

**Министерство образования и науки Смоленской области  
Муниципальное бюджетное общеобразовательное учреждение  
«Средняя общеобразовательная школа № 2» г. Сафоново  
Смоленской области**

Принята на заседании  
педагогического совета  
Протокол № 1 от 29.08.2024 г.



**ТОЧКА РОСТА**

**Дополнительная общеобразовательная общеразвивающая  
программа технической направленности  
«Мастер Arduino. Профильный уровень»**

**Срок реализации – 1 год  
Возраст детей – 15 - 17 лет**

Программа составлена педагогом  
дополнительного образования  
Егоровой Мариной Евгеньевной

г. Сафоново, 2024 г.

## Пояснительная записка

Направленность программы – **техническая**.

Вид программы – модернизированная.

### **Актуальность программы**

Образовательная робототехника в настоящее время является предметом особого внимания. Актуальным вопросом является выявление и развитие молодых талантов, формирование инженерного и алгоритмического мышления у обучающихся. В текущий момент специалисты в области микропроцессорных систем и робототехнических комплексов будут востребованы на рынке труда. В настоящее время увеличивается интерес к робототехнике. Направление «Робототехника» входит в число приоритетных соревнований в движении «Профессионалы России»

Образовательная робототехника является одним из способов инженерного мышления. Внедрение основ робототехники поможет сформированию у школьников целостное представления о мире техники, устройствах конструкций, механизмах и машинах.

Объединение конструирования с программированием даёт возможность интегрировать предметные науки с развитием инженерного мышления через техническое творчество. Инженерное творчество и лабораторные исследования являются мощным инструментом синтеза знаний.

При получении базовых знаний в основах алгоритмизации и программирования необходимо показать учащимся их применения в реальной жизни не только в IT сфере, но и в экономической и социальной сфере. В этом и заключается практическая значимость данной программы.

Таким образом, актуальность программы заключается в соответствии изучаемого материала современным требованиям государственной политики в области образования касаясь вопросов подготовки на школьном уровне детей с высоко развитым техническим мышлением.

### **Педагогическая целесообразность программы**

Педагогическая целесообразность дополнительной образовательной программы заключается в создании особой развивающей среды для выявления и развития общих и творческих способностей обучающихся, что может способствовать не только их приобщению к творчеству, причем не только к декоративно-прикладному, но и раскрытию лучших человеческих качеств. Дополнительная образовательная программа опирается на следующие педагогические принципы:

- принцип доступности обучения — учет возрастных и индивидуальных особенностей;
- принцип поэтапного углубления знаний — усложнение учебного материала от простого к сложному при условии выполнения обучающимся предыдущих заданий;
- принцип комплексного развития — взаимосвязь и взаимопроникновение разделов (блоков) программы;
- принцип совместного творческого поиска в педагогической деятельности;
- принцип личностной оценки каждого обучающегося без сравнения с другими детьми, помогающий им почувствовать свою неповторимость и значимость для группы.

Профильный уровень программы подразумевает изучение языка программирования и на основе приобретенных знаний разработку и программирование устройств на базе Arduino. Обучающиеся могут попробовать себя в качестве разработчика роботизированных технических устройств. Данный метод обучения способствует профориентации учащихся и направлен на выявление интереса к сфере робототехники.

Arduino даёт возможность ученику освоить основные приёмы конструирования и программирования управляемых электронных устройств и получить необходимые знания и

навыки для дальнейшей самореализации в области инженерии, изобретательства, информационных технологий и программирования.

### **Адресат программы**

Программа рассчитана для обучающихся 15-17 лет. Программа доступна для детей, проявивших выдающиеся способности (одаренные), детей с ограниченными возможностями здоровья (нарушение зрения и слуха), детей, находящихся в трудной жизненной ситуации.

Программа предназначена для обучающихся, интересующихся техникой и компьютерными технологиями, одаренных учащихся и направлена на обеспечение дополнительной теоретической и практической подготовки по направлению «Робототехника и программирование» и развитие творческих, интеллектуальных и исследовательских способностей.

**Срок освоения программы:** 1 год.

**Объем программы:** 68 академических часов.

**Режим занятий:** 2 академический час в неделю.

**Учебная группа:** 5 – 8 учащихся.

**Форма организации образовательного процесса:** очная.

### **Условия реализации программы**

#### ***Техническое обеспечение образовательного процесса:***

1. Персональный компьютер (ноутбук).
2. Колонки.
3. Проектор мультимедийный.
4. Электронный конструктор Arduino с набором датчиков и компонентов.

#### **Формы проведения занятий:**

- лекция;
- практическое занятие;
- самостоятельная работа.

### **Цель программы**

Развитие научно-технического и творческого потенциала личности ученика посредством обучения основам программирования и разработке умных устройств.

#### **Задачи:**

##### ***образовательные:***

- познакомить учащихся с базовым кругом специальных знаний в области робототехники согласно программе: принципами и возможностями конструирования, проектирования, программирования объектов техники на базе набора Arduino;
- формировать и расширять кругозор учащихся в области робототехники, способствовать становлению устойчивого познавательного интереса к современным технологиям конструирования, программирования и использования роботизированных устройств;

- обучить приемам коллективного проектирования, конструирования и программирования объектов техники согласно программе с использованием набора Arduino;
- выработать навыки применения средств ИКТ в повседневной жизни, при выполнении индивидуальных и коллективных проектов;
- обучить основам проектирования умных с использованием микропроцессоров на базе Arduino;
- обучить основам программирования в среде Arduino IDE;

***развивающие:***

- развитие конструкторских навыков;
- развитие логического алгоритмического мышления;
- развивать познавательную активность учащихся посредством включения в проектную деятельность;
- развивать интерес учащихся к различным областям электроники, программирования и роботостроения;

***воспитательные:***

- воспитание интереса к техническим видам творчества;
- развитие коммуникативной компетенции: навыков сотрудничества в коллективе, малой группе (в паре), участия в беседе, обсуждении;
- развитие социально-трудовой компетенции: воспитание трудолюбия, самостоятельности, умения доводить начатое дело до конца;
- формирование и развитие информационной компетенции: навыков работы с различными источниками информации, умения самостоятельно искать, извлекать и отбирать необходимую для решения учебных задач информацию.

**Планируемые результаты.**

**Личностные результаты:**

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;
- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

**Метапредметные результаты:**

- умение принимать и сохранять учебную задачу;
- умение планировать последовательность шагов алгоритма для достижения цели;
- умение ставить цель (создание творческой работы), планировать достижение этой цели;

- умение осуществлять итоговый и пошаговый контроль по результату;
- способность адекватно воспринимать оценку наставника и других обучающихся;
- умение различать способ и результат действия;
- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учёта характера сделанных ошибок;
- умение в сотрудничестве ставить новые учебные задачи;
- умение осуществлять поиск информации в индивидуальных информационных архивах обучающегося, информационной среде образовательного учреждения, федеральных хранилищах информационных образовательных ресурсов;
- умение использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- умение осуществлять анализ объектов с выделением существенных и несущественных признаков;
- умение проводить сравнение, классификацию по заданным критериям;
- умение устанавливать аналогии, причинно-следственные связи;
- умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- умение выслушивать собеседника и вести диалог;
- способность признавать возможность существования различных точек зрения и право каждого иметь свою;
- умение планировать учебное сотрудничество с наставником и другими обучающимися: определять цели, функции участников, способы взаимодействия.

### **Предметные результаты**

- знать назначение модулей и датчиков конструктора Arduino;
- знать основы языка программирования C/C++ в среде Arduino IDE;
- уметь собирать модели умных устройств на базе Arduino;
- уметь читать схемы соединений и составлять их для дальнейшей разработки умных устройств;
- уметь подключать микроконтроллер к ПК, проводить компиляцию программного кода и его загрузку на микроконтроллер;
- уметь исправлять неисправности при работе устройств, собранных на базе Arduino.

### **Форма контроля**

Наблюдение, устный опрос, письменный опрос, демонстрация.

### **Виды контроля и формы аттестации**

1. Входной контроль.
2. Текущий контроль.
3. Итоговый контроль.

### **Используемые педагогические технологии**

**Коллективно – творческая деятельность** - комплексная педагогическая технология, объединяющая в себе формы образования, воспитания и эстетического общения. Ее результат – общий успех, оказывающий положительное влияние как на коллектив в целом, так и на каждого учащегося в отдельности.

**Личностно – ориентированное обучение** – это такое обучение, которое ставит главным - самобытность ребенка, его самооценку субъектность процессов обучения. Цель личностно – ориентированного обучения состоит в том, чтобы заложить в ребенке механизмы самореализации, саморазвития, саморегуляции самовоспитания и другие, необходимые для

становления самобытного образа и диалогического взаимодействия с людьми, природой, культурой, цивилизацией.

**Проблемное обучение** – создание в учебной деятельности проблемных ситуаций и организация активной самостоятельной деятельности учащихся по их разрешению, в результате чего происходит творческое овладение знаниями, умениями, навыками, развиваются мыслительные способности.

**Проектные методы обучения** – работа по данной методике дает возможность развивать индивидуальные творческие способности учащихся, более осознанно подходить к профессиональному и социальному самоопределению.

**Исследовательские методы в обучении** – дают возможность учащимся самостоятельно пополнять свои знания, глубоко вникать в изучаемую проблему и предполагать пути ее решения, что важно при формировании мировоззрения. Это важно для определения индивидуальной траектории развития каждого школьника.

**Здоровьесберегающие технологии - образовательные технологии»** по определению Н.К. Смирнова, - это все те психолого-педагогические технологии, программы, методы, которые направлены на воспитание у учащихся культуры здоровья, личностных качеств, способствующих его сохранению и укреплению, формирование представления о здоровье как ценности, мотивацию на ведение здорового образа жизни.

### Учебный план

| № п/п               | Название раздела,<br>темы                             | Количество часов |        |          | Формы<br>аттестации/контроля          |
|---------------------|---|------------------|--------|----------|---------------------------------------|
|                     |   | Всего            | Теория | Практика |                                       |
| 1                   | <i><b>Введение</b></i>                                | 1                | 1      | 0        | Входной контроль                      |
| 2                   | <i><b>Основы<br/>программирования<br/>Arduino</b></i> | 45               | 24     | 21       | Текущий контроль                      |
| 3                   | <i><b>Проектирование<br/>устройств на Arduino</b></i> | 22               | 2      | 20       | Текущий контроль<br>Итоговый контроль |
| <i><b>ИТОГО</b></i> |   | 68               | 27     | 41       |                                       |

## Содержание учебного плана

### Раздел 1. Введение

*Теория:* Вводное занятие. Основы безопасной работы. Инструктаж по технике безопасности. Микроконтроллеры и их применение. Умные устройства и умный дом.

*Практика:* не планируется.

### Раздел 2. Основы программирования Arduino

*Теория:* Среда Arduino IDE: установка, внешний вид, меню, назначение и основные принципы работы. Скейтч: определение, создание, компиляция и загрузка. Библиотеки Arduino. Тестирование устройств. Язык программирования C/C++ в Arduino IDE. Операторы *setup()* и *loop()*. Синтаксис ( ; , {}, //, /\*\*/ ). Типы данных (*boolean, char, byte, int, unsigned int, word, long, unsigned long, float, double, string* - массив символов, *String* - объект класса, массив (*array*), *void*). Преобразование типов данных (*char(), byte(), int(), long(), float()*). Константы (*HIGH / LOW, INPUT / OUTPUT, true / false*, целочисленные константы, константы с плавающей запятой). Арифметические операторы (=, +, -, \*, /, %). Операторы сравнения (==, !=, <, >, <=, >=). Логические операторы (&& (И), || (ИЛИ), ! (Отрицание)). Унарные операторы (++ , -- , += , -= , \*= , /=). Математические и тригонометрические функции (*min(), max(), abs(), constrain(), map(), pow(), sq(), sqrt(), sin(), cos(), tan()*). Генераторы случайных значений (*randomSeed(), random()*). Управляющие операторы (*if, if...else, for, switch case, while, do... while, break, continue, return, goto*). Область видимости переменных. Квалификаторы переменных (*static, volatile, const*). Цифровой ввод/вывод (*pinMode(), digitalWrite(), digitalRead()*). Аналоговый ввод/вывод (*analogRead(), analogReference(), analogWrite()*). Дополнительные функции ввода/вывода (*tone(), noTone(), shiftOut(), pulseIn()*). Работа со временем (*millis(), micros(), delay(), delayMicroseconds()*). Внешнее прерывание (*attachInterrupt(), detachInterrupt()*) и функции передачи данных *serial*.

*Практика:* Установка и запуск программы Arduino IDE. Узечение меню среды Arduino IDE. Изучение синтаксиса в Arduino IDE. Работа с типами данных..Преобразование типов данных. Работа с константами. Работа с арифметическими операторами. Работа с операторами сравнения. Работа с логическими операторами. Работа с унарными операторами. Работа с математическими функциями. Работа с тригонометрическими функциями. Генерирование случайных чисел. Работа с управляющими операторами. Работа с областью видимости переменных и квалификаторами. Работа с цифровым вводом/выводом. Работа с аналоговым вводом/выводом. Работа с дополнительными функциями ввода/вывода. Работа со временем. Работа с внешним прерыванием. Работа с функциями передачи данных в Arduino IDE.

### Раздел 3. Конструирование

*Теория:* Состав набора Arduino. Сенсоры, модули, датчики и другие компоненты набора Arduino. Правила подключения компонентов к плате микроконтроллера. Плата Arduino UNO/, Arduino Nano, Arduino Micro, Arduino Mega, NodeMCU. Назначение выходов на плате. Схема соединений.

*Практика:* Сборка устройства управления работой светодиода, его программирование и тестирование. Сборка модели метеостанции, ей прошивка и тестирование. Дальномер ультразвуковой: сборка, программирование и тестирование. Управление сервоприводом джойстиком: сборка, программирование, прошивка и тестирование. Проектирование устройства



на основе датчика дыма: сборка, программирование, прошивка и тестирование. Разработка устройства передачи изображения с модуля камеры OV7670 на ПК: сборка, программирование, прошивка и тестирование. Выполнение проекта по замыслу: подбор оборудования, проектирование схемы соединения, сборка устройства, написание кода, компиляция кода, загрузка скейтча, тестирование устройства.

## Календарный учебный график

| № п/п  | Месяц* | Число* | Время проведения занятий* | Форма занятий        | Кол-во часов | Тема занятия  | Место проведения | Форма контроля |
|--|--------|--------|---------------------------|----------------------|--------------|---|------------------|----------------|
| <b>Раздел 1. Введение. Техника безопасности</b>  |        |        |                           |                      |              |   |                  |                |
| 1.   |        |        |                           | Лекция               | 1            | Введение в робототехнику. Правила техники безопасности                      | Каб. 10          | Устный опрос   |
| <b>Раздел 2. Основы программирования Arduino</b> |        |        |                           |                      |              |   |                  |                |
| 2.   |        |        |                           | Лекция               | 1            | Среда Arduino IDE. Установка программы и её запуск                          | Каб. 10          | Устный опрос   |
| 3.   |        |        |                           | Практическое занятие | 1            | Установка среды Arduino IDE. Изучение меню программы                        | Каб. 10          | Наблюдение     |
| 4.   |        |        |                           | Лекция               | 1            | Правила загрузки скейтча на платы. Техника безопасности. Библиотеки Arduino | Каб. 10          | Устный опрос   |
| 5.   |        |        |                           | Лекция               | 1            | Тестирование схемы устройства   | Каб. 10          | Устный опрос   |
| 6.   |        |        |                           | Лекция               | 1            | Язык программирования C/C++ в Arduino IDE.                                  | Каб. 10          | Устный опрос   |
| 7.   |        |        |                           | Лекция               | 1            | C/C++ в Arduino IDE: операторы данные и функции                             | Каб. 10          | Устный опрос   |
| 8.   |        |        |                           | Лекция               | 1            | C/C++ в Arduino IDE: синтаксис  | Каб. 10          | Устный опрос   |
| 9.   |        |        |                           | Практическое занятие | 1            | Изучение синтаксиса в Arduino IDE   | Каб. 10          | Наблюдение     |
| 10.  |        |        |                           | Лекция               | 1            | C/C++ в Arduino IDE: типы данных  | Каб. 10          | Устный опрос   |
| 11.  |        |        |                           | Практическое занятие | 1            | Работа с типами данных в Arduino IDE  | Каб. 10          | Наблюдение     |
| 12.  |        |        |                           | Лекция               | 1            | C/C++ в Arduino IDE: преобразование типов данных                            | Каб. 10          | Устный опрос   |
| 13.  |        |        |                           | Практическое занятие | 1            | Преобразование типов данных в Arduino IDE                                   | Каб. 10          | Наблюдение     |
| 14.  |        |        |                           | Лекция               | 1            | C/C++ в Arduino IDE: константы  | Каб. 10          | Устный опрос   |
| 15.  |        |        |                           | Практическое занятие | 1            | Работа с константами в Arduino IDE  | Каб. 10          | Наблюдение     |
| 16.  |        |        |                           | Лекция               | 1            | C/C++ в Arduino IDE: арифметические операторы                               | Каб. 10          | Устный опрос   |
| 17.  |        |        |                           | Практическое занятие | 1            | Работа с арифметическими операторами в Arduino IDE                          | Каб. 10          | Наблюдение     |

|     |  |  |                      |   |  |         |                  |
|-----|--|--|----------------------|---|--|---------|------------------|
| 18. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: операторы сравнения                               | Каб. 10 | Устный опрос     |
| 19. |  |  | Практическое занятие | 1 | Работа с операторами сравнения в Arduino IDE                           | Каб. 10 | Наблюдение       |
| 20. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: логические операторы                              | Каб. 10 | Устный опрос     |
| 21. |  |  | Практическое занятие | 1 | Работа с логическими операторами в Arduino IDE                         | Каб. 10 | Наблюдение       |
| 22. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: унарные операторы                                 | Каб. 10 | Устный опрос     |
| 23. |  |  | Практическое занятие | 1 | Работа с унарными операторами в Arduino IDE                            | Каб. 10 | Наблюдение       |
| 24. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: математические функции                            | Каб. 10 | Устный опрос     |
| 25. |  |  | Практическое занятие | 1 | Работа с математическими функциями в Arduino IDE                       | Каб. 10 | Наблюдение       |
| 26. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: тригонометрические функции                        | Каб. 10 | Устный опрос     |
| 27. |  |  | Практическое занятие | 1 | Работа с тригонометрическими функциями в Arduino IDE                   | Каб. 10 | Наблюдение       |
| 28. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: генераторы случайных чисел                        | Каб. 10 | Устный опрос     |
| 29. |  |  | Практическое занятие | 1 | Генерирование случайных чисел в Arduino IDE                            | Каб. 10 | Наблюдение       |
| 30. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: управляющие операторы                             | Каб. 10 | Устный опрос     |
| 31. |  |  | Практическое занятие | 1 | Работа с управляющими операторами в Arduino IDE                        | Каб. 10 | Наблюдение       |
| 32. |  |  | Практическое занятие | 1 | Работа с управляющими операторами в Arduino IDE                        | Каб. 10 | Наблюдение       |
| 33. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: область видимости переменных и квалификаторы      | Каб. 10 | Письменный опрос |
| 34. |  |  | Практическое занятие | 1 | Работа с областью видимости переменных и квалификаторами в Arduino IDE | Каб. 10 | Наблюдение       |
| 35. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: цифровой ввод/вывод                               | Каб. 10 | Устный опрос     |
| 36. |  |  | Практическое занятие | 1 | Работа с цифровым вводом/выводом в Arduino IDE                         | Каб. 10 | Наблюдение       |
| 37. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: аналоговый ввод/вывод                             | Каб. 10 | Устный опрос     |
| 38. |  |  | Практическое занятие | 1 | Работа с аналоговым вводом/выводом в Arduino IDE                       | Каб. 10 | Наблюдение       |
| 39. |  |  | Лекция               | 1 | C/C++ в Arduino IDE: дополнительные функции ввода/вывода               | Каб. 10 | Устный опрос     |

|  |  |  |  |                      |   |  |         |                  |
|--|--|--|--|----------------------|---|--|---------|------------------|
| 40.  |  |  |  | Практическое занятие | 1 | Работа с дополнительными функциями ввода/вывода в Arduino IDE    | Каб. 10 | Наблюдение       |
| 41.  |  |  |  | Лекция               | 1 | C/C++ в Arduino IDE: работа со временем                          | Каб. 10 | Устный опрос     |
| 42.  |  |  |  | Практическое занятие | 1 | Работа со временем в Arduino IDE                                 | Каб. 10 | Наблюдение       |
| 43.  |  |  |  | Лекция               | 1 | C/C++ в Arduino IDE: внешнее прерывание                          | Каб. 10 | Устный опрос     |
| 44.  |  |  |  | Практическое занятие | 1 | Работа с внешним прерыванием в Arduino IDE                       | Каб. 10 | Наблюдение       |
| 45.  |  |  |  | Лекция               | 1 | C/C++ в Arduino IDE: функции передачи данных                     | Каб. 10 | Устный опрос     |
| 46.  |  |  |  | Практическое занятие | 1 | Работа с функциями передачи данных в Arduino IDE                 | Каб. 10 | Наблюдение       |
| <b>Раздел 3. Проектирование устройств на Arduino</b> |  |  |  |                      |   |  |         |                  |
| 47.  |  |  |  | Лекция               | 1 | Сенсоры, модули, датчики и другие компоненты набора Arduino      | Каб. 10 | Устный опрос     |
| 48.  |  |  |  | Лекция               | 1 | Правила подключения компонентов к плате микроконтроллера         | Каб. 10 | Письменный опрос |
| 49.  |  |  |  | Практическое занятие | 1 | Управление работой светодиода: сборка устройства                 | Каб. 10 | Наблюдение       |
| 50.  |  |  |  | Практическое занятие | 1 | Управление работой светодиода: программирование и прошивка       | Каб. 10 | Демонстрация     |
| 51.  |  |  |  | Практическое занятие | 1 | Метеостанция: сборка устройства                                  | Каб. 10 | Наблюдение       |
| 52.  |  |  |  | Практическое занятие | 1 | Метеостанция: программирование и прошивка                        | Каб. 10 | Демонстрация     |
| 53.  |  |  |  | Практическое занятие | 1 | Дальномер ультразвуковой : сборка устройства                     | Каб. 10 | Наблюдение       |
| 54.  |  |  |  | Практическое занятие | 1 | Дальномер ультразвуковой : программирование и прошивка           | Каб. 10 | Демонстрация     |
| 55.  |  |  |  | Практическое занятие | 1 | Управление сервоприводом джойстиком: сборка устройства           | Каб. 10 | Наблюдение       |
| 56.  |  |  |  | Практическое занятие | 1 | Управление сервоприводом джойстиком: программирование и прошивка | Каб. 10 | Демонстрация     |
| 57.  |  |  |  | Практическое занятие | 1 | Сборка устройства с датчиком дыма                                | Каб. 10 | Наблюдение       |

|     |  |  |  |                        |   |  |         |              |
|-----|--|--|--|------------------------|---|--|---------|--------------|
| 58. |  |  |  | Практическое занятие   | 1 | Программирование и прошивка устройства с датчиком дыма                         | Каб. 10 | Демонстрация |
| 59. |  |  |  | Практическое занятие   | 1 | Передача изображения с модуля камеры OV7670 на ПК: сборка устройства           | Каб. 10 | Наблюдение   |
| 60. |  |  |  | Практическое занятие   | 1 | Передача изображения с модуля камеры OV7670 на ПК: программирование устройства | Каб. 10 | Наблюдение   |
| 61. |  |  |  | Практическое занятие   | 1 | Передача изображения с модуля камеры OV7670 на ПК: прошивка и тестирование     | Каб. 10 | Демонстрация |
| 62. |  |  |  | Самостоятельная работа | 1 | Выбор темы проекта. Подбор оборудования  | Каб. 10 | Наблюдение   |
| 63. |  |  |  | Самостоятельная работа | 1 | Проектирование схемы соединения устройства                                     | Каб. 10 | Наблюдение   |
| 64. |  |  |  | Самостоятельная работа | 1 | Сборка устройства по замыслу   | Каб. 10 | Наблюдение   |
| 65. |  |  |  | Самостоятельная работа | 1 | Сборка устройства по замыслу   | Каб. 10 | Наблюдение   |
| 66. |  |  |  | Самостоятельная работа | 1 | Программирование и прошивка устройства по замыслу                              | Каб. 10 | Наблюдение   |
| 67. |  |  |  | Самостоятельная работа | 1 | Программирование и прошивка устройства по замыслу                              | Каб. 10 | Наблюдение   |
| 68. |  |  |  | Практическое занятие   | 1 | Демонстрация работы устройства   | Каб. 10 | Демонстрация |

\*Месяц, число и время проведения занятий – согласно утвержденному расписанию.

## Методическое обеспечение программы

### Информационное обеспечение

#### *Подборка видеоматериала по темам программы*

1. История робототехники: от 5 века до н.э. до современности  
<https://www.youtube.com/watch?v=u9betZrEiqI>
2. Как работает процессор?  
[https://www.youtube.com/watch?v=qlhZrMg3\\_Tk](https://www.youtube.com/watch?v=qlhZrMg3_Tk)
3. Программирование Ардуино с нуля. Arduino для начинающих  
[https://youtu.be/E0b2yukhZ\\_Q](https://youtu.be/E0b2yukhZ_Q)
4. Уроки Ардуино #0 - что такое Arduino, куда подключаются датчики и как питать Ардуино  
<https://www.youtube.com/watch?v=nrczO8tWJNg&t=205s>

#### *Инструкции по работе с Arduino и примеры проектов:*

1. Официальный сайт компании Arduino <https://arduino.ru/>.
2. ArduinoMaster.Ru. Проекты ардуино для начинающих  
<https://arduinomaster.ru/projects/proekty-arduino-dlya-nachinayushhih/>
3. Робототехника18.рф. Ардуино проекты для начинающих <https://clck.ru/tjRSE>
4. Программные коды для выполнения практической части программы (в электронном виде в кабинете).

#### *Инструкции по технике безопасности:*

1. Инструкция по охране труда обучающихся (вводный инструктаж).
2. Инструкция правилам безопасного поведения учащихся в ОУ.
3. Инструкция по пожарной безопасности.
4. Инструкция по электробезопасности.
5. Инструкция по правилам безопасности при обнаружении неизвестных пакетов, взрывоопасных предметов.
6. Инструкция правила безопасного поведения при угрозе террористического акта.

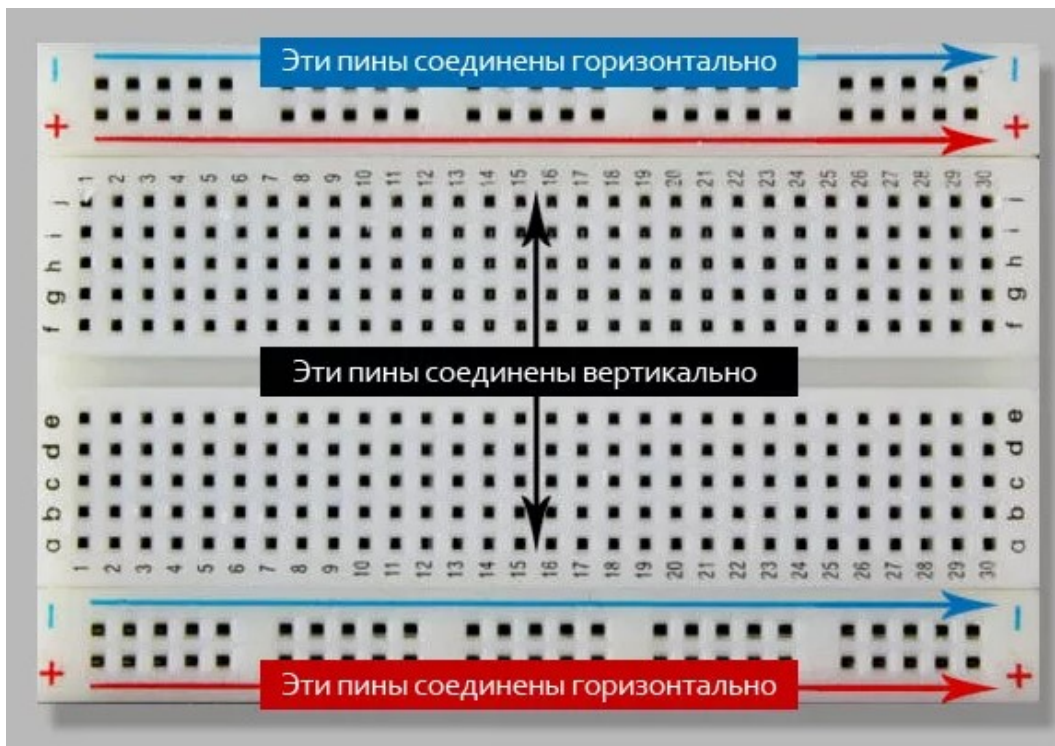
#### *Мультимедийные презентации по темам:*

1. Введение в образовательную робототехнику.
2. Платформа Arduino. Микроконтроллеры на базе Arduino.
3. Среда Arduino IDE.
4. Скейтч. Загрузка скейтча.
5. Тестирование схемы устройств на базе Arduino.
6. Язык программирования C/C++ в Arduino IDE.
7. C/C++ в Arduino IDE: операторы данные и функции.
8. C/C++ в Arduino IDE: синтаксис.
9. C/C++ в Arduino IDE: типы данных.
10. C/C++ в Arduino IDE: преобразование типов данных.
11. C/C++ в Arduino IDE: константы.
12. C/C++ в Arduino IDE: арифметические операторы.
13. C/C++ в Arduino IDE: операторы сравнения.
14. C/C++ в Arduino IDE: логические операторы.
15. C/C++ в Arduino IDE: унарные операторы.

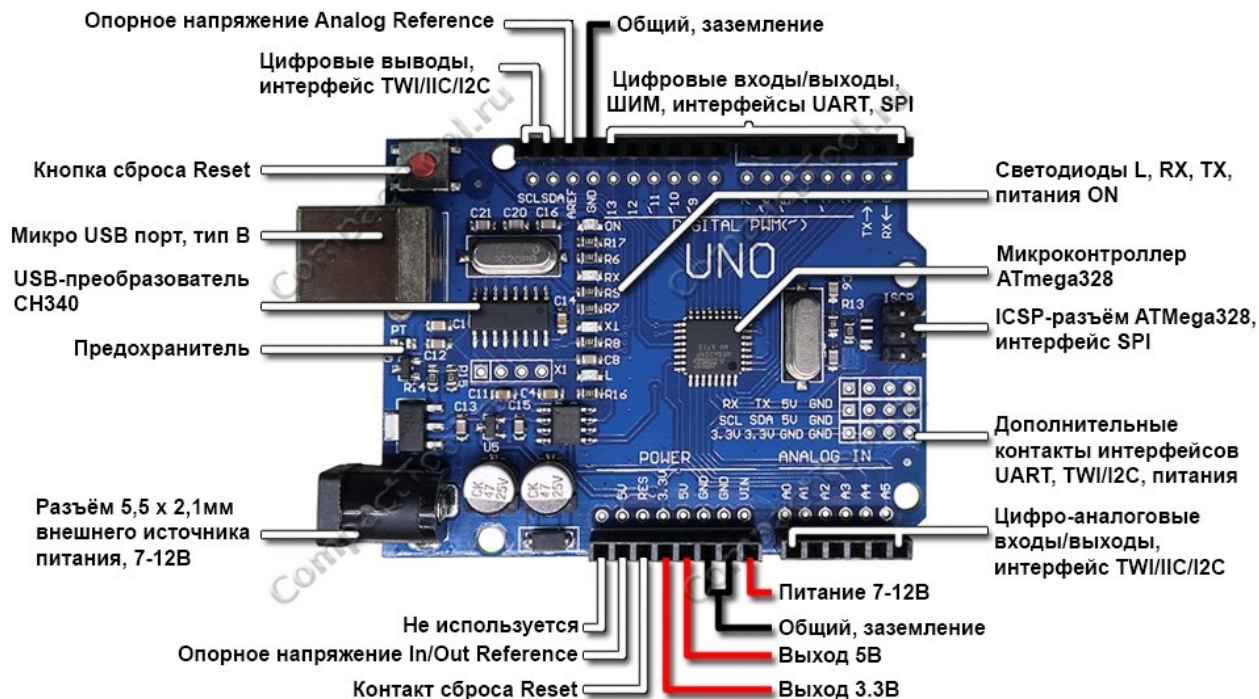
16. C/C++ в Arduino IDE: математические функции.
17. C/C++ в Arduino IDE: генераторы случайных чисел.
18. C/C++ в Arduino IDE: управляющие операторы.
19. C/C++ в Arduino IDE: область видимости переменных и квалификаторы.
20. C/C++ в Arduino IDE: цифровой ввод/вывод, аналоговый ввод/вывод, дополнительные функции ввода/вывода.
21. C/C++ в Arduino IDE: работа со временем.
22. C/C++ в Arduino IDE: внешнее прерывание.
23. C/C++ в Arduino IDE: функции передачи данных.
24. Сенсоры, модули, датчики и другие компоненты набора Arduino
25. Правила подключения компонентов к плате микроконтроллера.

# Раздаточный материал

## 1. Макетная плата

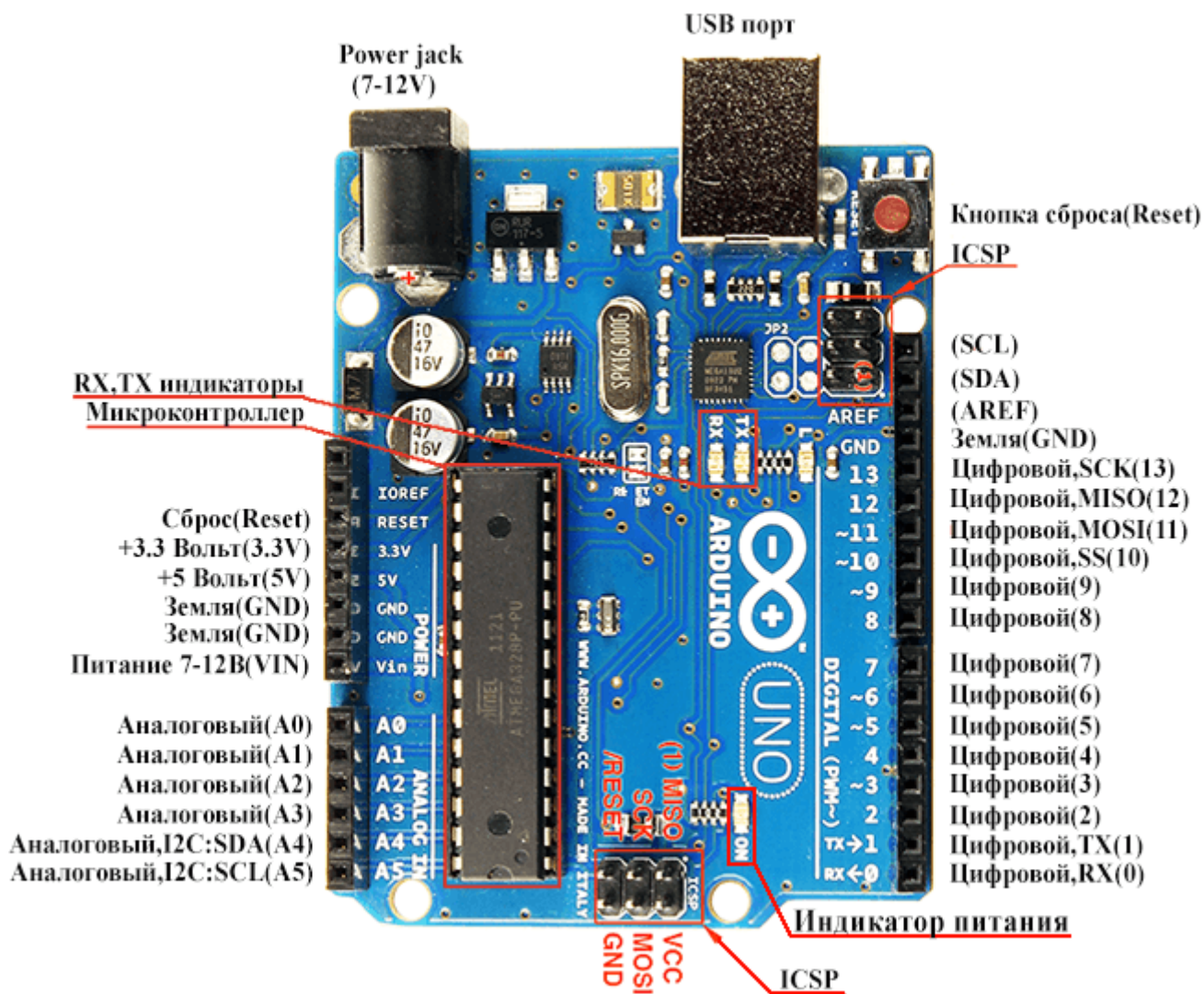


## 2. Состав платы Arduino UNO





### 3. Распиновка платы Arduino UNO



## Материал для практических работ по разделу 2 «Основы программирования Arduino»

### 1. Справочник языка Arduino

(для перехода на сайт с примерами использования нажмите на гиперссылку «Пример использования»)

| Язык Arduino                 | Пример  | Описание   |
|------------------------------|---|--|
| <b>Операторы</b>             |   |  |
| <b>setup()</b>               | <pre>void setup() {   pinMode(3, INPUT); }</pre>  | <p>Функция используется для инициализации переменных, определения режимов работы выводов на плате и т.д. Функция запускается только один раз, после каждой подачи питания на микроконтроллер.</p> <p>► <a href="#">Пример использования</a></p>  |
| <b>loop()</b>                | <pre>void loop() {   digitalWrite(3, HIGH);   delay(1000);   digitalWrite(3, LOW);   delay(1000); }</pre>   | <p>Функция loop крутится в цикле, позволяя программе совершать вычисления и реагировать на них. Функции setup() и loop() должны присутствовать в каждом скетче, даже если эти операторы в программе не используются.</p> <p>► <a href="#">Пример использования</a></p>   |
| <b>Управляющие операторы</b> |   |  |
| <b>if</b>                    | <pre>... if (x &gt; 100) digitalWrite(3, HIGH); if (x &lt; 100) digitalWrite(3, LOW); ...</pre>   | <p>Оператор if используется в сочетании с операторами сравнения (=, !=, &lt;, &gt;) и проверяет, достигнута ли истинность условия. Например, если значение переменной x больше 100, то включается светодиод на выходе 13, если меньше — светодиод выключается.</p> <p>► <a href="#">Пример использования</a></p>   |
| <b>if..else</b>              | <pre>... if (x &gt; 100) digitalWrite(3, HIGH); else digitalWrite(3, LOW); ...</pre>  | <p>Оператор else позволяет сделать проверку отличную от указанной в if, чтобы осуществлять несколько взаимо исключающих проверок. Если ни одна из проверок не получила результат ИСТИНА, то выполняется блок операторов в else.</p> <p>► <a href="#">Пример использования</a></p>  |
| <b>switch...case</b>         | <pre>... switch (x) {   case 1: digitalWrite(3, HIGH);   case 2: digitalWrite(3, LOW);   case 3: break;   default: digitalWrite(4, HIGH); } ...</pre> | <p>Подобно if, оператор switch управляет программой, позволяя задавать действия, которые будут выполняться при разных условиях. Break является командой выхода из оператора, default выполняется, если не выбрана ни одна альтернатива.</p> <p>► <a href="#">Пример использования</a></p>  |
| <b>for</b>                   | <pre>void setup() {   pinMode(3, OUTPUT); } void loop() {   for (int i=0; i &lt;= 255; i++){     analogWrite(3, i);     delay(10);   } }</pre>        | <p>Конструкция for используется для повторения операторов, заключенных в фигурные скобки. Например, плавное затемнение светодиода. Заголовок цикла for состоит из трех частей: for (initialization; condition; increment) — initialization выполняется один раз, далее проверяется условие condition, если условие верно, то выполняется приращение increment. Цикл повторяется пока не станет ложным condition.</p> <p>► <a href="#">Пример использования</a></p> |
| <b>while</b>                 | <pre>void loop() {   while (x &lt; 10)</pre>  | <p>Оператор while используется, как цикл, который будет выполняться, пока условие в круглых скобках является истиной. В примере оператор</p>   |

|                                     |   |  |
|-------------------------------------|---|--|
|                                     | <pre>{ x = x + 1; Serial.println(x); delay(200); }</pre>  | <p>цикла while будет повторять код в скобках бесконечно до тех пор, пока x будет меньше 10.</p> <p>► <a href="#">Пример использования</a></p>  |
| <b>do...while</b>                   | <pre>void loop() { do { x = x + 1; delay(100); Serial.println(x); } while (x &lt; 10); delay(900); }</pre>  | <p>Оператор цикла do...while работает так же, как и цикл while. Однако, при истинности выражения в круглых скобках происходит продолжение работы цикла, а не выход из цикла. В приведенном примере, при x больше 10 операция сложения будет продолжаться, но с паузой 1000 мс.</p> <p>► <a href="#">Пример использования</a></p> |
| <b>break<br/>continue</b>           | <pre>switch (x) { case 1: digitalWrite(3, HIGH); case 2: digitalWrite(3, LOW); case 3: break; case 4: continue; default: digitalWrite(4, HIGH); }</pre> | <p>Break используется для принудительного выхода из циклов switch, do, for и while, не дожидаясь завершения цикла. Оператор continue пропускает оставшиеся операторы в текущем шаге цикла.</p> <p>► <a href="#">Пример использования</a></p>   |
| <b>Синтаксис</b>                    |   |  |
| <b>;</b><br>(точка с запятой)       | <pre>... digitalWrite(3, HIGH); ...</pre>   | <p>Точка с запятой используется для обозначения конца оператора. Забытая в конце строки точка с запятой приводит к ошибке при компиляции.</p>  |
| <b>{<br/>}</b><br>(фигурные скобки) | <pre>void setup() { pinMode(3, INPUT); }</pre>  | <p>Открывающая скобка "{" должна сопровождаться закрывающей скобкой "}". Непарные скобки могут приводить к скрытым и непонятным ошибкам при компиляции скетча.</p>   |
| <b>//</b><br>(комментарий)          | <pre>x = 5; // комментарий</pre>  | <p>Комментарии используются для напоминания, как работает программа. Они игнорируются компилятором и не экспортируются в процессор, не занимая место в памяти микроконтроллера.</p>  |
| <b>#define</b>                      | <pre>#define ledPin 3</pre>   | <p>Директива #define позволяет дать имя константе. Директива служит исключительно для удобства и улучшения читаемости программы.</p> <p>► <a href="#">Пример использования</a></p>   |
| <b>#include</b>                     | <pre>// библиотека для серво #include &lt;Servo.h&gt;</pre>   | <p>Директива #include используется для включения сторонних библиотек в скетч. Помните, что директивы #include и #define, не требуют точки запятой.</p> <p>► <a href="#">Пример использования</a></p>   |
| <b>Типы данных</b>                  |   |  |
| <b>boolean</b>                      | <pre>boolean val = false;</pre>   | <p>Переменная boolean может принимать значение — true или false. Каждая переменная типа boolean занимает один байт в памяти микроконтроллера.</p> <p>► <a href="#">Пример использования</a></p>  |
| <b>char</b>                         | <pre>// оба значения эквивалентны char val = 'A'; char val = '65';</pre>  | <p>Тип данных char хранит символьное значение и занимает в памяти 1 байт. Символы пишутся в одинарных кавычках, например: 'A', но в памяти символы хранятся в виде чисел.</p> <p>► <a href="#">Пример использования</a></p>  |
| <b>byte</b>                         | <pre>byte val = 255;</pre>  | <p>byte — без знаковый тип данных для хранения</p>   |

|                     |  |  |
|---------------------|--|--|
|                     |  | чисел в диапазоне от 0 до 255. Переменная занимает в памяти 1 байт.<br>▶ <a href="#">Пример использования</a>  |
| <b>int</b>          | <code>int val = 32767;</code>          | Тип данных для хранения целых чисел. Переменная типа <code>int</code> хранит целочисленные 16-битные значения в диапазоне от -32768 до 32767.<br>▶ <a href="#">Пример использования</a>  |
| <b>unsigned int</b> | <code>unsigned int val = 65535;</code> | Переменная типа <code>unsigned int</code> также может хранить двухбайтовые значения. Но вместо отрицательных чисел хранит только положительные значения в большом диапазоне от 0 до 65535.<br>▶ <a href="#">Пример использования</a>   |
| <b>float</b>        | <code>float val = 25.1547;</code>      | Переменная типа <code>float</code> служит для хранения чисел с десятичным разделителем. Числа с плавающей точкой позволяют более точно описать аналоговые величины, чем целые числа. Точность дробных чисел составляет 6-7 знаков — это общее количество цифр, а не количество цифр после запятой.<br>▶ <a href="#">Пример использования</a> |

## 2. Справочник по теме «Функции»

(для перехода на сайт с примерами использования нажмите на название функции)

### 1) ЦИФРОВОЙ ВВОД/ВЫВОД

- [pinMode\(\)](#) // Установить режим работы пина на вход или выход.
- [digitalWrite\(\)](#) // Подает HIGH или LOW на цифровой пин.
- [digitalRead\(\)](#) // Считывает значение с заданного пина.

### 2) АНАЛОГОВЫЙ ВВОД/ВЫВОД

- [analogRead\(\)](#) // Считывает значение с указанного контакта.
- [analogReference\(\)](#) // Определяет опорное напряжение.
- [analogWrite\(\)](#) // Выдает аналоговую величину на контакт.

### 3) ФУНКЦИИ РАБОТЫ СО ВРЕМЕНЕМ

- [millis\(\)](#) // Возвращает количество миллисекунд с момента начала выполнения текущей программы.
- [micros\(\)](#) // Возвращает количество микросекунд с момента начала выполнения текущей программы.
- [delay\(\)](#) // Останавливает выполнение программы на заданное количество миллисекунд.
- [delayMicroseconds\(\)](#) // Останавливает выполнение программы на заданное количество микросекунд.

### 4) МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

- [min\(\)](#) // Возвращает наименьшее из двух значений.
- [max\(\)](#) // Возвращает наибольшее из двух значений.
- [abs\(\)](#) // Возвращает модуль числа.
- [constrain\(\)](#) // Возвращает число, так чтобы оно было в области допустимых значений
- [map\(\)](#) // Пропорционально переносит значение из текущего диапазона в новый.
- [pow\(\)](#) // Вычисляет значение возведенное в заданную степень.
- [sq\(\)](#) // Возвращает квадрат числа.
- [sqrt\(\)](#) // Вычисляет квадратный корень числа.

### 5) ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ

- [sin\(\)](#) // Возвращает синус угла, заданного в радианах.
- [cos\(\)](#) // Возвращает косинус угла, заданного в радианах.
- [tan\(\)](#) // Возвращает тангенс угла, заданного в радианах.

### 6) ГЕНЕРАТОРЫ СЛУЧАЙНЫХ ЗНАЧЕНИЙ

- [randomSeed\(\)](#) // Инициализирует генератор псевдослучайных чисел.

- b. [random\(\)](#) // Возвращает псевдослучайное число.
- 7) ДОПОЛНИТЕЛЬНЫЕ ФУНКЦИИ ВВОДА/ВЫВОДА
  - a. [tone\(\)](#) // Генерирует сигнал прямоугольной формы.
  - b. [noTone\(\)](#) // Останавливает сигнал, созданный командой `tone()`.
  - c. [shiftOut\(\)](#) // Последовательно выводит байт информации на порт ввода/вывода.
  - d. [pulseIn\(\)](#) // Считывает длину сигнала на заданном порту.
- 8) ВНЕШНИЕ ПРЕРЫВАНИЯ
  - a. [attachInterrupt\(\)](#) // Задаёт функцию обработки внешнего прерывания.
  - b. [detachInterrupt\(\)](#) // Выключает обработку внешнего прерывания.
- 9) ФУНКЦИИ ПЕРЕДАЧИ ДАННЫХ
  - a. [Serial](#) // Набор функций **Serial** служит для связи устройства Ардуино с компьютером или другими устройствами.

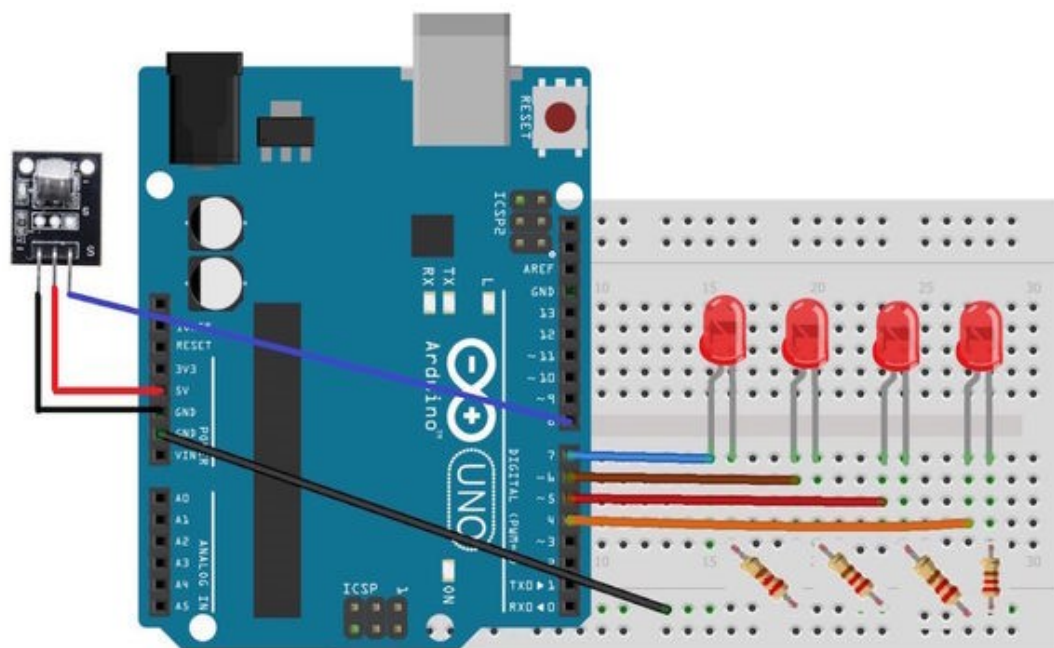
## Материал для практических работ (схема соединений и код программы) по разделу 3 «Проектирование устройств на Arduino»

### 1. Управление работой светодиода

*Вариант 1: Управление работой светодиода через ИК-пульт*

КОД: [http://arduino.zl3p.com/projects/led\\_remote\\_control](http://arduino.zl3p.com/projects/led_remote_control)

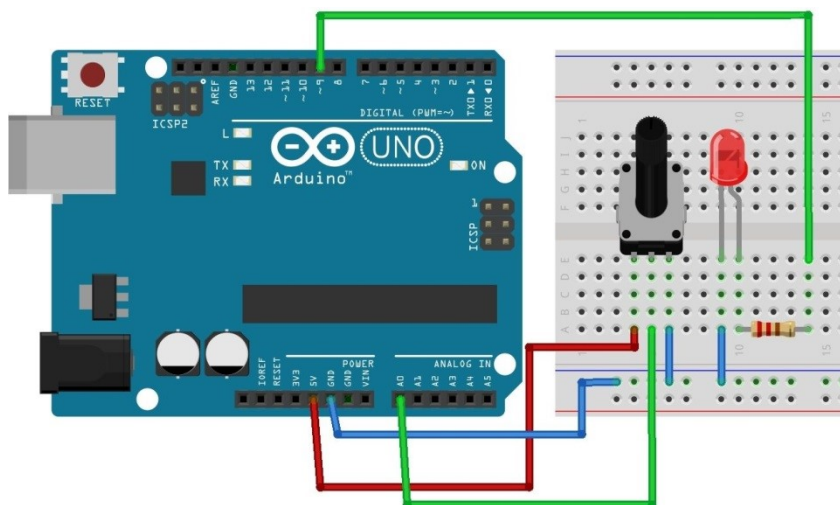
СХЕМА:



*Вариант 2: Управление работой светодиода через потенциометр*

КОД: <http://edurobots.ru/2014/04/arduino-potenciometr/>

СХЕМА:



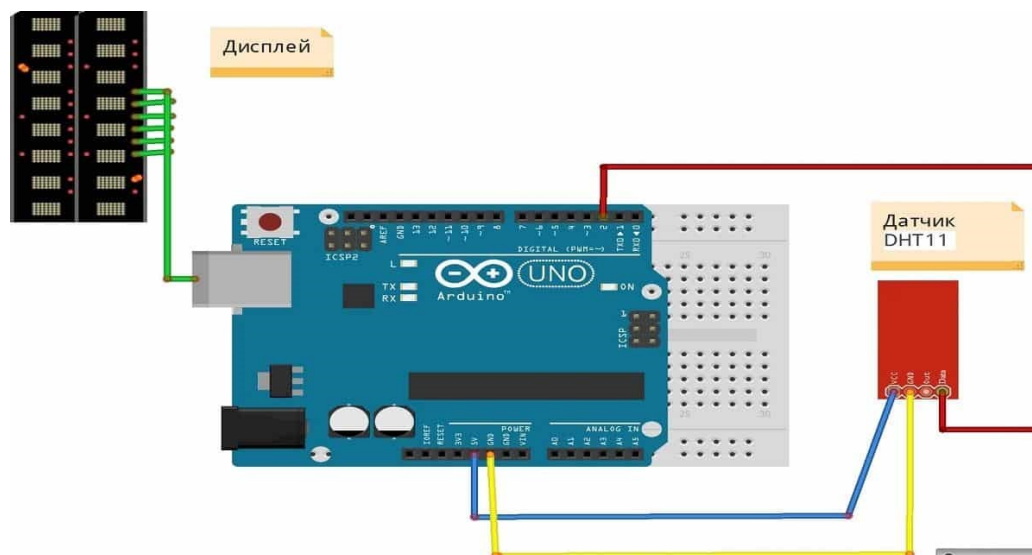


## 2. «Метеостанция»

### Вариант 1 (на дисплее)

КОД: <https://www.asutpp.ru/meteostantsiya-na-arduino-svoimi-rukami.html>

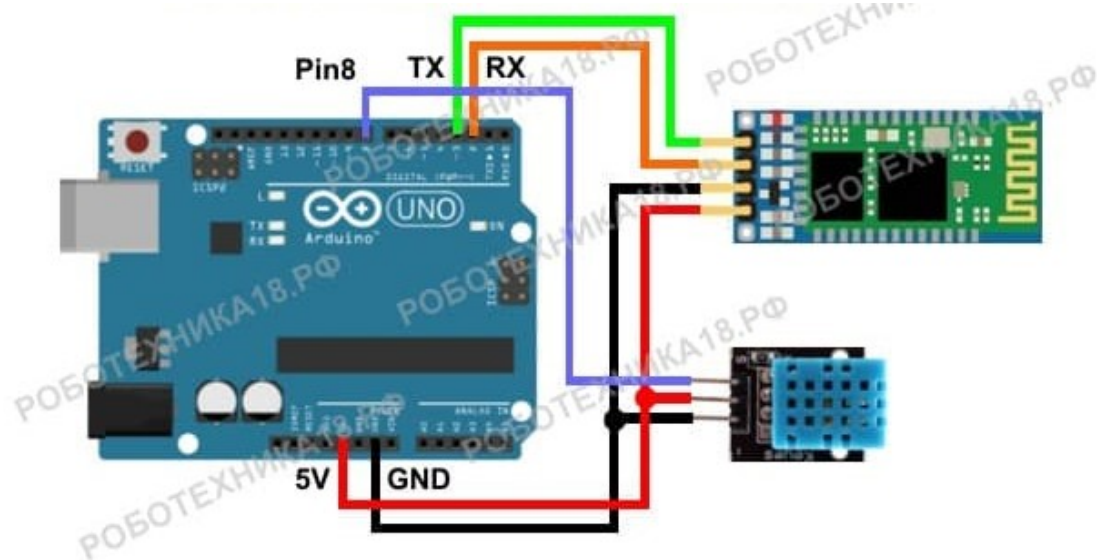
СХЕМА:



### Вариант 2 (с передачей данных по Bluetooth)

КОД: <https://clck.ru/uQuom>

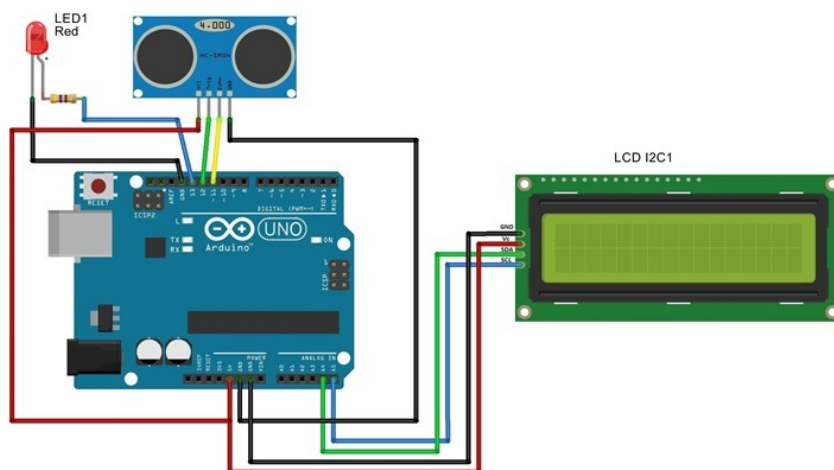
СХЕМА:



### 3. Дальномер ультразвуковой

КОД : <http://publicatorbar.ru/2016/04/18/control-lcd1602-ultrazvukovoj-datchik-hc-sr04/>

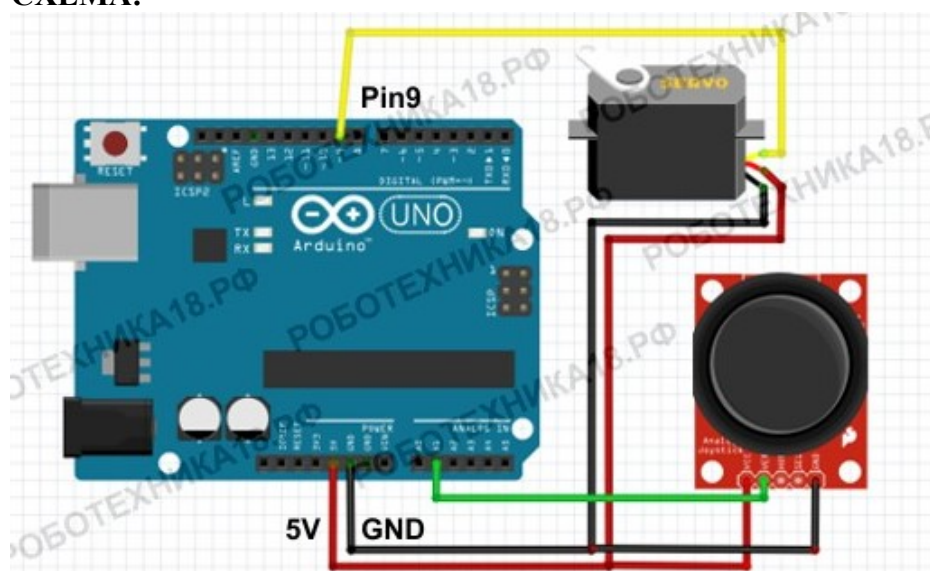
СХЕМА:



### 4. Управление сервоприводом джойстиком:

КОД : <https://clck.ru/uRstP>

СХЕМА:

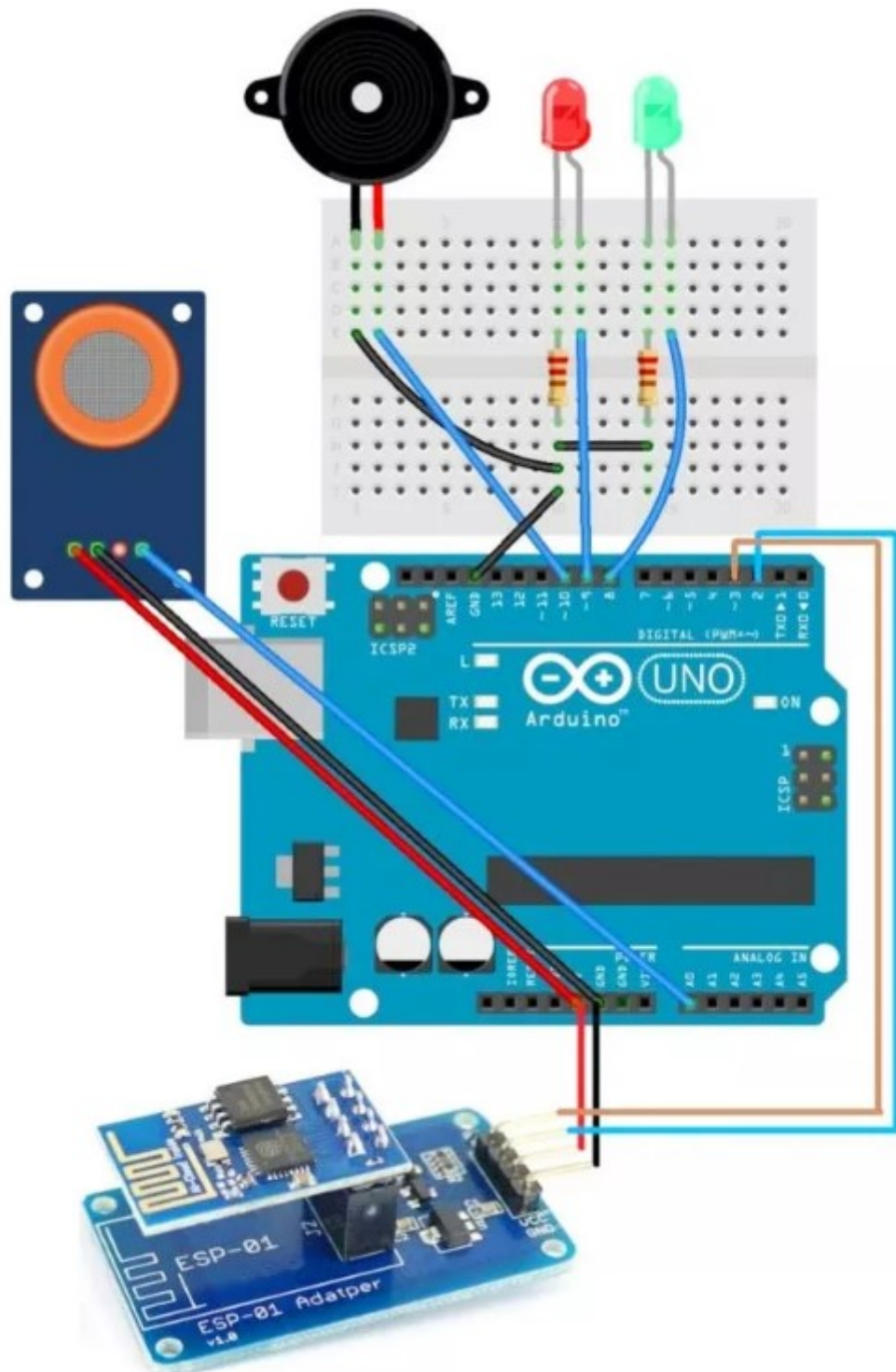




## 5. Устройство обнаружения дыма.

КОД : <https://arduinoplus.ru/dimovaya-signalizacia-arduino/>

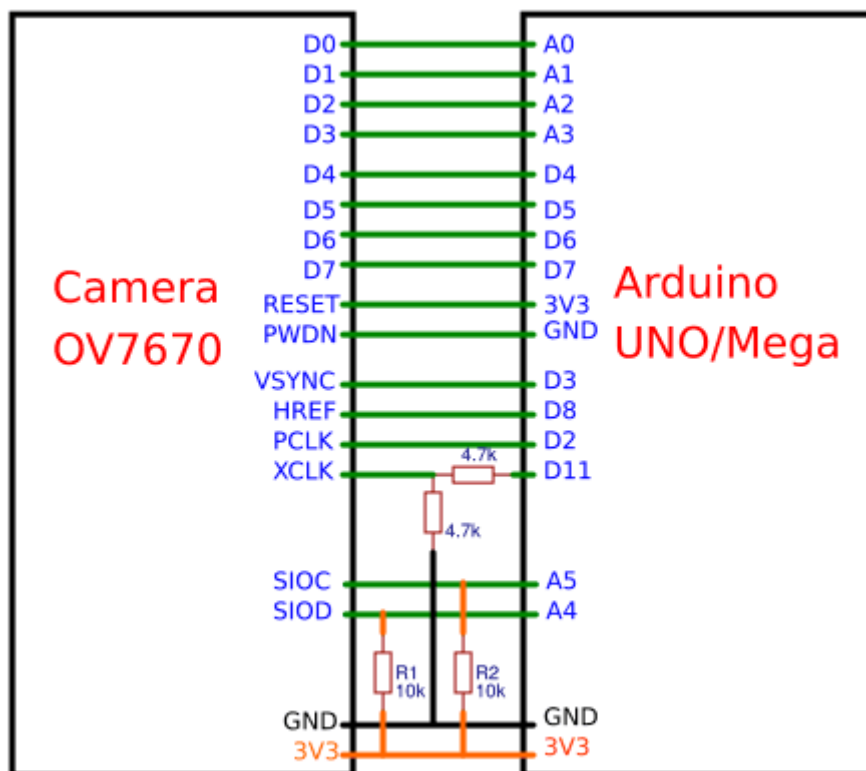
СХЕМА:



## 6. Передача изображения с модуля камеры OV7670 на ПК

КОД : <https://microkontroller.ru/arduino-projects/podklyuchenie-kamery-ov7670-k-arduino-uno/>

СХЕМА:



## Контрольно-измерительные материалы

### 1. Тест для входного контроля (примерная работа)

Работа состоит из 5 тестовых заданий, среди ответов на которые может быть несколько правильных.

1. На источнике тока написано: input  $U = 220\text{ В}$  и output  $U = 6,3\text{ В}$ . Какое напряжение МОЖЕТ выдать на выходе данный источник.

- 1) 220 В
- 2) 2 В
- 3) 6,3 В
- 4) 10 В

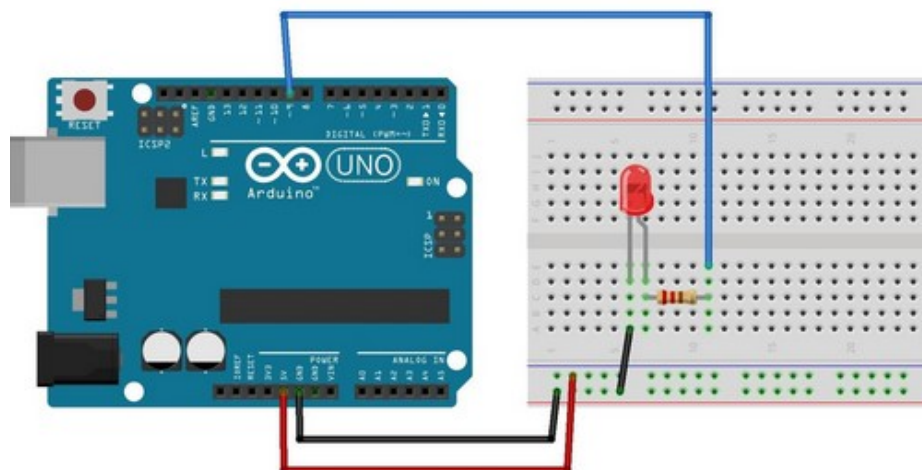
2. Что означает выход GND на плате?

- 1) нулевой потенциал (земля)
- 2) выход для подключения питания 5В
- 3) цифровой выход
- 4) аналоговый выход

3. На каком языке программирования основан язык программирования плат Arduino?

- 1) C/C++
- 2) Python
- 3) C#
- 4) Java

4. Для какой цели в данной схеме используется резистор, последовательно соединенный со светодиодом?



- 1) Для уменьшения силы тока текущего через светодиод
- 2) Для увеличения яркости свечения светодиода
- 3) Для увеличения силы тока текущего через светодиод

5. Какова правильная полярность подключения светодиода?

- 1) Длинная ножка (анод) к «минусу» питания, короткая ножка (катод) – к «плюсу»
- 2) Длинная ножка (катод) к «плюсу» питания, короткая ножка (анод) – к «минусу»
- 3) Длинная ножка (анод) к «плюсу» питания, короткая ножка (катод) – к «минусу»
- 4) Принципиальных отличий нет

## 2. Письменная работа по теме «Основы программирования Arduino» (примерная работа)

Работа содержит 3 задания, требующие подробного решения

### Вариант 1.

**Задание 1.** Перед вами часть программы в среде Arduino IDE. Поясните данную часть кода.

```
for(i = 100; i <= 1000; i+=100) { }
```

**Задание 2.** Предложите алгоритм нахождения наибольшего числа в массиве.

**Задание 3.** Запишите, как будет выглядеть конструкция «Функция  $y=0$ , если  $x <= 0$  и равна  $x+4$ , если  $x > 0$ » в Arduino IDE.

### Вариант 2.

**Задание 1.** Перед вами часть программы в среде Arduino IDE. Поясните данную часть кода.

```
if (a >= 10 && a <= 20) { }
```

**Задание 2.** Предложите алгоритм нахождения в тексте слов, содержащих в себе следующие символы «ава».

**Задание 3.** Запишите, как будет выглядеть конструкция «Функция  $y$  равна 0, если  $x$  больше 0 и равна  $x^2$ , если  $x$  меньше или равен 0» в Arduino IDE.

## 3. Письменная работа по теме «Проектирование устройств на Arduino» (примерная работа)

Работа содержит 2 задания, требующие подробного решения

### Вариант 1.

**Задание 1.** Предложите идею устройства, главным датчиком которого является датчик цвета TCS230. Опишите его назначение.

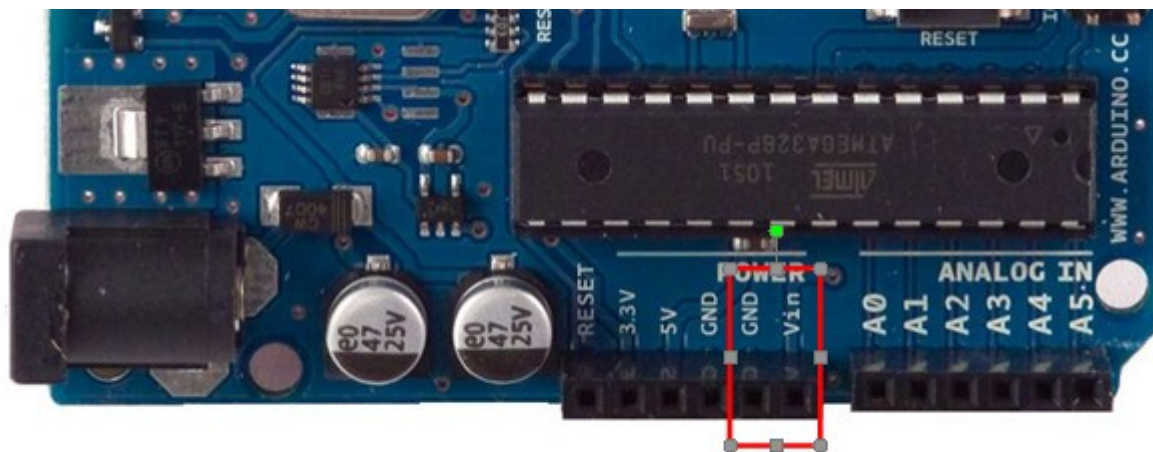
**Задание 2.** Опишите назначение выходов, отмеченных на плате Arduino UNO красным квадратом.



## Вариант 2.

**Задание 1.** Предложите идею устройства, главным датчиком которого является датчик цвета Датчик наклона и вибрации SW-520D. Опишите его назначение.

**Задание 2.** Опишите назначение выходов, отмеченных на плате Arduino UNO красным квадратом.



## Список литературы

### Для учителя:

1. Асмолов А. Г., Бурменская Г. В., Володарская И. А. и др. Формирование универсальных учебных действий в основной школе: от действия к мысли. Система заданий: пособие для учителя - 2-е изд. - М. : Просвещение, 2011. — 159 с.
2. Белиовская Л.Г., Белиовский А.Е. Програмируем микрокомпьютер NXT в LabVIEW. – М.: ДМК Пресс, 2010. – 280с
3. Волкова С. И. «Конструирование», - М: «Просвещение», 2009 .
4. Вязовов С.М., Калягина О.Ю., Слезин К.А.. Соревновательная робототехника: приёмы программирования в среде EV3: учебно-практическое пособие. – М.: Издательство «Перо», 2014. – 132с.
5. Гайсина И.Р. Развитие робототехники в школе [Текст] / И.Р.Гайсина // Педагогическое мастерство (II): материалы междунар. заоч. науч. конф. (г. Москва, декабрь 2012 г.). — М.: Буки-Веди, 2012. — С. 105- 107
6. Горский В.А. Техническое конструирование. – М.: Дрофа, 2010.- 112 с. 11. Злаказов А.С. Уроки Лего-конструирования в школе: методическое пособие. – М.: БИНОМ. Лаборатория знаний, 2011. – 120с.
7. Катцен С. PIC-микроконтроллеры. Все, что вам необходимо знать/ пер. с англ. Евстифеева А.В. — М.: Додэка-XXI, 2008- 656 с.
8. Кравченко А.В. 10 практических устройств на AVR-микроконтроллерах. — М.: Издательский дом «Додэка-XXI», К. «МК-Пресс», 2008. — 224с
9. Муллен, Р. HTML4: Справочник программиста .– СПб: Питер Ком, 2015. – 304 с.
10. Петин В.А. Arduino и Raspberry Pi в проектах Internet of Things. - СПб.: БХВ-Петербург, 2016. -320 с.: ил. -(Электроника).
11. Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам», утв. приказом Минпросвещения России от 09.11.2018 № 196
12. Программа развития воспитательной компоненты, Письмо МО РФ от 13.05.2013 №ИР-352/09
13. Ревич Ю.В. Занимательная микроэлектроника. — СПб.: БХВ-Петербург, 2007. — 592с. 11.
14. Федеральный закон от 29.2012 № 273 «Об образовании в Российской Федерации».
15. Филиппов С.А. Робототехника для детей и родителей. – СПб.: Наука, 2013. 319 с
16. Халамов В.Н. и др. Образовательная робототехника во внеурочной деятельности: учебно-методическое пособие - Челябинск: Взгляд, 2011. – 96с. : ил.;
17. Эванс Б. Arduino блокнот программиста /пер. с англ. В.Н.Гололобов (электронная книга)
18. Юревич Е.И. Основы робототехники – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2005. – 416 с.
19. Юревич Е.И. Основы робототехники – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2005. – 416 с.: ил.;

### Для обучающихся:

1. Блум Джереми. Изучаем Arduino: инструменты и методы технического волшебства: Пер. с англ. – СПб.: БХВ-Петербург, 2015 – 336 с.
2. Быстрый старт. Первые шаги по освоению Arduino (методическое пособие к набору).

3. Петин В.А. Практическая энциклопедия Arduino. – М.: ДМК Пресс, 217. – 152 с.
4. Ревич Ю.В. Занимательная микроэлектроника. – СПб.: БХВ-Петербург, 2007. – 592с.
5. Филиппов С.А. Робототехника для детей и родителей – СПб.: Наука, 2011. – 263 с.:

### **Интернет ресурсы:**

1. ArduinoMaster.Ru. Проекты ардуино для начинающих [Электронный ресурс] — Режим доступа: <https://arduinomaster.ru/projects/proekty-arduino-dlya-nachinayushhih/>
2. Амперка. База знаний Амперки [Электронный ресурс] — Режим доступа: <http://wiki.amperka.ru/>
3. Единая коллекция цифровых образовательных ресурсов [Электронный ресурс]. / режим доступа <http://school-collection.edu.ru>
4. Каталог сайтов по робототехнике - полезный, качественный и наиболее полный сборник информации о робототехнике. [Электронный ресурс] — Режим доступа: <http://robotics.ru/>
5. Официальный сайт компании Arduino[Электронный ресурс] — Режим доступа: <https://arduino.ru/>.
6. Робототехника18.рф. Ардуино проекты для начинающих [Электронный ресурс] — Режим доступа: <https://clck.ru/tjRSE>
7. Федеральный центр информационно-образовательных ресурсов (ФЦИОР) [Электронный ресурс]. / режим доступа <http://fcior.edu.ru>