание и характеристики резистора.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

**Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

## **Лабораторная работа 8**

### **Цифровые выводы Arduino. Широтно-импульсная модуляция (ШИМ)**

**Цель работы:** изучение возможностей управления вводом/выводом с использованием цифровых выводов Arduino на примере подключения кнопки, рассмотрение способов обработки команды нажатия на кнопку, в том числе для изменения уровня ШИМ-сигнала.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### **Теоретическая часть**

*Содержание:*

1. Цифровые выводы Arduino и функции управления вводом/выводом.
2. Подключение кнопки (режим INPUT\_PULLUP).
3. Подключение кнопки без использования промежуточной переменной buttonState.
4. Подключение кнопки с использованием условного оператора if.
5. Подключение кнопки с использованием директивы #define.
6. Подключение кнопки в режиме INPUT.

7. Подключение кнопки в режиме INPUT с переключением состояния светодиода.
8. Понятие о широтно-импульсной модуляции.
9. Подключение кнопки в режиме INPUT для управления яркостью светодиода.

### 1. Цифровые выводы Arduino и функции управления вводом/выводом.

Для связи с внешними элементами в контроллере Arduino UNO существуют 14 цифровых выводов. Каждый вывод может быть определен программой как вход или выход. У цифрового выхода есть только два состояния: высокое и низкое. Высокое состояние соответствует напряжению на выходе порядка 5В, низкое состояние – 0В. Выход допускает подключение нагрузки с током до 40мА. Когда вывод определен как вход, считав его состояние, можно определить уровень напряжения на входе. При напряжении близком к 5В (реально более 3В) будет считано высокое состояние, соответствующее константе HIGH. При напряжении близком к 0 (менее 1,5В) будет считано низкое состояние, или константа LOW. Светодиод мы должны подключить к выводу, определив его как выход, а кнопка подключается к выводу с режимом вход.

Для работы с цифровыми выводами в системе Arduino есть 3 встроенные функции. Они позволяют установить режим вывода, считать или установить вывод в определенное состояние. Для определения состояния выводов в этих функциях используются константы HIGH и LOW, которые соответствуют высокому и низкому уровню сигнала.

1. Функция **pinMode(pin, mode)** – устанавливает режим вывода (вход или выход).

Аргументы: pin – номер вывода; mode – режим вывода.

*mode = INPUT* вывод определен как вход, подтягивающий резистор отключен

*mode = INPUT\_PULLUP* вывод определен как вход, подтягивающий резистор подключен

*mode = OUTPUT* вывод определен как выход.

Функция не возвращает ничего.

2. Функция **digitalWrite(pin, value)** – устанавливает состояние выхода (высокое или низкое).

Аргументы: pin – номер вывода; value – состояние выхода.

*value = LOW* устанавливает выход в низкое состояние

*value = HIGH* устанавливает выход в высокое состояние

Функция не возвращает ничего.

3. Функция **digitalRead(pin)** – считывает состояние входа.

Аргументы: pin – номер вывода.

Возвращает состояние входа:

*digitalRead(pin) = LOW* низкий уровень на входе

*digitalRead(pin) = HIGH* высокий уровень на входе

### 2. Подключение кнопки (режим INPUT\_PULLUP).

Для простоты соединений будем использовать встроенный светодиод, который нами применялся в предыдущей работе. Он подключен к цифровому выводу 13. Кнопку подключаем к любому другому выводу, например, 2. Аппаратная часть схемы подключения кнопки должна обеспечивать уровни напряжений 0В при нажатой кнопке и 5В при свободной.

При отжатой кнопке резистор формирует на выводе 5В, а при нажатой – вход замыкается на землю. Все выводы платы имеют внутри контроллера резисторы, подключенные к 5В. Их можно программно включать или отключать от выводов. Сопротивление этих резисторов порядка 20-50 кОм. В итоге схема подключения будет выглядеть так:

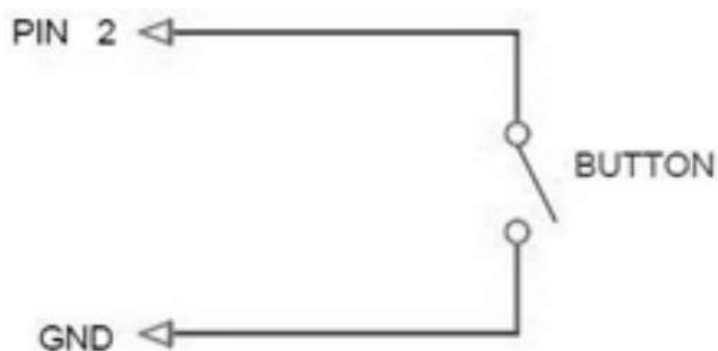


Рис. 113. Принципиальная схема подключения кнопки

Кнопку можно припаять на проводах к разъему, а можно установить ее на макетную плату без пайки.

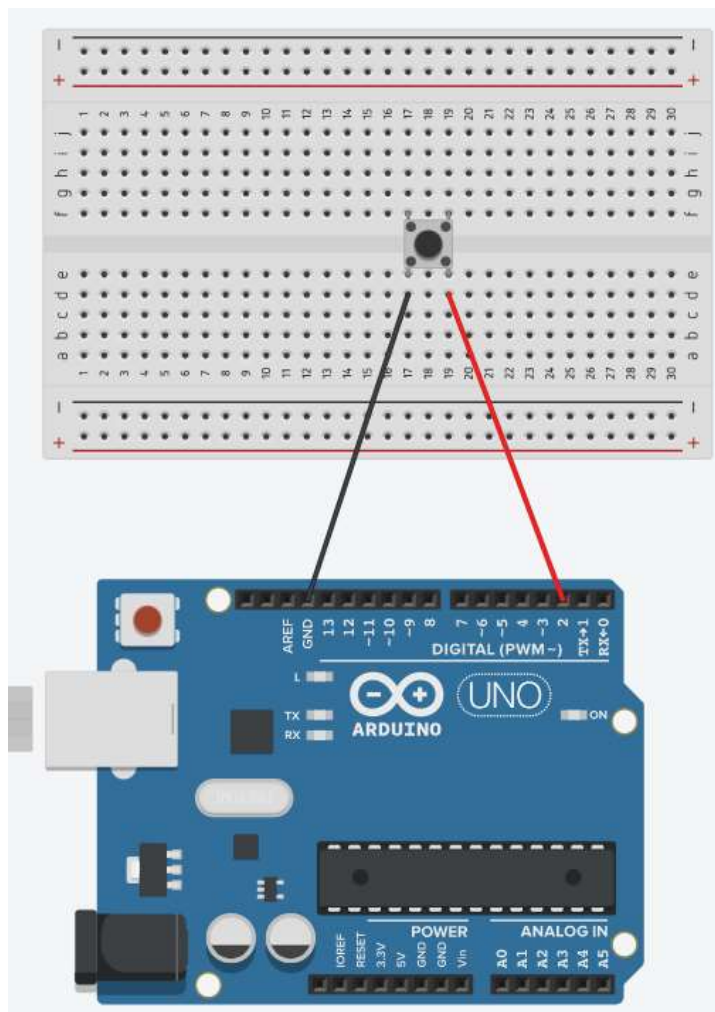


Рис. 114. Схема подключения кнопки

Для работы кнопки, соберите предложенную схему и введите следующий код в среде разработки:

```
// C++ code
// Включение встроенного светодиода при нажатии кнопки
boolean buttonState; //– создание глобальной переменной buttonState
void setup() {
  pinMode(13, OUTPUT); //– определение порта 13 как выхода для светодиода
  pinMode(2, INPUT_PULLUP); //– определение порта 2 как входа для кнопки
}
void loop() {
```

```

    buttonState = digitalRead(2); //– считывание состояния 2 порта и запись данного
значения в переменную buttonState
    buttonState = ! buttonState; //– инверсия переменной buttonState
    digitalWrite(13, buttonState); //– установка состояния порта 2 в соответствии со
значением buttonState
}

```

Для хранения промежуточного значения состояния кнопки создаем переменную buttonState с типом boolean. Это логический тип данных. Переменная может принимать одно из двух значений: *true* (истинно) или *false* (ложно). В нашем случае – светодиод светится и не светится.

### **3. Подключение кнопки без использования промежуточной переменной buttonState.**

Далее измените программу следующим образом. Участок кода программы

```

buttonState = digitalRead(2); //– считывание состояния 2 порта и запись данного
значения в переменную buttonState
buttonState = ! buttonState; //– инверсия переменной buttonState
digitalWrite(13, buttonState); //– установка состояния порта 2 в соответствии со
значением buttonState

```

можно записать без использования промежуточной переменной buttonState.

```
digitalWrite(13, ! digitalRead(2) );
```

Проверьте работу программы с внесенными изменениями. В качестве аргумента для функции digitalWrite() выступает функция digitalRead(). Хороший стиль – это именно такой вариант. Не требуются дополнительные переменные, меньше текст. Т.е. функцию можно использовать как аргумент другой функции. Функции можно вызывать из функций.

### **4. Подключение кнопки с использованием условного оператора if.**

Другой вариант этой же программы, использующий условный оператор if (также необходимо изменить код и проверить функциональность программы):

```

// C++ code
// Включение встроенного светодиода при нажатии кнопки
boolean buttonState; //– создание глобальной переменной buttonState
void setup() {
  pinMode(13, OUTPUT); //– определение порта 13 как выхода для светодиода
  pinMode(2, INPUT_PULLUP); //– определение порта 2 как входа для кнопки
}
void loop() {
  if ( digitalRead(2) == LOW ) digitalWrite(13, HIGH);
  else digitalWrite(13, LOW);
}

```

В цикле loop() проверяется состояние вывода 2 (кнопка), и если оно низкое (LOW), то на выводе 13 (светодиод) формируется высокое состояние (HIGH). В противном случае состояние светодиода низкое (LOW).

### **5. Подключение кнопки с использованием директивы #define.**

Во всех примерах для функций ввода/вывода мы указывали аргумент pin, определяющий номер вывода, в виде конкретного числа – константы. Мы помнили, что константа 2 это номер вывода кнопки, а 13 – номер вывода светодиода. Гораздо удобнее работать с символьными именами. Для этого в языке C существует директива, связывающая идентификаторы с константами, выражениями.

Директива #define определяет идентификатор и последовательность символов, которая подставляется вместо идентификатора, каждый раз, когда он встречается в тексте программы.

В общем виде она выглядит так:

*#define* имя последовательность\_символов

Если в наших программах мы напишем:

#define LED\_PIN 13 //– номер вывода светодиода равен 13

то каждый раз, когда в программе встретится имя LED\_PIN, при трансляции вместо него будет подставлены символы 13. Функция включения светодиода выглядит так:

digitalWrite(LED\_PIN, HIGH);

Окончательный вариант программы с использованием #define:

// C++ code

// Включение встроенного светодиода при нажатии кнопки

#define LED\_PIN 13 //– номер вывода светодиода равен 13

#define BUTTON\_PIN 2 //– номер вывода кнопки равен 2

void setup() {

pinMode(LED\_PIN, OUTPUT); // определение порта 13 (LED\_PIN) как выхода для светодиода

pinMode(BUTTON\_PIN, INPUT\_PULLUP); // определение порта 2 (BUTTON\_PIN) как входа для кнопки

}

void loop() {

digitalWrite(LED\_PIN, ! digitalRead(BUTTON\_PIN) ); // установка состояния порта 13 (LED\_PIN) в соответствии с инвертированным значением состояния порта 2 (BUTTON\_PIN)

}

Необходимо запомнить, что директива #define не выполняет операций, она только задает постоянные значения. Т.к. это псевдооператор, то после данной директивы точку с запятой ставить не нужно. Обычно имена для директивы define задают в верхнем регистре с использованием разделителя (нижнее подчеркивание).

## **6. Подключение кнопки в режиме INPUT.**

Рассмотрим еще один способ подключения кнопки к плате Arduino (режим INPUT) и возможности обработки сигнала нажатия на кнопку.

Кнопка установлена так, что каждый ее контакт подключен к разным линиям (+ и -) макетной платы. Также подключен резистор на 10 кОм, который притягивает контакт к земле. Это нужно для того, чтобы на 8 контакте не было наводок. Если убрать этот резистор из схемы, то светодиод будет загораться при шевелении проводов или кнопки (в режиме INPUT).

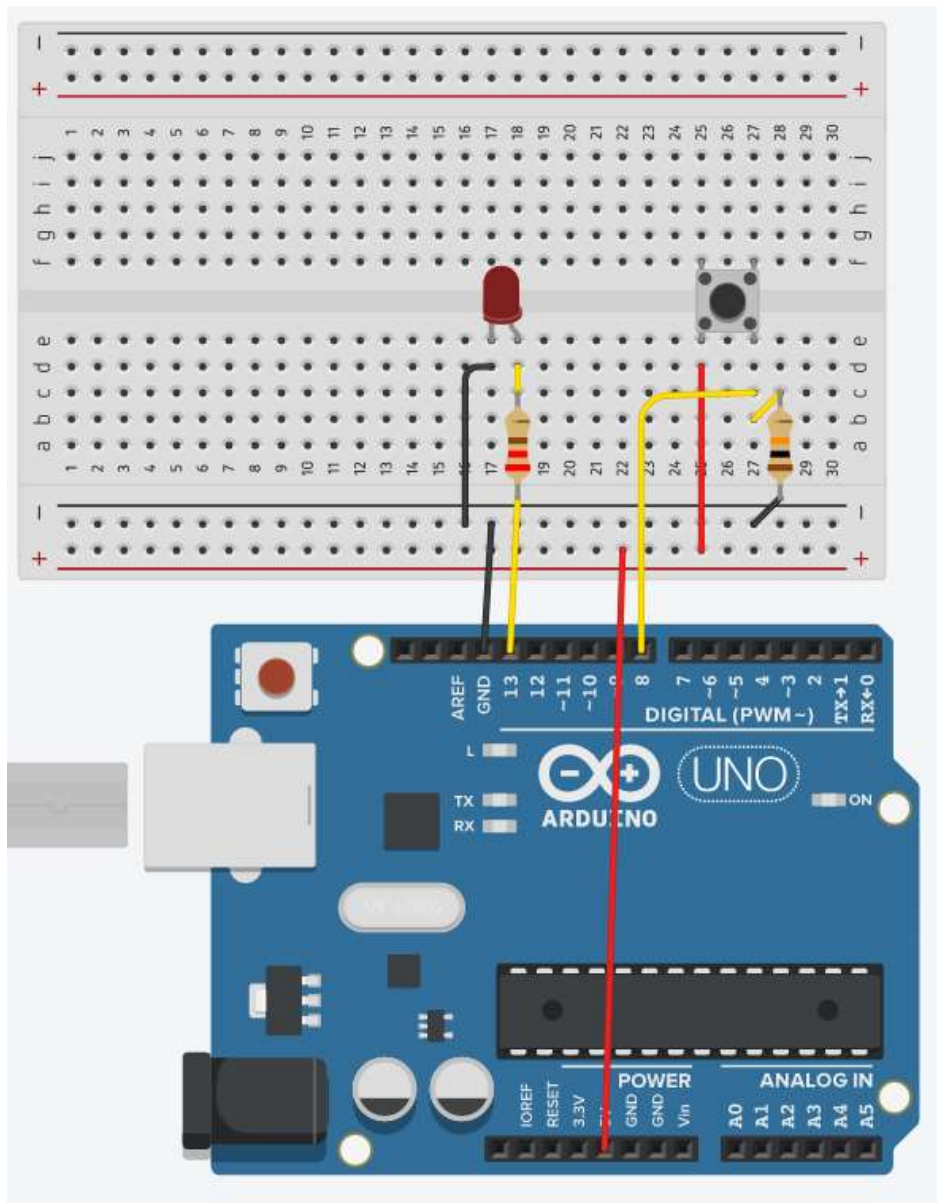


Рис. 115. Сборка схемы с подключением кнопки в режиме Input  
Внесите следующий код в среду разработки:

```
// C++ code
// Включение встроенного светодиода при нажатии кнопки
int switchPin = 8; // задание переменной для порта 8 (кнопка)
int ledPin = 13; // задание переменной для порта 13 (встроенный светодиод)
void setup() {
  pinMode(switchPin, INPUT); // определение порта 2 (switchPin) как входа для кнопки
  pinMode(ledPin, OUTPUT); // определение порта 13 (ledPin) как выхода для
  светодиода}
void loop() {
  // включение встроенного светодиода при нажатии кнопки
  if(digitalRead(switchPin) == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

}

В начале программы были объявлены переменные с номерами портов кнопки и светодиода. Затем функция `setup()` задает, как используется порт – в качестве входа или выхода. В цикле `loop()` с помощью оператора ветвления `if` проверяется условие – подается ли на порт 8 высокий уровень сигнала. В случае, если да, то светодиод включается. В противном случае светодиод выключается.

#### **7. Подключение кнопки в режиме INPUT с переключением состояния светодиода.**

Далее усложним код. Сделаем так, чтобы каждое нажатие кнопки переключало состояние светодиода, т.е. при нажатии на кнопку светодиод загорался и гас только при следующем нажатии. Схема остается прежней, программу необходимо изменить следующим образом:

```
// C++ code
// Переключение состояния встроенного светодиода при нажатии кнопки
int switchPin = 8; // задание переменной для порта 8 (кнопка)
int ledPin = 13; // задание переменной для порта 13 (встроенный светодиод)
// задание переменных для хранения состояния кнопки и светодиода
boolean lastButton = LOW;
boolean currentButton = LOW;
boolean ledOn = false;
void setup() {
  pinMode(switchPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
// функция игнорирования шума и подавлениядребезга контактов
boolean debounce(boolean last) {
  boolean current = digitalRead(switchPin);
  if(last != current) {
    delay(5);
    current = digitalRead(switchPin);
  }
  return current;
}
void loop() {
  currentButton = debounce(lastButton);
  if(lastButton == LOW && currentButton == HIGH) {
    ledOn = !ledOn;
  }
  lastButton = currentButton;
  digitalWrite(ledPin, ledOn);
}
```

Здесь добавлены переменные для хранения состояния светодиода и кнопки. Так же мы создали новую функцию для подавлениядребезга контактов `debounce()`. Код в цикле `loop()` тоже немного изменился. Теперь в условном операторе мы проверяем нажата ли кнопка и если нажата, то меняем состояние светодиода на противоположное. Потом меняем переменную с последним состоянием на текущее состояние кнопки и включаем или выключаем светодиод.

#### **8. Понятие о широтно-импульсной модуляции.**

Часто в робототехнике возникает необходимость плавно управлять каким-то процессом, будь то яркость светодиода, мощность обогревателя или скорость вращения моторчика.

Очевидно, что управление напрямую связано с изменением напряжения на потребителе: и светодиод будет по-другому светить, и моторчик крутиться с другой скоростью.

Но управлять напряжением может только ЦАП – цифро-аналоговый преобразователь, а в нашем микроконтроллере встроенного ЦАП нет, у нас есть только цифровой сигнал, т.е. он либо высокий уровень (1), либо низкий (0).

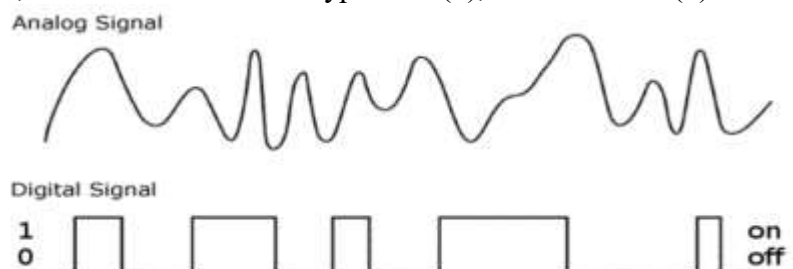


Рис. 116. Аналоговый и цифровой сигналы

Можно ли добиться плавного управления цифровым сигналом? Представьте себе вентилятор, вращающийся на полной мощности, напряжение постоянно. Представим теперь, что секунду напряжение подаётся, и секунду – нет, и так продолжается «по кругу». Вентилятор начнёт крутиться в два раза медленнее, но мы скорее всего будем замечать моменты включения и выключения, особенно если вентилятор маленький. Большой вентилятор более инертен и там можно даже не заметить изменений скорости в пределах двух секунд. Можно теперь включать напряжение на 0.5 секунды, а на остальные 1.5 секунды – выключать. Вентилятор будет крутиться со скоростью 25% от максимальной. Мы с вами смогли представить так называемый ШИМ сигнал, широтно-импульсную модуляцию.

Представим себе электрический импульс А с амплитудой напряжения от 0В до 5В, длиной 1 мс и периодом повторения 10мс (т.е. в течение 1 мс напряжение на линии, по которой проходит импульс А составляет 5В, затем в течение 9мс составляет 0В, и так повторяется каждые 10 мс). Представим, что мы увеличиваем длительность импульса А до 2мс (пусть теперь это будет импульс Б), а повторяется он точно те же каждые 10мс. Задача изменения длительности импульса А с 1мс до импульса Б 2 мс и есть задача ШИМ.

Модуляция означает изменение параметров колебания (частоты, амплитуды, фазы). Широтно-импульсная модуляция – изменение скважности импульсов при постоянной частоте. Скважность – то же самое, что протяженность, т.е. в нашем примере это изменение протяженности импульса с 1мс до 2мс. На графике изображены три сигнала. Сигналы модулируются ШИМом, который генерирует и регулирует скважность импульсов.

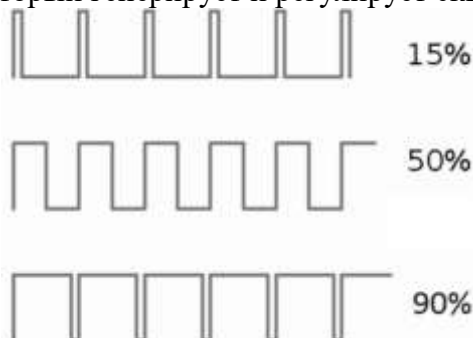


Рис. 117. Широтно-импульсная модуляция с регулировкой скважности

Скважность на графике сверху –15%. То есть, за один период, равный 100%, 15% времени выдается логическая единица (напряжение +3В либо +5В). 75% времени выдается

логический ноль (отсутствие напряжения в линии – 0В). На среднем графике скважность 50% – 50% времени выдается логическая 1, 50% выдается логический 0. На графике снизу скважность 90%. 90% – 1. 10% – 0.

Если подключить светодиод к ШИМ регулятору, то в случае с первым графиком светодиод будет светиться слабо.

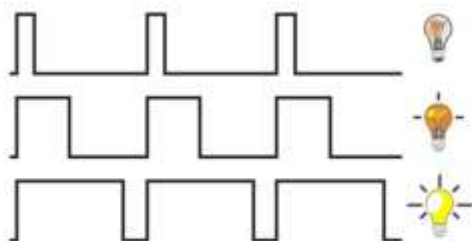


Рис. 118. Изменение яркости светодиода в зависимости от скважности

С графиком 2 свечение светодиода будет ярче чем с 1, но сам светодиод будет светиться на 50% своей мощности. В случае с 3-им графиком, яркость свечения светодиода будет настроена на 90% и будет близка к максимальной.

Широтно-импульсная модуляция позволяет превратить цифровые линии ввода/вывода общего назначения в аналоговые выходы Arduino.

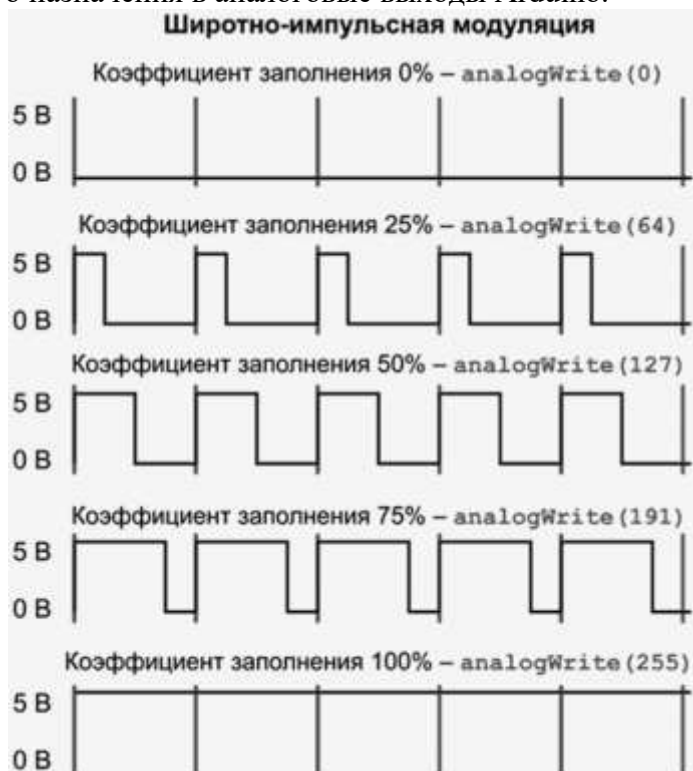


Рис. 119. ШИМ с различными значениями скважности

Таким образом, микроконтроллеры обычно не могут выдавать произвольное напряжение. Они могут выдать либо напряжение питания (например, 5 В), либо землю (т.е. 0 В). Но уровнем напряжения управляет многое: например, яркость светодиода или скорость вращения мотора. Для симуляции неполного напряжения и используется ШИМ (англ. Pulse Width Modulation или PWM).

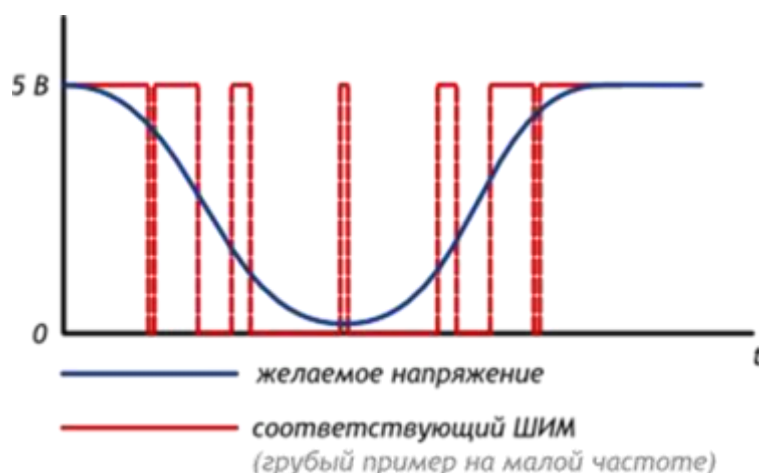


Рис. 120. Связь скважности ШИМ и желаемого напряжения

Выход микроконтроллера переключается между землёй и источником питания тысячи раз в секунду. Или, как ещё говорят, имеет частоту в тысячи герц. Глаз не замечает мерцания более 50 Гц, поэтому нам кажется, что светодиод не мерцает, а горит в полсилы. Аналогично, разогнанный мотор не может остановить вал за миллисекунды, поэтому ШИМ-сигнал заставит вращаться его в неполную силу.

В Arduino Uno есть 6 пинов для работы с ШИМ: 3, 5, 6 и 9–11, что позволяет выводить аналоговое напряжение в виде ШИМ-сигнала из цифровых значений. Для генерации аналогового напряжения на одном из выводов ШИМ платы Arduino требуется конфигурирование этого вывода с помощью команды `pinMode(pin, mode)` и последующего вызова функции `analogWrite(pin, value)`, где

`pin` – указывает номер вывода ШИМ на разъеме платы Arduino,

`value` – значение, задающее выходное напряжение ШИМ (8-битное число в диапазоне 0-255).

На сконфигурированном однажды выходе будет постоянно присутствовать сигнал ШИМ частотой 490 Гц с заданными параметрами до следующего вызова функции `analogWrite()` (или `digitalRead()`, или `digitalWrite()`) для этого же вывода микроконтроллера.

## 9. Подключение кнопки в режиме INPUT для управления яркостью светодиода.

Далее с использованием ШИМ будем управлять яркостью светодиода. Используем в качестве основы предыдущий проект, но теперь необходимо подключить светодиод к выходу, который поддерживает ШИМ.

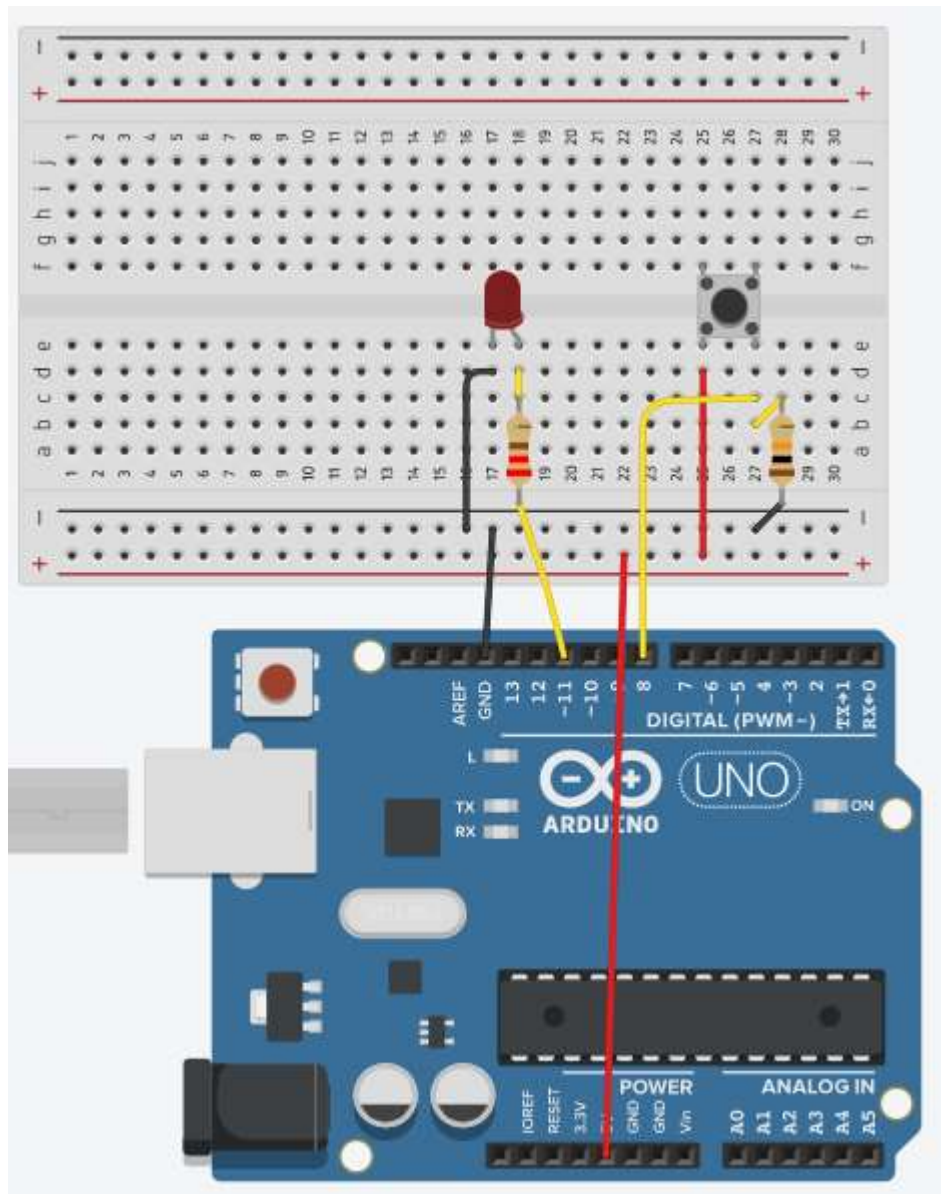


Рис. 121. Сборка схемы для регулирования уровня яркости светодиода

Для подключения светодиода был взят порт номер 11. Он предназначен в том числе для выдачи ШИМ-сигнала. Со светодиодом используется резистор номиналом 220 Ом.

```
// C++ code
// Изменение яркости светодиода при нажатии кнопки
int switchPin = 8; // задание переменной для порта 8 (кнопка)
int ledPin = 11; // задание переменной для порта 11 (светодиод)
// задание переменных для хранения состояния кнопки и светодиода
boolean lastButton = LOW;
boolean currentButton = LOW;
int ledLevel = 0;
void setup() {
  pinMode(switchPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
// функция игнорирования шума и подавления дребезга контактов
boolean debounce(boolean last) {
  boolean current = digitalRead(switchPin);
```

```

if(last != current) {
  delay(5);
  current = digitalRead(switchPin);
}
return current;
}
void loop() {
  currentButton = debounce(lastButton);
  if(lastButton == LOW && currentButton == HIGH) {
    ledLevel = ledLevel + 51;
  }
  lastButton = currentButton;
  if(ledLevel > 255) ledLevel = 0;
  analogWrite(ledPin, ledLevel);
}

```

В данной программе переменная ledPin равна 11. Переменная ledLevel предназначена для хранения значения ШИМ-сигнала. Каждый раз при нажатии кнопки значение переменной ledLevel будет изменяться на 51 единицу. Функция для подавления дребезга контактов debounce() не была изменена. В цикле используется функция analogWrite().

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

### Задания

1. Изменить программу для управления яркостью светодиода так, чтобы светодиод работал на 3-х уровнях яркости.
2. Создать программу управления включением/выключением 2-х светодиодов с помощью 2-х соответствующих кнопок.

### Указания по порядку выполнения работы

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

### Содержание отчета

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Опишите назначение функции `pinMode`. Какие аргументы необходимы для задания данной функции?
2. Опишите назначение функций `digitalWrite` и `digitalRead`. Какие аргументы необходимы для задания данных функций?
3. Опишите назначение директивы `#define`.
4. Когда и почему возникает короткое замыкание в электрической цепи, в чем его опасность?
5. Что такое подтягивающий резистор, и зачем он нужен?
6. Как избавиться от дребезга контактов программным способом?

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

### **Лабораторная работа 9**

#### **Аналоговые входы платы Arduino и резистивные датчики**

**Цель работы:** исследование возможностей работы аналоговых входов платы Arduino на примере подключения и программирования резистивных датчиков (потенциометра и фоторезистора).

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

## Теоретическая часть

### Содержание:

1. Аналоговые входы и аналого-цифровой преобразователь платы Arduino.
2. Резистивные датчики.
3. Подключение потенциометра и считывание значений датчика с использованием библиотеки Serial.
4. Управление яркостью светодиода с помощью потенциометра.
5. Фоторезистор.
6. Светильник с автоматическим включением.
7. Светильник с изменяющейся яркостью.

### 1. Аналоговые входы и аналого-цифровой преобразователь платы Arduino.

Плата Arduino UNO содержит 6 аналоговых входов, предназначенных для измерения напряжения сигналов. Данные выводы платы могут работать в режиме, как дискретных выводов, так и аналоговых входов. Эти выводы имеют номера от 14 до 19. Изначально они настроены как аналоговые входы, и обращение к ним можно производить через имена A0-A5. В любой момент их можно настроить на режим дискретных выходов.

```
pinMode(A3, OUTPUT); // установка режима дискретного вывода для A3  
digitalWrite(A3, LOW); // установка низкого состояния на выходе A3
```

Чтобы вернуть в режим аналогового входа:

```
pinMode(A3, INPUT); // установка режима аналогового входа для A3
```

Измерение напряжения на входах производится *аналого-цифровым преобразователем (АЦП)* с коммутатором на 6 каналов. АЦП имеет разрешение 10 бит, что соответствует коду на выходе преобразователя 0-1023. Погрешность измерения не более 2 единиц младшего разряда.

Для сохранения максимальной точности (10 разрядов) необходимо, чтобы внутреннее сопротивление источника сигнала не превышало 10 кОм. Это требование особенно важно при использовании резисторных делителей, подключенных к аналоговым входам платы. Сопротивление резисторов делителей не может быть слишком большим.

Программные функции аналогового ввода:

Функция ***analogRead()***

```
int analogRead(port)
```

считывает значение напряжения на указанном аналоговом входе. Входное напряжение диапазона от 0 до уровня источника опорного напряжения (часто 5В) преобразовывается в код от 0 до 1023.

Если аналоговый вход не подключен, то значения возвращаемые функцией `analogRead()` могут принимать случайные значения.

### 2. Резистивные датчики.

Последовательно подключённые резисторы делят поступающее на них напряжение в определённой пропорции.

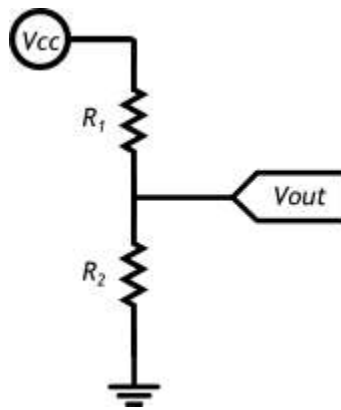


Рис. 122. Последовательное подключение резисторов

Расчёт пропорции производится следующим образом. Сила тока, протекающая через резисторы одинакова, т.к. они соединены последовательно, и по закону Ома может быть рассчитана как:

$$I = \frac{V_{CC}}{R_1 + R_2}$$

По тому же закону Ома можно вычислить напряжение  $V_{out}$ , которое падает на резисторе  $R_2$ :

$$V_{out} = U_2 = I \times R_2 = \frac{R_2 \cdot V_{CC}}{R_1 + R_2}$$

Из полученной формулы видно, что чем больше  $R_2$  относительно  $R_1$ , тем большее напряжение падает на нём.

Если вместо  $R_2$  использовать не постоянный резистор, а датчик, который меняет своё сопротивление,  $V_{out}$  будет зависеть от измеряемого значения.

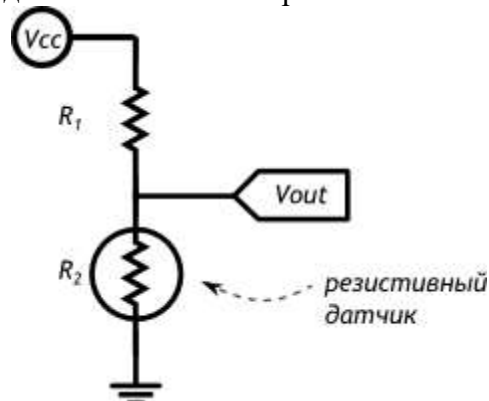


Рис. 123. Подключение резистивного датчика

Микроконтроллер умеет измерять напряжение. Таким образом, мы можем использовать свойства делителя напряжения для получения показаний от сенсора.

Примеры резистивных датчиков:

#### 1. Термистор.

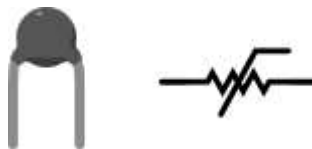


Рис. 124. Термистор

Термистор изменяет своё сопротивление в зависимости от собственной температуры.

#### 2. Фоторезистор.

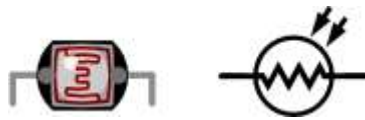


Рис. 125. Фоторезистор

Фоторезистор (англ. Light Dependent Resistor или сокращённо LDR) изменяет своё сопротивление в зависимости от силы света, попадающего на его керамическую «змейку».

### 3. Потенциометр.

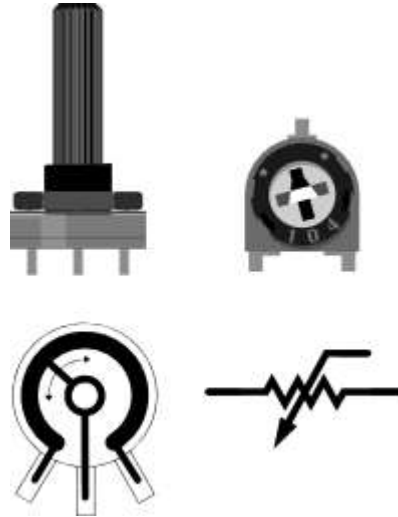


Рис. 126. Потенциометр

Потенциометр ещё называют переменным резистором, триммером. Это делитель из двух резисторов в одном корпусе. Поэтому у него 3 ноги: питание, выход, земля. На выводы потенциометра подается напряжение (5V и GND), а средний вывод радиоэлемента является подвижным контактом. При вращении ручки потенциометра, напряжение сигнала на среднем выводе будет меняться от своего максимального значения (5 Вольт) до нуля.

Таким образом, соотношение R1 и R2 меняется поворотом ручки. От 100% в пользу R1 до 100% в пользу R2.

### 3. Подключение потенциометра и считывание значений датчика с использованием библиотеки Serial.

*Потенциометр Ардуино* служит для регулировки или настройки различных параметров в электрической цепи – мощности, напряжения, громкости звука и т.д. Рассмотрим, как подключить переменный резистор к Ардуино правильно, и вывести значения порта, к которому подключен потенциометр, с использованием библиотеки Serial.

Соберите предложенную ниже схему и напишите код:

```
// C++ code
int analogPin = 0; // определение порта потенциометра
int val = 0; // хранение значения, считываемого с потенциометра
void setup()
{
  Serial.begin(9600) // соединение контроллера с внешними устройствами
  // (компьютером) с помощью библиотеки Serial на скорости 9600 бод или бит/сек
}
Void loop()
{
  val = analogRead(analogPin); // считывание значения порта потенциометра
  Serial.println(val); // вывод на экран компьютера значения порта потенциометра
}
```

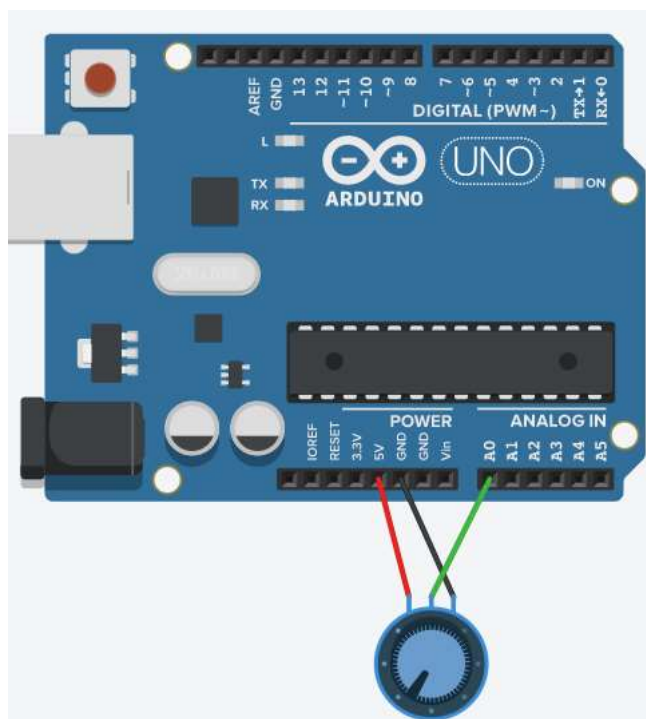


Рис. 127. Сборка схемы с потенциометром

#### 4. Управление яркостью светодиода с помощью потенциометра.

Далее представим примеры программ для регулировки яркости светодиода. Для регулировки яркости светодиода с помощью переменного резистора, следует считывать данные с данного радиоэлемента, подключив его к аналоговому входу. В зависимости от поворота ручки потенциометра необходимо в линейной зависимости менять яркость светодиода.

*Компоненты, необходимые для реализации проекта:*

- 1 плата Arduino
- 1 беспаячная макетная плата
- 1 светодиод
- 1 резистор номиналом 220 Ом
- 1 потенциометр
- провода

*Принципиальная схема выглядит следующим образом:*

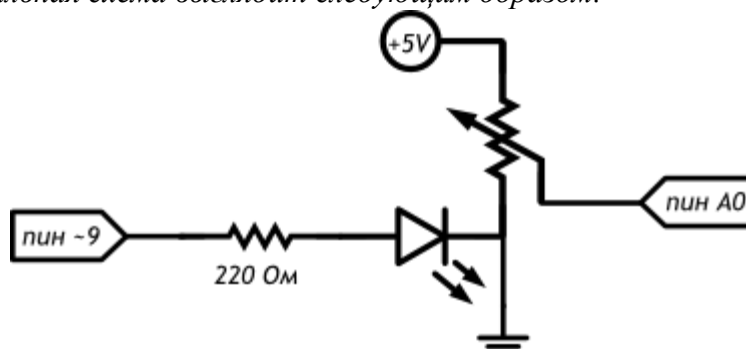


Рис. 128. Принципиальная схема подключения потенциометром для управления яркостью светодиода

*Ниже представлена схема сборки данного устройства:*

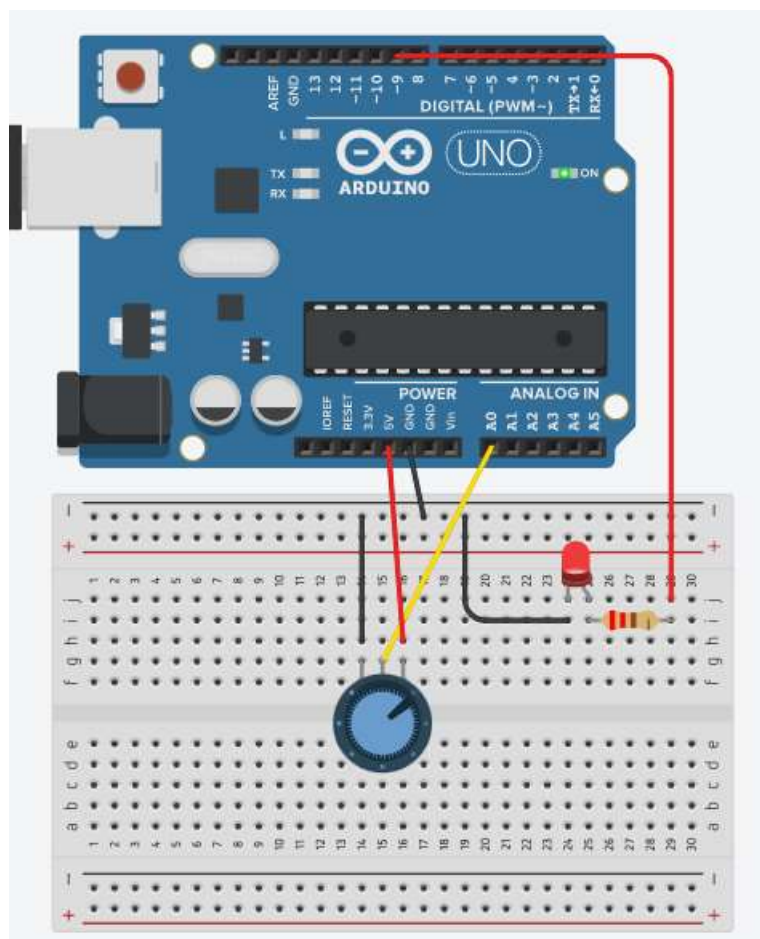


Рис. 129. Схема сборки с потенциометром для управления яркостью светодиода

Чтобы применять меньше проводов и сделать схему более удобной, можно использовать соединение к длинной рельсе «-» макетки («-» и «+» платы условны, значение зависит от подключения, обозначения используются для удобства). В нашем случае светодиод и потенциометр подсоединены к «-» макетной платы, а к рельсе, в свою очередь, подведена «земля» от микроконтроллера.

Не важно, какая из крайних ножек потенциометра будет подключена к 5 В, а какая к GND, поменяется только направление, в котором нужно крутить ручку для увеличения напряжения. Запомните, что сигнал мы считываем со средней ножки.

Итак, для считывания аналогового сигнала, принимающего широкий спектр значений, а не просто 0 или 1, как цифровой, подходят только порты, помеченные на плате как «ANALOG IN» и пронумерованные с префиксом А. Для Arduino Uno – это А0-А5.

*Введите следующий код в среду разработки:*

```
// C++ code
#define LED_PIN 9 // определение порта светодиода
#define POT_PIN A0 // определение порта потенциометра
void setup()
{
  Serial.begin(9600); // соединение контроллера с внешними устройствами
  // (компьютером) с помощью библиотеки Serial на скорости 9600 бод или бит/сек
  pinMode(LED_PIN, OUTPUT); // определение порта светодиода как выхода
  pinMode(POT_PIN, INPUT); // определение порта потенциометра как входа
}
void loop()
{
```

```

int rotation, brightness; // переменные для хранения значений
// угла поворота потенциометра (в диапазоне от 0 до 1023)
// и яркости светодиода (в диапазоне от 0 до 255)
rotation = analogRead(POT_PIN); // присваивание переменной rotation значения порта
потенциометра
brightness = rotation / 4; // перевод значения в диапазоне от 0 до 1023 в диапазон
// от 0 до 255 для возможности передачи порту 9, поддерживающему ШИМ
// и присваивание переменной brightness полученного значения
analogWrite(LED_PIN, brightness); // задание яркости светодиода в зависимости от
brightness
Serial.println(brightness); // вывод значения brightness через последовательный
интерфейс
}

```

С помощью директивы `#define` мы сказали компилятору заменять идентификатор `POT_PIN` на `A0` – номер аналогового входа. Вы можете встретить код, где обращение к аналоговому порту будет по номеру без индекса `A`. Такой код будет работать, но во избежание путаницы с цифровыми портами используйте индекс.

Переменным принято давать названия, начинающиеся со строчной буквы.

Чтобы использовать переменную, необходимо ее объявить, что мы и делаем инструкцией:

```
int rotation, brightness;
```

Для объявления переменной необходимо указать ее тип, здесь – `int` (от англ. *integer*) – целочисленное значение в диапазоне от -32 768 до 32 767

Переменные одного типа можно объявить в одной инструкции, перечислив их через запятую, что мы и сделали

Функция `analogRead(pinA)` возвращает целочисленное значение в диапазоне от 0 до 1023, пропорциональное напряжению, поданному на аналоговый вход, номер которого мы передаем функции в качестве параметра `pinA`

Обратите внимание, как мы получили значение, возвращенное функцией `analogRead()`: мы просто поместили его в переменную `rotation` с помощью оператора присваивания `=`, который записывает то, что находится справа от него в ту переменную, которая стоит слева.

## 5. Фоторезистор.

**Фоторезистор** (англ. *Light Dependent Resistor* или сокращённо *LDR*) – это один из видов резистивных датчиков, который изменяет своё сопротивление в зависимости от силы света, попадающего на его керамическую «змейку».

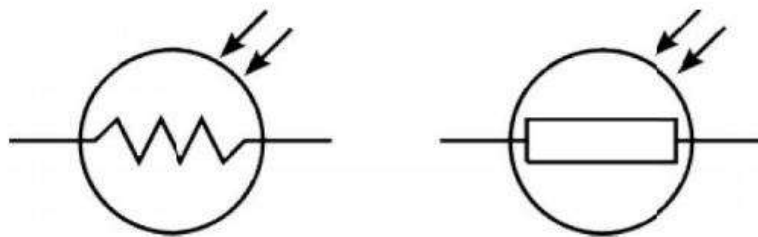


Рис. 130. Схемное обозначение фоторезистора

У некоторых видов данного полупроводникового прибора под воздействием света электрическое сопротивление может снижаться на два-три порядка. Основной частью

фоторезисторов является полупроводниковый элемент (например, сернистый свинец или сернистый кадмий), расположенный так, чтобы на него попадал свет.

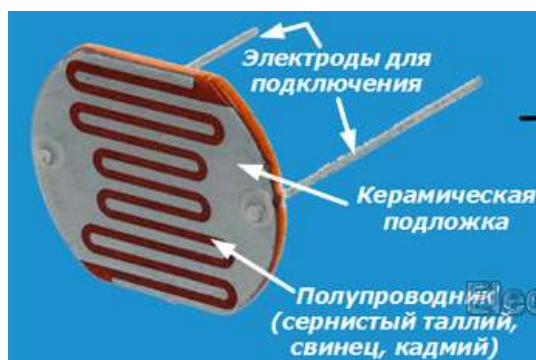


Рис. 131. Устройство фоторезистора

Явлением **фотопроводимости** называется увеличение электропроводности полупроводника под воздействием электромагнитного излучения. Принцип работы фоторезистора основан на появлении подвижных носителей заряда (электронов) в результате поглощения полупроводником световой энергии, вследствие чего уменьшается его сопротивление, т.е. возникает дополнительная электропроводность. Получая от фоторезистора информацию, плата может включать или выключать реле, запускать двигатели, отсылать сообщения. Естественно, при этом мы должны правильно подключить датчик.

Устройство и принцип действия фоторезистора максимально просты, поэтому данные полупроводниковые приборы сегодня широко применяются во многих отраслях науки и техники. Это объясняется высокой чувствительностью фоторезисторов, малыми габаритами и простотой конструкции приборов, долговечностью в работе, а также возможностью обеспечить бесконтактные измерения и контроль.

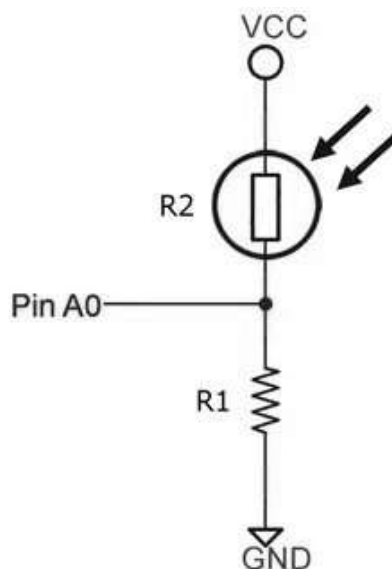


Рис. 132. Принципиальная схема подключения фоторезистора

Основным недостатком фоторезисторов является чувствительность к спектру. В зависимости от типа падающего света сопротивление может меняться на несколько порядков. К минусам также относится низкая скорость реакции на изменение освещённости. Если свет мигает – датчик не успевает отреагировать. Если же частота изменения довольно велика – резистор вообще перестанет «видеть», что освещённость меняется.

## 6. Светильник с автоматическим включением.

В следующем проекте рассматривается подключение фоторезистора к плате Arduino. В зависимости от показаний фоторезистора, т.е. в зависимости от освещения, будет происходить автоматическое включение и выключение светодиода либо будет изменяться яркость светодиода. Также в данной работе будет рассмотрен оператор ветвления *if* ... *else* и его реализация на языке C++.

*Компоненты, необходимые для реализации проекта:*

- плата Arduino Uno;
- макетная плата;
- 1 фоторезистор;
- 1 светодиод;
- 1 резистор 220 Ом;
- 1 резистор 880 Ом;
- провода.

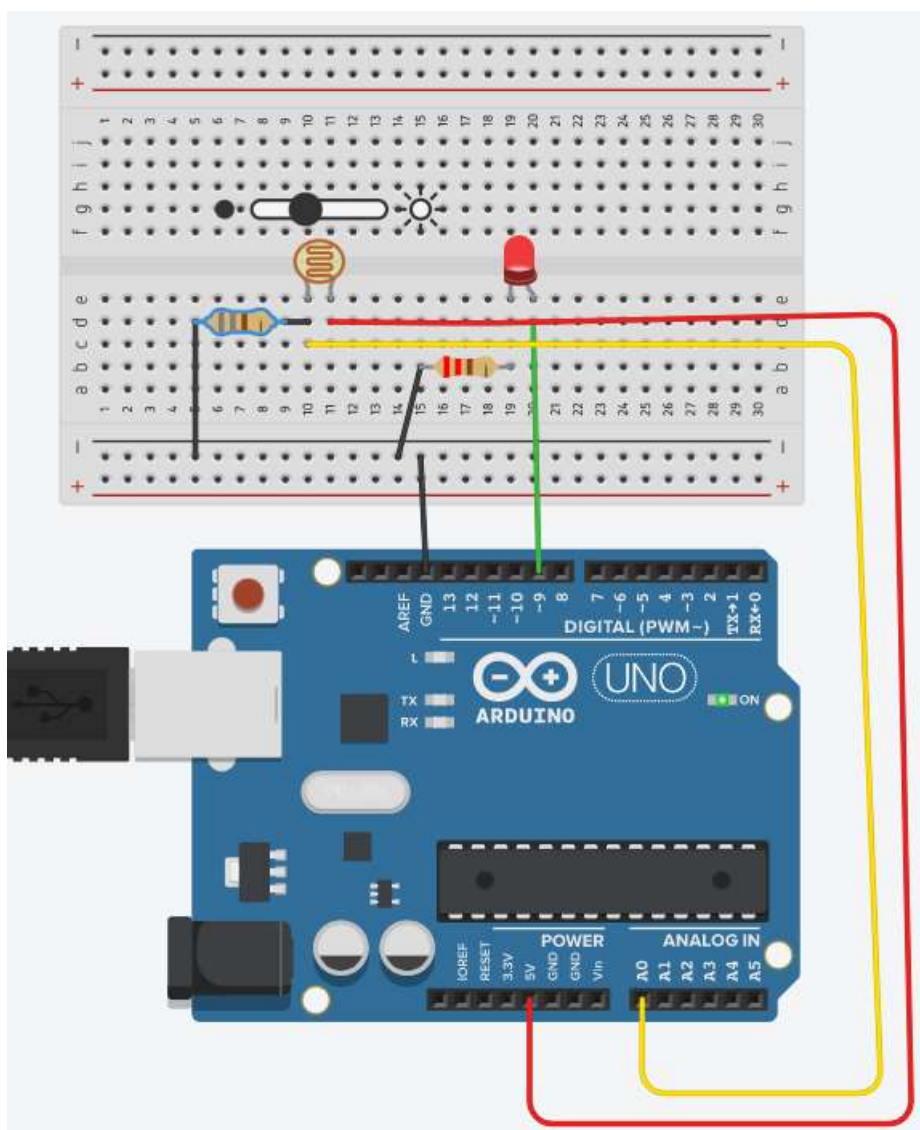


Рис. 133. Схема подключения фоторезистора на макетной плате

Фоторезистор будет играть роль переменного сопротивления для включения и выключения светильника. Принцип работы схемы в том, что в электрической цепи будет меняться сопротивление, в зависимости от освещенности в помещении, а значит, будут меняться данные на аналоговом входе A0. После сборки схемы с фоторезистором, создайте следующую программу:

```

// C++ code
#define SENSOR A0 // определение номера порта фоторезистора
#define LED 9 // определение номера порта светодиода
unsigned int value = 0; // переменная для хранения цифрового значения аналогового
порта A0
// тип данных unsigned int округляет значения и принимает только положительные
числа
void setup() {
  pinMode(LED, OUTPUT); // определение порта светодиода как выхода
  pinMode(SENSOR, INPUT); // определение порта потенциометра как входа
}
void loop() {
  value = analogRead(SENSOR); // присваивание переменной value значения порта
потенциометра
  if (value<500) digitalWrite(LED, HIGH); // если value <500, включаем светодиод
  if (value>500) digitalWrite(LED, LOW); // если value>500, выключаем светодиод
}

```

В первой строчке программы имя SENSOR присваивается для аналогового входа A0 с помощью функции **#define**.

Оператор **unsigned int** указывает, что значение value может принимать только положительное целое число, а начальное значение value равно нулю.

Оператор ветвления if задает команды, которые будут выполняться, если условие истинно. После оператора else следуют команды, которые будут выполняться в случае ложности условия.

### 7. Светильник с изменяющейся яркостью.

Для изменения яркости светодиода в зависимости от освещения и сборки «умного» светильника необходимо схему оставить прежней, а в коде записать:

```

// C++ code
#define SENSOR A0 // определение номера порта фоторезистора
#define LED 9 // определение номера порта светодиода
unsigned int value = 0; // переменная для хранения цифрового значения аналогового
порта A0
// тип данных unsigned int округляет значения и принимает только положительные
числа
int data; // переменная для хранения значения яркости светодиода (порт 9)
void setup() {
  pinMode(LED, OUTPUT); // определение порта потенциометра как входа
  pinMode(SENSOR, INPUT); // определение порта светодиода как выхода
  Serial.begin(9600); // соединение контроллера с внешними устройствами
  // (компьютером) с помощью библиотеки Serial на скорости 9600 бод или бит/сек
}
void loop() {
  value = analogRead(SENSOR); // присваивание переменной value значения порта
потенциометра
  data = 255-value / 4; // для перевода значения value из диапазона 0...1023
  // в диапазон 0...255 делим значение value на 4
  // для того, чтобы яркость светодиода была обратно пропорциональна освещенности
  // производим вычитание из 255 полученного значения
  analogWrite(LED, data); // задание порту 9 (LED) значения яркости светодиода
}

```

```
Serial.println(data); // вывод значения data (яркости) через последовательный
интерфейс
}
```

В данной программе пин 9 использует ШИМ.

Меняя значение резистора для подключения фоторезистора, можно «сдвигать» уровень чувствительности в «темную» или «светлую» сторону. Так, 10 кОм даст быстрое переключение наступления света. В случае 1кОм датчик света будет более точно определять высокий уровень освещенности.

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

### Задания

1. В проекте «Управление яркостью светодиода с помощью потенциометра» измените номер порта светодиода – 9 на 11. Произведите необходимые изменения в программе так, чтобы схема работала.

2. Создайте на основе схемы «Управление яркостью светодиода с помощью потенциометра» еще один проект. Для этого добавьте к схеме устройства второй светодиод, резистор на 220 Ом и соберите аналогичную схему на этой же макетной плате, подключив второй светодиод к пину номер 3 и другому входу «земля». Измените программу так, чтобы светодиоды мигали в противофазу: первый выключен, второй горит максимально ярко и до противоположного состояния.

3. Основываясь на схему проекта «Управление яркостью светодиода с помощью потенциометра», подключите к порту X еще один светодиод. Измените код таким образом, чтобы второй светодиод светился на 1/Y от яркости первого

*Варианты:*

| Номер варианта | X  | Y  |
|----------------|----|----|
| 1.             | 3  | 2  |
| 2.             | 5  | 3  |
| 3.             | 6  | 4  |
| 4.             | 10 | 5  |
| 5.             | 11 | 6  |
| 6.             | 3  | 7  |
| 7.             | 5  | 8  |
| 8.             | 6  | 9  |
| 9.             | 10 | 10 |
| 10.            | 11 | 11 |
| 11.            | 3  | 12 |
| 12.            | 5  | 13 |
| 13.            | 6  | 14 |

4. В схеме проекта «Светильник с изменяющейся яркостью» измените номинал резистора для подключения фоторезистора. Сделайте выводы об изменении работы

программы. Составьте таблицу зависимости значения резистора и значения переменной data.

5. Возьмите 3 светодиода: зеленый, желтый, красный. Соберите схему и напишите программу, которая решает следующую задачу: при освещенности в диапазоне 0-400 светится красный светодиод, 401-800 желтый, 801-1023 зеленый.

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.

2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.

3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.

4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.

5. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Можем ли мы при сборке схемы подключить светодиод и потенциометр напрямую к разным входам GND микроконтроллера?

2. В какую сторону нужно крутить переменный резистор для увеличения яркости светодиода?

3. Зачем мы делим значение, полученное с аналогового входа перед тем, как задать яркость светодиода? Что будет, если этого не сделать?

4. Что такое фоторезистор?

5. Объясните явление фотопроводимости.

6. Опишите тип данных unsigned int.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

## Лабораторная работа 10

### Проектирование устройств с датчиком температуры на базе Arduino

**Цель работы:** создание проектов Arduino с использованием датчика температуры, программирование работы датчика температуры.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

*Содержание:*

1. Общая характеристика термодатчиков.
2. Подключение и программирование датчика температуры.
3. Индикация значения температуры с помощью RGB-светодиода.

#### **1. Общая характеристика термодатчиков.**

Существуют различные интегральные датчики температуры с аналоговым выходом по напряжению.

При всем разнообразии этих устройств, им присущи следующие общие качества:

- напряжение на выходе линейно пропорционально температуре;
- датчики имеют калиброванный масштабный коэффициент зависимости выходного напряжения от температуры, дополнительной калибровки не требуют.

Т.е. для измерения температуры с помощью датчиков такого типа необходимо измерить напряжение на выходе и через масштабный коэффициент пересчитать его в температуру.

Существует множество термодатчиков, которые можно отнести к этой категории: LM35; TMP35; TMP36; TMP37.

Принципиальные отличия перечисленных датчиков друг от друга заключается в том, что:

- TMP36 – единственный из перечисленных термодатчиков, способен измерять отрицательную температуру.
- датчики имеют различные диапазоны измерения температуры.

*Замечание:* Есть схема включения LM35, позволяющая измерять отрицательные температуры. Но она сложнее в реализации и требует дополнительного питания. Лучше для отрицательных температур использовать TMP36.

Основные параметры термодатчиков LM35, TMP35, TMP36, TMP37 представлены в таблице:

| Тип           | Диапазон измерения температуры, °C | Смещение напряжения на выходе, мВ | Масштабный коэффициент, мВ/°C | Напряжение на выходе при +25 °C, мВ |
|---------------|------------------------------------|-----------------------------------|-------------------------------|-------------------------------------|
| LM35, LM35A   | 0 ... +150                         | 0                                 | 10                            | 250                                 |
| LM35C, LM35CA | 0 ... +110                         | 0                                 | 10                            | 250                                 |
| LM35D         | 0 ... +100                         | 0                                 | 10                            | 250                                 |
| TMP35         | +10 ... +125                       | 0                                 | 10                            | 250                                 |
| TMP36         | -40 ... +125                       | 500                               | 10                            | 750                                 |
| TMP37         | +5 ... +100                        | 0                                 | 20                            | 500                                 |

У всех термодатчиков напряжение на выходе может быть только положительным, но за счет смещения TMP36 способен измерять отрицательную температуру. Нулевое напряжение на его выходе соответствует температуре – 40 °C, а при выходном напряжении 0,5 В температура будет равна 0 °C.

Датчик температуры TMP36 является твердотельным, то есть для измерения температуры в нем не используется ртуть как в термометрах или термисторы (резисторы, чувствительные к температуре).

Вместо этого в нем используется тот факт, что при возрастании температуры напряжение на диоде возрастает в известной пропорции. Технически, на самом деле это падение напряжения между базой и эмиттером ( $V_{be}$ ) транзистора. После усиления такого изменения напряжения легко получить аналоговый сигнал, который будет прямо пропорционален температуре.

Датчики TMP36 не имеют движущихся частей, они точны, никогда не изнашиваются и могут работать в жестких условиях окружающей среды. Кроме того, они очень дешевые и очень просты в использовании.

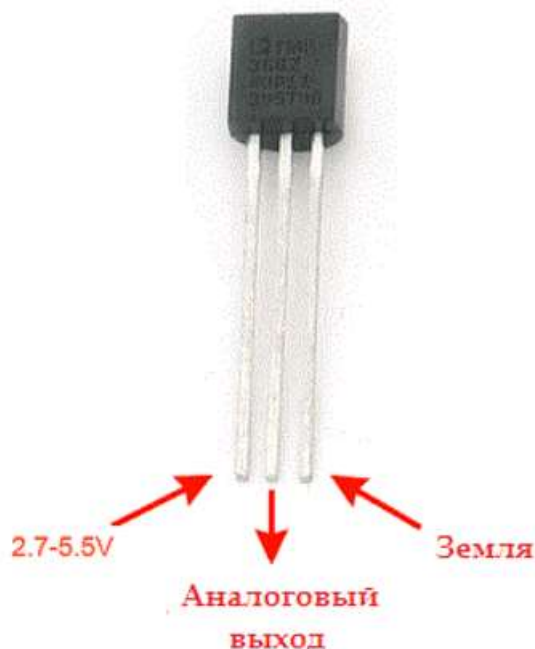


Рис. 134. Датчик TMP36

Выходной диапазон напряжения TMP36 составляет от 0.1В (-40°C) до 2.0В (150°C), но после 125°C точность измерения уменьшается. Подаваемое напряжение от 2.7В до 5.5В, потребление тока всего 0.05 мА.

## 2. Подключение и программирование датчика температуры.

Используем датчик температуры TMP36 для измерения температуры и зарегистрируем выход тремя светодиодами.

Arduino будет интерпретировать сигналы с аналогового входа температурного датчика TMP36, используя встроенный аналого-цифровой преобразователь (АЦП), доступ к которому осуществляется через аналоговые выводы А0-А5.

*Компоненты, необходимые для реализации проекта:*

- 1 плата Arduino Uno;
- 1 макетная плата;
- 3 светодиода для создания индикаторной «гистограммы»;
- 3 одинаковых резистора (220 Ом);
- 1 датчик температуры TMP36;
- провода.

*Принципиальная схема выглядит следующим образом:*

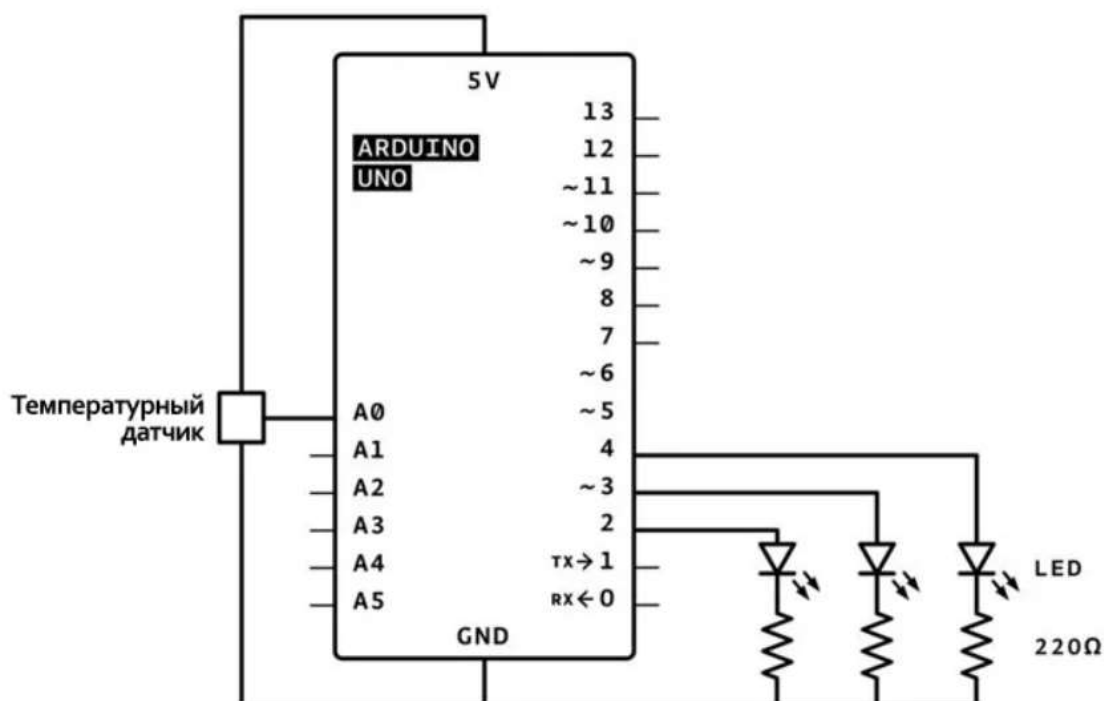


Рис. 135. Принципиальная схема подключения датчика температуры

*Соберите на макетной плате схему:*

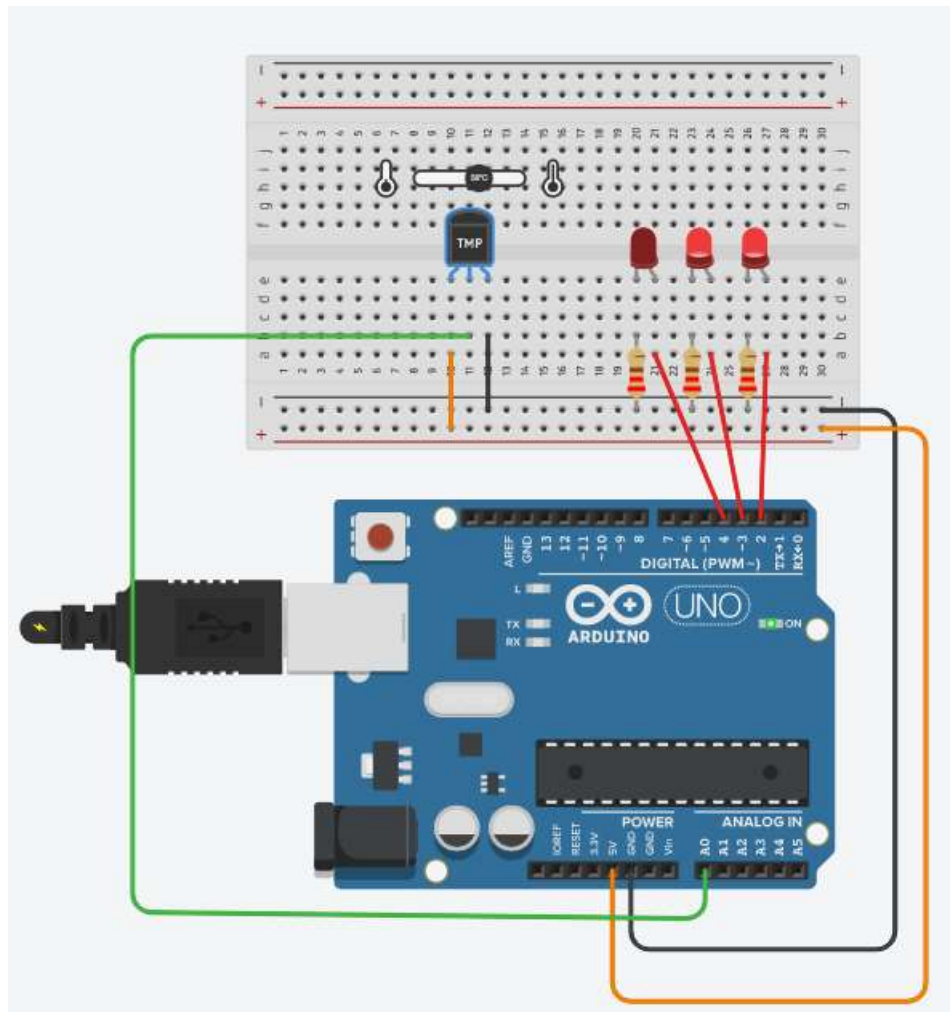


Рис. 136. Индикация светодиодов в зависимости от показаний датчика температуры

Датчик температуры создает изменяющийся сигнал напряжения в зависимости от температуры, которую он определяет. Он имеет три контакта: один, который соединяется с землей, другой, который подключается к 5 вольтам, и третий, который выдает переменное напряжение на Arduino, аналогичный аналоговому сигналу от потенциометра.

Существует несколько различных моделей датчиков температуры. Модель TMP36 удобна, потому что ее выходное напряжение прямо пропорционально температуре в градусах Цельсия.

В схеме можно увидеть, что температурный датчик подключен к источнику питания (5 вольт), заземлению (0 В) и аналоговому выходу A0. По мере повышения температуры выход, подключенный к A0, увеличивает свое напряжение.

*После сборки схемы, создайте следующую программу:*

```
// C++ code
int baselineTemp = 0; // создание переменной для хранения порогового значения
температуры
int celsius = 0; // создание переменной для хранения значения температуры в градусах
по Цельсия
int fahrenheit = 0; // создание переменной для хранения значения температуры в
градусах по Фаренгейту
void setup()
{
  pinMode(A0, INPUT); // определение порта A0 как входа
  Serial.begin(9600); // включение монитора последовательного порта
```

```

pinMode(2, OUTPUT); // определение порта 2 как выхода
pinMode(3, OUTPUT); // определение порта 3 как выхода
pinMode(4, OUTPUT); // определение порта 4 как выхода
}
void loop()
{
baselineTemp = 40; // определение пороговой температуры
celsius = map((analogRead(A0) - 20) * 3.04, 0, 1023, -40, 125); // измерение
температуры в градусах по Цельсия
fahrenheit = ((celsius * 9) / 5 + 32); // перевод в градусы по Фаренгейту
Serial.print(celsius); // вывод значения celsius через монитор последовательного порта
Serial.print(« C, «);
Serial.print(fahrenheit); // вывод значения fahrenheit через монитор последовательного
порта
Serial.println(« F»);
if (celsius < baselineTemp) // если celsius<пороговой температуры
{
digitalWrite(2, LOW); // задание низкого уровня (0) для порта 2
digitalWrite(3, LOW); // задание низкого уровня (0) для порта 3
digitalWrite(4, LOW); // задание низкого уровня (0) для порта 4
}
if (celsius >= baselineTemp && celsius < baselineTemp + 10) // если значение
// пороговой температуры <= celsius < значение пороговой температуре+10
{
digitalWrite(2, HIGH); // задание высокого уровня для порта 2
digitalWrite(3, LOW); // задание низкого уровня для порта 3
digitalWrite(4, LOW); // задание низкого уровня для порта 4
}
if (celsius >= baselineTemp + 10 && celsius < baselineTemp + 20) // если значение
// пороговой температуры+10 <= celsius < значение пороговой температуре+20
{
digitalWrite(2, HIGH); // задание высокого уровня для порта 2
digitalWrite(3, HIGH); // задание высокого уровня для порта 3
digitalWrite(4, LOW); // задание низкого уровня для порта 4
}
if (celsius >= baselineTemp + 20) // если celsius >= значение пороговой
температуре+20
{
digitalWrite(2, HIGH); // задание высокого уровня для порта 2
digitalWrite(3, HIGH); // задание высокого уровня для порта 3
digitalWrite(4, HIGH); // задание высокого уровня для порта 4
}
delay(1000); // ожидание в 1 секунду
}

```

Перед setup() создаются переменные для хранения целевой базовой температуры, а также значения датчика. Они типа int, потому что они являются целыми числами.

Внутри setup контакты сконфигурированы с помощью функции pinMode(). Pin A0 сконфигурирован как вход, поэтому мы можем «слушать» электрическое состояние датчика температуры. Пины 2, 3 и 4 сконфигурированы как выходы для управления светодиодами.

В основном цикле baselineTemp устанавливается на целевую температуру в 40 градусов С°.

Строка кода

```
celsius = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```

необходима для перевода единиц измерения датчика Arduino в градусы Цельсия.

Индикация светодиодам следующая – 1) холодно (40°-50°); тепло (50°-60°); жарко (60°-70°).



Рис. 137. Индикация светодиодам

Для преобразования температуры в Фаренгейты необходимо выполнить преобразование со значением, указанным в градусах Цельсия:  $F = (C * 9) / 5 + 32$ .

### 3. Индикация значения температуры с помощью RGB-светодиода.

Используем датчик температуры для измерения температуры и зарегистрируем значение температуры с помощью RGB-светодиода.

*Компоненты, необходимые для реализации проекта:*

- 1 плата Arduino Uno;
- 1 макетная плата;
- 1 RGB-светодиод для индикации температуры;
- 1 резистор (1 кОм);
- 1 датчик температуры TMP36;
- провода.

*Соберите на макетной плате схему:*

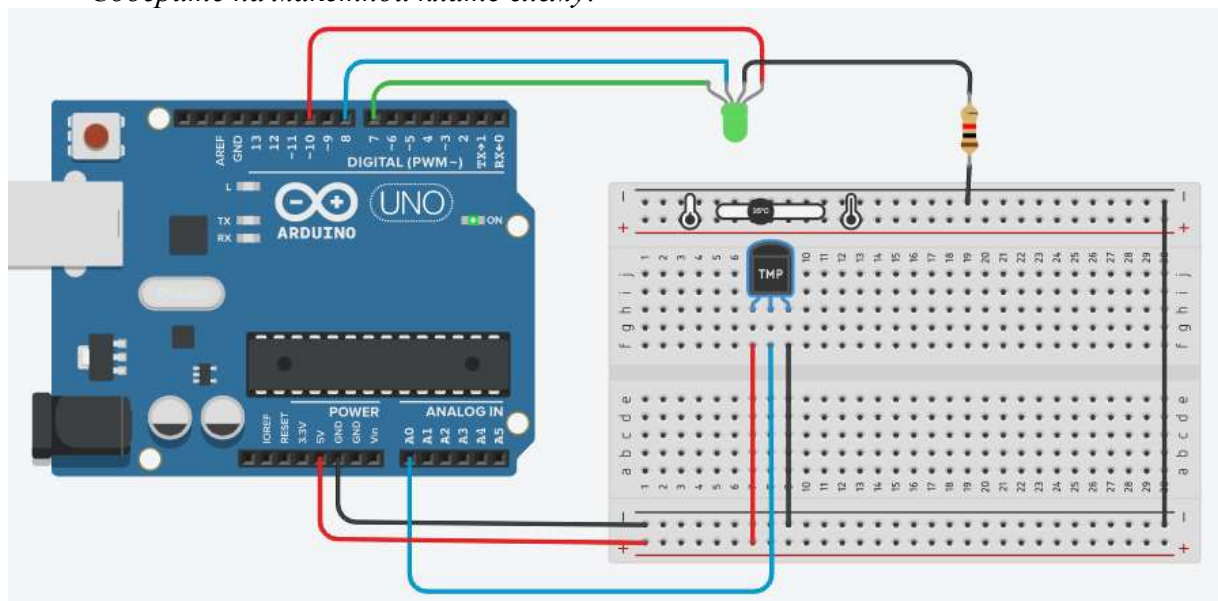


Рис. 138. Индикация значения температуры с помощью RGB-светодиода  
*После сборки схемы, создайте следующую программу:*

```
// C++ code
void setup()
{
  pinMode(A0, INPUT); // определение порта A0 как входа
  Serial.begin(9600); // включение монитора последовательного порта
  pinMode(10, OUTPUT); // определение порта 10 как выхода
  pinMode(7, OUTPUT); // определение порта 7 как выхода
  pinMode(8, OUTPUT); // определение порта 8 как выхода
}
void loop()
{
  Serial.println(analogRead(A0)); // вывод значения аналогового порта A0 через
монитор последовательного порта
  // индикация температуры как высокой, если значение порта A0 > 300
  if (analogRead(A0) > 300) {
    digitalWrite(10, HIGH); // задание высокого уровня (1) для порта 10
    delay(1000); // ожидание в 1с
    digitalWrite(10, LOW); // задание низкого уровня (0) для порта 10
    delay(1000); // ожидание в 1с
  }
  // индикация температуры как средней, если значение порта 100 >= A0 < 300
  if (analogRead(A0) >= 100 && analogRead(A0) < 300) {
    digitalWrite(7, HIGH); // задание высокого уровня для порта 7
    delay(1000); // ожидание в 1с
    digitalWrite(7, LOW); // задание низкого уровня для порта 7
    delay(1000); // ожидание в 1с
  } else // иначе
  {
    // индикация температуры как низкой, если значение порта A0 < 100
    digitalWrite(8, HIGH); // задание высокого уровня для порта 8
    delay(1000); // ожидание в 1с
    digitalWrite(8, LOW); // задание низкого уровня для порта 8
    delay(1000); // ожидание в 1с
  }
}
```

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

#### **Задания**

1. Измените код в программе «Подключение и программирование датчика температуры», отрегулировав разницу температурных порогов в первой программе на меньший диапазон (2, 4, 6, вместо 10, 20, 30).

### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

### **Контрольные вопросы**

1. Назовите основные характеристики датчика температуры TMP36?
2. При каком напряжении работает датчик температуры TMP36?
3. Как перевести значения аналогового сигнала датчика в значения температуры в градусах Цельсия?
4. Приведите формулу перевода значения температуры в градусах Цельсия в градусы по Фаренгейту.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

## **Лабораторная работа 11**

### **Программирование пьезодинамика и воспроизведение мелодий в проектах Arduino**

**Цель работы:** рассмотрение способов подключения и программирования работы пьезодинамика в проектах Arduino.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

**Знает:** способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы

организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

*Содержание:*

1. Основные понятия.
2. Подключение пьезодинамика.
3. Плавное изменение частоты звука.
4. Сборка и программирование терменвокса.
5. Программа, изменяющая громкость звучания.
6. Мелодии для зуммера.

#### 1. Основные понятия.

**Пьезодинамик** (пьезоизлучатель звука, зуммер, англ. buzzer) переводит переменное напряжение в колебание мембраны, которая в свою очередь создаёт звуковую волну. Это акустическое устройство для воспроизведения звука, использующее пьезоэлектрический эффект.



Рис. 139. Пьезодинамик

По сравнению с традиционными электромагнитными преобразователями звука, пьезоизлучатели имеют простую конструкцию. Пьезокерамический излучатель состоит из металлической пластины, на которую нанесена пьезоэлектрическая керамика, имеющая токопроводящее напыление. Пластина и напыление являются контактами пьезоизлучателя.

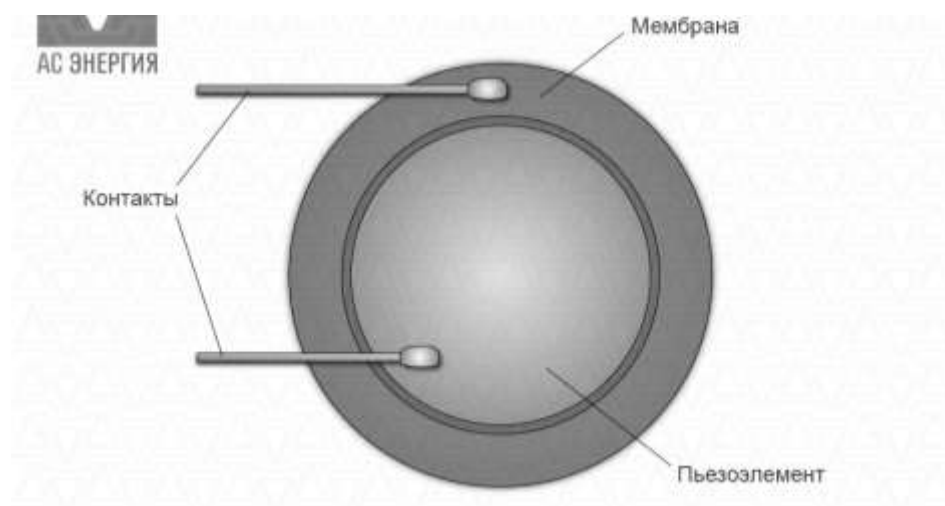


Рис. 140. Компоненты пьезодинамика

Принцип действия излучателей основан на эффекте, открытом братьями Кюри в 1880 г. В пьезокристаллах под действием механических сил на сдвиг, изгиб или кручение образуются электрические заряды. Кроме «прямого» эффекта существует и обратный эффект – если подать электричество на кристалл, то он начнет деформироваться. При частых колебаниях кристалла создается звуковая волна с заданной частотой. При этом происходят удары о металлическую пластинку, которая и производит «шум» нужной частоты. *Т.е. под действием электрического поля возникает механическое движение мембраны, которое и вызывает слышимые нами звуковые волны.*

Пьезоизлучатели получили широкое распространение: их используют в различных устройствах – будильниках, телефонах, игрушках и в другой технике. Часто при создании проектов Arduino требуется звуковой сигнал при включении устройства на плате или при других случаях. Пьезодинамик быстро может заставить схему издавать нужные звуки – сигнализировать, пищать или проигрывать мелодию. Управлять пьезодинамиком можно, подавая на устройство питание, включить-выключить с большой частотой. Управляя длительностью и частотой можно добиться разных звуков, например, сирены или сыграть нужную мелодию. Для уменьшения громкости сигнала пьезодинамика Ардуино можно использовать резисторы с разным номиналом, включая их в электрическую цепь.

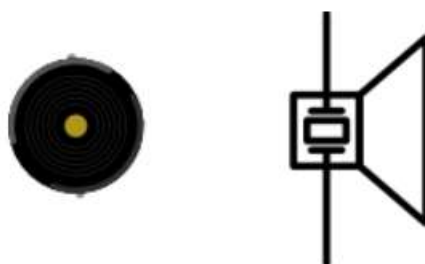


Рис. 140. Схемное обозначение пьезодинамика

Основные характеристики пьезодинамика:

|                                        |    |         |
|----------------------------------------|----|---------|
| Рекомендуемое (номинальное) напряжение | V  | Вольт   |
| Громкость (на заданном расстоянии)     | P  | Децибел |
| Пиковая частота                        | fP | Герц    |
| Ёмкость                                | C  | Фарад   |

Амплитудно-частотная характеристика:

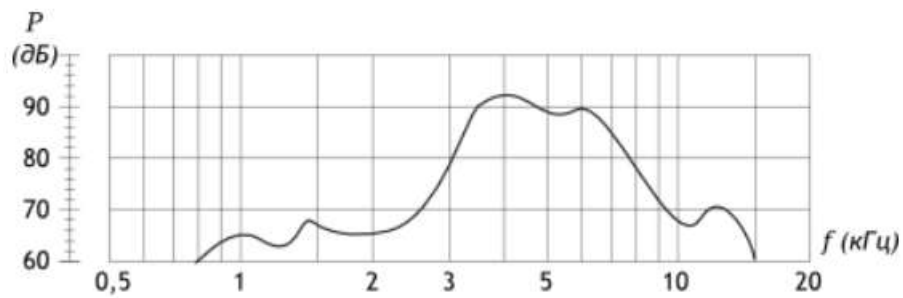


Рис. 141. Амплитудно-частотная характеристика пьезодинамика

Амплитудно-частотная характеристика (АЧХ) определяет громкость звука в зависимости от частоты управляющего сигнала, который и определяет высоту звучащей ноты.

Идеальная АЧХ — это прямая, т.е. одинаковая громкость вне зависимости от частоты. Но мир не идеален и разные виды излучателей имеют разные отклонения от идеала.

Для звучания нужно подавать на динамик квадратную волну. Какой частоты будет волна, такой частоты будет и звук.

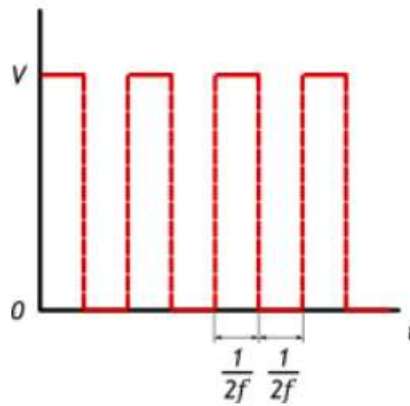


Рис. 142. Задание частоты сигнала пьезодинамика

Пьезодинамик потребляет всего пару мА, поэтому можно смело подключать его прямо к микроконтроллеру.

Подключение напрямую:

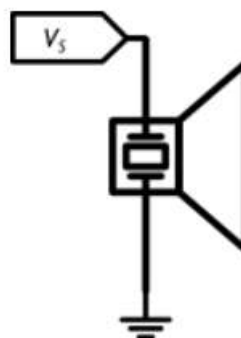


Рис. 143. Принципиальная схема подключения пьезодинамика

Подключение с регулировкой громкости:

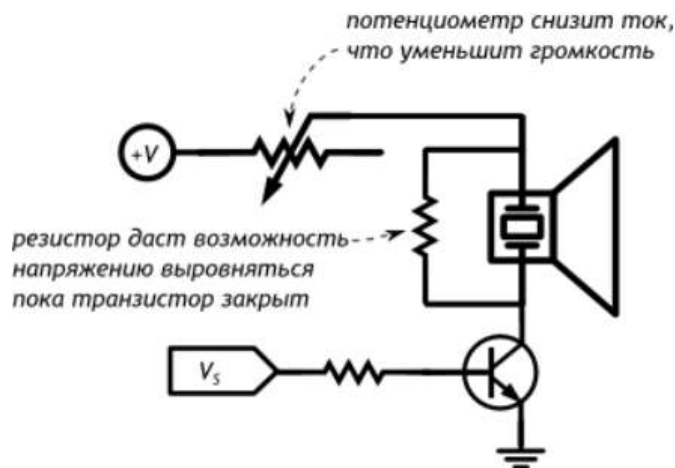


Рис. 144. Принципиальная схема подключения пьезодинамика с регулировкой громкости

Различают активный и пассивный пьезоизлучатель. Принцип действия у них одинаков, но в активном нет возможности менять частоту звучания, хотя сам звук громче и подключение проще. Активный зуммер будет сам генерировать звук, используя свой генератор, и для него требуется напряжение постоянного тока. Пассивный зуммер требует ШИМ сигнала, чтобы генерировать звук. Чтобы идентифицировать активный зуммер, приложите к нему постоянное напряжение, если пьезодинамик запищит, то он является активным.

## 2. Подключение пьезодинамика.

Один из контактов пьезодинамика подключается к земле (GND), другой – к любому пину с ШИМ (3, 5, 6, 9, 10, 11).

Схема на макетной плате:

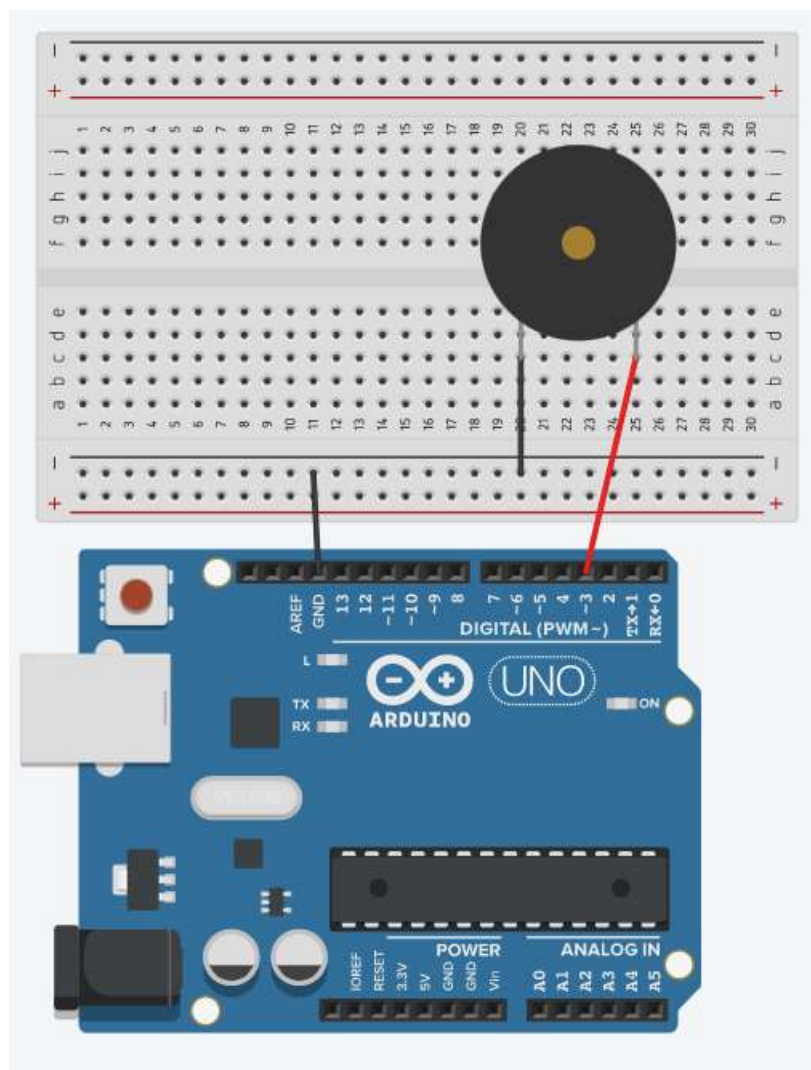


Рис. 145. Схема подключения пьезодинамика

После сборки схемы, создайте следующую программу:

```
// C++ code
void setup() {
  pinMode(3, OUTPUT); // определение порта 3 как выхода
}
void loop() {
  tone(3, 600); // включение на пьезодинамик 600 Гц
  delay(1000); // задание паузы в 1 секунду (1000 мс = 1 секунда)
  tone(3, 900); // включение на пьезодинамик 900 Гц
  delay(1000); // задание паузы в 1 секунду
  noTone(3); // отключение пьезодинамика (порт 11)
  delay(1000); // задание паузы в 1 секунду
}
```

Воспроизведение звука на Ардуино выполняется функцией `tone()`, где в скобках указывается номер пина и частота звука. Чтобы отключить звук на пьезодинамике необходимо использовать функцию `noTone()`.

### 3. Плавное изменение частоты звука.

Для плавного изменения частоты зуммера измените код следующим образом:

```
// C++ code
void setup() {
```

```

pinMode(3, OUTPUT); // определение порта 3 как выхода
}
void loop() {
for (int x = 0; x < 500 ; x++) // увеличение частоты звука
{
tone (3, x);
delay(1);
}
for (int x = 500; x > 0 ; x--) // уменьшение частоты звука
{
tone (3, x);
delay(1);
}
}
}

```

Для изменения частоты активного зуммера Ардуино используется цикл for, с помощью которого мы перебираем частоту звука от 0 до 500 и обратно.

#### 4. Сборка и программирование терменвокса.

В этом эксперименте мы имитируем действие музыкального инструмента терменвокс: изменяем высоту звучания бесконтактным путем, больше или меньше закрывая от света фоторезистор.

Оригинальный инструмент был изобретён в 1920 году Львом Сергеевичем Терменом, советским изобретателем, инженером-электромехаником и музыкантом. Данное изобретение можно воспроизвести с помощью нехитрой электроники.

*Компоненты, необходимые для реализации проекта:*

- 1 плата Arduino Uno;
- 1 макетная плата;
- 1 пьезопищалка;
- 1 резистор номиналом 10 кОм;
- 1 фоторезистор;
- провода.

*Принципиальная схема:*

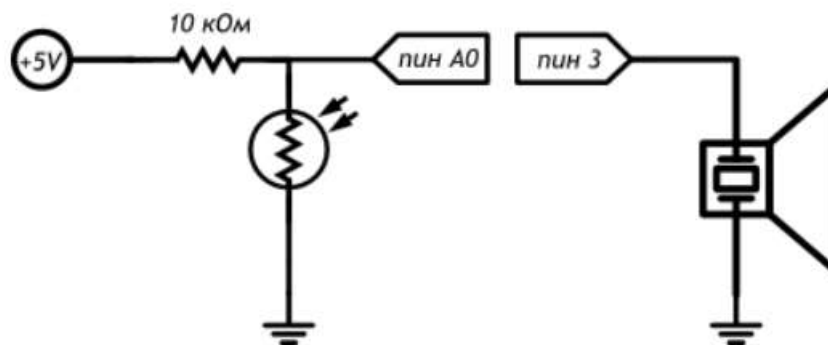


Рис. 146. Принципиальная схема терменвокса  
Схема на макетной плате:

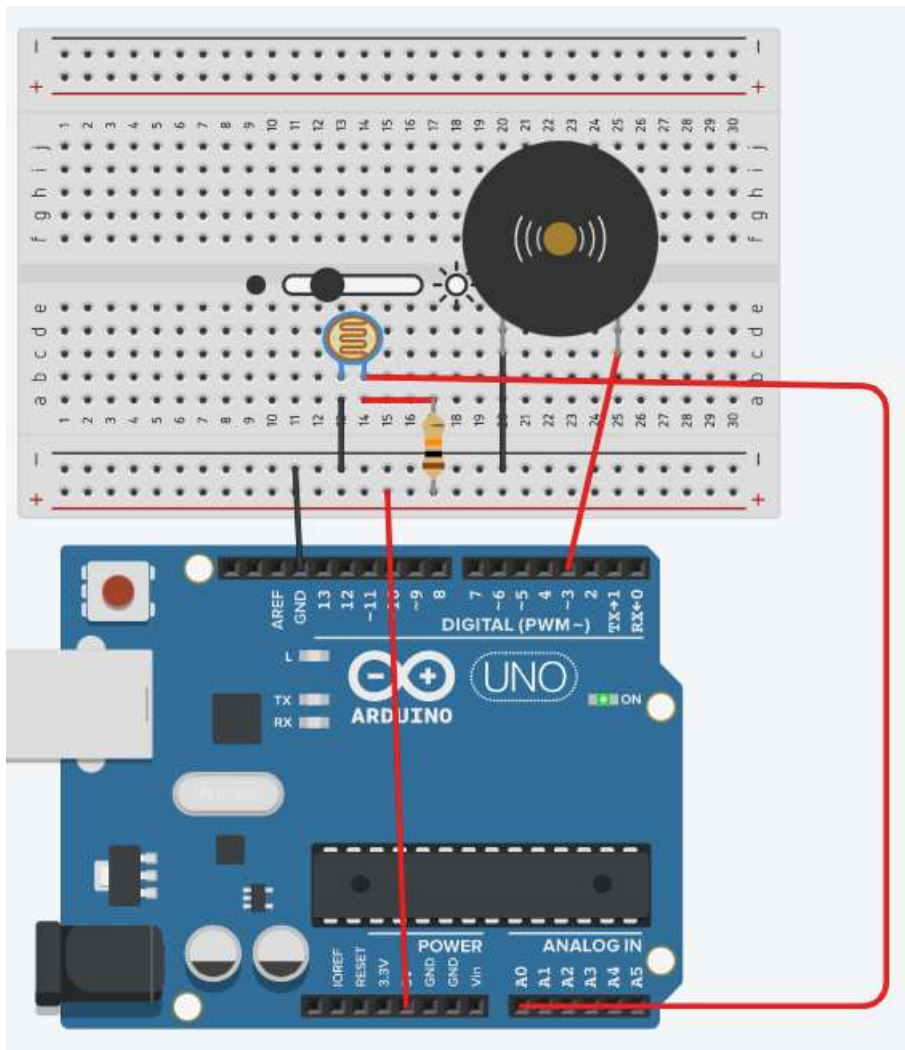


Рис. 147. Схема сборки терменвокса

В данной схеме используется резистор 10 кОм.

Полярность пьезопищалки роли не играет: можно подключать любую из ее ножек к земле, любую к порту микроконтроллера. Также не играет роли полярность фоторезистора (как и обычного резистора). Его можно устанавливать любой стороной.

Использовать стандартную функцию `tone()` можно только на одной ноге Arduino в одно время; а также использование `tone()` мешает использованию ШИМ на ногах 3 и 11. Используйте данную функцию только тогда, когда вам необходима какая-либо частота и не нужен ШИМ на 3 и 11 ножках Arduino. Например, вызванная для пина «5» или любого другого, функция `tone()` будет мешать работе выводов «3» и «11». Также если к Ардуино подключены несколько пьезоизлучателей, то одновременно будет работать только один. Чтобы включить излучатель на другом выводе, нужно прервать звук на текущем, вызвав функцию `noTone()`. Еще одно ограничение: нельзя извлечь звук частотой ниже 31 Гц.

Фоторезистор – делитель напряжения: он помещается в позицию R2 – между аналоговым входом и землей. Так можно получить резистивный фотосенсор.

После сборки схемы с пьезодинамиком, создайте следующую программу:

// C++ code

`#define BUZZER_PIN 3` // определение номера порта пьезодинамика

`#define LDR_PIN A0` // определение номера порта фоторезистора

`void setup()`

`{`

`pinMode (BUZZER_PIN, OUTPUT);` // определение порта пьезодинамика как выхода

```

    }
    void loop()
    {
        int val, frequency; // переменные для хранения значений освещённости и частоты
        val = analogRead (LDR_PIN); // присваивание переменной val значения порта A0
        (потенциометр)
        frequency = map (val, 0, 1023, 3500, 4500);
        // нахождение частоты звучания пьезодинамика в герцах (ноты)
        // с использованием функции проекции (англ. map), которая
        // переводит значение из одного диапазона в другой, строя пропорцию
        // диапазон значения освещенности порта потенциометра [0; 1023]
        // переводится в значение частоты [3500; 4500] от 3,5 до 4,5 кГц
        tone (BUZZER_PIN, frequency, 20);
        // задание пьезодинамику полученного значения частоты
        // звучание в течение 20 миллисекунд
        // при следующем прогоне цикла loop значение частоты
        // будет изменяться в зависимости от освещенности,
    }

```

Функция `map(value, fromLow, fromHigh, toLow, toHigh)` возвращает целочисленное значение из интервала `[toLow, toHigh]`, которое является пропорциональным отображением содержимого `value` из интервала `[fromLow, fromHigh]`.

Верхние границы `map` не обязательно должны быть больше нижних и могут быть отрицательными. К примеру, значение из интервала `[1, 10]` можно отобразить в интервал `[10,-5]`.

Если при вычислении значения `map` образуется дробное значение, оно будет отброшено, а не округлено.

Функция `map` не будет отбрасывать значения за пределами указанных диапазонов, а также масштабирует их по заданному правилу.

Если вам нужно ограничить множество допустимых значений, используйте функцию `constrain(value, from, to)`, которая вернет:

- ✓ `value`, если это значение попадает в диапазон `[from, to]`
- ✓ `from`, если `value` меньше него
- ✓ `to`, если `value` больше него

Функция `tone(pin, frequency, duration)` заставляет пьезопищалку, подключенную к порту `pin`, издавать звук высотой `frequency` герц на протяжении `duration` миллисекунд.

Параметр `duration` не является обязательным. Если его не передать, звук включится навсегда. Чтобы его выключить, вам понадобится функция `noTone(pin)`. Ей нужно передать номер порта с пищалкой, которую нужно выключить

Вызов `tone` для уже звучащего порта обновит частоту и длительность звучания.

### **5. Программа, изменяющая громкость звучания.**

В программе, изменяющей только громкость звучания, не используется функция `tone()`:

```

// C++ code
int soundPin=3; // определение номера порта пьезодинамика
void setup() {
    pinMode(3, OUTPUT); // // определение порта пьезодинамика как выхода
}
void loop() {
    analogWrite (soundPin, 50); // включение пьезодинамика на значение громкости 50

```

```

delay(1000); // задание паузы в 1 секунду (1000 мс = 1 секунда)
analogWrite (soundPin, 0); // выключение пьезодинамика
delay(1000); // задание паузы в 1 секунду
}

```

Этот код задает частоту включения звука один раз в две секунды. Для корректной работы устройства необходимо задать номер PIN, определить его как «выход». Управление аппаратными ШИМ осуществляется с помощью системной функции `analogWrite()`. Используя функцию `analogWrite()`, нельзя изменять тональность звука. Пьезоизлучатель всегда будет звучать на частоте примерно 980 Гц. Это значение совпадает с частотой работы выводов с ШИМ на платах Ардуино и аналогов. С помощью функции `analogWrite()` частота ШИМ изменяется в диапазоне 0 до 255 и соответствует коэффициенту заполнения импульса от 0 до 100 %.

Т.е. функция `analogWrite` использует в качестве аргументов номер вывода и уровень, который изменяет свое значение от 0 до 255. Это все по причине того, что ШИМ-выводы Arduino имеют ЦАП (цифроаналоговый преобразователь) 8-бит. Изменяя этот параметр, пользователь меняет громкость зуммера на небольшую величину. Для полного выключения следует прописать в порте значение «0».

## 6. Мелодии для зуммера.

Для того чтобы разнообразить работу с новым проектом и добавить в него «развлекательный» элемент, можно задавать определённый набор частот звука, делая его созвучным некоторым знаменитым композициям из песен и кинофильмов. Разнообразные скетчи для таких мелодий можно найти в интернете. Приведем пример программы, проигрывающей заданную мелодию:

```

// C++ code
/*
 * Использование пищалки для создания мелодии!
 * Зуммер является электромеханическим
 * компонентом, который можно использовать в качестве источника
 * звука. Внутри зуммера есть катушка, провода и небольшой магнит.
 * Когда ток течет через катушку, она становится намагниченной и
 * притягивает мембрану к магниту, создавая крошечный «щелчок».
 * Когда это происходит с частотой тысячи раз в секунду, создается
 * звуковой тон.
 * Arduino имеет встроенную команду с именем tone(), которая
 * заставляет зуммер издавать звуки различной частоты.
 * Эта программа знает частоты определенных нот, что позволяет
 * создавать музыкальные мелодии.
 */
// Этот скетч использует зуммер для воспроизведения мелодий.
// Команда Arduino tone() будет издавать звуки определенной частоты.
// Мы создаем функцию, которая сопоставляет символ нотной гаммы
// («До-ре-ми-фа...До») соответствующей частоте из следующей таблицы:
// note    frequency
// c       262 Hz
// d       294 Hz
// e       330 Hz
// f       349 Hz
// g       392 Hz
// a       440 Hz
// b       494 Hz

```

```

// C 523 Hz
const int buzzerPin = 3;
// Мы создали массив с нотами, которые хотим воспроизвести,
// измените эти значения, чтобы создать свои мелодии!
// Длина должна равняться общему количеству нот и пауз

const int songLength = 18;
// Обозначение нот представляет собой массив из текстовых символов,
// соответствующим нотам в песне. Пробел означает паузу (пустую ноту)
char notes[] = «cdfda ag cdfdg gf»; // пробелы означают паузы
// Ритм задается массивом из длительности нот и пауз между ними.
// «1» – четвертная нота, «2» – половинная, и т.д.
// Не забывайте, что пробелы должны быть тоже определенной длины.
int beats[] = {1,1,1,1,1,1,4,4,2,1,1,1,1,1,4,4,2};
// «tempo» это скорость проигрывания мелодии.
// Для того, чтобы мелодия проигрывалась быстрее, вы
// должны уменьшить следующее значение.
int tempo = 150;
void setup()
{
  pinMode(buzzerPin, OUTPUT);
}
void loop()
{
  int i, duration;
  for (i = 0; i < songLength; i++) // пошаговое воспроизведение
  // из массива
  {
    duration = beats[i] * tempo; // длительность нот/пауз в ms
    if (notes[i] == ' ') // если нота отсутствует?
    {
      delay(duration); // тогда не большая пауза
    }
    else // в противном случае играть
    {
      tone(buzzerPin, frequency(notes[i]), duration);
      delay(duration); // ждать пока проигрывается
    }
    delay(tempo/10); // маленькая пауза между нотами
  }
  // Мы хотим, чтобы мелодия проиграла всего один раз, так что здесь остановимся
  окончательно:
  while(true){}
  // Если же вы хотите, чтобы мелодия играть снова и снова,
  // Удалить вышеуказанное заявление
}
int frequency(char note)
{
  // Эта функция принимает символ ноты (a-g), и возвращает
  // частоту в Гц для функции tone().
  int i;

```

```

const int numNotes = 8; // количество хранимых нот
// Следующие массивы содержат символы нот и соответствующие им
// частоты. Последний символ «C» (нота «ДО») в верхнем регистре
// (большая), это сделано для того чтобы отличить ее от первой
// ноты «c», более низкого тона. Если вы хотите добавить больше
// нот, вы должны будете использовать уникальный символ для
// каждой новой ноты.
// Каждый «char» (символ), мы заключаем в одинарные кавычки.
char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
int frequencies[] = { 262, 294, 330, 349, 392, 440, 494, 523 };
// Теперь мы будем искать во всем массиве, символ ноты и если
// находим, возвращаем частоту для этой ноты.
for (i = 0; i < numNotes; i++) // пошаговый перебор нот
{
    if (names[i] == note)      // если находим
    {
        return(frequencies[i]); // возвращаем частоту
    }
}
return(0); // Поиск символа не дал результата? Но, необходимо
// вернуть какое-то значение, так вернем 0.
}

```

При написании собственных мелодий пригодится знание частот нот и длительностей интервалов, используемых в стандартной нотной записи.



Рис. 148. Клавиши одной октавы  
Частота нот и длительность для пьезодинамика Ардуино (для первой октавы):

| TONE    |   | FREQUENCY<br>(in hertz) |   | DELAY<br>VALUE |
|---------|---|-------------------------|---|----------------|
| Do (C)  | → | 261.63                  | → | 1915 $\mu$ s   |
| Re (D)  | → | 293.66                  | → | 1700 $\mu$ s   |
| Mi (E)  | → | 329.63                  | → | 1519 $\mu$ s   |
| Fa (F)  | → | 349.23                  | → | 1432 $\mu$ s   |
| Sol (G) | → | 392.00                  | → | 1275 $\mu$ s   |
| La (A)  | → | 440.00                  | → | 1136 $\mu$ s   |
| Si (B)  | → | 493.88                  | → | 1014 $\mu$ s   |

Рис. 149. Частота нот и длительность нот первой октавы  
Частота нот для пьезодинамика Ардуино:

| Ноты      | Суббконтр-<br>октава | Контр-<br>октава | Большая | Малая  | Первая | Вторая | Третья  | Четвертая | Пятая   |
|-----------|----------------------|------------------|---------|--------|--------|--------|---------|-----------|---------|
| ДО        | 16,35                | 32,70            | 65,41   | 130,82 | 261,63 | 523,26 | 1046,52 | 2093,04   | 4186,08 |
| ДО диез   | 17,32                | 34,65            | 69,30   | 138,59 | 277,18 | 554,36 | 1108,72 | 2217,44   | 4434,88 |
| РЕ        | 18,35                | 36,71            | 73,42   | 146,83 | 293,66 | 587,32 | 1174,64 | 2349,28   | 4698,56 |
| РЕ диез   | 19,45                | 38,89            | 77,78   | 155,57 | 311,13 | 622,26 | 1244,52 | 2489,04   | 4978,08 |
| МИ        | 20,60                | 41,20            | 82,41   | 164,82 | 329,63 | 659,26 | 1318,52 | 2637,04   | 5274,08 |
| ФА        | 21,83                | 43,65            | 87,31   | 174,62 | 349,23 | 698,46 | 1396,92 | 2793,84   | 5587,68 |
| ФА диез   | 23,12                | 46,25            | 92,50   | 185,00 | 369,99 | 739,98 | 1479,96 | 2959,92   | 5919,84 |
| СОЛЬ      | 24,50                | 49,00            | 98,00   | 196,00 | 392,00 | 784,00 | 1568,00 | 3136,00   | 6272,00 |
| СОЛЬ диез | 25,96                | 51,91            | 103,83  | 207,65 | 415,30 | 830,60 | 1661,20 | 3322,40   | 6644,80 |
| ЛЯ        | 27,50                | 55,00            | 110,00  | 220,00 | 440,00 | 880,00 | 1760,00 | 3520,00   | 7040,00 |
| ЛЯ диез   | 29,14                | 58,27            | 116,54  | 233,08 | 466,16 | 932,32 | 1864,64 | 3729,28   | 7458,56 |
| СИ        | 30,87                | 61,74            | 123,47  | 246,94 | 493,88 | 987,76 | 1975,52 | 3951,04   | 7902,08 |

Рис. 150. Частота нот и длительность нот

Приведем еще один пример простой программы, проигрывающей мелодию:

```
// C++ code
#define buzzer 3 // подключаем пьезоэлемент к контакту 5 ШИМ
void setup()
{
}
// tone(номер пина, частота в Гц, длительность в мсек)
void loop()
{
tone(buzzer, 440, 250);
delay(250);
tone(buzzer, 165, 250);
delay(250);
tone(buzzer, 440, 250);
delay(250);
delay(250);
}
```

```

tone(buzzer, 440, 250);
delay(250);
tone(buzzer, 165, 250);
delay(250);
tone(buzzer, 440, 250);
delay(250);
delay(250);

tone(buzzer, 330, 125);
delay(125);
tone(buzzer, 294, 125);
delay(125);
tone(buzzer, 262, 125);
delay(124);
tone(buzzer, 494, 125);
delay(125);
tone(buzzer, 440, 125);
delay(125);
tone(buzzer, 494, 125);
delay(125);
tone(buzzer, 262, 125);
delay(125);
tone(buzzer, 294, 125);
delay(125);

tone(buzzer, 330, 250);
delay(250);
tone(buzzer, 165, 250);
delay(250);
tone(buzzer, 440, 250);

delay(5000);
}

```

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

#### Задания

1. Создайте программу, выдающую с помощью азбуки Морзе позывной SOS: три точки, три тире, три точки.
2. Создайте программу, которая выводит сигнал Вашего имени с помощью пьезодинамика.
3. Измените код программы терменвокса так, чтобы с падением освещенности звук становился ниже (например, падал от 5 кГц до 2,5 кГц)

4. Измените код программы так, чтобы звук терменвокса раздавался не непрерывно, а 10 раз в секунду с различными паузами.
5. Добавьте в программу включение последовательного порта для отображения значений высоты тона.
6. Воспроизведите с помощью Arduino Вашу любимую мелодию.

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Что такое пьезодинамик?
2. Можем ли мы регулировать яркость светодиода, подключенного к 11-му порту, во время звучания пьезопищалки?
3. Что изменится в работе терменвокса, если заменить резистор на 10 кОм резистором на 100 кОм? Попробуйте ответить без эксперимента. Затем отключите питание, замените резистор и проверьте.
4. Как будет работать вызов `tone` без указания длительности звучания?
5. Можно ли устроить полифоническое звучание с помощью функции `tone`?

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

### **Лабораторная работа 12**

#### **Подключение и программирование семисегментного индикатора и ЖК-дисплея**

**Цель работы:** исследование семисегментного индикатора и ЖК-дисплея для проектирования устройств на базе Arduino.

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

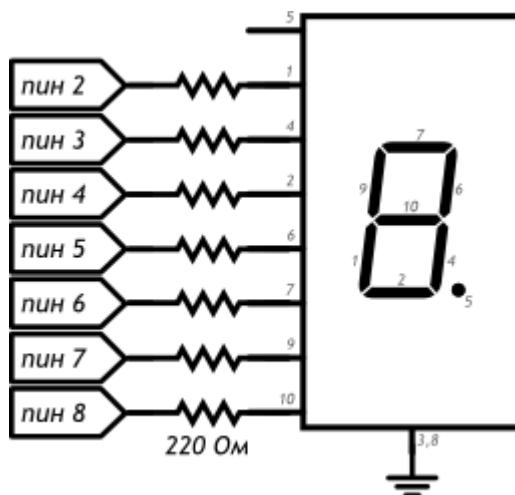
*Содержание:*

1. Подключение семисегментного индикатора для создания секундомера.
2. Характеристика и режимы работы ЖК-дисплея 16×2.
3. Подключение ЖК-дисплея 16×2 для вывода приветствия.
4. Вывод приветствия «Hello, world» на ЖК-дисплее 16×2.
5. Подключение ЖК-дисплея 16×2 с регулировкой контрастности.
6. Вывод значений температуры на ЖК-дисплей 16×2.

В этом проекте мы создаем секундомер, который считает от 0 до 9.

- 1 плата Arduino Uno;
- 1 макетная плата;
- 1 семисегментный индикатор;
- 7 резисторов номиналом 220 Ом;
- провода.

Принципиальная схема выглядит следующим образом:



120

Перед сборкой схемы представьте в письменном виде изображение каждой цифры от 0 до 9 с указанием номеров сегментов индикатора, которые должны быть включены для отображения данной цифры, например, для цифры «0» включены сегменты под номерами 1, 2, 4, 6, 7, 9, а сегмент под номером 10 выключен. Далее для каждой цифры укажите номера пинов, которым соответствуют данные сегменты.

Соберите на макетной плате схему:

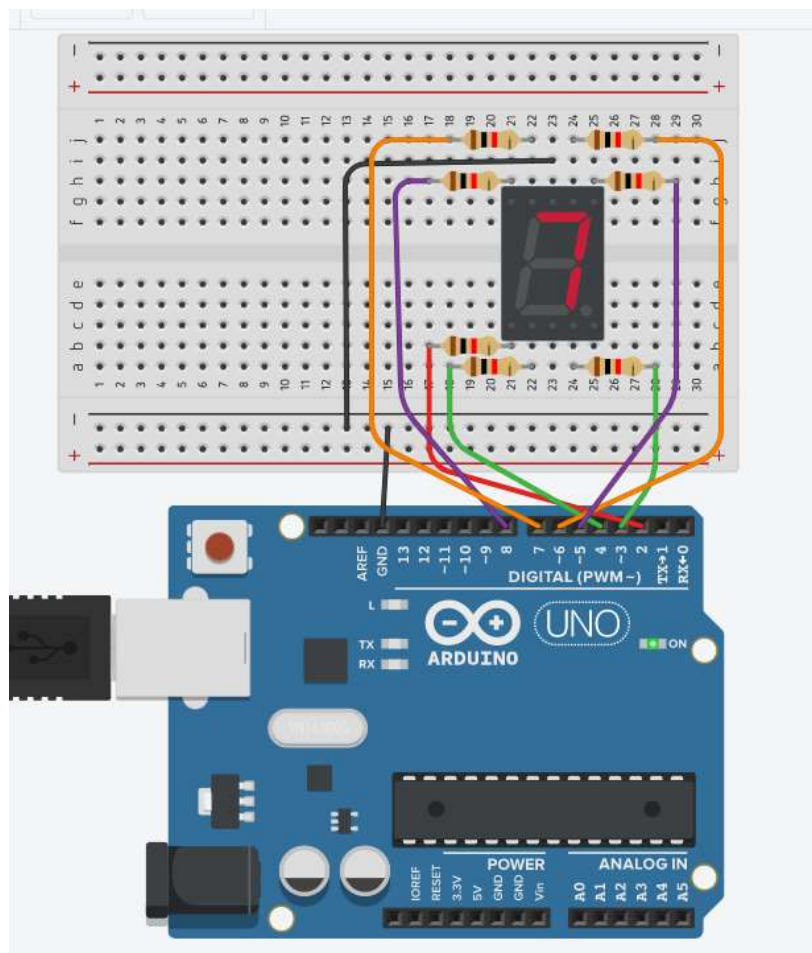


Рис. 152. Схема подключения семисегментного индикатора

Во время сборки схемы, сопоставьте сегменты индикатора с номерами его ножек, а те, в свою очередь, с пинами Arduino, к которым они подключаются.

Выводы 3 и 8 семисегментного индикатора являются катодами, к земле можете подключать любой из них.

Вывод 5 индикатора – это точка. Мы не используем её в этом проекте.

Сегменты индикатора – просто светодиоды, поэтому для работы с каждым из них применяется резистор номиналом 220 Ом.

После сборки схемы, создайте следующую программу:

```
// C++ code
#define FIRST_SEGMENT_PIN 2 // т.к. используются порты от 2 по 8
#define SEGMENT_COUNT 7 // т.к. количество сегментов равно 7
// префикс «0b» означает, что целое число за ним записано в
// в двоичном коде. Единицами мы обозначим номера сегментов
// индикатора, которые должны быть включены для отображения
// арабской цифры. Всего цифр 10, поэтому в массиве 10 чисел.
// Нам достаточно всего байта (англ. byte, 8 бит) для хранения
// комбинации сегментов для каждой из цифр.
```

```

byte numberSegments[10] = {
0b00111111, 0b00001010, 0b01011101, 0b01011110, 0b01101010,
0b01110110, 0b01110111, 0b00011010, 0b01111111, 0b01111110,
};
void setup()
{
for (int i = 0; i < SEGMENT_COUNT; ++i)
pinMode(i + FIRST_SEGMENT_PIN, OUTPUT);
}
void loop()
{
// определяем число, которое собираемся отображать. Пусть им
// будет номер текущей секунды, зацикленный на десятке
int number = (millis() / 1000) % 10;
// получаем код, в котором зашифрована арабская цифра
int mask = numberSegments[number];
// для каждого из 7 сегментов индикатора
for (int i = 0; i < SEGMENT_COUNT; ++i) {
// определяем: должен ли он быть включён. Для этого
// считываем бит (англ. read bit), соответствующий текущему
// сегменту «i». Истина – он установлен (1), ложь – нет (0)
boolean enableSegment = bitRead(mask, i);
// включаем/выключаем сегмент на основе полученного значения
digitalWrite(i + FIRST_SEGMENT_PIN, enableSegment);
}
}

```

Создан массив типа byte: каждый его элемент это 1 байт, 8 бит, может принимать значения от 0 до 255.

Символы арабских цифр закодированы состоянием пинов, которые соединены с выводами соответствующих сегментов: 0, если сегмент должен быть выключен, и 1, если включен.

Millis() – возвращает количество миллисекунд, прошедших с запуска. Возвращает unsigned long, от 1 до 4 294 967 295 миллисекунд (~50 суток), имеет разрешение 1 миллисекунда, после переполнения сбрасывается в 0.

Запись number = (millis() / 1000) % 10 означает, что переменной number присваивается остаток от деления значения (millis() / 1000) на 10.

В переменную mask помещается тот элемент массива numberSegments, который соответствует текущей секунде, вычисленной в предыдущей инструкции.

В цикле for мы пробегаем по всем сегментам, извлекая с помощью встроенной функции bitRead нужное состояние для текущего пина, в которое его и приводим с помощью digitalWrite и переменной enableSegment.

bitRead(x, n) возвращает boolean значение: n-ый бит справа в байте x.

## 2. Характеристика и режимы работы ЖК-дисплея 16×2.

**Жидкокристаллический дисплей 16×2** нужен в проектах для вывода строк символов, например, показаний датчиков, отображения простых меню, подсказок и приветствий. На экран выводятся символы размером 5×8 пикселей. ЖК-дисплей 16×2 стоит недорого, есть различные модификации с разными цветами подсветки, можно легко скачать готовые библиотеки для скетчей Ардуино. Недостатком этого экрана является тот факт, что дисплей имеет 16 цифровых выводов, из которых обязательными являются минимум 6. Поэтому использование ЖК-дисплея добавляет серьезные ограничения для плат Arduino

Uno. Если контактов не хватает, то придется использовать плату Arduino Mega или же сэкономить контакты, в том числе за счет подключения дисплея через I2C-переходник (использование I2C уменьшает количество проводов до 4, а занятых пинов до 2).

Существуют и другие символьные ЖК-дисплеи: 16×4, 16×1, 20×4 и т. д., также основанные на контроллере параллельного интерфейса HD44780 от Hitachi. В ЖК-дисплеях используются жидкие кристаллы для создания видимого изображения. Когда к подобному кристаллу прикладывается ток, то он становится непрозрачным, перекрывая подсветку, которая находится за экраном. В результате эта конкретная область становится темной по сравнению с другой. Подобным образом на экране отображаются символы.

*Технические характеристики ЖК-дисплея 16×2:*

Символьный тип отображения, есть возможность загрузки символов;

Технология: LCD (Liquid Crystal Display)

Формат: 16x2 символов;

Цвет: монохромный;

Светодиодная подсветка;

Цвет подсветки (зависит от модели): красный / зелёный / синий / чёрный / янтарный;

Цвет символов (зависит от модели): зелёный / чёрный / белый;

Индикация: 2 строки по 16 символов

Контроллер HD44780;

Интерфейс: параллельный 4/8 бит;

Напряжение питания: 5В;

Максимальный ток потребления: 1 мА;

Потребляемый ток подсветки: 100 мА;

Диапазон рабочих температур от -20С до +70С, диапазон температур хранения от -30С до +80С;

Габариты: 84×44×13 мм;

Угол обзора 180 градусов.

*Назначение выводов ЖК-дисплея 16×2:*

**GND** – должен быть подключен к земле Arduino.

**VCC** – это вывод питания для ЖК-дисплея, к которому подключается 5-вольтовый контакт Arduino.

**Vo** (LCD Contrast) – вывод контролирует контрастность и яркость ЖК-дисплея. Используя простой делитель напряжения с потенциометром, мы можем точно отрегулировать контрастность.

**RS** (Register Select) – этот вывод позволяет Arduino сообщать ЖК-дисплею, отправляются команды или данные. В основном этот вывод используется для дифференциации команд от данных. Например, когда на выводе RS установлено значение LOW, мы отправляем команды на ЖК-дисплей (например, установить курсор в определенном месте, очистить дисплей, сдвинуть дисплей вправо и т. д.). Когда вывод RS установлено значение HIGH, мы отправляем данные/символы на ЖК-дисплей.

**R/W** (Read/Write) – вывод предназначен для контроля того, что необходимо сделать – считать данные или передать их на ЖК-дисплей. Поскольку мы просто используем этот ЖК-дисплей в качестве устройства вывода, то достаточно на этот вывод подать HIGH уровень, тем самым мы перейдем в режим записи.

**EN** (Enable) – вывод используется для включения дисплея. Это означает, что когда на этом выводе установлено значение LOW ЖК-дисплей не реагирует на то, что происходит с R/W, RS и линиями шины данных. Когда же на этом выводе HIGH, ЖК-дисплей обрабатывает входящие данные.

**D0-D7** (Data Bus) – это выводы, по которым передаются 8-битные данные на дисплей. Например, если мы хотим отобразить символ «А» в верхнем регистре, мы отправляем на LCD дисплей 0100 0001 (в соответствии с таблицей ASCII).

**АК** (Anode & Cathode) – используются для управления подсветкой LCD дисплея.

*Дисплей может работать в двух режимах передачи данных:*

8-битный режим – используются и младшие, и старшие биты (DB0-DB7);

4-битный режим – используются только младшие биты (DB4-DB7).

Использовать восьмибитный режим не целесообразно. Для его работы требуется 4 дополнительных вывода, а выигрыш по скорости небольшой. Быстрота связана с тем, что в 8-битном режиме данные передаются за один раз. А в 4-битном режиме байт разделяется на 2 части, один из них сдвигается на 4 бита вправо и выполняются 2 операции записи.

### **3. Подключение ЖК-дисплея 16×2 для вывода приветствия.**

*Принципиальная схема подключения ЖК-дисплея 16×2 с регулировкой контрастности (используется потенциометр):*

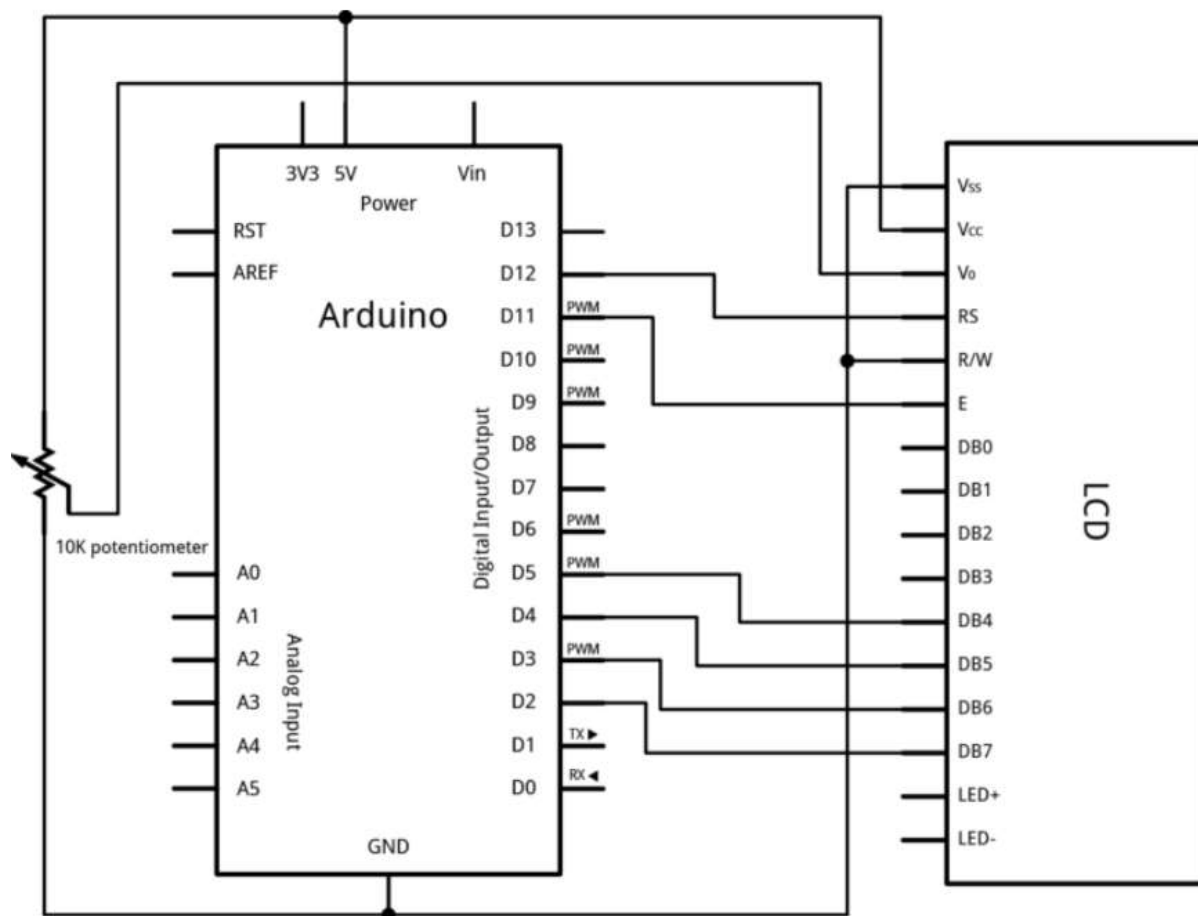


Рис. 153. Принципиальная схема подключения ЖК-дисплея 16×2

*Компоненты, необходимые для реализации проекта (без потенциометра, его подключим позже):*

- 1 плата Arduino Uno;
- 1 макетная плата;
- 1 ЖК-дисплей 16×2;
- 1 резистор номиналом 330 Ом;
- провода.

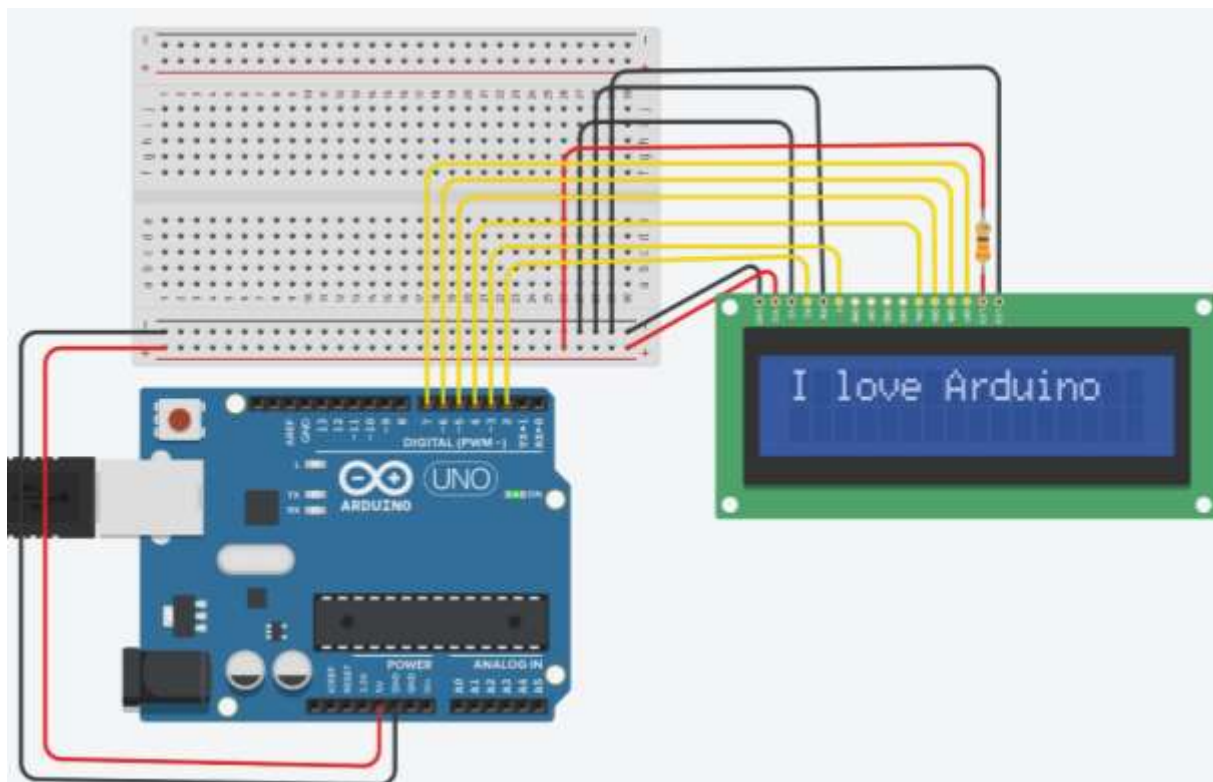


Рис. 154. Схема подключения ЖК-дисплея 16×2

*Добавьте код:*

```
// C++ code
#include <LiquidCrystal.h>
// Библиотека для работы с текстовым дисплеем
LiquidCrystal lcd(2,3,4,5,6,7);

void setup() {
  lcd.begin(16,2);
  lcd.print(«I love Arduino»);
}

void loop() {}
```

Для работы с символьным графическим дисплеем можно воспользоваться библиотекой `LiquidCrystal`, которая входит в стандартный набор Arduino IDE (поддерживается также в Tinkercad) и предназначена для работы по 8-битному (4-битному) параллельному интерфейсу.

*Со списком всех функций управления дисплеем для библиотеки `LiquidCrystal` и функциями управления курсором можно ознакомиться в Интернете.*

#### **4. Вывод приветствия «Hello, world» на ЖК-дисплее 16×2.**

*Выведем еще одно приветствие, используя ту же схему:*

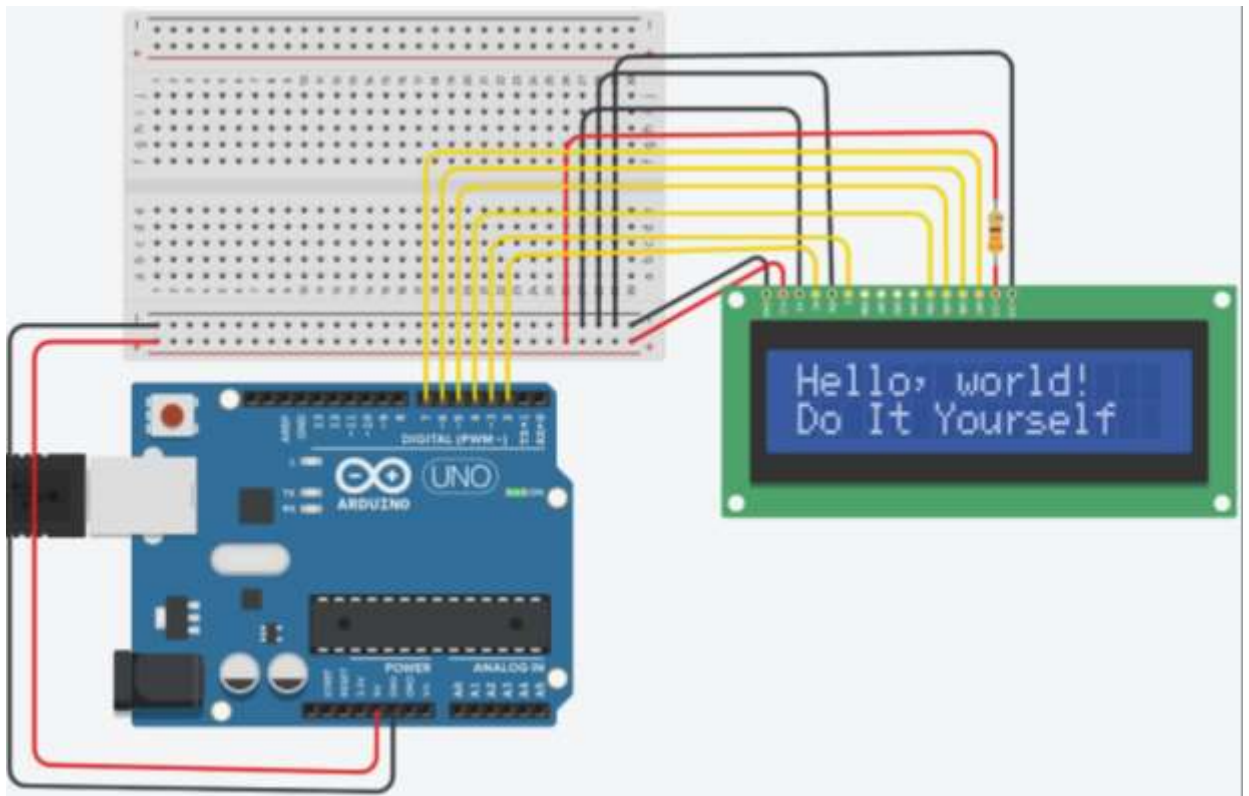


Рис. 155. Вывод приветствия «Hello, world» на ЖК-дисплее 16×2

*Наберите следующий код:*

```
// C++ code
// Библиотека для работы с текстовым дисплеем
#include <LiquidCrystal.h>
// Задаём имя пинов дисплея
constexpr byte PIN_RS = 2;
constexpr byte PIN_EN = 3;
constexpr byte PIN_DB4 = 4;
constexpr byte PIN_DB5 = 5;
constexpr byte PIN_DB6 = 6;
constexpr byte PIN_DB7 = 7;
/*
 * Создаём объект для работы с текстовым дисплеем
 * LiquidCrystal lcd(RS, EN, DB4, DB5, DB6, DB7);
 */
LiquidCrystal lcd(PIN_RS, PIN_EN, PIN_DB4, PIN_DB5, PIN_DB6, PIN_DB7);
void setup() {
  // Устанавливаем размер экрана
  // Количество столбцов и строк
  lcd.begin(16, 2);
  // Устанавливаем курсор в колонку 0 и строку 0
  lcd.setCursor(0, 0);
  // Печатаем первую строку
  lcd.print(«Hello, world!»);
  // Устанавливаем курсор в колонку 0 и строку 1
  lcd.setCursor(0, 1);
  // Печатаем вторую строку
  lcd.print(«Do It Yourself»);
}
```

```

}
void loop() {}

```

*constexpr* – спецификатор типа, введённый в стандарт программирования языка C++11 для обозначения константных выражений, которые могут быть вычислены во время компиляции кода.

### 5. Подключение ЖК-дисплея 16×2 с регулировкой контрастности.

Добавим потенциометр для регулировки уровня контрастности. Для этого необходимо подключить крайние выводы потенциометра к 5В и GND, а центральный контакт (бегунок) потенциометра к контакту 3 ЖК-дисплея. При повороте ручки потенциометра контрастность будет изменяться.

*Схема выглядит следующим образом:*

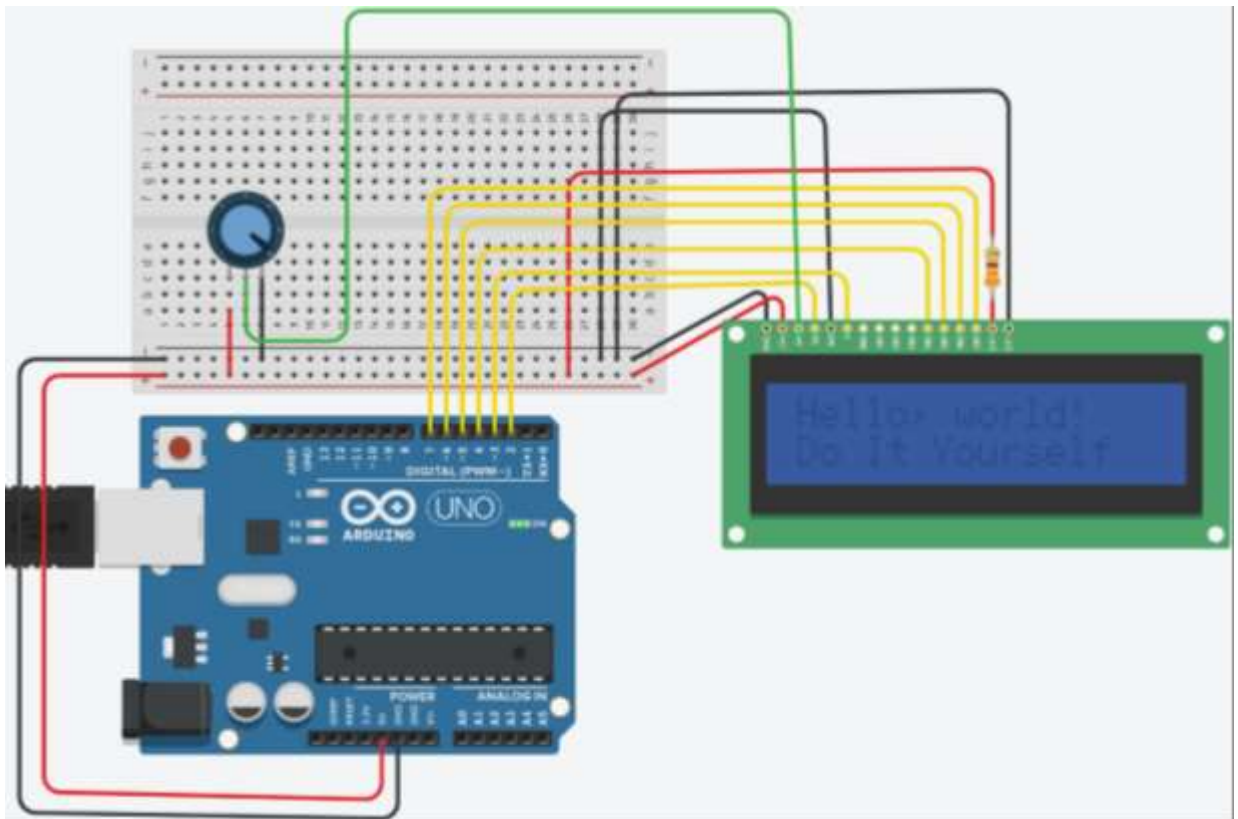


Рис. 156. Подключение ЖК-дисплея 16×2 с регулировкой контрастности  
Код при этом изменять не нужно.

### 6. Вывод значений температуры на ЖК-дисплей 16×2.

В следующем проекте показания датчика температуры будут выводиться на ЖК-дисплей 16×2.

*Соберите схему с использованием датчика температуры:*

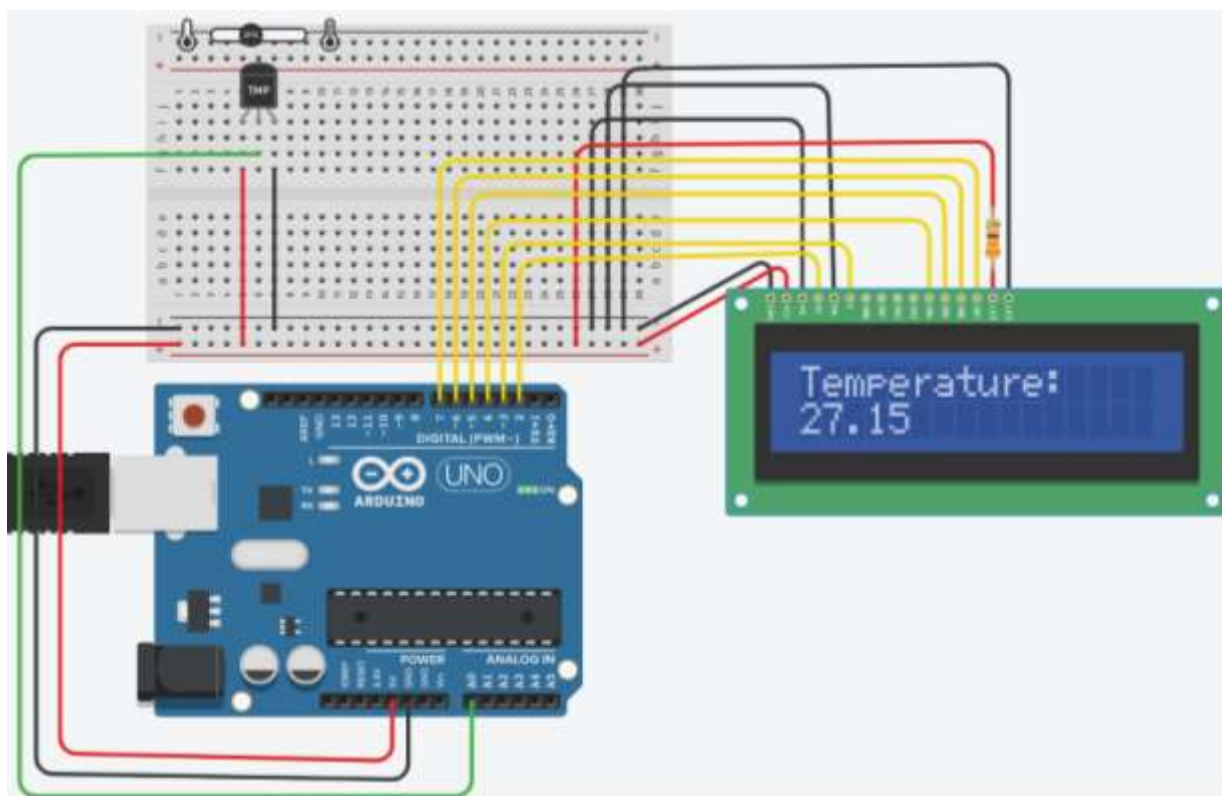


Рис. 157. Подключение ЖК-дисплея для вывода показаний датчика температуры

*Наберите код:*

```
// C++ code
#include <LiquidCrystal.h>
#define pinTemp A0
// Библиотека для работы с текстовым дисплеем
LiquidCrystal lcd (2,3,4,5,6,7);
void setup() {
  Serial.begin(9600);
  pinMode(pinTemp,INPUT);
  lcd.begin(16,2);
  lcd.print(«Temperature:»);
}
void loop()
{
  float temp=((analogRead(pinTemp)*(5.0/1024))-0.5)/0.01;
  lcd.setCursor(0, 2);
  lcd.print(temp);
  Serial.println (temp);
  delay(1000);
}
```

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

### **Задания**

1. Измените код программы «Секундомер», чтобы индикатор отсчитывал десятые доли секунды.
2. Поменяйте программу «Секундомер» так, чтобы вместо символа «0» отображался символ «А».
3. Дополните схему и программу «Секундомер» таким образом, чтобы сегмент-точка включался при прохождении четных чисел и выключался на нечетных.
4. Добавьте к схеме «Вывод значений температуры на ЖК-дисплей 16×2» потенциометр для регулировки контрастности дисплея.
5. Создайте новый проект, подключите фоторезистор и отобразите на ЖК-дисплее показания, снятые с фоторезистора.

### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.
2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.
3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.
4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.
5. Подвести итоги выполненной работы.

### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

### **Контрольные вопросы**

1. К какой ножке семисегментного индикатора подключается земля?
2. Каким образом хранятся закодированные символы цифр семисегментного индикатора?
3. Каким образом выводится символ на семисегментный индикатор?
4. Вспомните назначение выводов ЖК-дисплея.
5. Какие режимы передачи данных используются при функционировании ЖК-дисплея?
6. Назовите технические характеристики ЖК-дисплея.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими

предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

**Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

### **Лабораторная работа 13**

#### **Применение ультразвукового датчика в проектах Arduino**

**Цель работы:** исследование ультразвукового датчика для проектирования устройств на базе Arduino.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

#### **Теоретическая часть**

*Содержание:*

1. Общие сведения об ультразвуковом датчике.
2. Электронная рулетка.
3. Ультразвуковой датчик HC-SR04.
4. Изменение яркости светодиода в зависимости от показаний ультразвукового дальномера.

##### **1. Общие сведения об ультразвуковом датчике.**

Ультразвуковой датчик – это устройство, которое использует ультразвуковые волны для измерения расстояния до объекта. Ультразвуковые датчики, представляющие собой тандемы микрофона и динамика, посылают и принимают сверхвысокочастотные звуковые волны для определения расстояния до объекта.



Рис. 158. Ультразвуковой датчик

При этом сверхвысокочастотные звуковые волны ультразвукового датчика отражаются от поверхности объекта.

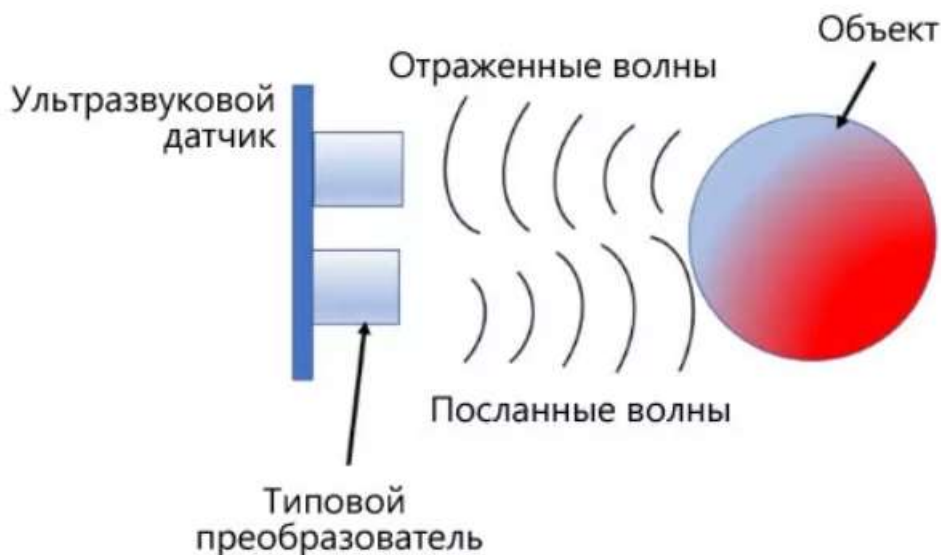


Рис. 159. Функционирование ультразвукового датчика

Скорость звука составляет 343 м/с или 1224 км/ч. Однако этот показатель может отличаться, в зависимости от среды, через которую проходит звук. Как правило, в газообразных веществах скорость звука меньше, чем в жидкостях, в свою очередь в жидкостях меньше, чем в твёрдых телах. Если брать газы и жидкости, то скорость распространения звука будет еще зависеть от температуры, увеличиваясь по мере нагревания.

Ультразвуковые датчики расстояния Ардуино очень востребованы в робототехнических проектах из-за своей относительной простоты, достаточной точности и доступности. Они могут быть использованы как приборы, помогающие объезжать препятствия, моделировать карту помещения и сигнализировать о приближении или удалении объектов.

## 2. Электронная рулетка.

В данной работе будет использоваться ультразвуковой датчик (PING))) от компании Parallax ввиду его наличия в среде Tinkercad. Он определяет дистанцию до ближайшего объекта, который находится перед ним (поисковый диапазон варьируется от 2 см до 3 м). Принцип его работы заключается в следующем – он отправляет пучок ультразвуковых волн, а затем ждет эха, создаваемого «отпружиниванием» этого сигнала от находящегося впереди

объекта. Arduino отправляет датчику короткий импульс, чтобы запустить процесс обнаружения, а затем через *тот же контакт* ждет импульса при помощи функции `pulseIn()`. Продолжительность ответного импульса эквивалентна времени, которое уходит на то, чтобы ультразвук дошел до объекта и вернулся к датчику. При помощи скорости звука это время можно конвертировать в дистанцию.

Так как в основу принципа действия положен ультразвук, то такой датчик не подходит для определения расстояния до звукопоглощающих объектов. Оптимальными для измерения являются предметы с ровной гладкой поверхностью.

*Принципиальная схема подключения ультразвукового датчика:*

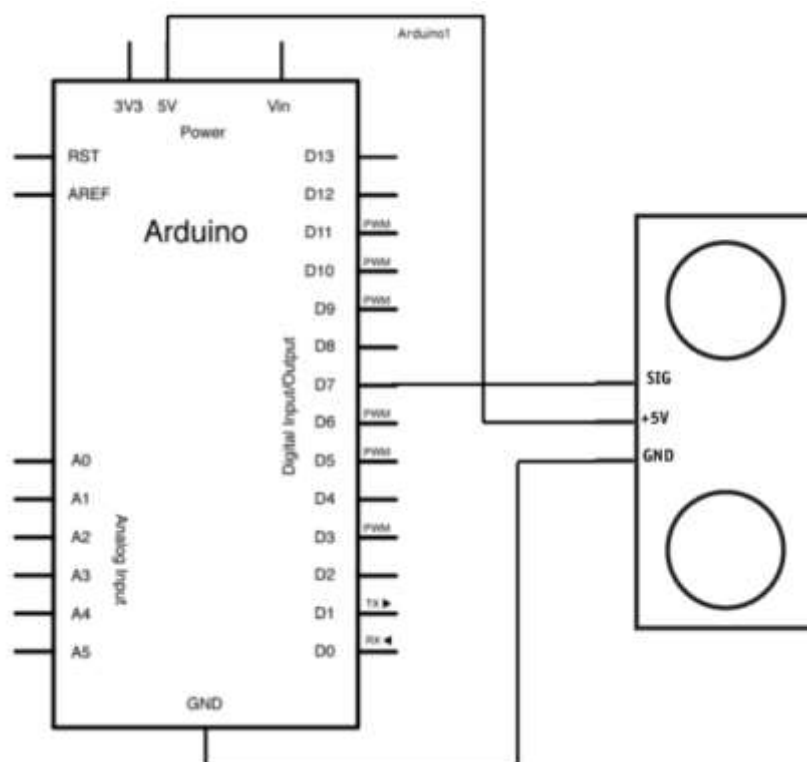


Рис. 160. Функциональная схема подключения ультразвукового датчика

*Компоненты, необходимые для реализации проекта:*

- плата Arduino Uno;
- макетная плата;
- ультразвуковой датчик расстояния (PING))) (Tinkercad) или ультразвуковой датчик HC-SR04;
- провода.

*Схема на макетной плате:*

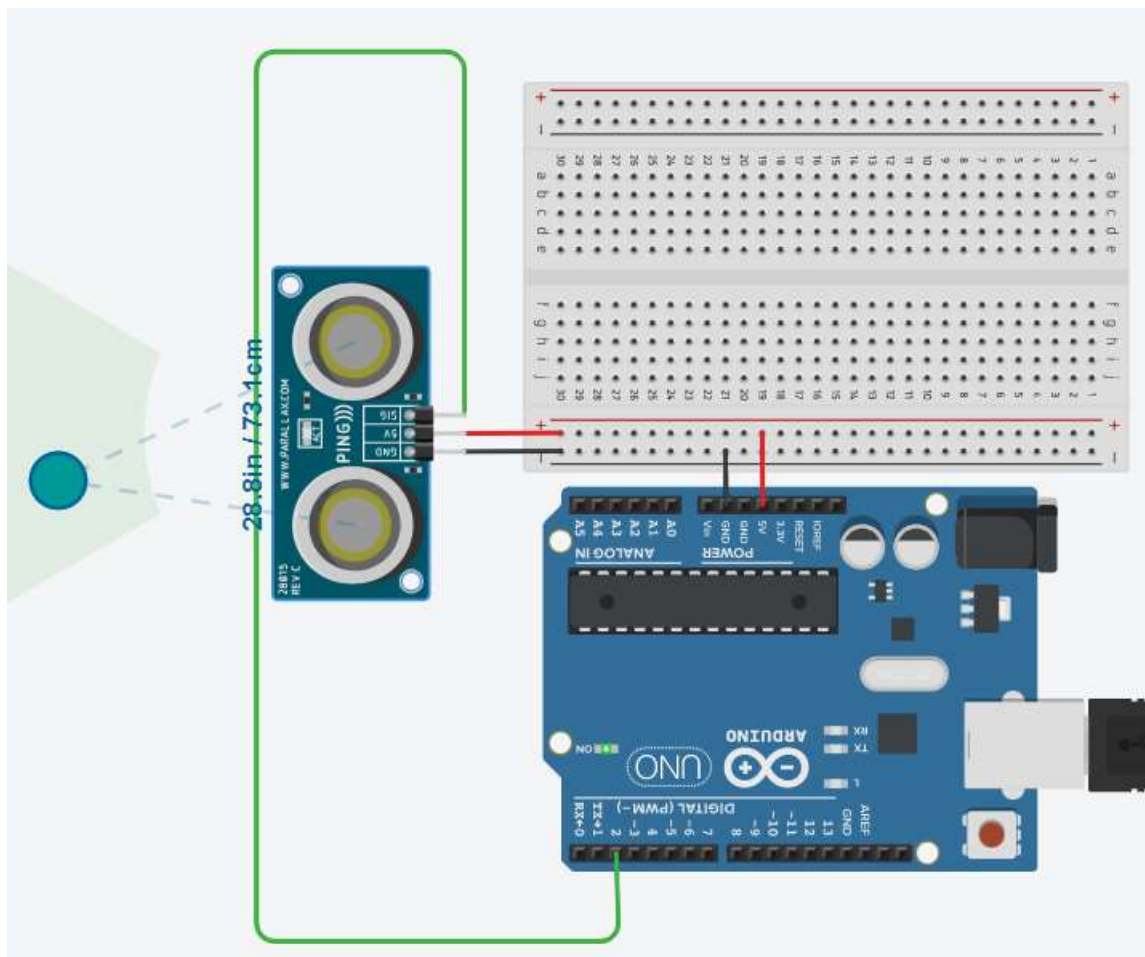


Рис. 161. Схема подключения 3-контактного ультразвукового датчика Контакт «Питание» PING))) подключен к +5V, контакт «Земля» подключен к заземлению, контакт «Сигнал» подключен к цифровому контакту 2.

*Далее наберите следующий код:*

```
// C++ code
const int pingPin = 2;
void setup() {
  Serial.begin(9600);
}
void loop()
{
  long duration, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(4);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(4);
  digitalWrite(pingPin, LOW);
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(1000);
}
```

```

}
long microsecondsToCentimeters(long microseconds)
{
return microseconds / 29 / 2;
}

```

**Функция `pulseIn()`** считывает длину сигнала на заданном порту (HIGH или LOW). Например, если задано считывание HIGH функцией `pulseIn()`, функция ожидает пока на заданном порту не появится HIGH. Когда HIGH получен, включается таймер, который будет остановлен, когда на порту вход/выхода будет LOW. Функция `pulseIn()` возвращает длину сигнала в микросекундах. Функция возвращает 0, если в течение заданного времени (таймаута) не был зафиксирован сигнал на порту.

Возможны некоторые погрешности в измерении длинных сигналов. Функция может измерять сигналы длиной от 10 микросекунд до 3 минут.

**Синтаксис функции `pulseIn()`:**

```

pulseIn(pin, value)
pulseIn(pin, value, timeout)

```

**Параметры функции `pulseIn()`:**

`pin`: номер порта вход/выхода, на котором будет ожидаться сигнал (int);  
`value`: тип ожидаемого сигнала – HIGH или LOW;  
`timeout` (опционально): время ожидания сигнала (таймаут) в микросекундах; по умолчанию – одна секунда (unsigned long).

**Возвращаемое значение функции `pulseIn()`:**

Длина сигнала в микросекундах или 0, если сигнал не получен до истечения таймаута. (unsigned long)

**Пример:**

```

int pin = 7;
unsigned long duration;
void setup()
{
pinMode(pin, INPUT);
}
void loop()
{
duration = pulseIn(pin, HIGH);
}
Запись return microseconds / 29 / 2;

```

Скорость звука составляет 343 м/с или 1224 км/ч при температуре 20 °С, 50% влажности и атмосферном давлении на уровне моря. Т.е. 34300 см/с или 0,0343 см/мкс. Соответственно, звуку потребуется 29,2 микросекунды, чтобы пройти один сантиметр (0,0343 см/мкс\*29,2мкс=1см). Поэтому можно получить расстояние исходя из времени между излучением и приемом импульса с помощью следующего уравнения:

*Расстояние (см) = время (мкс)/29,2/2.*

Далее применяется деление на 2, так как за это время звук проходит расстояние туда и обратно. Поэтому мы делим ее на два, чтобы получить реальную дистанцию до объекта.

Сенсор получает сигнал эха, и выдаёт расстояние, которое кодируется длительностью электрического сигнала на выходе датчика. Следующий импульс может быть излучён, только после исчезновения эха от предыдущего. Это время называется периодом цикла. Рекомендованный период между импульсами должен быть не менее 50 мс. Измеренное расстояние до объекта пропорционально ширине эха и может быть рассчитано по формуле, приведённой выше.

Таким образом, получена схема электронной рулетки, с помощью которой можно измерить расстояние до различных объектов.

Ультразвуковые дальномеры часто используются при создании роботов, в проектах автоматизации технологических объектов, в промышленности. Широкое применение они нашли в сфере строительства, ремонта, в производстве мебели.

### 3. Ультразвуковой датчик HC-SR04.

Далее рассмотрим схему сборки с 4-контактным ультразвуковым датчиком HC-SR04:

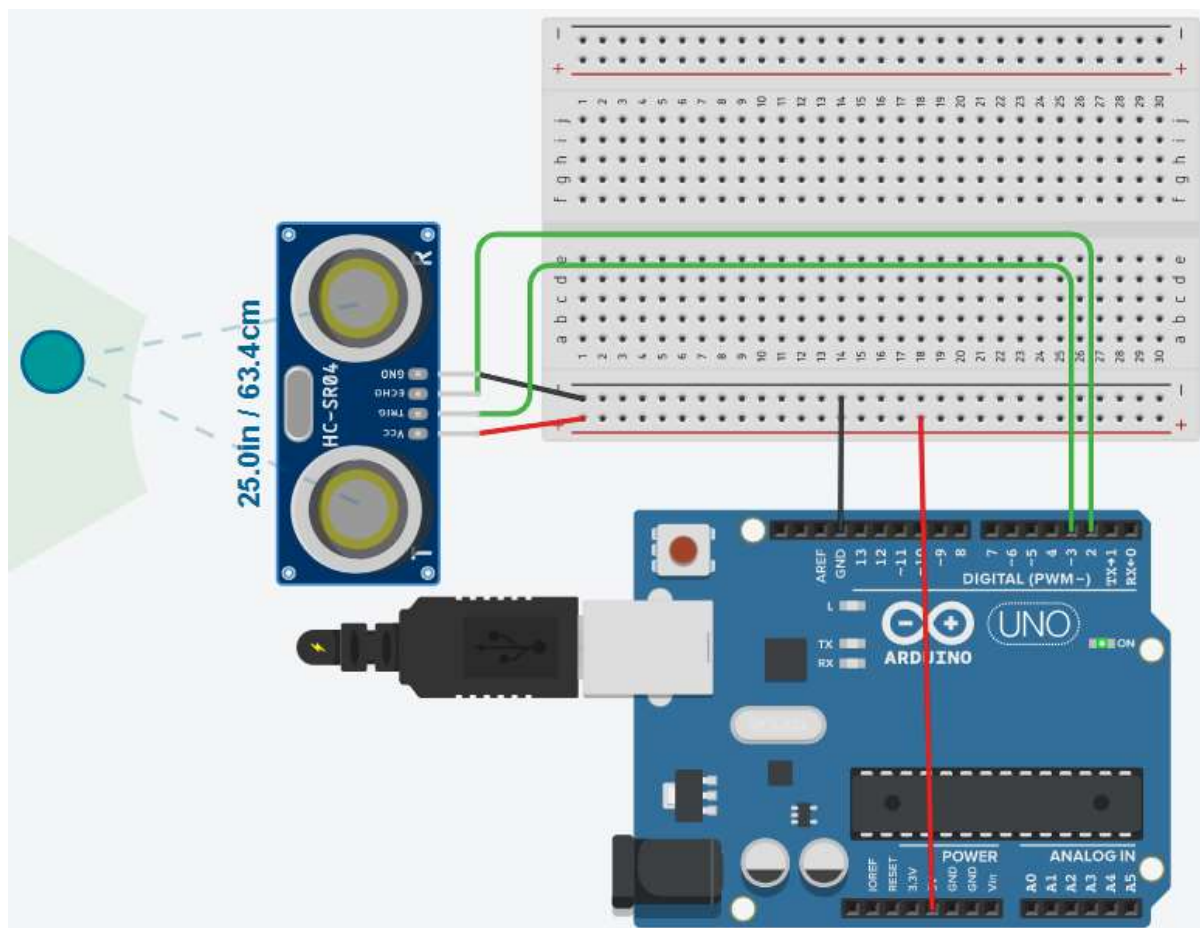


Рис. 162. Схема подключения 4-контактного ультразвукового датчика

Датчик HC-SR04 – это датчик расстояния, который использует ультразвук для определения расстояния до объекта в диапазоне от 2 до 450 см. Он отличается небольшими размерами, низким энергопотреблением, хорошей точностью и отличной ценой.

В случае с 4-контактным дальномером работать нужно будет отдельно с ножкой для подачи сигнала для запуска измерения (TRIG) и с ножкой для приема сигнала о получении отраженного эхо (ECHO). Код для работы датчика HC-SR04 выглядит следующим образом:

```
// C++ code
#define PIN_TRIG 3
#define PIN_ECHO 2
long duration, cm;
void setup() {
  // Инициализируем взаимодействие по последовательному порту
  Serial.begin (9600);
  //Определяем входы и выходы
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
}
```

```

}
void loop() {
  // Сначала генерируем короткий импульс длительностью 2-5 микросекунд.
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(5);
  digitalWrite(PIN_TRIG, HIGH);
  // Выставив высокий уровень сигнала, ждем около 10 микросекунд. В этот момент
  датчик будет посылать сигналы с частотой 40 КГц.
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);
  // Время задержки акустического сигнала на эхолотаторе.
  duration = pulseIn(PIN_ECHO, HIGH);
  // Теперь осталось преобразовать время в расстояние
  cm = (duration / 2) / 29.1;
  Serial.print(«Distance: «);
  Serial.print(cm);
  Serial.println(«cm»);

  // Задержка между измерениями для корректной работы скетча
  delay(250);
}

```

#### 4. Изменение яркости светодиода в зависимости от показаний ультразвукового дальномера.

Добавим в первую схему светодиод, яркость которого будет обратно пропорциональна значению расстояния, которое получено с ультразвукового датчика.

*Схема выглядит следующим образом:*

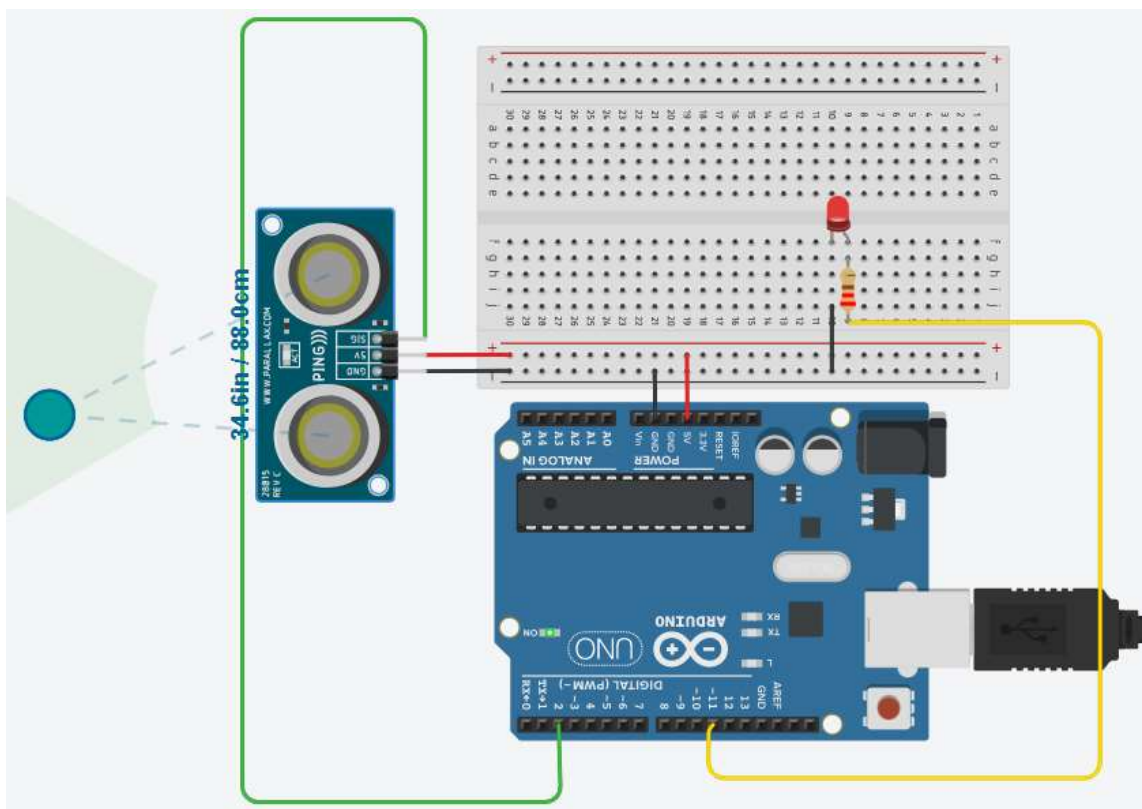


Рис. 163. Изменение яркости светодиода в зависимости от показаний  
ультразвукового дальномера

*Код программы:*

```
// C++ code
#define ledPin 11 // порт светодиода
const int pingPin = 2;
int brightness;
void setup() {
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT); //назначаем порт на выход
}
void loop()
{
  long duration, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(4);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(4);
  digitalWrite(pingPin, LOW);
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print(«cm»);
  Serial.println();
  if (cm <= 255) {
    brightness=255-cm;
    analogWrite(ledPin, brightness);
  }
  Serial.print(brightness);
  Serial.println();
  delay(1000);
}
long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
```

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

### Задания

1. Произведите изменения в программе «Электронная рулетка» таким образом, чтобы помимо расстояния в сантиметрах, программа также рассчитывала расстояние в дюймах и выводила полученные значения в монитор последовательного интерфейса.

2. Добавьте к схеме «Ультразвуковой датчик HC-SR04» светодиод, который включается при показаниях ультразвукового датчика, равных более 80 см.

3. Доработайте схему «Ультразвуковой датчик HC-SR04» таким образом, чтобы при показаниях ультразвукового датчика менее 50 см, включался предупреждающий звуковой сигнал.

4. Подключите RGB-светодиод к схеме «Электронная рулетка» для индикации значений, получаемых с ультразвукового датчика: если расстояние до объекта больше 50 см, то горит красный светодиод, расстояние составляет от 50 см до 100 см – синий светодиод, расстояние больше 100 см – горит зеленый светодиод.

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.

2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.

3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.

4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.

5. Подвести итоги выполненной работы.

#### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

#### **Контрольные вопросы**

1. Объясните принцип работы ультразвукового датчика.

2. Что называют ультразвуком?

3. Назовите основные отличия в программировании работы ультразвуковых датчиков расстояния (PING))) и HC-SR04.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

#### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

## Лабораторная работа 14

### Виды и назначение двигателей в устройствах на базе Arduino

**Цель работы:** изучение видов, назначения двигателей, способов их подключения и программирования для проектирования устройств на базе Arduino.

**Знания, умения и владения, приобретаемые обучающимися в результате освоения темы в рамках формируемых компетенций:**

*Знает:* способы организации образовательной деятельности обучающихся для развития интереса и мотивации к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ; компоненты информационной образовательной среды и возможности их дидактического применения, принципы, способы и методы организации информационной образовательной среды в рамках обучения информатике и ИКТ.

*Умеет:* корректно организовать учебную и исследовательскую деятельность обучающихся в образовательном процессе по информатике и ИКТ с использованием приемов, направленных на поддержание познавательного интереса; обосновывать и включать образовательные ресурсы в информационную образовательную среду, содержание и процесс обучения информатике и ИКТ.

*Владеет:* приемами развития познавательного интереса и повышения мотивации обучающихся к учебному предмету в рамках урочной и внеурочной деятельности по информатике и ИКТ с применением современных информационных технологий; умениями по проектированию электронных образовательных ресурсов по информатике и ИКТ, в частности, для их дальнейшего использования в электронном обучении на базе дистанционных технологий.

### Теоретическая часть

*Содержание:*

1. Принцип работы двигателей.
2. Двигатели постоянного тока.
3. Серводвигатели.
4. Подключение двигателя постоянного тока (DC motor).
5. Подключение сервопривода.

#### 1. Принцип работы двигателей.

Моторы используют электромагнитные силы для обеспечения движения, преобразования электроэнергии в механическую энергию. *Магнитные поля создают силу, которая может перемещать объекты.* Каждый магнит имеет магнитное поле с северным и южным полюсом. Если вы попытаетесь приблизить два северных полюса двух магнитов, они будут отталкиваться. То же самое произойдет, если вы попытаетесь приблизить два южных полюса. Если полюса одинаковые, они будут отталкиваться друг от друга. Если же вы приблизите северный полюс одного магнита к южному полюсу другого магнита, они притянутся с определенным усилием. То есть, противоположные полюса притягиваются друг к другу.

Электродвигатель использует свойства магнитов отталкиваться и притягиваться для генерации движения. В обычном электродвигателе два магнита: *постоянный и переменный*. Переменный магнит называется электромагнитом. Электромагнит создается с помощью пропускания электрического тока через проводник. *Постоянный магнит постоянно имеет магнитное поле (северный и южный полюса), а электромагнит генерирует магнитное поле только, когда через него пропускают электрический ток.* Сила магнитного

поля электромагнита может быть увеличена с помощью повышения тока, проходящего через проводник или методом формирования нескольких обмоток проводника.



Рис. 164. Моторы

В электродвигателе электромагнит устанавливается на ось таким образом, что он может *свободно вращаться внутри магнитного поля постоянного магнита*. В момент, когда через проводник проходит электрический ток, переменное магнитное поле взаимодействует со статическим магнитным полем магнита, возникают силы отталкивания и притяжения. *Это заставляет электромагнит вращаться, возникает движение.*

Основные узлы электродвигателя:

- постоянный магнит (магниты), в случае, когда он установлен неподвижно, называется **статором**;
- внутри статора есть катушка, которая установлена на оси и вращается, называется **ротором**.

## 2. Двигатели постоянного тока.

Двигатели постоянного тока (DC motor) во многом являются самыми простыми электродвигателями. Большинство коллекторных («щеточных») двигателей работают одинаково. Есть ротор и статор. Есть магниты на статоре и катушка на роторе с магнитным полем, которое генерируется с помощью подачи на нее силы тока. Есть щетки внутри мотора, которые заставляют двигаться ротор.

При использовании источника постоянного тока, для управления подобным двигателем практически ничего не надо. Скорость его вращения зависит от силы тока, которая поступает на катушки от источника питания к коллектору или коммутатору, который позволяет изменять направления вращения (он осуществляет переключение между обмотками).

## ДВИГАТЕЛЬ ПОСТОЯННОГО ТОКА

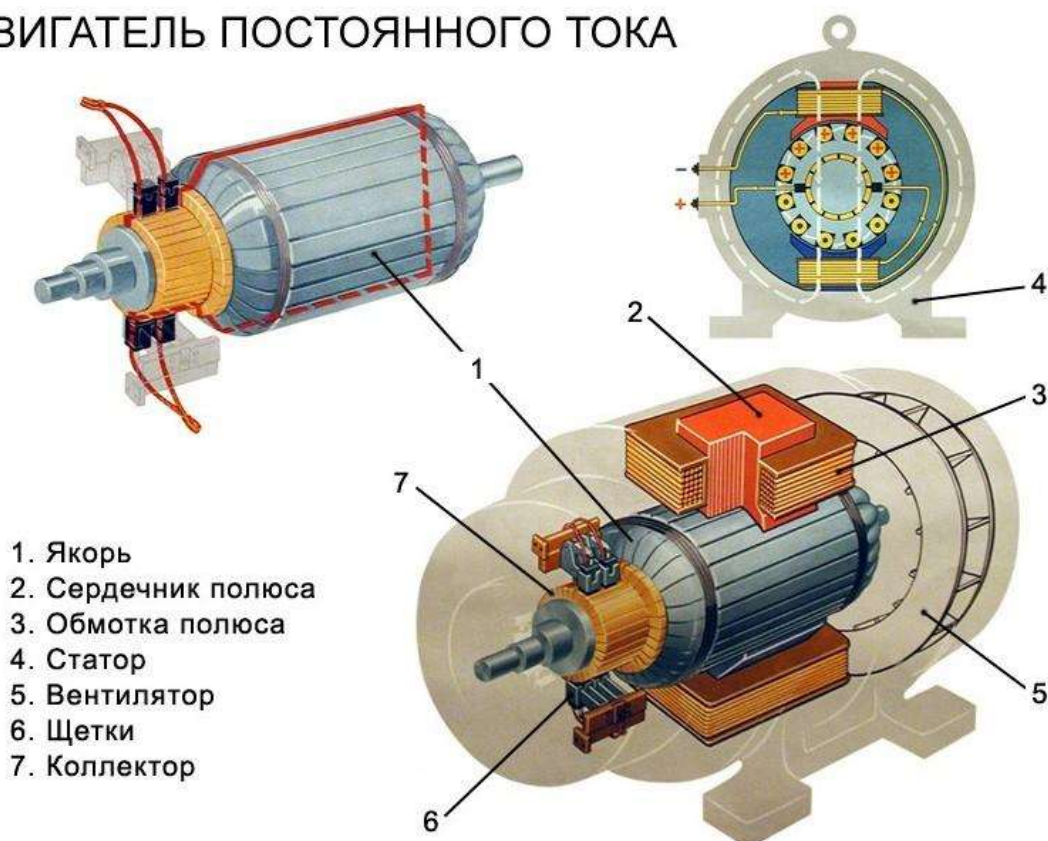


Рис. 165. Устройство двигателя постоянного тока

Для вращения оси двигателя в противоположном направлении, также достаточно подключить контакты от источника питания к двигателю наоборот.

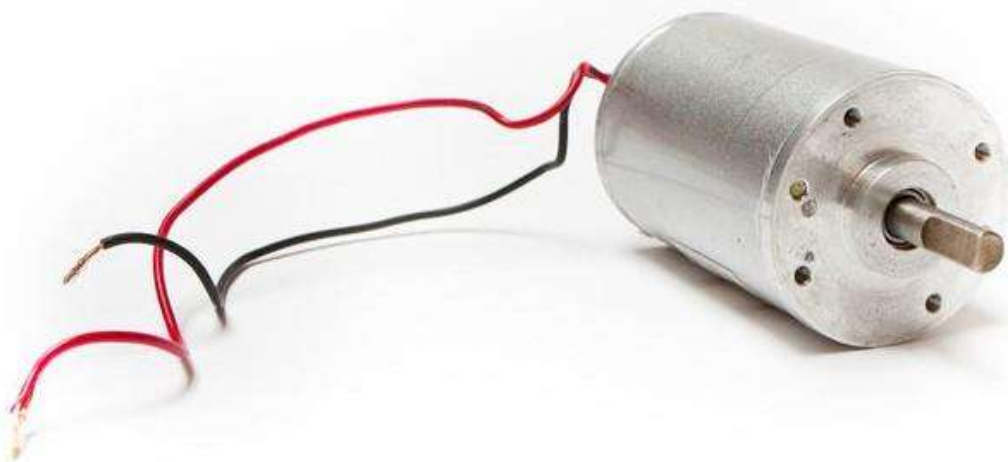


Рис. 166. Общий вид двигателя постоянного тока

### 3. Серводвигатели.

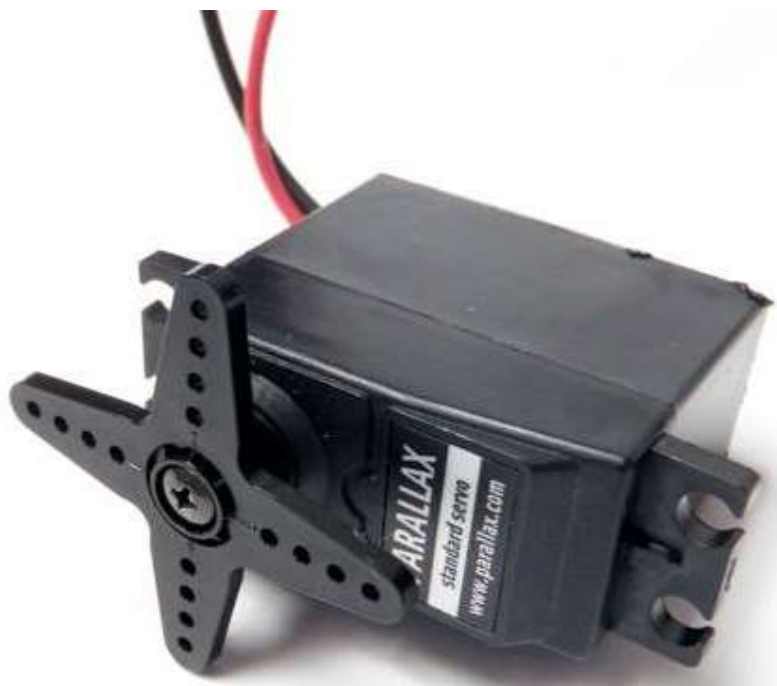


Рис. 167. Общий вид серводвигателя

Обычные серводвигатели используются для поворота и удерживания определенной позиции в диапазоне от 0 до 180 градусов. Благодаря этому они находят широкое применение в робототехнике, приводах позиционирования. В производстве серводвигатели используются в модулях рулевого управления автомобилями, лодками, в механизмах фокусировки современных видеокамер.

В большинстве случаев серводвигатели имеют три провода: питание, земля и сигнал. Обычно провод питания красного цвета, земля – черного или коричневого. Сигнал – желтый, оранжевый или белый.

В серводвигателях, которые, например, используются в системах управления радиоуправляемых машин, электродвигатель подключен к потенциометру. Стандартный приемник/передатчик посылает ШИМ-сигналы на серву. Электроника (небольшая плата-контроллер) внутри серводвигателя, переводит ширину импульса в положение. Когда на серву поступает сигнал к повороту, питание будет подаваться на мотор до тех пор, пока потенциометр не достигнет положения, которое будет соответствовать заданному предварительно через приемник/передатчик.

Сигнал управления является цифровым ШИМ сигналом с частотой 50 Гц. Каждые 20 миллисекунд подается цифровой импульс управления. Продолжительность (ширина) импульса находится в диапазоне от 1.0 миллисекунды до 2.0 миллисекунд. 1.5 – середина диапазона. Большая ширина импульса может использоваться для дополнительного хода, который выходит за нормальный рабочий диапазон.

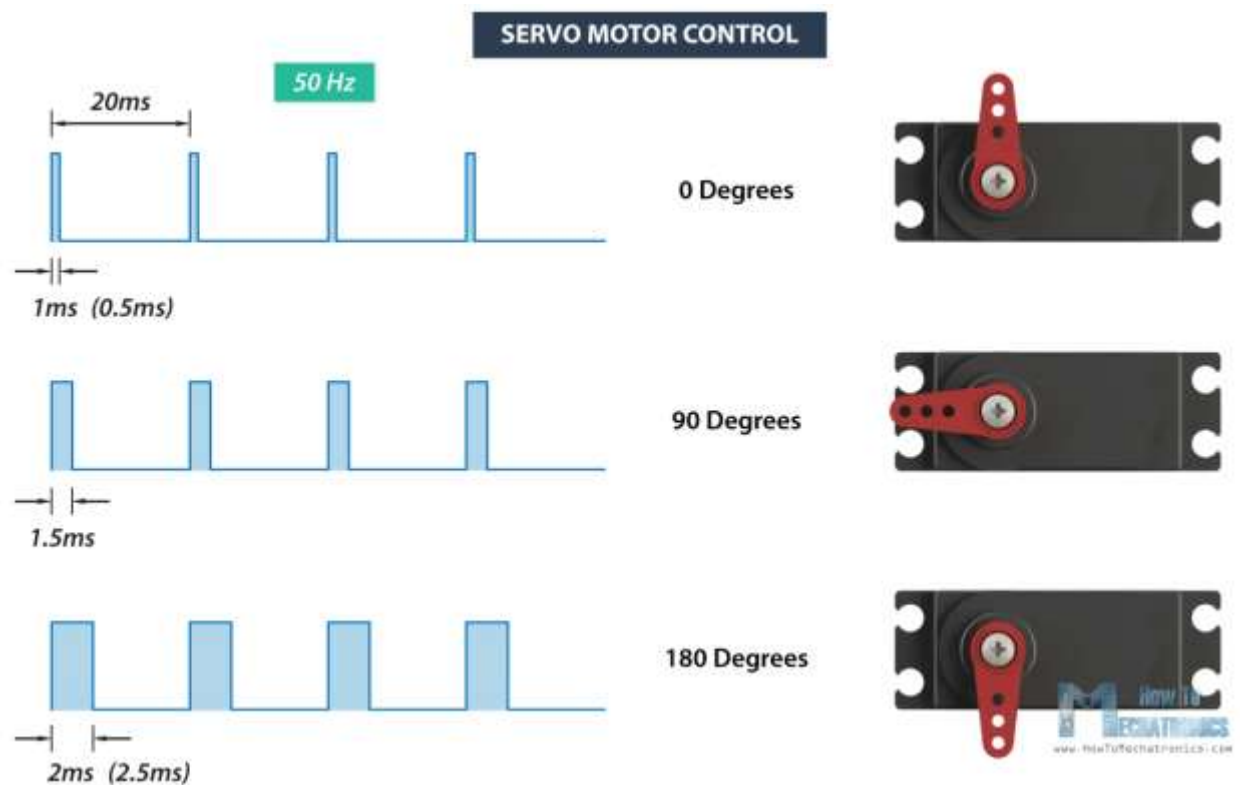


Рис. 168. Зависимость сигнала управления серводвигателя от ШИМ

Импульс шириной 1.5 миллисекунды обычно устанавливает серводвигатель в «нейтральное» положение или 90 градусов. Импульс шириной 1.25 миллисекунд может установить его в 0 градусов, а импульс 1.75 миллисекунды в 180 градусов. Физические ограничения и время для различных серводвигателей могут отличаться, но его нейтральное положение всегда соответствует 1.5 миллисекундам.

#### 4. Подключение двигателя постоянного тока (DC motor).

*Соберем простую схему со светодиодом и мотором:*

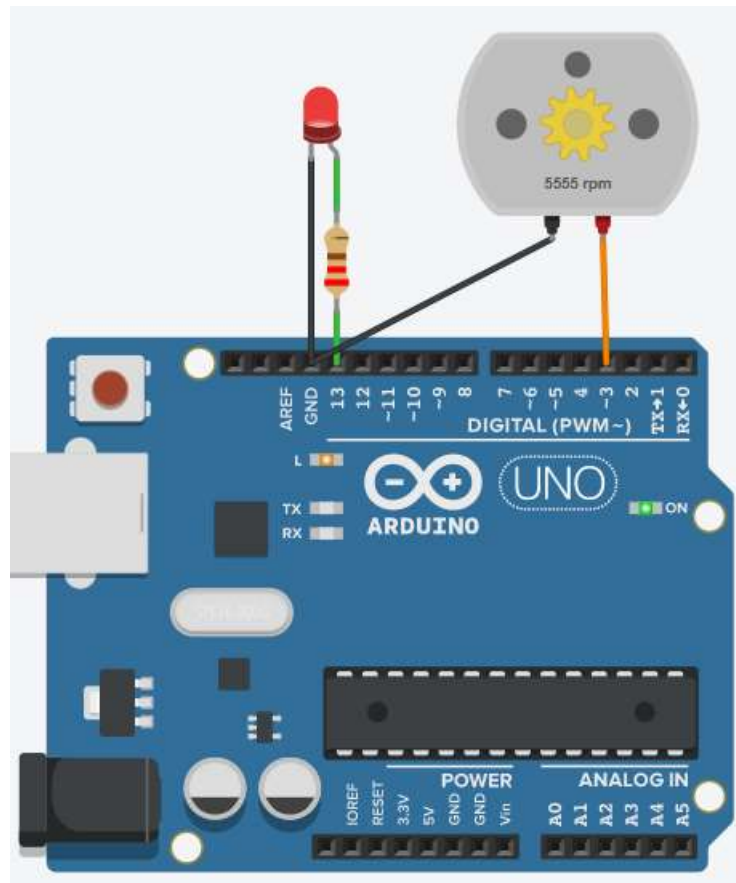


Рис. 169. Схема подключения двигателя постоянного тока

Наберите код:

```
// C++ code
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(3, OUTPUT);
}
void loop()
{
  digitalWrite(13, HIGH);
  analogWrite(3, 255);
}
```

В данной программе на пин 3, поддерживающий ШИМ, подается максимальное значение 255, что приводит к движению мотора с частотой вращения 5555 оборотов в минуту (revolutions per minute).

Далее поработаем с направлением вращения мотора. В следующей программе направление вращения мотора будем задавать с помощью кнопки: при нажатой кнопке мотор будет вращаться в противоположном направлении (против часовой стрелки), а при отсутствии нажатия – мотор вращается в прямом направлении (по часовой стрелке). Индикация светодиодов также будет указывать на направление вращения.

Схема выглядит следующим образом:

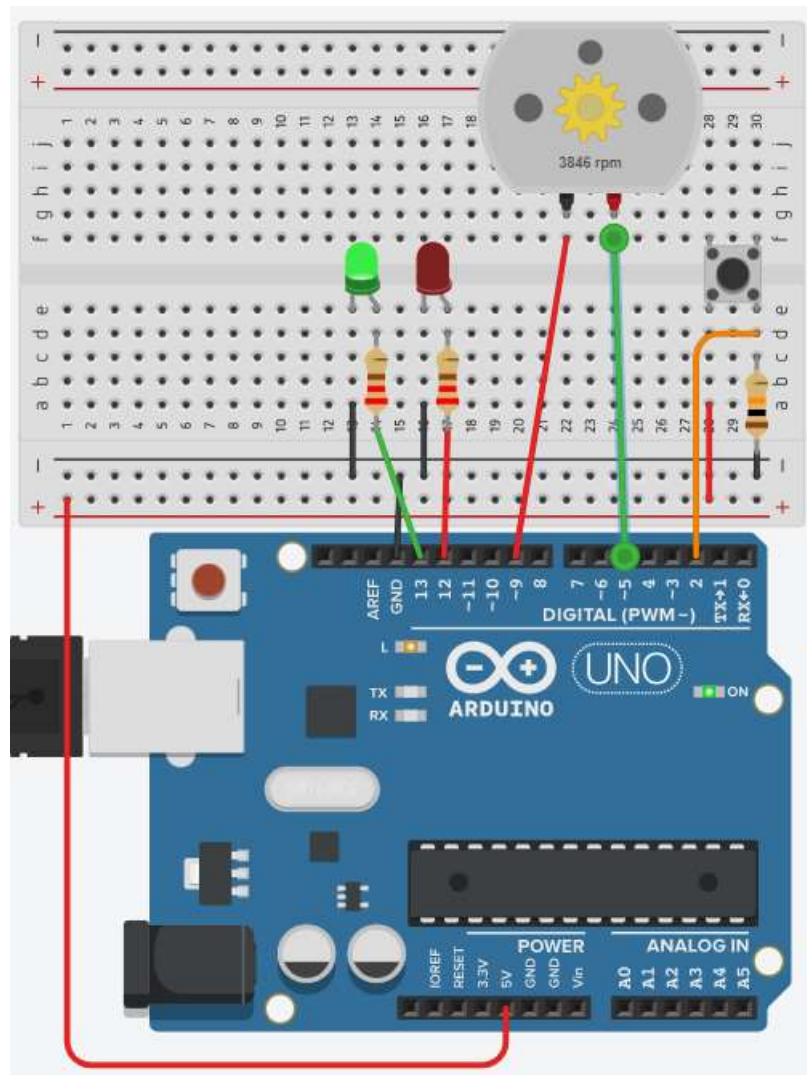


Рис. 170. Задание направления вращения мотора с помощью кнопки

*Kод:*

```
// C++ code
int buttonState=0;
void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(5, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  buttonState=digitalRead(2);
  if (buttonState==0) {
    digitalWrite(13,1);
    digitalWrite(12,0);
    digitalWrite(9,0);
    digitalWrite(5,1);
  }
}
```

```

else {
digitalWrite(12,1);
digitalWrite(13,0);
digitalWrite(9,1);
digitalWrite(5,0);
}
delay(100);
Serial.println(buttonState);
}

```

В предыдущей программе применялся условный оператор if ... else (полное ветвление). Далее рассмотрим условный оператор switch case на примере программы работы трех моторов.

*Схема:*

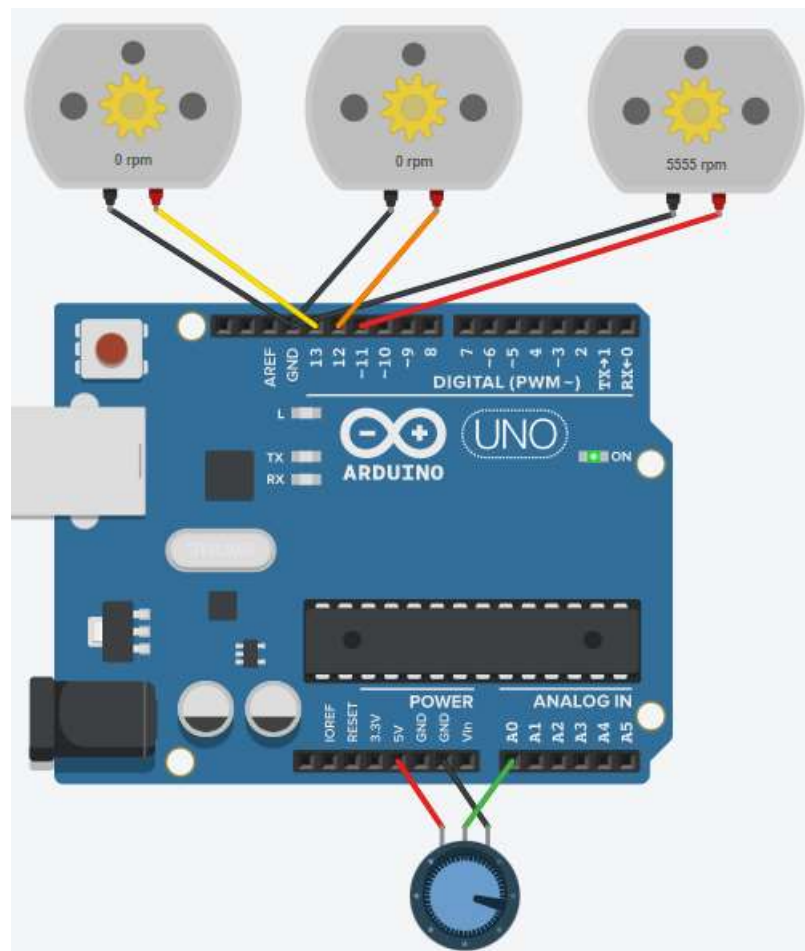


Рис. 171. Схема работы трех моторов

*Код:*

```

// C++ code
int sensorValue = 0;
void setup()
{
pinMode(A0, INPUT);
Serial.begin(9600);
pinMode(11,OUTPUT);
}

```

```

pinMode(12,OUTPUT);
pinMode(13,OUTPUT);
}
void loop()
{
sensorValue = analogRead(A0);
Serial.println(sensorValue);
delay(10);
switch(sensorValue){
case 225:
{
digitalWrite(11,1);
}
break;
case 777:
{
digitalWrite(11,1);
digitalWrite(12,1);
}
break;
case 880:
{
digitalWrite(11,1);
digitalWrite(12,1);
digitalWrite(13,1);
}
break;
default:
{
digitalWrite(11,0);
digitalWrite(12,0);
digitalWrite(13,0);
}
}
}
}

```

## **5. Подключение сервопривода.**

Соберем схему и напишем код для работы сервомотора.

*Схема:*

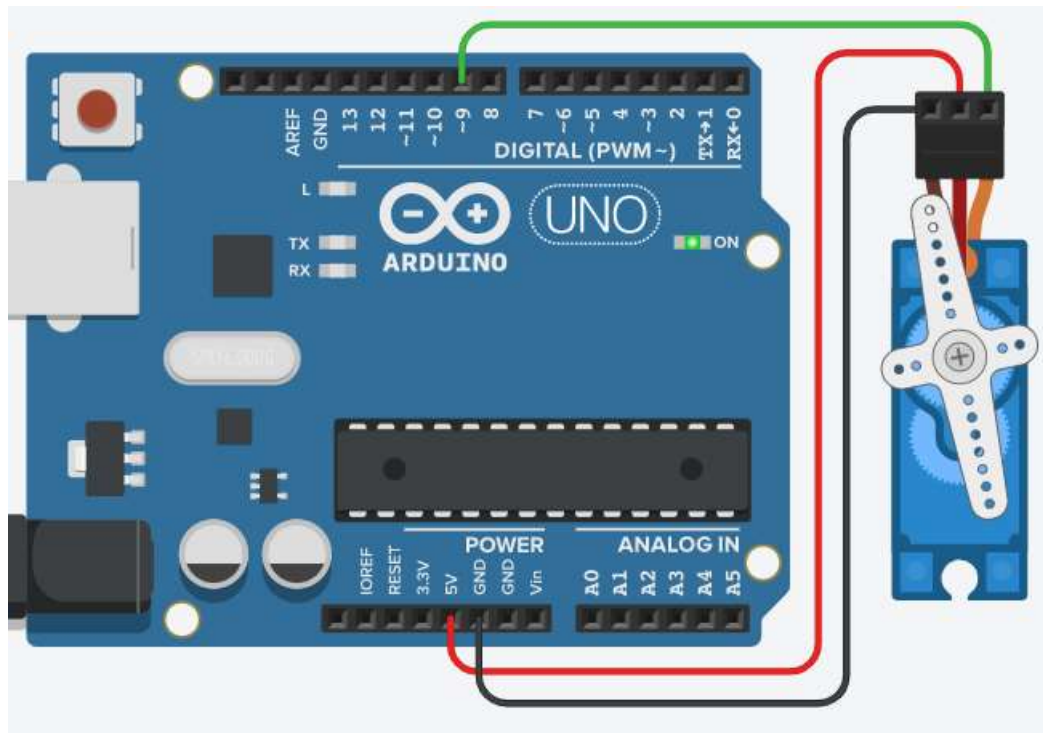


Рис. 172. Схема подключения сервопривода

Далее наберем следующий код:

```
// C++ code
void setup()
{
  pinMode(9,OUTPUT);
}
void loop()
{
  digitalWrite(9,HIGH);
  delayMicroseconds(600);
  digitalWrite(9,LOW);
  delayMicroseconds(2000);
}
```

**Функция `delayMicroseconds(time)`** формирует задержку на указанное количество микросекунд (мкс). 1 секунда = 1 000 000 мкс.

`time` принимает тип данных `unsigned int` и может приостановить выполнение на срок от 4 до 16383 мкс (меньше чем максимум для этого типа данных) с разрешением 4 мкс. Работает не на таймере, а на пропуске тактов процессора, поэтому может работать в прерывании и при отключенных прерываниях. Иногда не совсем корректно работает с переменными, поэтому нужно стараться использовать константы (`const` или просто число). Часто используется в библиотеках для эмуляции цифровых интерфейсов связи.

Замените в строчке кода

`delayMicroseconds(600);`

значение 600 на 2500. **Что изменилось в работе программы?**

Далее будем использовать предыдущую схему, но изменим код таким образом, чтобы сервомотор плавно поворачивался на 180° и также плавно возвращался в прежнее положение. Для этого используем следующий код с циклом `for`:

```
// C++ code
void setup()
```

```

{
  pinMode(9,OUTPUT);

}
void loop()
{
  for (int i=0; i<2500; i++)
  {
    digitalWrite(9,HIGH);
    delayMicroseconds(i);
    digitalWrite(9,LOW);
    delayMicroseconds(2000);
  }
  for (int i=0; i<2500; i++)
  {
    digitalWrite(9,HIGH);
    delayMicroseconds(2500-i);
    digitalWrite(9,LOW);
    delayMicroseconds(2000);
  }
}

```

Эту же программу реализуем **с помощью цикла while – цикла с предусловием** (схема остается прежней, код для обратного вращения мотора добавить самостоятельно):

```

// C++ code
int position=0;
void setup()
{
  pinMode(9,OUTPUT);
}
void loop()
{
  while (position<2500)
  {
    position++;
    digitalWrite(9,HIGH);
    delayMicroseconds(position);
    digitalWrite(9,LOW);
    delayMicroseconds(2000);
  }
}

```

Данную программу также можно реализовать **с помощью цикла do while – цикла с постусловием** (схема остается прежней):

```

// C++ code
int position=0;
void setup()
{
  pinMode(9,OUTPUT);
  Serial.begin(9600);
}

```

```

}
void loop()
{
do
{
position=position+10;
digitalWrite(9,HIGH);
delayMicroseconds(500+position);
digitalWrite(9,LOW);
delayMicroseconds(2500);
Serial.println(position);
}
while (position<2500);
delay(1000);
digitalWrite(12,LOW);
Serial.println(«stop»);
}

```

Для сборки следующей схемы используем потенциометр, который позволит регулировать угол поворота сервомотора посредством поворота ручки потенциометра:

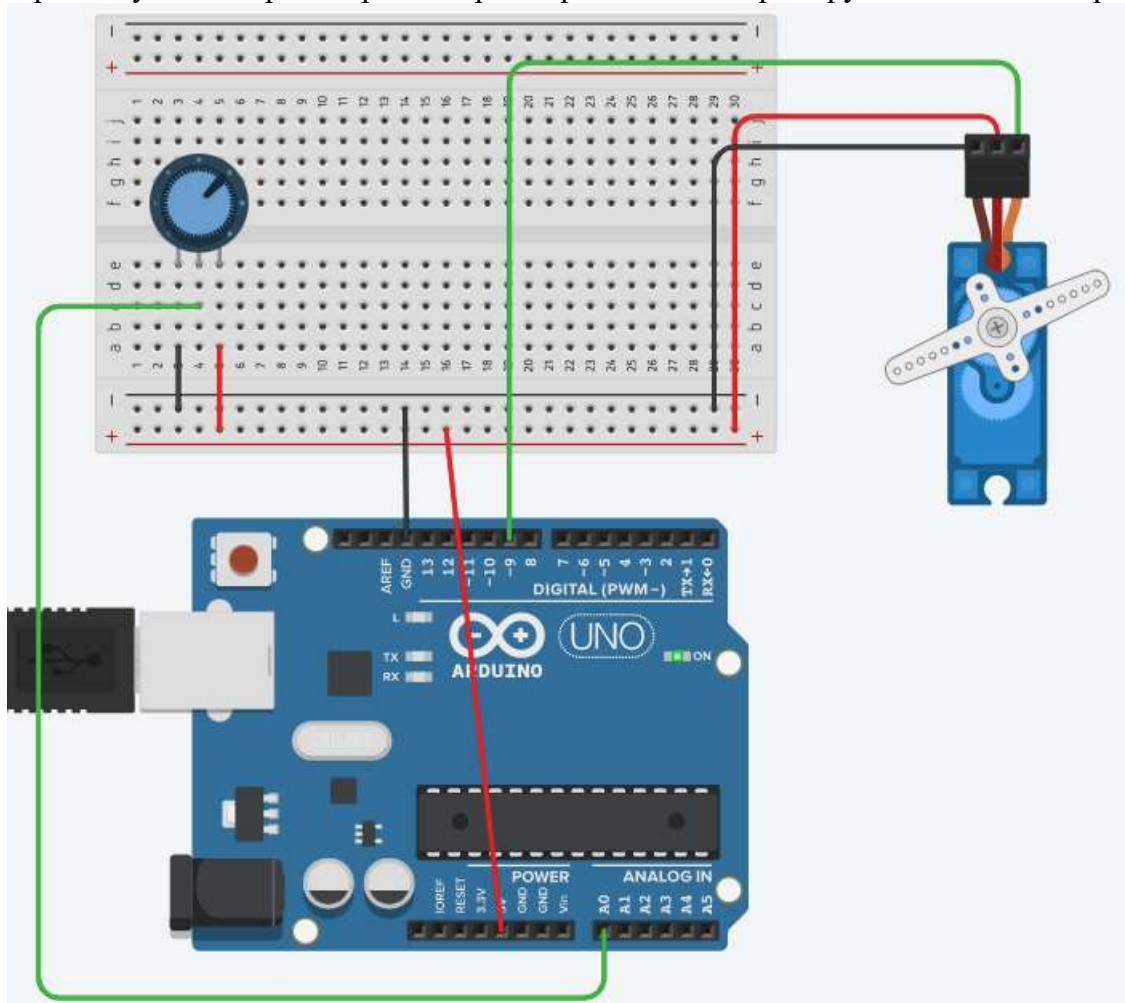


Рис. 173. Подключение потенциометра для регулировки угла поворота сервомотора  
Код работы:

```

// C++ code
#include <Servo.h>

```

```

Servo myservo;
int potpin=A0;
int val;
void setup()
{
  myservo.attach(9);
  Serial.begin(9600);
}
void loop()
{
  val=analogRead(potpin);
  val=map(val,0,1023,0,180);
  myservo.write(val);
  delay(15);
  Serial.println(val);
}

```

Данная программа обеспечивает управление серводвигателем, который подключен к 9-му ШИМ пину платы Arduino. Потенциометр, в свою очередь, подключен к аналоговому пину A0 и обеспечивает управление положением серводвигателя, посылая напряжение, пропорциональное сопротивлению на пин A0 платы Arduino. Скетч интерпретирует сигнал в импульсы, которые посылаются на серводвигатель. Двигатель обеспечивает поворот вала в соответствии с положением стрелки потенциометра.

В работе программы используется библиотека Servo. Для использования данной библиотеки в программе нужна команда `#include <Servo.h>`, затем нужно задать переменную типа servo:

```
#include <Servo.h>
```

```
Servo myservo;
```

**Приведем некоторые методы данной библиотеки:**

**attach()**

указывает вывод, к которому подключен сервопривод.

Синтаксис:

```
servo.attach(pin, min, max);
```

Параметры:

pin – обязательный параметр, цифровой пин, к которому подключен сигнальный провод сервопривода;

min – необязательный параметр, ширина импульса в микросекундах, соответствующая минимальному (угол 0 градусов) положению сервопривода. (по умолчанию 544);

max – необязательный параметр, ширина импульса в микросекундах, соответствующая максимальному (угол 180 градусов) положению сервопривода;

Возвращаемые значения: нет.

**write()**

поворачивает сервопривод на заданный угол. Для сервоприводов постоянного вращения устанавливает скорость и направление вращения.

Синтаксис:

```
servo.write(angle);
```

Параметры:

angle – обязательный параметр. Устанавливает угол от 0 до 180 градусов. При использовании сервопривода постоянного вращения значение 90 используется для

неподвижного состояния. Значение 0 для максимальной скорости кручения в одну сторону, а 180 для максимальной скорости кручения в другую сторону.

Возвращаемые значения: нет.

#### **writeMicroseconds()**

поворачивает сервопривод на угол, заданный в микросекундах. С сервоприводами постоянного вращения работает по таком же принципу, как и функция write().

Синтаксис:

```
servo.writeMicroseconds(ms);
```

Параметры:

ms – обязательный параметр, значение в микросекундах.

Возвращаемые значения: нет

#### **read()**

возвращает текущее положение сервопривода.

Синтаксис:

```
servo.read();
```

Параметры: нет

Возвращаемые значения: Int от 0 до 180.

При подключении серво через attach(pin) диапазон длины импульса устанавливается стандартный: 544.. 2400 мкс (задан в библиотеке). Если есть желание настроить серво идеально, чтобы она работала на весь диапазон – нужно попробовать покрутить её через writeMicroseconds() в крайних значениях диапазона и найти минимум и максимум, при которых серво «упирается».

**Оборудование и материалы.** Для выполнения лабораторной работы необходима компьютерная техника с возможностью подключения к сети «Интернет», в частности для работы в онлайн-среде моделирования Arduino – Tinkercad. Также необходима возможность подключения к электронной информационно-образовательной среде университета. Дополнительное оборудование – стартовый набор Arduino UNO, включающий плату и другие электронные компоненты для разработки устройств.

**Программное обеспечение** – операционная система Windows 8 или аналогичные, браузер для доступа к сети Интернет, онлайн-среда моделирования Tinkercad, среда программирования Arduino IDE.

#### **Задания**

1. К схеме «Подключение двигателя постоянного тока (DC motor)» добавьте второй мотор, который будет работать с мощностью 1/3 от мощности первого мотора. Отобразите значения соответствующих ШИМ-портов в мониторе последовательного интерфейса.

2. В программе с циклом *for* подберите такое значение времени (вместо значения 2500), чтобы сервомотор поворачивался на 90°.

3. С использованием библиотеки Servo и метода attach прокрутить сервопривод быстро на 180° и обратно.

4. С использованием библиотеки Servo и метода attach прокрутить сервопривод плавно на 90° и обратно.

#### **Указания по порядку выполнения работы**

1. Изучить теоретическое введение.

2. Ответить письменно на контрольные вопросы, приведенные в конце лабораторной работы.

3. Изучить и представить в среде программирования все примеры программ, рассмотренных в теоретической части.

4. Составить программы в соответствии с заданиями (если представлены варианты заданий, выполнить их в соответствии с заданным преподавателем вариантом). Подключить устройство и провести тестирование работы программы.

5. Подвести итоги выполненной работы.

### **Содержание отчета**

Отчет по лабораторной работе оформляется письменно в тетради. Защита осуществляется по отчёту, представленному студентом, и с демонстрацией выполненного задания на компьютере, а также с подключенным к персональному компьютеру роботом, выполняющим действия в соответствии с программой. Письменный отчет должен содержать:

- 1) Порядковый номер и название лабораторной работы.
- 2) Цель работы.
- 3) Ответы на вопросы, приведенные в конце работы в качестве контрольных.
- 4) Формулировку заданий, по которым осуществлялась разработка программ.

### **Контрольные вопросы**

1. Объясните принцип работы двигателей. Для каких функций предназначены постоянный и переменный магниты в составе двигателей? Как они называются?
2. Из каких компонентов состоит двигатель постоянного тока?
3. Как функционирует двигатель постоянного тока?
4. Для чего предназначены серводвигатели?
5. Разъясните принцип работы серводвигателей.

**Указания по технике безопасности.** Указания по технике безопасности при выполнении данной лабораторной работы являются общепринятыми при работе пользователей ПК. Приведем правила техники безопасности: 1) нельзя самостоятельно пытаться производить ремонт персонального компьютера, устанавливать или удалять программное обеспечение; 2) в случае возникновения неисправностей или каких-либо вопросов по работе с оборудованием необходимо обратиться к администратору или другим сотрудникам лаборатории; 3) не разрешено трогать электрические розетки металлическими предметами и нарушать другие правила при работе с электрооборудованием; 4) на рабочем месте необходимо соблюдать чистоту; 5) нельзя проносить в лабораторию пищу и напитки.

### **Список литературы, рекомендуемый к использованию по данной теме:**

основная литература – [3, 4, 6];

дополнительная литература – [3, 15, 18, 19, 20, 21, 26, 30].

### **СПИСОК ЛИТЕРАТУРЫ**

#### *Перечень основной литературы*

1. Дополнительное образование детей: история и современность: учеб. пособие для академического бакалавриата / отв. ред. А. В. Золотарева. – 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2018. – 267 с.
2. История и теория дополнительного образования: учебное пособие для академического бакалавриата / Б. А. Дейч [и др.]; под редакцией Б. А. Дейча. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2017. – 335 с.
3. Никитина, Т. В. Образовательная робототехника как направление инженерно-технического творчества школьников: Учебное пособие / Никитина Т. В. – Челябинск: Челябинский государственный педагогический университет, 2014. – 171 с.
4. Основы робототехники: учебное пособие / В. С. Глухов, А. А. Дикой, Р. А. Галустов, И. В. Дикая. – Армавир: Армавирский государственный педагогический университет, 2019. – 308 с.
5. Пономарева, Ю. С. Практикум по основам робототехники. Задачи для LEGO MINDSTORMS NXT и EV3: учебно-методическое пособие / Ю. С. Пономарева, Т. В. Шемелова. – Волгоград: Волгоградский государственный социально-педагогический университет, 2016. – 36 с.
6. Юревич Е.И. Основы робототехники / Е.И. Юревич. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2005. – 416 с.

#### *Перечень дополнительной литературы*

1. Абраухова В. В. Педагогика в системе дополнительного образования детей и взрослых: учебное пособие / В. В. Абраухова. – Москва; Берлин: Директ-Медиа, 2020. – 52 с. : ил.
2. Бабаева Ю.Д., Лейтес Н.С. Психология одаренности детей и подростков: учеб. пособие для студ. высш. и сред. пед. учеб. заведений/ под ред. Н.С. Лейтес. – М.: Академия, 2000. – 336с.
3. Блум Д. Изучаем Arduino®: инструменты и методы технического волшебства / Джереми Блум – С-П.: БХВПетербург, 2017. – 336 с.
4. Буйлова Л. Н. Технология разработки и оценки качества дополнительных общеобразовательных общеразвивающих программ: новое время – новые подходы. – М.: Пед. общ-во России, 2015 г. – 272 с.
5. Воротников С.А. Информационные устройства робототехнических систем: учеб. пособие / С.А. Воротников; под. ред. С.Л. Зенкевича, А.С. Ющенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 384 с.
6. Вязовов С.М., Калягина О.Ю., Слезин К.А. Соревновательная робототехника: приемы программирования в среде EV3: учеб.-практ. пособие. М.: Издательство «Перо», 2014. – 128 с.
7. Гончаревич И. Ф. Основы робототехники. Механизмы выдвижения и поворота робота-погрузчика с пневмоприводом: Методические рекомендации / Гончаревич И. Ф. – Москва: Московская государственная академия водного транспорта, 2014. – 62 с.
8. Злаказов А.С. Уроки Лего-конструирования в школе: методическое пособие / А.С. Злаказов, А.Г. Горшков, С.Г. Шевалдина. – М: БИНОМ. Лаборатория знаний. – 2011. – 120 с.
9. Иванов А. А. Основы робототехники: учеб. пособие / А.А. Иванов. – М.: Форум, 2012. – 224 с.: ил.
10. Киселёв М. М. Робототехника в примерах и задачах: курс программирования механизмов и роботов / М. М. Киселёв, М. М. Киселёв. – Москва: СОЛОН-ПРЕСС, 2017. – 136 с.
11. Кузьмина М. В. др. Робототехника в школе как ресурс подготовки инженерных кадров будущей России // сборник методических материалов для работников образования в условиях реализации Федеральных государственных образовательных стандартов (по итогам областных семинаров и курсов повышения квалификации по образовательной робототехнике) [Электронный ресурс]: ИРО Кировской области, 2017. – 179 с.
12. Кулаков Д. Б. Роботы и робототехника: лабораторный практикум: учебное пособие / Д. Б. Кулаков, Б. Б. Кулаков. – Москва: Российский университет дружбы народов, 2018. – 124 с.
13. Методика непрерывного профессионального развития кадров сферы дополнительного образования детей: учебное пособие / А. В. Золотарева [и др.] ; под научной редакцией А. В. Золотаревой. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2019. – 239 с.
14. Образовательная и соревновательная робототехника в условиях реализации Федеральных государственных образовательных стандартов: учебно-методическое пособие / Авт.-сост. М.В. Кузьмина; КО ГОАУ ДПО «ИРО Кировской области». – Киров: 2019. – 159 с.
15. Образовательная робототехника во внеурочной деятельности: учебно-методическое пособие / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011. – 96 с.
16. Образовательная робототехника на уроках информатики и физики в средней школе / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011. – 157 с. : ил.

17. Образовательная робототехника Электронный ресурс: Учебно-методический комплекс дисциплины / сост.: А. С. Соболевский, Э. Ф. Шарипова. – Челябинск: Челябинский государственный педагогический университет, 2014. – 32 с.

18. Проведение научно-ознакомительной практики с использованием модульной микропроцессорной платформы Arduino: учебно-методическое пособие / сост.: А. А. Афонин, А. С. Захаров; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль: ЯрГУ, 2018. – 60 с.

19. Пшихопов В.Х., Медведев М.Ю., Костюков В.А., Гайдук А.Р., Федоренко Р.В., Гуренко Б.В., Крухмалев В.А., Медведева Т.Н. Проектирование роботов и робототехнических систем: Учебное пособие – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 196 с.

20. Ревич Ю.В. Занимательная электроника / Юрий Ревич. – 4-е изд., перераб. и доп. – С-П.: БХВ-Петербург, 2017. – 639 с.

21. Рыбак, Л. А. Роботы и робототехнические комплексы: Учебное пособие / Рыбак Л. А. – Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2013. – 84 с.

22. Сборник учебно-методических материалов по образовательной робототехнике: Опыт образовательных учреждений Томской области / сост. О.С. Нетесова. – Томск: Издательство Томского государственного педагогического университета, 2016. – 172 с.

23. Семакин И.Г. Преподавание базового курса информатики в средней школе: метод. пособие / И.Г. Семакин, Т.Ю. Шеина. – 2-е изд., испр. и доп. – М.: БИНОМ. Лаборатория знаний, 2004. – 540 с.

24. Сергеева Т. Ф., Пронина Н. А., Сечкарева Е. В. Система работы с одаренными детьми: теория и практика. Ростов на/Д.: Изд-во «Феникс», 2011. – 284, [2] с.: ил.

25. Соревновательная деятельность региональных ресурсных центров технического творчества для детей и молодежи на базе социально ориентированных НКО на примере Программы «Робототехника» [Электронный ресурс]. – 2013. – 38 с.

26. Ступина Е.Е., Ступин А.А., Чупин Д.Ю., Каменев Р.В. Основы робототехники: учебное пособие. – Новосибирск: Агентство «Сибпринт», 2019. – 160 с

27. Филиппов С. А. Уроки робототехники. Конструкция. Движение. Управление / С. А. Филиппов; составители А. Я. Щелкунова. – 4-е изд. – Москва: Лаборатория знаний, 2022. – 191 с

28. Филиппов С.А. Робототехника для детей и родителей / С.А. Филиппов. – СПб.: Наука, 2013. – 319 с.

29. Щетинская А.И., Тавстуха О.Г., Болотова М.И. Теория и практика современного дополнительного образования детей: учебное пособие / А. И. Щетинская, О. Г. Тавстуха, М. И. Болотова; Мин-во образования и науки РФ; Оренбург, гос. пед. ун-т. – Оренбург: Изд-во ОПТУ, 2006. – 404 с.

30. Ярнольд С. Arduino для начинающих: самый простой пошаговый самоучитель / Стюарт Ярнольд – М.: Э, 2017. – 253 с.

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

1 <http://engjournal.ru/catalog/machin/hydro/682.html> - Кулаков Б.Б., Кулаков Д.Б., Беляев В.В. Антропоморфные роботы как новая сфера применения гидроприводов. Инженерный журнал: наука и инновации, 2013, вып. 4.

2 <http://fea.ru/news/6245> - центр компьютерного инжиниринга СПбПУ

3 <http://genius.pstu.ru/joomla/> - открытый информационный социально-образовательный Портал «Одарённые дети России XXI века»

4 <http://zakon-ob-obrazovanii.ru> - Федеральный закон "Об образовании в Российской Федерации" N 273-ФЗ от 29 декабря 2012 года с изменениями 2018 года.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для обучающихся по организации и проведению самостоятельной работы  
по дисциплине

**«ОРГАНИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ»  
(«ОБРАЗОВАТЕЛЬНАЯ РОБОТОТЕХНИКА»)»**

44.03.05 Педагогическое образование (с двумя профилями подготовки)  
направленность (профиль):  
«Информатика и математика»

**Ставрополь, 2026**

## СОДЕРЖАНИЕ

|                                                                                                               |    |
|---------------------------------------------------------------------------------------------------------------|----|
| 1. Общие положения                                                                                            | 3  |
| 2. Цель и задачи самостоятельной работы                                                                       | 4  |
| 3. Технологическая карта самостоятельной работы студента                                                      | 5  |
| 4. Порядок выполнения самостоятельной работы студентом                                                        | 5  |
| 4.1. Методические рекомендации по работе с учебной литературой                                                | 5  |
| 4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям                           | 7  |
| 4.3. Методические рекомендации по самопроверке знаний                                                         | 7  |
| 4.4. Методические рекомендации по написанию научных текстов (докладов, докладов, эссе, научных статей и т.д.) | 7  |
| 4.5. Методические рекомендации по выполнению исследовательских проектов                                       | 10 |
| 4.6. Методические рекомендации по подготовке к экзаменам и зачетам                                            | 13 |
| 5. Контроль самостоятельной работы студентов                                                                  | 14 |
| 6. Список литературы для выполнения СРС                                                                       | 14 |

## 1. Общие положения

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- написание докладов;
- подготовка к семинарам, практическим и лабораторным работам, их оформление;
- составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);
- выполнение учебно-исследовательских работ, проектная деятельность;
- подготовка практических разработок и рекомендаций по решению проблемной ситуации;
- выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;
- компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;
- выполнение курсовых работ (проектов) в рамках дисциплин;
- выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине **«Организация дополнительного образования»** направлена на формирование следующих **компетенций**:

| Код    | Формулировка:                                                                                                                                                                            |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ПК-3   | Способен формировать развивающую образовательную среду для достижения личностных, предметных и метапредметных результатов обучения средствами преподаваемых учебных предметов            |
| ПК-12* | Способен использовать современные цифровые технологии, специализированное программное обеспечение, методы искусственного интеллекта и машинного обучения в профессиональной деятельности |

*\* Номер компетенции изменить в соответствии с образовательной программой*

## 2. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование набора общенаучных, профессиональных и специальных компетенций будущего бакалавра по соответствующему направлению подготовки

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

### 3. Технологическая карта самостоятельной работы студента

| Коды реализуемых компетенций                      | Вид деятельности студентов                       | Средства и технологии оценки | Объем часов, в том числе (астр.) |                                    |       |
|---------------------------------------------------|--------------------------------------------------|------------------------------|----------------------------------|------------------------------------|-------|
|                                                   |                                                  |                              | СРС                              | Контактная работа с преподавателем | Всего |
| ПК-3 (ПК-3.1, ПК-3.2)                             | Самостоятельное изучение литературы и источников | Собеседование                | 29,5                             | 0.5                                | 30    |
| ПК-3 (ПК-3.1, ПК-3.2)<br>ПК-12 (ПК-12.1, ПК-12.2) | Подготовка к лабораторным занятиям               | Защита ЛР                    | 29,5                             | 0.5                                | 30    |
| ПК-12 (ПК-12.1, ПК-12.2)                          | Написание реферата/доклада                       | Защита доклада               | 23,5                             | 0.5                                | 24    |
| Итого за семестр                                  |                                                  |                              | 82,5                             | 1,5                                | 84    |
| Итого                                             |                                                  |                              | 82,5                             | 1,5                                | 84    |

### 4. Порядок выполнения самостоятельной работы студентом

#### 4.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста**:

- информационно-поисковый (задача – найти, выделить искомую информацию)
- усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)
- аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)
- творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

*Основные виды систематизированной записи прочитанного:*

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

*Методические рекомендации по составлению конспекта:*

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

#### *4.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям*

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

#### *4.3. Методические рекомендации по самопроверке знаний*

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

#### *4.4. Методические рекомендации по написанию научных текстов (докладов, докладов, эссе, научных статей и т.д.)*

Перед тем, как приступить к написанию научного текста, важно разобраться, какова истинная цель вашего научного текста - это поможет вам разумно распределить свои силы и время.

Во-первых, сначала нужно определиться с идеей научного текста, а для этого необходимо научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными

идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного). Во-вторых, научиться организовывать свое время, ведь, как известно, свободное (от всяких глупостей) время – важнейшее условие настоящего творчества, для него наконец-то появляется время. Иногда именно на организацию такого времени уходит немалая часть сил и талантов.

Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно (чтобы и самому понятно было), а также стремясь структурировать свой текст. Каждый раз надо представлять, что ваш текст будет кто-то читать и ему захочется сориентироваться в нем, быстро находить ответы на интересующие вопросы (заодно представьте себя на месте такого человека). Понятно, что работа, написанная «сплошным текстом» (без заголовков, без выделения крупным шрифтом наиболее важным мест и т. п.), у культурного читателя должна вызывать брезгливость и даже жалость к автору (исключения составляют некоторые древние тексты, когда и жанр был иной и к текстам относились иначе, да и самих текстов было гораздо меньше – не то, что в эпоху «информационного взрыва» и соответствующего «информационного мусора»).

Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Доклад – это самостоятельное исследование студентом определенной проблемы, комплекса взаимосвязанных вопросов.

Доклад не должна составляться из фрагментов статей, монографий, пособий. Кроме простого изложения фактов и цитат, в доклад е должно проявляться авторское видение проблемы и ее решения.

Рассмотрим основные этапы подготовки  
а студентом.

Выполнение доклада начинается с выбора темы.

Затем студент приходит на первую консультацию к руководителю, которая предусматривает:

- обсуждение цели и задач работы, основных моментов избранной темы;
- консультирование по вопросам подбора литературы;
- составление предварительного плана.

Следующим этапом является работа с литературой. Необходимая литература подбирается студентом самостоятельно.

После подбора литературы целесообразно сделать рабочий вариант плана работы. В нем нужно выделить основные вопросы темы и параграфы, раскрывающие их содержание.

Составленный список литературы и предварительный вариант плана уточняются, согласуются на очередной консультации с руководителем.

Затем начинается следующий этап работы – изучение литературы. Только внимательно читая и конспектируя литературу, можно разобраться в основных вопросах темы и подготовиться к самостоятельному (авторскому) изложению содержания доклада. Конспектируя первоисточники, необходимо отразить основную идею автора и его позицию по исследуемому вопросу, выявить проблемы и наметить задачи для дальнейшего изучения данных проблем.

Систематизация и анализ изученной литературы по проблеме исследования позволяют студенту написать работу.

Рабочий вариант текста доклада предоставляется руководителю на проверку. На основе рабочего варианта текста руководитель вместе со студентом обсуждает возможности доработки текста, его оформление. После доработки доклад сдается на кафедру для его оценивания руководителем.

*Требования к написанию доклада*

Написание 1 доклада является обязательным условием выполнения плана СРС по любой дисциплине профессионального цикла.

Тема доклада может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

Доклад должен быть написан научным языком.

Объем доклада должен составлять 20-25 стр.

*Структура доклада:*

- Введение (не более 3-4 страниц). Во введении необходимо обосновать выбор темы, ее актуальность, очертить область исследования, объект исследования, основные цели и задачи исследования.

- Основная часть состоит из 2-3 разделов. В них раскрывается суть исследуемой проблемы, проводится обзор мировой литературы и источников Интернет по предмету исследования, в котором дается характеристика степени разработанности проблемы и авторская аналитическая оценка основных теоретических подходов к ее решению. Изложение материала не должно ограничиваться лишь описательным подходом к раскрытию выбранной темы. Оно также должно содержать собственное видение рассматриваемой проблемы и изложение собственной точки зрения на возможные пути ее решения.

- Заключение (1-2 страницы). В заключении кратко излагаются достигнутые при изучении проблемы цели, перспективы развития исследуемого вопроса

- Список использованной литературы (не меньше 10 источников), в алфавитном порядке, оформленный в соответствии с принятыми правилами. В список использованной литературы рекомендуется включать работы отечественных и зарубежных авторов, в том числе статьи, опубликованные в научных журналах в течение последних 3-х лет и ссылки на ресурсы сети Интернет.

- Приложение (при необходимости).

*Требования к оформлению:*

- текст с одной стороны листа;
- шрифт Times New Roman;
- кегль шрифта 14;
- межстрочное расстояние 1,5;
- поля: сверху 2,5 см, снизу – 2,5 см, слева - 3 см, справа 1,5 см;
- доклад должен быть представлен в сброшюрованном виде.

*Порядок защиты доклада:*

Защита доклада проводится на практических занятиях, после окончания работы студента над ним и исправления всех недочетов, выявленных преподавателем в ходе консультаций. На защиту доклада отводится 5-7 минут времени, в ходе которого студент должен показать свободное владение материалом по заявленной теме. При защите доклада приветствуется использование мультимедиа-презентации.

*Оценка доклада*

Доклад оценивается по следующим критериям:

- соблюдение требований к его оформлению;
- необходимость и достаточность для раскрытия темы приведенной в тексте доклада информации;
- умение студента свободно излагать основные идеи, отраженные в докладе;
- способность студента понять суть задаваемых преподавателем и сокурсниками вопросов и сформулировать точные ответы на них.

*Критерии оценки:*

Оценка «отлично» выставляется студенту, если в докладе студент исчерпывающе, последовательно, четко и логически стройно излагает материал; свободно справляется с задачами, вопросами и другими видами применения знаний; использует для написания

доклада современные научные материалы; анализирует полученную информацию; проявляет самостоятельность при написании доклада.

*Оценка «хорошо»* выставляется студенту, если качество выполнения доклада достаточно высокое. Студент твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопросы по теме доклада.

*Оценка «удовлетворительно»* выставляется студенту, если материал доклада излагается частично, но пробелы не носят существенного характера, студент допускает неточности и ошибки при защите доклада, дает недостаточно правильные формулировки, наблюдаются нарушения логической последовательности в изложении материала.

*Оценка «неудовлетворительно»* выставляется студенту, если он не подготовил доклад или допустил существенные ошибки. Студент неуверенно излагает материал доклада, не отвечает на вопросы преподавателя.

#### *Описание шкалы оценивания*

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

| Уровень выполнения контрольного задания | Рейтинговый балл (в % от максимального балла за контрольное задание) |
|-----------------------------------------|----------------------------------------------------------------------|
| Отличный                                | 100                                                                  |
| Хороший                                 | 80                                                                   |
| Удовлетворительный                      | 60                                                                   |
| Неудовлетворительный                    | 0                                                                    |

#### *4.5. Методические рекомендации по выполнению исследовательских проектов*

Исследовательская проектная работа – это групповая работа, для выполнения которой необходим выбор и приложение научной методики к поставленной задаче, получение собственного теоретического или экспериментального материала, на основании которого необходимо провести анализ и сделать выводы об исследуемом явлении. Выполнение проекта – это всегда коллективная, творческая практическая работа, предназначенная для получения определенного продукта или научно-технического результата. Такая работа подразумевает четкое, однозначное формирование поставленной задачи, определение сроков выполнения намеченного, определение требований к разрабатываемому объекту.

Выполнение 1 группового проекта является обязательным условием выполнения самостоятельной работы по любой дисциплине профессионального цикла. Тема проектного задания может быть выбрана студентом из предложенных в рабочей программе или фонде оценочных средств дисциплины, либо определена самостоятельно, исходя из интересов студента (в рамках изучаемой дисциплины). Выбранную тему необходимо согласовать с преподавателем.

#### *Требования по выполнению и оформлению проекта*

При выполнении проекта приветствуется работа в группе (2-3 человека). Проект – это исследовательская работа, в ходе которой студенты должны продемонстрировать владение навыками научного исследования, умения проводить анализ, обобщать информацию, делать выводы, предлагать свои решения проблемы, рассматриваемой в проекте.

При подготовке материалов проекта студенты должны продемонстрировать владение современными методами компьютерной обработки данных.

*Критерии оценки работы участника проекта.*

Для каждого из участников проекта оцениваются:

- профессиональные теоретические знания в соответствующей области;
- умение работать со справочной и научной литературой, осуществлять поиск необходимой информации в Интернет;
- умение работать с техническими средствами;
- умение пользоваться соответствующими выполняемому проекту информационными технологиями;
- умение готовить материалы проекта для презентации: составлять и редактировать тексты, формировать презентацию проекта;
- умение работать в команде;
- умение публично представлять результаты собственной деятельности;
- коммуникабельность, инициативность, творческие способности.

*Критерии выставления оценки участникам проекта*

| Оценка    | Профессиональные компетенции                                                                                                                                                                                       | Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий | Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)                    | Отчетность                                                                   |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| «Отлично» | Работа выполнена на высоком профессиональном уровне. Представленный материал в основном фактически верен, допускаются негрубые фактические неточности. Студент свободно отвечает на вопросы, связанные с проектом. | Технические средства и информационные технологии освоены и использованы для реализации проекта полностью                     | Студент проявил инициативу, творческий подход, способность к выполнению сложных заданий, навыки работы в коллективе, организационные способности. | Проект представлен полностью и в срок.                                       |
| «Хорошо»  | Работа выполнена на достаточно высоком профессиональном уровне. Допущено до 4–5 фактических ошибок. Студент отвечает на вопросы, связанные с проектом, но недостаточно полно.                                      | Обнаруживаются некоторые ошибки в использовании соответствующих технических средств и информационных технологий              | Студент достаточно полно, но без инициативы и творческих находок выполнил возложенные на него задачи.                                             | Проект представлен достаточно полно и в срок, но с некоторыми недоработками. |

| Оценка                | Профессиональные компетенции                                                                                                                                                                                   | Компетенции, связанные с использованием соответствующих выполняемому проекту технических средств и информационных технологий    | Иные универсальные компетенции (коммуникабельность, инициативность, умение работать в «команде», управленческие навыки и т.д.)     | Отчетность                                                           |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|
| «Удовлетворительно»   | Уровень недостаточно высок. Допущено до 8 фактических ошибок. Студент может ответить лишь на некоторые из заданных вопросов, связанных с проектом.                                                             | Обнаруживает недостаточное владение навыками работы с техническими средствами и соответствующим и информационным и технологиями | Студент выполнил большую часть возложенной на него работы.                                                                         | Проект сдан со значительным опозданием (более недели) и не полностью |
| «Неудовлетворительно» | Работа не выполнена или выполнена на низком уровне. Допущено более 8 фактических ошибок. Ответы на связанные с проектом вопросы обнаруживают непонимание предмета и отсутствие ориентации в материале проекта. | Навыков работы с техническими средствами нет, информационные технологии не освоены                                              | Студент практически не работал, не выполнил свои задачи или выполнил лишь отдельные не существенные поручения в групповом проекте. | Проект не сдан.                                                      |

*Студенты должны:* защитить проект в режиме презентации, предъявить файлы выполненного проекта, уметь рассказать о технологиях, использованных ими при выполнении проекта, дать оценку работы каждого члена группы (если проект групповой).

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

| Уровень выполнения контрольного задания | Рейтинговый балл (в % от максимального балла за контрольное задание) |
|-----------------------------------------|----------------------------------------------------------------------|
| Отличный                                | 100                                                                  |
| Хороший                                 | 80                                                                   |
| Удовлетворительный                      | 60                                                                   |

#### *4.6. Методические рекомендации по подготовке к экзаменам и зачетам*

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Экзаменационная сессия - это серия экзаменов, установленных учебным планом. Между экзаменами интервал 3-4 дня. Не следует думать, что 3-4 дня достаточно для успешной подготовки к экзаменам.

В эти 3-4 дня нужно систематизировать уже имеющиеся знания. На консультации перед экзаменом студентов познакомят с основными требованиями, ответят на возникшие у них вопросы. Поэтому посещение консультаций обязательно.

Требования к организации подготовки к экзаменам те же, что и при занятиях в течение семестра, но соблюдаться они должны более строго. Во-первых, очень важно соблюдение режима дня; сон не менее 8 часов в сутки, занятия заканчиваются не позднее, чем за 2-3 часа до сна. Оптимальное время занятий - утренние и дневные часы. В перерывах между занятиями рекомендуются прогулки на свежем воздухе, неустойчивые занятия спортом. Во-вторых, наличие хороших собственных конспектов лекций. Даже в том случае, если была пропущена какая-либо лекция, необходимо во время ее восстановить (переписать ее на кафедре), обдумать, снять возникшие вопросы для того, чтобы запоминание материала было осознанным. В-третьих, при подготовке к экзаменам у студента должен быть хороший учебник или конспект литературы, прочитанной по указанию преподавателя в течение семестра. Здесь можно эффективно использовать листы опорных сигналов.

Вначале следует просмотреть весь материал по сдаваемой дисциплине, отметить для себя трудные вопросы. Обязательно в них разобраться. В заключение еще раз целесообразно повторить основные положения, используя при этом листы опорных сигналов.

Систематическая подготовка к занятиям в течение семестра позволит использовать время экзаменационной сессии для систематизации знаний.

### **Контроль самостоятельной работы студентов**

Контроль самостоятельной работы проводится преподавателем в аудитории.

Предусмотрены следующие виды контроля: собеседование, оценка доклада, оценка презентации, оценка участия в круглом столе, оценка выполнения проекта.

Подробные критерии оценивания компетенций приведены в Фонде оценочных средств для проведения текущей и промежуточной аттестации.

### **Список литературы для выполнения СРС**

#### *Перечень основной литературы*

1. Дополнительное образование детей: история и современность: учеб. пособие для академического бакалавриата / отв. ред. А. В. Золотарева. – 2-е изд., испр. и доп. – М.: Издательство Юрайт, 2018. – 267 с.
2. История и теория дополнительного образования: учебное пособие для академического бакалавриата / Б. А. Дейч [и др.]; под редакцией Б. А. Дейча. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2017. – 335 с.
3. Никитина, Т. В. Образовательная робототехника как направление инженерно-технического творчества школьников: Учебное пособие / Никитина Т. В. – Челябинск: Челябинский государственный педагогический университет, 2014. – 171 с.
4. Основы робототехники: учебное пособие / В. С. Глухов, А. А. Дикой, Р. А. Галустов, И. В. Дикая. – Армавир: Армавирский государственный педагогический университет, 2019. – 308 с.
5. Пономарева, Ю. С. Практикум по основам робототехники. Задачи для LEGO MINDSTORMS NXT и EV3: учебно-методическое пособие / Ю. С. Пономарева, Т. В. Шемелова. – Волгоград: Волгоградский государственный социально-педагогический университет, 2016. – 36 с.
6. Юревич Е.И. Основы робототехники / Е.И. Юревич. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2005. – 416 с.

*Перечень дополнительной литературы*

1. Абраухова В. В. Педагогика в системе дополнительного образования детей и взрослых: учебное пособие / В. В. Абраухова. – Москва; Берлин: Директ-Медиа, 2020. – 52 с. : ил.
2. Бабаева Ю.Д., Лейтес Н.С. Психология одаренности детей и подростков: учеб. пособие для студ. высш. и сред. пед. учеб. заведений/ под ред. Н.С. Лейтес. – М.: Академия, 2000. – 336с.
3. Блум Д. Изучаем Arduino®: инструменты и методы технического волшебства / Джереми Блум – С-П.: БХВПетербург, 2017. – 336 с.
4. Буйлова Л. Н. Технология разработки и оценки качества дополнительных общеобразовательных общеразвивающих программ: новое время – новые подходы. – М.: Пед. общ-во России, 2015 г. – 272 с.
5. Воротников С.А. Информационные устройства робототехнических систем: учеб. пособие / С.А. Воротников; под. ред. С.Л. Зенкевича, А.С. Ющенко. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 384 с.
6. Вязовов С.М., Калягина О.Ю., Слезин К.А. Соревновательная робототехника: приемы программирования в среде EV3: учеб.-практ. пособие. М.: Издательство «Перо», 2014. – 128 с.
7. Гончаревич И. Ф. Основы робототехники. Механизмы выдвижения и поворота робота-погрузчика с пневмоприводом: Методические рекомендации / Гончаревич И. Ф. – Москва: Московская государственная академия водного транспорта, 2014. – 62 с.
8. Злаказов А.С. Уроки Лего-конструирования в школе: методическое пособие / А.С. Злаказов, А.Г. Горшков, С.Г. Шевалдина. – М: БИНОМ. Лаборатория знаний. – 2011. – 120 с.
9. Иванов А. А. Основы робототехники: учеб. пособие / А.А. Иванов. – М.: Форум, 2012. – 224 с.: ил.
10. Киселёв М. М. Робототехника в примерах и задачах: курс программирования механизмов и роботов / М. М. Киселёв, М. М. Киселёв. – Москва: СОЛОН-ПРЕСС, 2017. – 136 с.
11. Кузьмина М. В. др. Робототехника в школе как ресурс подготовки инженерных кадров будущей России // сборник методических материалов для работников образования в условиях реализации Федеральных государственных образовательных стандартов (по итогам областных семинаров и курсов повышения квалификации по

- образовательной робототехнике) [Электронный ресурс]: ИРО Кировской области, 2017. – 179 с.
12. Кулаков Д. Б. Роботы и робототехника: лабораторный практикум: учебное пособие / Д. Б. Кулаков, Б. Б. Кулаков. – Москва: Российский университет дружбы народов, 2018. – 124 с.
  13. Методика непрерывного профессионального развития кадров сферы дополнительного образования детей: учебное пособие / А. В. Золотарева [и др.] ; под научной редакцией А. В. Золотаревой. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2019. – 239 с.
  14. Образовательная и соревновательная робототехника в условиях реализации Федеральных государственных образовательных стандартов: учебно-методическое пособие / Авт.-сост. М.В. Кузьмина; КО ГОАУ ДПО «ИРО Кировской области». – Киров: 2019. – 159 с.
  15. Образовательная робототехника во внеурочной деятельности: учебно-методическое пособие / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011. – 96 с.
  16. Образовательная робототехника на уроках информатики и физики в средней школе / В.Н. Халамов (рук.) [и др.]. – Челябинск: Взгляд, 2011. – 157 с. : ил.
  17. Образовательная робототехника Электронный ресурс: Учебно-методический комплекс дисциплины / сост.: А. С. Соболевский, Э. Ф. Шарипова. – Челябинск: Челябинский государственный педагогический университет, 2014. – 32 с.
  18. Проведение научно-ознакомительной практики с использованием модульной микропроцессорной платформы Arduino: учебно-методическое пособие / сост.: А. А. Афонин, А. С. Захаров; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль: ЯрГУ, 2018. – 60 с.
  19. Пшихопов В.Х., Медведев М.Ю., Костюков В.А., Гайдук А.Р., Федоренко Р.В., Гуренко Б.В., Крухмалев В.А., Медведева Т.Н. Проектирование роботов и робототехнических систем: Учебное пособие – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 196 с.
  20. Ревич Ю.В. Занимательная электроника / Юрий Ревич. – 4-е изд., перераб. и доп. – С-П.: БХВ-Петербург, 2017. – 639 с.
  21. Рыбак, Л. А. Роботы и робототехнические комплексы: Учебное пособие / Рыбак Л. А. – Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2013. – 84 с.
  22. Сборник учебно-методических материалов по образовательной робототехнике: Опыт образовательных учреждений Томской области / сост. О.С. Нетесова. – Томск: Издательство Томского государственного педагогического университета, 2016. – 172 с.
  23. Семакин И.Г. Преподавание базового курса информатики в средней школе: метод. пособие / И.Г. Семакин, Т.Ю. Шеина. – 2-е изд., испр. и доп. – М.: БИНОМ. Лаборатория знаний, 2004. – 540 с.
  24. Сергеева Т. Ф., Пронина Н. А., Сечкарева Е. В. Система работы с одаренными детьми: теория и практика. Ростов на/Д.: Изд-во «Феникс», 2011. – 284, [2] с.: ил.
  25. Соревновательная деятельность региональных ресурсных центров технического творчества для детей и молодежи на базе социально ориентированных НКО на примере Программы «Робототехника» [Электронный ресурс]. – 2013. – 38 с.
  26. Ступина Е.Е., Ступин А.А., Чупин Д.Ю., Каменев Р.В. Основы робототехники: учебное пособие. – Новосибирск: Агентство «Сибпринт», 2019. – 160 с
  27. Филиппов С. А. Уроки робототехники. Конструкция. Движение. Управление / С. А. Филиппов; составители А. Я. Щелкунова. – 4-е изд. – Москва: Лаборатория знаний, 2022. – 191 с

28. Филиппов С.А. Робототехника для детей и родителей / С.А. Филиппов. – СПб.: Наука, 2013. – 319 с.
29. Щетинская А.И., Тавстуха О.Г., Болотова М.И. Теория и практика современного дополнительного образования детей: учебное пособие / А. И. Щетинская, О. Г. Тавстуха, М. И. Болотова; Мин-во образования и науки РФ; Оренбург, гос. пед. ун-т. – Оренбург: Изд-во ОПТУ, 2006. – 404 с.
30. Ярнольд С. Arduino для начинающих: самый простой пошаговый самоучитель / Стюарт Ярнольд – М.: Э, 2017. – 253 с.

**Методическая литература:**

1. Методические рекомендации для самостоятельной работы студентов по дисциплине «Организация дополнительного образования»
2. Методические указания к лабораторным работам по дисциплине «Организация дополнительного образования»

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

- 1 <http://engjournal.ru/catalog/machin/hydro/682.html> - Кулаков Б.Б., Кулаков Д.Б., Беляев В.В. Антропоморфные роботы как новая сфера применения гидроприводов. Инженерный журнал: наука и инновации, 2013, вып. 4.
- 2 <http://fea.ru/news/6245> - центр компьютерного инжиниринга СПбПУ
- 3 <http://genius.pstu.ru/joomla/> - открытый информационный социально-образовательный Портал «Одарённые дети России XXI века»
- 4 <http://zakon-ob-obrazovanii.ru> - Федеральный закон "Об образовании в Российской Федерации" N 273-ФЗ от 29 декабря 2012 года с изменениями 2018 года.