

Федеральное государственное автономное образовательное учреждение
высшего образования

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
(НИУ «БелГУ»)**

**ФАКУЛЬТЕТ МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНОГО
ОБРАЗОВАНИЯ**

**КАФЕДРА ИНФОРМАТИКИ, ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН И
МЕТОДИК ПРЕПОДАВАНИЯ**

**РАЗРАБОТКА ФАКУЛЬТАТИВНОГО КУРСА ДЛЯ УЧАЩИХСЯ 10-11
КЛАССОВ НА ТЕМУ «ОСНОВЫ РОБОТОТЕХНИКИ»**

Выпускная квалификационная работа
обучающегося по направлению подготовки 44.03.05 Педагогическое
образование профили Физика и математика
очной формы обучения, группы 02041301
Першина Вадима Леонидовича

Научный руководитель
к.ф.-м.н., доцент
Величко М. А.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 ТЕОРЕТИЧЕСКИЕ ДАННЫЕ, НЕОБХОДИМЫЕ ДЛЯ РАЗРАБОТКИ ФАКУЛЬТАТИВНОГО КУРСА.....	6
1.1 Основные понятия о факультативном курсе.....	6
1.2 Основные данные о плате Arduino UNO	13
1.2.1 Плата Arduino Uno	13
1.2.2 Распиновка и их значение	15
1.2.3 Подключение контроллера Arduino к ПК, и работа с платой	18
1.2.4 Память и её параметры на плате Arduino Uno	19
1.3 Методы обучения, используемые в процессе преподавания робототехники	13
1.4 Робототехника как средство формирования ключевых компетенций учащихся	25
2 ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА ПО РОБОТОТЕХНИКЕ	30
2.1 Методы и формы организации проектной деятельности по робототехнике	30
2.2 Факультативный курс «основы робототехники».....	32
2.3 Анализ педагогического эксперимента	52
ЗАКЛЮЧЕНИЕ	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	60
ПРИЛОЖЕНИЕ А	63
Лабораторная работа №5	63
Лабораторная работа №6	65
Лабораторная работа №7	66
Лабораторная работа №8	69
Лабораторная работа № 9.....	73

ВВЕДЕНИЕ

Человек – самое развитое существо на нашей планете. И достичь таких успехов человеку помог интеллект. Именно благодаря интеллекту человечество научилось использовать орудия труда для облегчения своей работы.

Сейчас на дворе XXI век, а человечество продолжает искать способы облегчить свою жизнь. И на данный момент этими «орудиями труда» являются информационные технологии. Уже сейчас мы используем: высокомоощные компьютеры, многофункциональные телефоны, беспроводные сети и много что ещё. Ко всему многообразию добавились ещё и роботы. Эти роботы являются прямыми помощниками в таких опасных местах как: вулканы; глубоководные океанические впадины; космос; промышленность, где человек не в силах находиться и самостоятельно выполнять некоторые ответственные работы (ядерные реакторы, химические заводы, металлургические заводы и т.д.); медицина, где точность важнейший из параметров без которых невозможно успешно завершить операцию или диагностировать заболевание; в армии; роботов так-же используют; в быту роботы тоже нашли свою нишу (пылесосы, мойщики стёкол и т.д.) и это далеко не полный список. В общем сфера деятельности роботов весьма и весьма обширна.

«Современные роботы, созданные на базе самых последних достижений науки и техники, применяются во всех сферах человеческой деятельности. Люди получили верного помощника, способного не только выполнять опасные для жизни человека работы, но и освободить человечество от однообразных рутинных операций» [12].

Робототехника — прикладная наука, занимающаяся разработкой автоматизированных технических систем и являющаяся важнейшей технической основой интенсификации производства [10].

Робототехника - это проектирование, конструирование и программирование всевозможных интеллектуальных механизмов – роботов. Роботами управляют процессоры, которых в наше время великое множество: от миниатюрных микропроцессоров, до гигантских компьютеров, которые управляют множеством действий одновременно. В случае с данным факультативным курсом в роли процессора будет выступать платформа Arduino[23].

Но есть и проблема. Роботами необходимо качественно управлять. Ведь развитие не стоит на месте. Каждый день выходят новые и новые гаджеты, датчики и всевозможные устройства, которые не каждый поймёт, как их использовать? И робототехника не исключение. Она объединяет множество естественных наук, от физики и информатики до биологии. Одним из вариантов решения данной проблемы является включение в образовательный процесс курсов робототехники. Введение в школу подобных курсов бы возможность для более качественно подготавливать молодое поколение к быстро меняющемуся миру информационных технологий. Также это способствовало лучшему изучению в школе дисциплин, непосредственно связанных с робототехникой.

Как показывает время миру нужны специалисты, обладающие знаниями в этой очень обширной области современной цивилизации. И начинать их подготовку необходимо как можно раньше. Но каким образом этого достичь? Ответ будет как донельзя простым. Именно со школы необходимо закладывать начальные знания о робототехнике, конструированию и программированию. Именно она должна прививать молодому поколению интерес к этой перспективной в наше время области развития.

Отсюда выявляется проблема исследования: разработка факультативного курса на тему: «Основы робототехники»

Объект исследования: учащиеся 10-11 классов.

Предмет исследования: плата Arduino Uno и наборы датчиков.

Цель исследования: разработать и апробировать факультативный курс в 10-11 классы средней общеобразовательной школы.

Задачи исследования:

1) Проанализировать литературу, обосновать возможность и актуальность подобного факультативного курса в современной образовательной среде;

2) Подобрать материал для курса в соответствии с требованиями образовательных учреждений и современных образовательных технологий.

3) Разработать факультативный курс на тему: «Основы робототехники»;

4) Апробировать часть разработанного факультативного курса и проанализировать его результаты.

5) По результатам апробирования произвести соответствующие выводы.

Структура дипломной работы. Работа состоит из введения, двух глав, заключения, списка используемой литературы, приложения.

1 ТЕОРЕТИЧЕСКИЕ ДАННЫЕ, НЕОБХОДИМЫЕ ДЛЯ РАЗРАБОТКИ ФАКУЛЬТАТИВНОГО КУРСА

1.1 Основные понятия о факультативном курсе

Факультативный курс (факультатив) (франц. facultatif - от лат. facultas - возможность) - необязательный учебный курс или предмет, изучаемый студентами вузов и учащимися средних учебных заведений по их желанию для углубления и расширения научно-теоретических знаний [29].

Еще на рубеже XIX и XX вв. некоторые педагоги поняли, что преподавание в общеобразовательной школе какого-либо предмета по обязательной единой общегосударственной программе становится более успешным, если его дополнить циклом необязательных для учащихся внепрограммных групповых занятий. Такие занятия должны были, прежде всего, учитывать «местные условия», а именно: реальные и потенциальные запросы и интересы конкретного коллектива учащихся данного класса, реальные возможности учителя вызвать и развить интерес учащихся к важным аспектам данного предмета, не охваченного обязательной программой. Так возникла идея факультативных занятий в школе [24].

Назначение факультативных курсов состоит в развитии их коммуникативных способностей, развивает навыки взаимодействия, самостоятельности при принятии решений, раскрывает их творческий потенциал. Дети и подростки лучше понимают, когда они что-либо самостоятельно создают или изобретают [1].

Факультативные курсы являются одной из форм дифференцированного обучения. Главной целью факультативных курсов по робототехнике является организованность занятости школьников во внеурочное время с учетом всестороннего развития личности (развитие навыков конструирования, развитие логического мышления).

Уникальность образовательной робототехники заключается в возможности объединить конструирование и программирование в одном курсе, что способствует интегрированию преподавания информатики, математики, физики, черчения, естественных наук с развитием инженерного мышления, через техническое творчество.

Основная задача факультативных курсов: учитывая интересы и склонности учащихся, расширить и углубить знания по робототехнике, обеспечить усвоение ими данного материала, ознакомить школьников с некоторыми общими идеями современной техники, раскрыть техническое творчество.

Факультативные курсы играют большую роль в наше время, бурное развитие технологий накладывает свой отпечаток и на образовательный процесс. Они позволяют прививать интерес учащихся к робототехнике и автоматизированным системам, расширяют знания по школьным дисциплинам непосредственно связанных с факультативным курсом.

Программа факультативных занятий по робототехнике составлена так, что все вопросы ее могут изучаться синхронно с изучением курсов математики, физики, информатике в школе.

Факультативные курсы школьники посещают по желанию, следовательно, педагогу необходимо создать условия, при которых способные ученики смогут реализовать свои возможности, а остальные учащиеся смогут решать посильные для них задачи или, пользуясь помощью учителя, более трудные задания [19].

Для того чтобы факультативные занятия по робототехнике были эффективными, необходимо их организовать там, где есть:

- 1) Высококвалифицированные учителя или другие специалисты, способные вести занятия на высоком научно-методическом уровне;
- 2) Не менее 15 учащихся, желающих изучать данный факультативный курс.

Если школа имеет классы с небольшой наполняемостью (что особенно характерно для некоторых сельских школ), то группы учащихся для факультативных занятий можно комплектовать по параллелям или из учащихся смежных классов (10 «А» -10 «Б» классы, 10-11 классы и т. п.) [11].

Запись учащихся на факультативные курсы производится на добровольных началах в соответствии с их интересами. Не следует принуждать учащихся обязательно изучать факультативные предметы. Особенно внимательно следует относиться к тем учащимся, которые встречаются трудности в изучении информатики или совмещают обучение в школе с другими видами занятий (спорт, музыка и т. д.). По окончании факультативного курса учащиеся сдают зачет (с оценкой), о чем делается отметка в аттестате.

Учитель несет полную ответственность за качество факультативных занятий; факультативные курсы вносятся в расписание и оплачиваются учителю. Примечательной особенностью факультативного курса является то, что программа курса для каждого класса составлена из ряда основных тем (независимых друг от друга). Однако содержание учебной работы учащихся на факультативных занятиях определяется не только знаниями по информатике, физики и математики, но и другими дисциплинами, а также

- 1) Различными методическими факторами;
- 2) Характером объяснения учителя;
- 3) Соотношением теории и учебных упражнений;
- 4) Содержанием познавательных вопросов и задач;
- 5) Сочетанием самостоятельной работы и коллективного обсуждения полученных каждым учащимся результатов.

Данный курс направлен на то, чтобы положить начало формированию у обучающихся целостного представления о мире техники, устройстве конструкций, механизмов и машин, их месте в окружающем мире. Реализация данного курса позволяет стимулировать интерес и любознательность, развивать способности к решению проблемных ситуаций

умению исследовать проблему, анализировать имеющиеся ресурсы, выдвигать идеи, планировать решения и реализовывать их, расширить технический и математический словари воспитанника.

Проведение факультативных курсов по робототехнике не означает отказа от других форм внеклассной работы (математические кружки, вечера, олимпиады и т. д.). Они должны дополнять эти формы работы с учащимися, которые интересуются робототехникой.

Требования к проведению факультативных курсов:

1) Преимущество в содержании, методах и формах организации занятий по робототехнике должна определяться целями всестороннего развития и воспитания учащихся.

2) Должно быть взаимосвязанное построение уроков и факультативных занятий по робототехнике.

3) Школа должно предоставить компьютерный класс для проведения факультативных курсов по робототехнике.

4) Не должно быть несогласованности с нормами организации работы общеобразовательной школы.

5) Главным критерием эффективности взаимосвязанного построения факультативных занятий по робототехнике должна быть результативность неразрывно связанных друг с другом процессов обучения, развития и воспитания школьников.

6) Факультативный курс носит практический характер; курсы по робототехнике целесообразно проводить, учитывая их функции – развивающую, воспитывающую и учебную.

Методические рекомендации по организации факультативных курсов:

а) Взаимосвязь в содержании, формах и методах организации учебной работы и факультативных курсов.

б) Обеспечение взаимосвязи (по содержанию) уроков и факультативных занятий.

в) Активизация самостоятельной работы учащихся.

г) Построение учебного процесса как совместной практической деятельности учащихся.

д) Использование наглядных пособий, применение конспект-таблиц на лекциях.

е) Использование практических заданий по темам на факультативных занятиях.

ж) Использование научно-исследовательских и опытно-конструкторских и технологических работ на факультативных курсах.

и) Принципы занимательности занятий.

к) Построение занятий проблемного изучения материала.

л) Прежде всего, факультативные курсы должны быть интересными, увлекательными для школьников. Хорошо известно, что занимательность изложений помогает раскрытию содержания сложных научных понятий и проблем. Заинтересованность поможет школьникам освоить факультативный курс, содержащиеся в нем идеи робототехники, науки, логику, и приемы творческой деятельности. В этом отношении цель учителя - добиться понимания учениками того, что они подготовлены к работе над сложными техническими разработками, однако для этого необходима заинтересованность, трудолюбие, усидчивость в организации своей работы [15].

Возможность работать со школьниками во внеурочное время, проявляющими повышенный интерес и способности к робототехнике, представляет собой одно из проявлений новой формы обучения - дифференцированного обучения.

В какой бы форме не проходились факультативные курсы по робототехнике, они должны быть для учащихся интересными, увлекательными, занимательными и дающими новые знания. Необходимо использовать естественную любознательность школьника для формирования устойчивого интереса к данному курсу.

Известный французский физик Луи де Бройль писал, что современная наука – «дочь удивления и любопытства, которые всегда являются ее скрытыми движущими силами, обеспечивающими ее непрерывное развитие».

Основными формами проведения факультативных курсов по робототехнике являются в настоящее время изложение узловых вопросов данного факультативного курса учителем (лекционным методом), семинары, собеседования (дискуссии), рефераты учащихся (как по теоретическим вопросам, так и по практическим частям), доклады учащихся и т. д.

Однако учителю не следует отдавать предпочтение какой-либо одной форме или методу изложения. Факультативный курс по робототехнике предусматривает отведение большего времени на практическую часть работы учащихся, следует все же чаще применять рефераты, доклады, семинары-дискуссии, чтение учебной и научно-популярной литературы и т. п.

Одной из возможных форм ведения факультативных курсов по робототехнике является разделение каждого занятия на две части. Первая часть посвящается изучению нового материала и самостоятельной работе учащихся по заданиям теоретического характера. Вторая часть каждого занятия включает в себя практическую работу, где учащиеся 10-11 классов могут проявить себя и свои творческие способности. Введение факультативных курсов может способствовать успешному переходу от форм и методов обучения в школе к формам и методам обучения в высших учебных заведениях [13].

Если представить изучаемый факультативный курс в виде серии последовательно расположенных задач. Решая последовательно все задачи самостоятельно или при незначительной помощи преподавателя, школьники постепенно изучают курс при большом личном участии, проявляя активность и самостоятельность, овладевая техникой логического мышления.

Определения либо включаются преподавателем в текст задачи, либо сообщаются особо. В необходимых случаях преподаватель проводит предварительную беседу или делает обобщения.

В настоящее время факультативные курсы по робототехнике внедряются в школьную программу как урок. На самих занятиях качество усвоения теории проверяется в процессе решения задач, практических и лабораторных работ. На курсах совершенно недопустимы такие формы работы, которые сковывали бы инициативу учащихся на занятиях. Занятие начинается с постановки упражнения для всех учащихся. За время, которое отводится на выполнение задачи или примера, учитель успевает проследить, кто и как справляется с заданием, при этом не торопит учащихся. Обычно, если не все, то некоторые из них выполняют задание в запланированное учителем время, а затем начинается разбор и теоретическое обоснование решений. Инициатива в оценке способов решения, в исправлении ошибок, в постановке вопросов представляется самим учащимся. В процессе этой работы достигается логическая точность в формулировках определений понятия или их свойств. В заключительном слове учитель дает мотивированную оценку знаний учащихся [5].

На курсах разрешается практиковать постановку докладов учащихся, где при написании докладов использовалась различная дополнительная литература, указанная учителем. Не следует увлекаться большим количеством докладов, в противном случае, у учителя просто не хватит времени для хорошей подготовки докладчиков.

Среднее общее образование призвано помочь учителю реализовать способности каждого ученика и создать условия для индивидуального развития школьников.

Чем разнообразнее образовательная среда, тем легче раскрыть индивидуальность личности ученика, а затем направить и скорректировать развитие школьника с учетом выявленных интересов, опираясь на его природную активность.

Личностно-ориентированное обучение строится на принципе вариативности, т.е. признания разнообразия содержания и форм обучения, выбор которых осуществляется с учетом развития ребенка и его педагогической поддержки. Пытаясь создать условия для личностно ориентированного обучения, школа предоставляет учащимся право выбора предметов и внеурочных курсов по интересам и склонностям.

1.2 Основные данные о плате Arduino UNO

1.2.1 Плата Arduino Uno

Плата Arduino Uno (см рисунок 1) — центр большой империи Arduino, самое популярное и самое доступное устройство Arduino. Средняя её цена 2000 рублей в интернет магазинах [17].

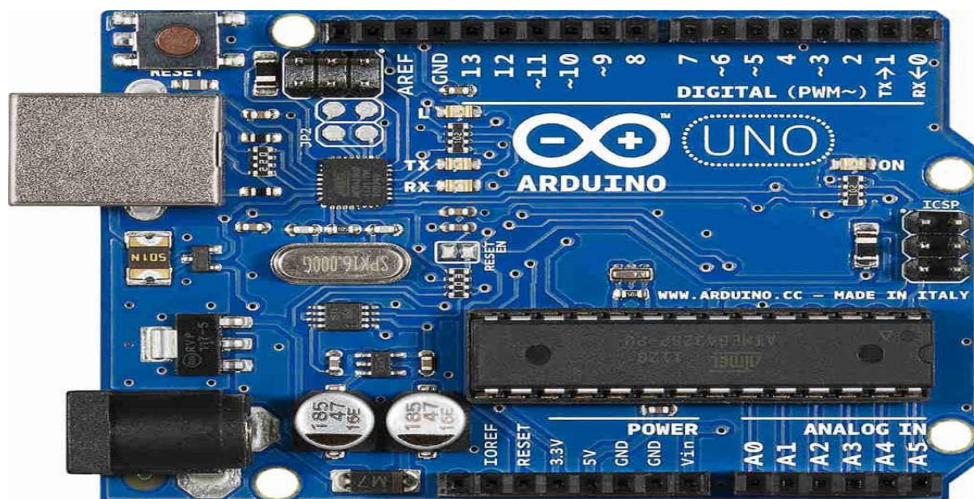


Рисунок 1 - Плата Arduino Uno. Вид сверху

Данная плата построена на итальянском микроконтроллере ATmega328, однако последнее поколение этих плат использует для связи с компьютером по USB микроконтроллер ATmega8U2. Платформа имеет 14

цифровых вход/выходов (6 из которых могут использоваться как выходы ШИМ), 6 аналоговых входов, кварцевый генератор 16 МГц, разъем USB, силовой разъем, разъем ICSP и кнопку перезагрузки (см рисунок 2).

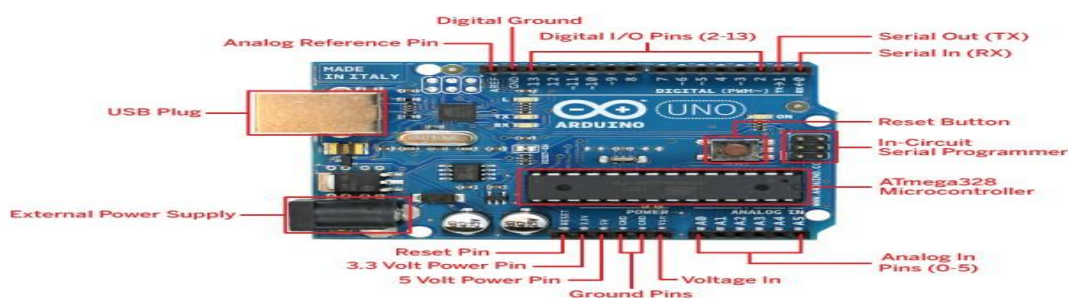


Рисунок 2 - Описание элементов платы Arduino Uno

Для работы необходимо подключить платформу к компьютеру посредством кабеля USB, либо подать питание при помощи адаптера AC/DC или батареи [22]. Ниже указаны официальные характеристики, представленные на сайте производителя (см. таблицу 1).

Таблица 1 Технические характеристики Arduino Uno

Микроконтроллер	ATmega328
Рабочее напряжение	5В
Входное напряжение питания	7-12В
Максимальное входное напряжение питания	6-20В
Цифровые входы/выходы	14 (из них 6 могут использоваться в качестве ШИМ-выходов)
Аналоговые входы	6
Постоянный ток через входы/выходы	40 мА
Постоянный ток для вывода 3,3 Вольт	50 мА
Флеш-память	32 Кбайта, но в тоже время 0.5 КБ используются загрузчиком
ОЗУ	2 КБ (ATmega328)
Энергозависимая память данных	1 КБ (ATmega328)
Тактовая частота	16 МГц

1.2.2 Распиновка и их значение

Пины – это ножки которые используются для того чтобы к плате можно было подключать внешние устройства (датчики, динамики и т.д.). работать они могут в двух режимах: Вход и Выход [25]. Каждый пин имеет встроенный резистор (20000-50000 Ом). Рекомендуемый ток – 20 милиАмпер, максимальный ток 40 милиАмпер на 1 пико секунду (см. рисунок 3)

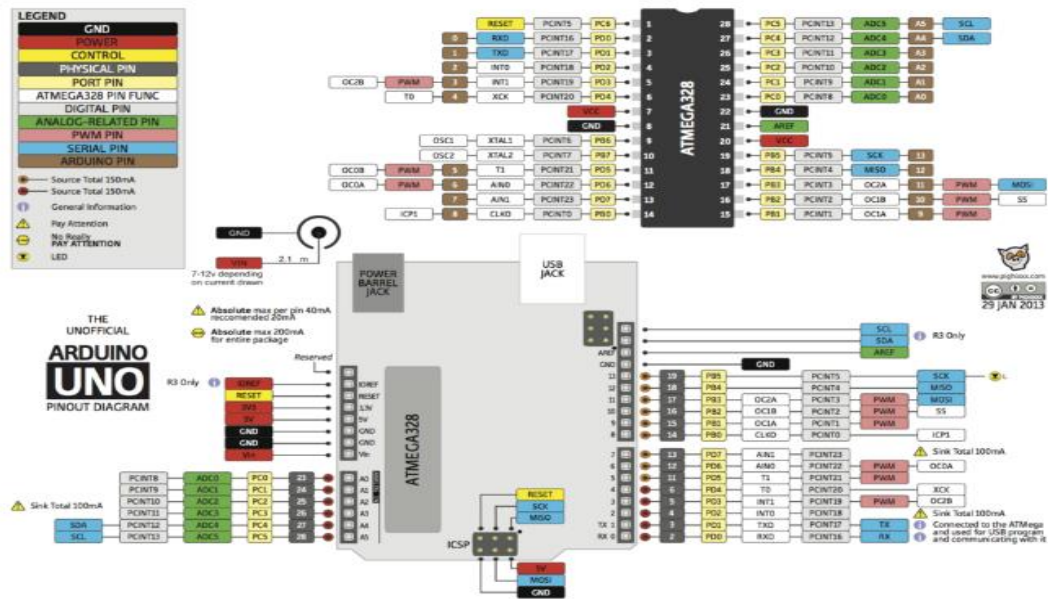


Рисунок 3 - Распиновка Платы Arduino Uno

Для комфортного использования и не загромождения платы разработчики на каждый пин установили несколько дополнительных функций (см. таблицу 2).

Таблица 2 Пины и их дополнительные функции

Пины	Контакты
0;1	RX; TX
10; 11; 12; 13	SS; MOSI; MISO; SCK
A4; A5	SDA;SCL

Пины от 0 до 13 - цифровые. Это означает, что эти пины управляются только двумя видами сигналов: высокий (high) и низкий (low). Также некоторые из пинов имеют возможность широко-импульсной модуляции, следовательно, эти пины могут управлять мощностью подаваемое на внешнее устройство [7]. Данные по всем цифровым пинам приведены ниже (см. таблицу 3).

Таблица 3 Цифровые пины на плате Arduino Uno

№ цифрового пина, и их обозначение в скетче	Специальное назначение	Широко-импульсной модуляции
0	RX	
1	TX	
2	Вход для прерываний	
3	Вход для прерываний	Есть
4	-	
5	-	Есть
6	-	Есть
7	-	
8	-	
9	-	Есть
10	SPI (SS)	Есть
11	SPI (MOSI)	Есть
12	SPI (MISO)	
12	SPI (SCK)	

Все цифровые пины, а их 14, могут быть настроены на два режима: вход и выход. Все они работают под напряжением 5 Вольт, также они имеют встроенные резисторы от 20000 до 50000 Ом, но эти резисторы необходимо включать в программной среде.

На платформе Arduino Uno установлено несколько устройств для осуществления связи с компьютером, другими устройствами Arduino или микроконтроллерами. ATmega328 поддерживают последовательный интерфейс UART TTL (5 В), осуществляемый выводами 0 (RX) и 1 (TX). Установленная на плате микросхема ATmega8U2 направляет данный

интерфейс через USB, программы на стороне компьютера "общаются" с платой через виртуальный COM порт. Прошивка ATmega8U2 использует стандартные драйвера USB COM. Мониторинг последовательной шины, программы Arduino позволяет посылать и получать текстовые данные при подключении к платформе. Некоторыми библиотеками можно создать последовательную передачу данных через любой из цифровых выводов Uno.

Пины с номерами от A0 до A5 – аналоговые. Подключенные к этим пинам устройства передают данные в аналого-цифровой преобразователь. На нашей плате этот аналого-цифровой преобразователь с разрешением на 10 бит, что даёт на выходе число от 0 до 1023.

Подключение внешних устройств к плате.

Как понятно из сказанного выше пины используются для подключения к ним внешних устройств. Самый простой способ проверить это, подсоединить светодиод (см. рисунок 4). Для этого используем два контакта: заземление или GND, и сигнальный [26].

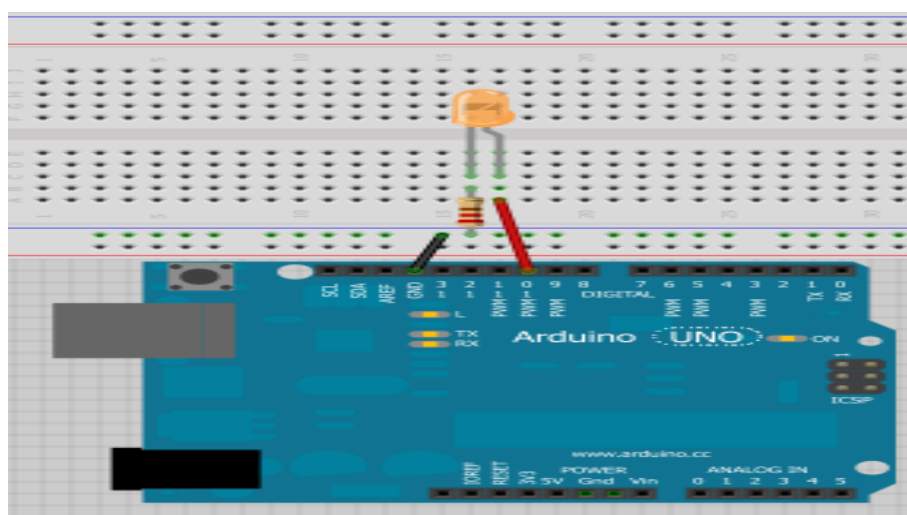


Рисунок 4 - подключение светодиода к плате.

Однако, необходимо чётко представлять, сколько энергии потребляет ваш датчик, поскольку использовать плату как источник питания следует только в том случае, если устройство не потребляет энергии больше чем

предельное напряжение контроллера. В противном случае у вас сгорят внутренние конденсаторы платы, и она перестанет функционировать.

1.2.3 Подключение контроллера Arduino к ПК, и работа с платой

На данный момент для того чтобы подключить Arduino к компьютеру необходимо использовать соответствующий плате разъём некоторые из них представлены ниже (см. рисунок 5). Соответственно: первый кабель «Принтер-USB» для плат Arduino Uno и Arduino Mega; второй кабель «mini-USB – USB» для платы Arduino Nano; третий кабель «micro-USB – USB» для платы Arduino Micro.



Рисунок 5 - Варианты кабелей для подключения платы к компьютеру

При подключении платы к компьютеру, программная среда автоматически воспринимает подключенные платы, подключённые к компьютеру. Для того чтобы программа смогла работать с платой необходимо задать номер порта, к которому подключен контроллер (см. Рисунок 6). После подключения необходимо выбрать нужный порт — для этого нужно установить нужный флажок [18].

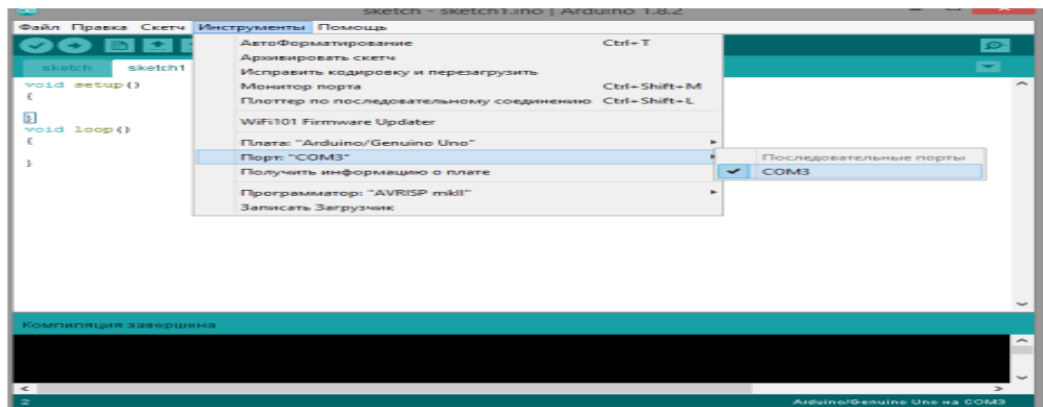


Рисунок 6 - Выбор нужного порта

Затем учащиеся выбирают тип контроллера Arduino.

Если контроллер выбран системой автоматически, необходимо проверить правильность этого выбора [20].

1.2.4 Память и её параметры на плате Arduino Uno

Плата Arduino Uno в стандартной комплектации имеет три типа памяти:

- Flash-память - Основная память. Сюда после компиляции записывается скетч. В Arduino Uno её 32 Кб.
- Оперативная память. При работе с платой, в этой памяти хранятся переменные данные, и созданные во время работы предметы. Как и в компьютере при выключении все данные хранящиеся в оперативной памяти пропадут.
 - Энергонезависимая память.
 - Энергонезависимая память объемом 1 Кб. В этой памяти хранятся данные даже если контроллер отключить от питания. Однако, чтобы пользоваться этой памятью, необходимо устанавливать дополнительные библиотеки, которые кстати в стоке стоят в среде программирования.

Следует помнить, что имеются модификации платы Arduino Uno, которые имеют большую память чем базовая модификация. Однако они тоже требуют для работы с такими платами необходимо скачивать библиотеки.

Программирование для платы Uno.

Начиная работать с платформой Ардуино помимо самой платы, нам также понадобится и среда программирования. Для обучения программированию и созданию скетчей (программ, написанных в данной среде программирования) подойдёт бесплатная среда Arduino IDE. Скачать её может абсолютно любой, перейдя на официальный сайт платформы [14].

Далее по списку идёт установка данной среды программирования на наш персональный компьютер. Ну тут всё просто, запускаем устанавливающий файл, указываем путь и идём пить чай. Пока мы пили чай программа уже установилась, и мы переходим на главный экран показанный (см. рисунок 7).

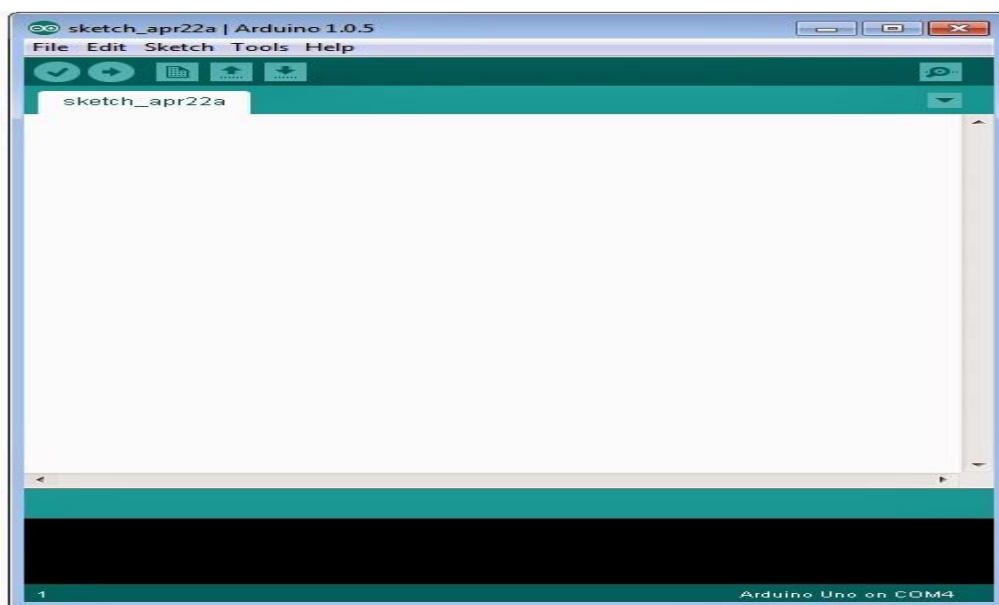


Рисунок 7 - Главный экран программной среды Arduino IDE

Дальше, нам необходимо понять на какой плате мы будем программировать. Узнать это просто: посмотрите на плату, и вы узнаете, как она называется. Далее необходимо выбрать нашу плату в среде разработки.

Чтобы это сделать кликните мышью в следующем порядке: инструменты – плата – выбираем из всего списка нужную нам плату. Теперь программная среда автоматически компилирует программный код, в такой вид, который поддерживает выбранная нами плата. Всё! Теперь любая написанная нами программа будет загружена в плату в одно касание по команде «загрузить» конечно если в ней нет ошибок, о чём при проверке нам сообщит программная среда.

Скетч – представляет собой цикл чаще всего бесконечный) в котором постоянно посылаются команды на пины, к которым подсоединены всякого рода датчики и благодаря всевозможным командам, которые вбиты в скетч формируется управление этими самыми датчиками. Данные скетчи могут находиться в библиотеках. Они либо встроены в среду, либо скачиваются в интернете. Написанный скетч закачивается в плату через USB предварительно скомпилированы, для того чтобы плата поняла что в неё загружено.

Автоматическая перезагрузка

При работе с платой Arduino Uno, старый скетч, который хранился в флэш-памяти автоматически стирается, поскольку плата сама перезагружается. Необязательно нажимать на кнопку перезагрузки. Это происходит из-за того, что одна из линий микроконтроллера, которая перезагружает плату, подключена к микросхеме управляющей потоком данных. Однако эту линию можно заблокировать, установив резистор между источником питания и данной линией.

Токовая защита разъема USB.

В плату установлен предохранитель, защищающий USB порт и саму плату в частности от короткого замыкания и сверхвысокого тока. Хотя сейчас компьютеры уже имеют такие предохранители, на плате он лишним не будет. Он просто даёт дополнительную защиту от всякого рода неожиданностей. Предохранитель настроен на ток больше чем 500 мА.

Плюсы платы Arduino Uno.

Первый и самый очевидный плюс: то, что плата сразу готова к работе. У вас нет необходимости собирать плату с нуля, разбираться в схемах, интерфейсе, заниматься программированием отдельных блоков. Нет необходимости часами сидеть и изучать инструкции для каждого из элементов. Мы просто берём, подключаем плату к компьютеру загружаем разработанный скетч и наслаждаемся с производением искусства созданного вашими собственными руками [20].

Лёгкость при работе, простота программирования, делают Arduino отличным средством обучения. Благодаря библиотекам, идущим в комплекте с платой. А если такой библиотеки нет, то её можно с лёгкостью скачать в сети интернет. Это значительно упрощает программирование скетча.

Размеры.

Некоторые считают, что размеры в семействе Ардуино – это недостатки. Но это не совсем правильно. В семействе Ардуино имеются и весьма маленькие платы (Mini; Nano). Однако для обучения программированию и конструированию это далеко не критичный фактор. Наоборот, обучающимся намного проще будет работать с довольно большими и увесистыми платами такими как Uno и Mega [27].

Все мы пользователи компьютеров. Но далеко не каждый знает, как он работает: мы просто запускаем нужную нам программу вы выполняем все необходимые нам действия. С Arduino – точно также самое: мы просто выполняем все необходимые нам действия (программирование, сборка схем) а плата всё уже делает сама. У нас нет необходимости забивать себе голову и изучать, как работает тот или иной блок платы. Всё время работы с платой пользователь концентрирует внимание на разработку проектов. Наличие готовых библиотек, а также модулей, включающих в себя датчики сразу же готовы к работе, что существенно облегчает жизнь не только профессиональным программистам, но и обычным любителям-экспериментаторам, которые просто хотят что-то сделать интересно, и собственными руками.

1.3 Методы обучения, используемые в процессе преподавания робототехники

Далее нам необходимо выбрать наилучшие методы обучения, с помощью которых учащиеся будут эффективно осваивать данный курс. Проанализировав множество методов обучения, мы выяснили, что для факультативного курса наилучшими будут следующие методы:

Метод проектов: Е. С. Полат видит в этом методе наилучший способ достижения дидактической цели. Проведя полную, детальную разработку проблемы, мы должны получить реальный, практический результат, который оформлен необходимым нам образом. И действительно в нашем факультативном курсе данный метод даст возможность учащимся самостоятельно работать над их проектами, тем самым развивать их творческие, познавательные и умственные способности. А это самые важные факторы при изучении такого курса как основы робототехники. Также самостоятельная работа способствует укреплению дисциплины, самоопределению как каждого учащегося, так и своей команды. Поскольку проект в робототехнике состоит из двух частей, то каждый учащийся сможет выбрать направление собственной деятельности. Одни программирует плату на компьютере, другие составляет и собирает схему установки [3].

Метод портфолио. Данный метод достаточно хорошо описан в книге «Медиаобразование: теоретические основы и опыт реализации. По мнению авторов, при создании своего портфолио каждый учащийся полностью анализирует свои достижения, представляют свои собственные возможности, и сами оценивают свои результаты [28].

Метод портфолио предполагает формирование структурированной папки, в которую помещают уже завершённые и специально оформленные работы. Они позволяют отразить образовательную биографию и уровень достижений ученика или группы учащихся. Этот метод помогает при формировании докладов на конференции школьников, при разработке

модели робота для выступления на соревнованиях различного уровня, при разработке плана на учебный период и т.д.

Метод взаимообучения своими истоками уходит в коллективный способ обучения. По мнению В. К. Дьяченко, обучение есть общение обучающихся и обучаемых. Вид общения определяет и организационную форму обучения. Исторический анализ показывает, что развитие способов обучения основывалось на применении различных видов общения. На занятиях элективного курса по конструированию и программированию роботов метод взаимообучения реализуется учениками самостоятельно, иногда даже без участия учителя. Разобравшись в решении какой-либо конструкторской задачи, учащиеся с удовольствием делятся своими знаниями с теми, кто испытывает затруднения при решении подобных задач. Таким образом, может сложиться ситуация, в которой учащиеся обучают самого учителя, что положительно влияет как на самооценку учеников, так и на отношения с учителем [8].

П. А. Юцявичене отмечает, что сущность метода модульного обучения состоит в том, что обучающийся самостоятельно может работать с предложенной ему индивидуальной программой, включающей в себя целевой план действий, банк информации и методическое руководство по достижению поставленных дидактических целей. В основе инвариантных программ, являющихся важным компонентом модульного обучения, лежат модули, представляющие собой профессионально значимые действия (учебные элементы). Достоинством модульной системы является гибкость, вариативность, возможность ее адаптации к изменяющимся условиям [30].

Целесообразно содержание факультативного курса по конструированию и программированию роботов разбить на следующие модули:

- 1) основы конструирования;
- 2) программирование;
- 3) решение прикладных задач.

Формирование структуры модулей может иметь циклический характер: тематика модулей повторяется через короткие (от недели до двух месяцев) или длинные (в пределах учебного года) промежутки времени. В темах конструирования и программирования одного временного периода удобно рассматривать задачи единых проектов, чтобы у учащихся сформировалось целостное представление о реализации той или иной модели робота.

Под проблемным обучением В. Оконь понимает совокупность таких действий, как организация проблемных ситуаций, формулирование проблем, оказание ученикам необходимой помощи в решении проблем, проверка правильности решений и руководство процессом систематизации и закрепления приобретенных знаний [16].

Метод проблемного обучения основан на создании проблемной мотивации и требует особого конструирования дидактического содержания материала, который должен быть представлен как цепь проблемных ситуаций. Этот метод позволяет активизировать самостоятельную деятельность учащихся, направленную на разрешение проблемной ситуации, в результате чего происходит творческое овладение знаниями, навыками, умениями и развитие мыслительных способностей. Практически каждую задачу, решаемую в процессе конструирования и программирования роботов, можно представить в качестве проблемной ситуации. Активизируя творческое и критическое мышление, учащиеся способны оптимизировать собственное решение задачи [9].

На практике в процессе реализации элективного курса по конструированию и программированию роботов наиболее продуктивным является применение совокупности нескольких методов обучения из вышеописанных [14].

1.4 Робототехника как средство формирования ключевых компетенций учащихся

Современная образовательная школа должна формировать целостную систему универсальных знаний, умений, навыков, а также опыт самостоятельной деятельности и личной ответственности обучающихся, то есть ключевые компетенции, определяющие современное качество содержания образования.

Для учителя это переход от передачи знаний к созданию условий для активного познания и получения детьми практического опыта. Для учащихся - переход от пассивного усвоения информации к активному ее поиску, критическому осмыслению, использованию на практике. Основной проблемой учителя является поиск средств и методов развития образовательных компетенций учащихся, как условие, обеспечивающее качественное усвоение программы.

Главными целевыми установками для учителя сегодня являются компетенции как результат образования, как интегрирующие начала «модели» выпускника школы.

В настоящее время возросла роль некоторых качеств личности, ранее необязательных для жизни в обществе, таких как: способность быстро ориентироваться в меняющемся мире, осваивать новые профессии и области знаний, умение находить общий язык с людьми самых разных профессий, культур и др. Эти качества получили название ключевых компетенций.

Под ключевыми компетентностями применительно к школьному образованию понимается способность учащихся самостоятельно действовать в ситуации.

Компетенция - это:

- 1) круг полномочий и прав, предоставляемых законом, уставом или договором конкретному лицу или организации в решении соответствующих вопросов;

2) совокупность определенных знаний, умений и навыков, в которых человек должен быть осведомлен и иметь практический опыт работы.

Компетентность - это умение активно использовать полученные личные и профессиональные знания, умения и навыки в практической деятельности.

Компетентностный подход выдвигает на первое место не информированность ученика, а способность организовывать свою работу. Запомнить и ответить - это накопление знаний; а применить свои знания и умения во вне учебной практической ситуации - это компетентность.

Смысл компетентностного подхода в том, что ученик должен осознавать постановку самой задачи, оценивать новый опыт, контролировать эффективность собственных действий. При таком подходе учебная деятельность периодически приобретает исследовательский или практико-преобразовательный характер [11].

Современные образовательные технологии обеспечивают включение в образовательный процесс специально организованной деятельности учащихся. Этот механизм компетентностного подхода хорошо моделируется внедрением курса робототехники в образовательный процесс.

Робототехника - прикладная наука, занимающаяся разработкой автоматизированных технических систем. Активная вовлеченность детей в конструирование физических объектов, способствует развитию понятийного и речевого аппарата, что в свою очередь, при правильной поддержке со стороны учителя, помогает детям лучше вникать в суть вещей и продолжать развиваться [9].

Робототехнику можно широко использовать при организации как учебного процесса, так и внеурочной деятельности. Образовательную робототехнику можно также применять на уроках информатики, биологии, физики, технологии и других предметах как ограниченно (демонстрации, наблюдения), так и при изучении отдельных тем по предмету.

При работе с платами по робототехнике используются межпредметные связи: физика и математика, информатика.

Меж предметные связи есть педагогическая категория для обозначения синтезирующих, интегративных отношений между объектами, явлениями и процессами реальной действительности, нашедших свое отражение в содержании, формах и методах учебно-воспитательного процесса и выполняющих образовательную, развивающую и воспитывающую функции.

Конструирование повышает мотивацию обучающихся к овладению новыми знаниями. Необходимо привлекать понятия из других предметов для расширения области практического применения теории, изучаемой в данном предмете. Использовать практические умения и навыки, полученные на уроках родственных предметов, для получения новых экспериментальных данных. У обучающихся появляется возможность повторять необходимые сведения по соответствующим предметам. При изучении нового учебного материала используются факты и понятия из разных учебных предметов. Обучающиеся самостоятельно воспроизводят отдельные знания фактического или теоретического характера из смежной дисциплины и привлекают факты и понятия, усвоенные ими на уроках одного предмета, для подтверждения вновь усваиваемых знаний на уроках другого. Самостоятельно привлекают теорию для объяснения изучаемых явлений на уроках другого учебного предмета. Все это позволяет повысить уровень сформированное ключевых компетенций по следующим параметрам:

Информационная компетенция: поиск информации по роботам-андроидам в сети Интернет, изучение найденных образцов моделей и анализ их конструкций.

Коммуникативная компетенция: подготовка сообщения, по теме возможной реализации найденных конструкций, внедрения новых элементов, подготовка сообщений отдельных учеников или групп учеников, коллективное обсуждение общего порядка работы при реализации проекта, групповая проектная работа, оценка деятельности каждого ученика.

Учебно-познавательная компетенция: создание модели андроида по известным схемам; программирование действий робота по образцу; исследовательская работа по моделированию конструкции; исследовательская работа по корректированию программ; оформление и защита работы, самостоятельное построение конструкции робота без схем и инструкций; программирование действий робота в зависимости от поставленной цели; демонстрация готовых моделей; проведение состязания между роботами и определение победителей; выявление удачных решений и недостатков конструкций.

Робототехника, представляя собой межпредметный курс, позволяет повысить уровень сформированное у обучающихся ключевых компетенций. Кроме того, работа с компьютерами, сборка роботов, проведение экспериментов по исследованию окружающей среды способствуют достижению результатов освоения образовательной программы общего образования, указанных в федеральных государственных образовательных стандартах, как владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем. Такая деятельность способствует достижению значительных результатов по учебным предметам [6].

2 ПРОЕКТНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА ПО РОБОТОТЕХНИКЕ

2.1 Методы и формы организации проектной деятельности по робототехнике

В связи с быстрым развитием технологий в наше время, обществу нужны люди, которые будут находить, исследовать, и решать проблемы, а не только исполнять указания. Таким образом школе необходимо дать учащимся не только знания, но и умения. Умения самостоятельно находить, всесторонне исследовать, и решать проблемы, поставленные ими самими, либо педагогом.

Для того чтобы решить этот вопрос, нам необходимо вводить курсы, созданные специально для решения этих задач. Но для того чтобы ввести курс в образовательное пространство школы необходимо было решить, какую оптимальную форму организации учебного процесса выбрать.

Начнём с того что курс «Основы робототехники» построен как факультативный, следовательно, эффективность обучения зависит и от организации занятий, проводимых с применением следующих методов по способу получения знаний, предложенных В.П. Беспалько:

1) Объяснительно-иллюстративный - предъявление информации различными способами (объяснение, рассказ, беседа, инструктаж, демонстрация, работа с технологическими картами и др);

2) Проектно-ориентированное обучение - это систематический учебный метод, вовлекающий учащихся в процесс приобретения знаний и умений с помощью широкой исследовательской деятельности, базирующейся на комплексных, реальных вопросах и тщательно проработанных заданиях.

3) Эвристический - метод творческой деятельности (создание творческих моделей и т.д.)

4) Проблемный - постановка проблемы и самостоятельный поиск её решения обучающимися;

5) Программированный - набор операций, которые необходимо выполнить в ходе выполнения практических работ (форма: компьютерный практикум, проектная деятельность);

6) Репродуктивный - воспроизводство знаний и способов деятельности (форма: собирание моделей и конструкций по образцу, беседа, упражнения по аналогу);

7) Поисковый - самостоятельное решение проблем;

8) Частично-поисковый - решение проблемных задач с помощью педагога;

9) Метод проблемного изложения - постановка проблемы педагогам, решение ее самим педагогом, соучастие обучающихся при решении [2].

Как мы видим, для более качественного обучения, метод проектов подходит как нельзя лучше. Под методом проектов понимают технологию организации образовательных ситуаций, в которых учащиеся ставят и решает собственные задачи, и технологию сопровождения самостоятельной деятельности учащегося.

Основные этапы разработки проекта Arduino:

1) Обозначение темы проекта.

2) Постановка целей и задач проекта.

3) Разработка и построения монтажной схемы на Arduino.

4) Составление программного скетча для необходимого нам результата.

5) Компиляция скетча в плату, проверка правильности работы. В противном случае поиск и устранение неполадок.

При таком подходе учащиеся учатся творчески подходить к постановке целей, ставить и решать задачи, сообща решать возможные проблемы, возникшие при решении конкретной задачи (будь то проблемы в составлении

схемы или программировании скетча). Данный подход благотворно влияет на развитие творческих способностей, умственное развитие. Но самое главное – учащиеся вместе решают поставленные ими задачи и решают возникшие проблемы, а это не что иное как работа в команде.

Таким образом мы видим, что данный курс, основанный на Arduino положительно сказывается на всём учебном процессе. С помощью данного факультативного курса у учащихся повышается интерес к изучению дисциплин, непосредственно связанных с изучением курса, таких как физика информатика, иностранный язык (в нашем случае английский).

2.2 Факультативный курс «основы робототехники»

Пояснительная записка

Актуальность факультативного курса заключается в том, что она направлена на формирование творческой личности, умеющей креативно, нестандартно мыслить. Технологические наборы Arduino UNO ориентированы на изучение основных физических принципов и базовых технических решений, лежащих в основе всех современных конструкций и устройств.

Данная программа предполагает решение инженерных и конструкторских задач, а также обучение объектно-ориентированному программированию и моделированию с использованием Arduino UNO. Использование Arduino UNO позволяют решать не только типовые задачи, но и нестандартные ситуации, исследовать датчики, вести собственные наблюдения. Кроме того, работа в команде способствует формированию умения взаимодействовать с соучениками, формулировать, анализировать, критически оценивать, отстаивать свои идеи. При дальнейшем освоении Arduino UNO становится возможным выполнение серьезных проектов,

развитие самостоятельного технического творчества, участие в соревнованиях по робототехнике.

Цель курса - научить использовать средства информационных технологий для решения конструкторских и межпредметных задач, способствовать успешной адаптации при переходе от пропедевтического курса информатики к базовому.

Задачи:

- Знакомство со средой программирования Arduino UNO, базовым и ресурсными наборами;
- Усвоение основ объектно-ориентированного программирования;
- Составление простых и сложных алгоритмов;
- Использование и программирование датчиков для исследования окружающей среды и выполнения поставленных задач;
- Проектирование и разработка собственных программ для решения стандартных и нестандартных задач;
- Создание собственных проектов, которые могут быть полезными в реальной жизни;
- Формирование умения работать в группе.

Планируемые результаты освоения программы

Личностные результаты:

- 1) Формирование способностей, обучающихся к саморазвитию, самообразованию и самоконтролю на основе мотивации к робототехнической и учебной деятельности;
- 2) Формирование современного мировоззрения, соответствующего современному развитию общества и науки;
- 3) Формирование коммуникативной и ИКТ-компетентности для успешной социализации, и самореализации в обществе.

Метапредметные результаты:

- а) Умение ставить и реализовывать поставленные цели;
- б) Умение самостоятельно планировать свою деятельность;

в) Умение выполнять и правильно оценивать результаты собственной деятельности;

г) Умение создавать, разрабатывать и реализовывать схемы, планы и модели для решения поставленных задач;

д) Умение устанавливать причинно-следственные связи и логически мыслить.

Предметные результаты:

1) Овладение простыми методами и формами обработки и анализа данных;

2) Формирование ИКТ- компетентности и информационной культуры;

3) Формирование умения автоматизировать и решать поставленные задачи, используя компьютер и технические устройства как инструмент.

Рабочая программа факультативного курса «Основы робототехники» (см. таблицу 4).

Таблица 4 рабочая программа факультативного курса.

п/п	Наименование темы	Количество часов		
		Всего	Теория	Практика
1	Вводное занятие. Правила техники безопасности	1	1	-
2	Лабораторная работа № 1 «Введение в Arduino»	1	1	-
3	Лабораторная работа № 2 «Знакомство с Arduino Uno. Управление светодиодом»	2	1	1
4	Лабораторная работа № 3 «Программирование в Arduino»	2	1	1
5	Лабораторная работа № 4 «Подключение RGB-светодиода»	2	1	1
6	Лабораторная работа № 5 «Модель свеча»	2	1	1
7	Лабораторная работа № 6 «Работа с жидкокристаллическим экраном»	3	1	2
8	Лабораторная работа № 7 «Передача данных в инфракрасном и ультразвуковом диапазонах»	3	1	2
9	Лабораторная работа № 8 «Работа сервопривода»	3	1	2

Продолжение таблицы 4

10	Лабораторная работа № 9 «Создаем робота»	3	1	2
11	Разработка и представление итогового проекта учащимися.	3	-	3
Итого		25	10	15

Ниже представлены первые три лабораторные работы, входящие в курс «Основы робототехники»

Лабораторная работа №1

Введение в Arduino.

Цель: Ознакомиться с микроконтроллерами Arduino, а также со средой программирования Arduino IDE

Краткая теория.

Появление первых микроконтроллеров ознаменовало начало новой эры в развитии микропроцессорной техники. Наличие в одном корпусе большинства системных устройств сделало микроконтроллер подобным обычному компьютеру. В отечественной литературе они даже назывались однокристальными микро-ЭВМ. Соответственно и желание использовать микроконтроллеры как обычные компьютеры появилось практически с их появлением. Но желание это сдерживалось многими факторами. Например, чтобы собрать устройство на микроконтроллере, необходимо знать основы схемотехники, устройство и работу конкретного процессора, уметь программировать на ассемблере и изготавливать электронную технику.

Потребуется также программаторы, отладчики и другие вспомогательные устройства. В итоге без огромного объема знаний и дорогостоящего оборудования не обойтись. Такая ситуация долго не позволяла многим любителям использовать микроконтроллеры в своих проектах. Сейчас, с появлением устройств, дающих возможность работать с микроконтроллерами без наличия серьезной материальной базы и знания многих предметов, все изменилось.

Примером такого устройства может служить проект Arduino итальянских разработчиков.

Обзор контроллеров семейства Arduino.

Основные версии плат Arduino представлены следующими моделями:

Due - плата на базе 32-битного ARM микропроцессора Cortex-M3 ARM SAM3U4E;

Leonardo - плата на микроконтроллере ATmega32U4;

Uno - самая популярная версия базовой платформы Arduino;

Duemilanove - плата на микроконтроллере ATmega168 или ATmega328;

Diecimila - версия базовой платформы Arduino USB;

Nano - компактная платформа, используемая как макет. Nano подключается к компьютеру при помощи кабеля USB Mini-B;

Mega ADK - версия платы Mega 2560 с поддержкой интерфейса USB-host для связи с телефонами на Android и другими устройствами с интерфейсом USB;

Mega2560 - плата на базе микроконтроллера ATmega2560 с использованием чипа ATmega8U2 для последовательного соединения по USB-порту;

Mega - версия серии Mega на базе микроконтроллера ATmega1280;

Arduino BT - платформа с модулем Bluetooth для беспроводной связи и программирования;

LilyPad - платформа, разработанная для переноски, может зашиваться в ткань;

Fio - платформа разработана для беспроводных применений. Fio содержит разъем для радио XBee, разъем для батареи LiPo и встроенную схему подзарядки;

Mini - самая маленькая платформа Arduino;

Pro - платформа, разработанная для опытных пользователей, может являться частью большего проекта:

Pro Mini - как и платформа Pro, разработана для опытных пользователей, которым требуется низкая цена, меньшие размеры и дополнительная функциональность.

Платы расширения Arduino.

Большую популярность плата Arduino приобрела не только из-за низкой стоимости, легкости разработки и программирования, но, главным образом, благодаря наличию плат расширения (так называемых шилдов), добавляющих

Arduino дополнительную функциональность. Шилды (кроме маленьких модулей и платы Lily Pad) подключаются к Arduino с помощью имеющихся на них штыревых разъемов.

Существует множество различных по функциональности шилдов, от простейших, предназначенных для макетирования, до сложных, представляющих собой отдельные многофункциональные устройства.

Среда программирования Arduino IDE.

Разработка собственных приложений на базе плат, совместимых с архитектурой Arduino, осуществляется в официальной бесплатной среде программирования Arduino IDE. Среда предназначена для написания, компиляции и загрузки собственных программ в память микроконтроллера, установленного на плате Arduino-совместимого устройства. Основой среды разработки является язык

Processing/Wiring - это фактически обычный C++, дополненный простыми и понятными функциями для управления вводом/выводом на контактах.

Существуют версии среды для операционных систем Windows, Mac OS и Linux.

Настройка среды Arduino IDE.

Среда разработки Arduino состоит из следующих частей: редактора программного кода; области сообщений; окна вывода текста; панели инструментов с кнопками часто используемых команд; нескольких меню.

Программа, написанная в среде Arduino, носит название скетч. Скетч пишется в текстовом редакторе, который имеет цветовую подсветку создаваемого программного кода. Во время сохранения и экспорта проекта в области сообщений появляются пояснения и информация об ошибках. Окно вывода текста показывает сообщения Arduino, включающие полные отчеты об ошибках и другую информацию. Кнопки панели инструментов позволяют проверить и записать программу, создать, открыть и сохранить скетч, открыть мониторинг последовательной шины.

Разрабатываемым скетчам дополнительная функциональность может быть добавлена с помощью библиотек, представляющих собой специальным образом оформленный программный код, реализующий некоторый функционал, который можно подключить к создаваемому проекту. Специализированных библиотек существует множество. Обычно библиотеки пишутся так, чтобы упростить решение той или иной задачи и скрыть от разработчика детали программно- аппаратной реализации. Среда Arduino IDE поставляется с набором стандартных библиотек: Serial, EEPROM, SPI, Wire и др. Выбор библиотеки в меню приведет к добавлению в исходный код следующей строки строки:

```
#include <имя библиотеки.h>
```

Эта директива подключает заголовочный файл с описанием объектов, функций и констант библиотеки, которые теперь могут быть использованы в проекте. Среда Arduino будет компилировать создаваемый проект вместе с указанной библиотекой. Перед загрузкой скетча требуется задать необходимые параметры в меню Сервис | Плата (Tools Board) - и Сервис | Последовательный порт (Tools | Serial Port). Современные платформы Arduino перезагружаются автоматически перед загрузкой. На старых платформах необходимо нажать кнопку перезагрузки. На большинстве плат во время процесса загрузки будут мигать светодиоды RX и TX. При загрузке скетча используется загрузчик (bootloader) Arduino – небольшая программа, загружаемая в микроконтроллер на плате. Она позволяет загружать

программный код без использования дополнительных аппаратных средств. Работа загрузчика распознается по миганию светодиода на цифровом выводе D13. Монитор последовательного порта (Serial Monitor) отображает данные, посылаемые в платформу Arduino (плату USB или плату последовательной шины). Для отправки данных необходимо ввести в соответствующее поле текст и нажать кнопку Послать (Send) или клавишу <Enter>. Затем следует из выпадающего списка выбрать скорость передачи, соответствующее значению `serial.baudin` в скетче.

Таким образом мы разобрали: что такое Arduino; какие бывают платы; как выглядит и как пользоваться средой программирования.

Лабораторная работа № 2

Знакомство с Arduino Uno. Управление светодиодом

Цели работы:

Цель 1: Познакомиться с платформой Arduino Uno

- 1) изучить состав платы, выучить назначение отдельных ее элементов
- 2) изучить способ (способы) подключения платформы к компьютеру
- 3) освоить программный интерфейс компьютер - Arduino, научиться загружать программный код (скетчи) на платформу Arduino

Цель 2: Познакомиться со структурой и основными командами программного кода (скетча) для Arduino на примере программы «Мигающий светодиод»

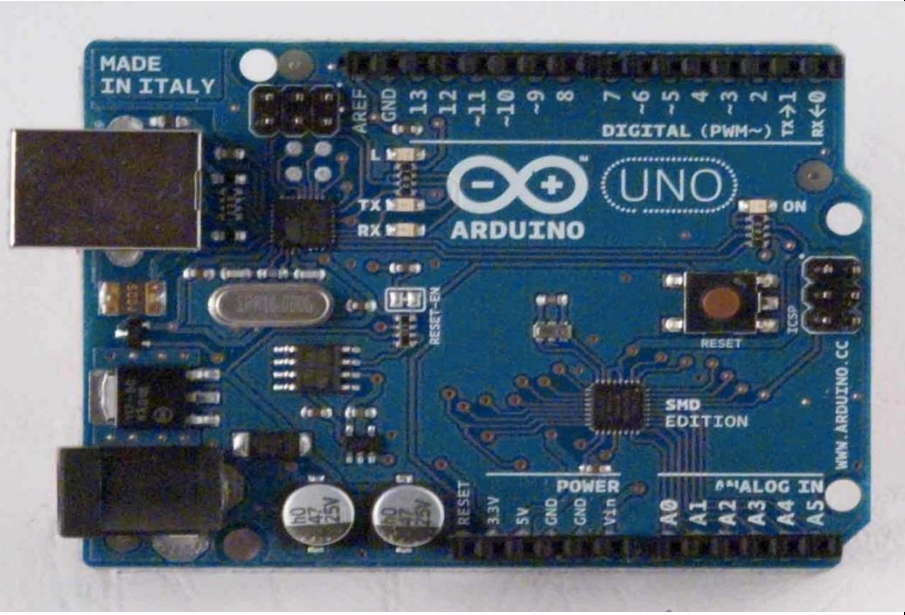
- 4) выучить назначение и синтаксис команд `pinMode()`, `digitalWrite()`, `delay()`
- 5) выучить назначение и правила оформления функций `setup()` и `loop()`
- 6) выучить правила объявления переменных в скетче Arduino
- 7) выучить правила оформления комментария

1 шаг. Начнём с беседы: вот вопросы которые мы должны обсудить (в парах каждый задает 7 вопросов из 14 и отвечает на 7 вопросов):

1. Где на платформе находится процессор?
2. Какой объем памяти микроконтроллера?
3. Где находится гнездо для подключения USB-кабеля? Для чего применяется соединение микроконтроллера с компьютером?
4. Где находится гнездо для подключения внешнего питания? Для чего применяется внешнее питание?
5. Сколько цифровых контактов (входов / выходов) есть на платформе? Где они расположены?
6. Почему некоторые цифровые контакты отмечены знаком ~ (тильда)?
7. Объясните, что такое “широотно-импульсная модуляция сигналов”
8. Сколько контактов аналогового входа есть на платформе? Где они находятся? Объясните, что такое аналоговый сигнал. Для чего используются контакты аналогового входа?
9. Где находятся контакты для доступа к питанию? Какое напряжение использует Arduino?
10. Где находятся контакты “земля”? Сколько таких контактов размещено на платформе?
11. Где находится кнопка сброса? Для чего она служит?
12. Где находится встроенный светодиод? Как он обозначен? К какому цифровому выходу он подключён?
13. Где находятся светодиоды, которые могут служить индикаторами загрузки программы? Как они обозначены?
14. Какой язык используется для программирования платформы?

2 шаг. Объединитесь в группу и отметьте цифрами на рисунке составляющие платформы (см. таблицу 5):

Таблица 5 Задание к второй лабораторной

<ol style="list-style-type: none"> 1. процессор 2. гнездо для подключения USB-кабеля 3. гнездо для подключения внешнего питания 4. цифровые контакты (входы / выходы) 5. контакты аналогового входа 6. контакты для доступа к питанию 7. контакты “земля” 8. кнопка сброса 9. встроенный светодиод 10. светодиоды-индикаторы загрузки программы 	
---	--

3 шаг: Первая программа для микроконтроллера “Мигающий светодиод”. Программа для встроенного светодиода.

Задача: написать программу, при выполнении которой светодиод 5 секунд горит, 1 секунду не горит.

Пошаговая инструкция.

1. Запустите среду программирования.
2. В окне редактора введите программу и прокомментируйте каждую строку написанного ниже кода. Запомните программу на своем диске с именем arduino1

*/**

Опишите здесь, что выполняет ваша программа

**/*

`int ledPin = 13; // ваши комментарии`

`void setup () { //`

`pinMode (ledPin, OUTPUT); //`

```

        }
void loop () {
digitalWrite (ledPin, HIGH); //
delay (5000);
digitalWrite (ledPin, LOW); //
delay (1000);
}

```

3. Подключите Arduino к компьютеру. Проверьте, что задействован нужный COM порт (Сервис - Последовательный порт). Светящаяся красным светом лампочка означает, что платформа находится под напряжением. Возможно, на платформе работает встроенная программа, которая заставляет встроенный светодиод ритмично мигать. Мы изменим частоту мигания.

4. Загрузите код на платформу. Добейтесь, чтобы схема работала так, как было задумано.

5. Измените частоту мигания. Убедитесь, что вы умеете управлять этими временными параметрами.

6. Предложить учащимся поэкспериментировать и изменяя параметры скетча изменить работу светодиода.

Таким образом мы выяснили расположение элементов платы и распиновку платы Arduino Uno, и выполнили простейшее упражнение с использованием среды разработки Arduino IDE.

Лабораторная работа №3

Тема: Программирование в Arduino.

2. Цель: Изучить основные правила программирования Arduino IDE, а также ознакомиться с функциями программирования

Краткая теория.

Цифровые выводы.

Выводы платформы Arduino могут работать как входы или как выходы. Также аналоговые входы Arduino (ATmega) могут конфигурироваться и работать так же. Как и цифровые порты ввода/вывода. Выводы Arduino настроены как порты ввода, поэтому не требуется декларации в функции `pinMode ()`. Сконфигурированные порты ввода находятся в высокоимпедансном состоянии. Это означает, что порт ввода дает слишком малую нагрузку на схему, в которую он включен. Для перевода порта ввода из одного состояния в другое требуется маленькое значение тока. Если к выводу ничего не подключено, то значения на нем будут принимать случайные величины, наводимые электрическими помехами. Если на порт ввода не поступает сигнал, то рекомендуется задать порту известное состояние. Это делается добавлением подтягивающих резисторов 10 кОм, подключающих вход либо к питанию +5 В. Либо к земле. Микроконтроллер ATmega имеет программируемые встроенные подтягивающие резисторы 20 кОм. Программирование данных резисторов осуществляется так:

```
pinMode (pin, INPUT); // назначить выводу порт ввода
digitalWrite (pin, HIGH); // включить подтягивающий резистор
```

Выводы, сконфигурированные как порты вывода находятся в низкоимпедансном состоянии. Данные выводы могут пропускать через себя достаточно большой ток. Выводы микросхемы ATmega могут быть источником тока до 40 мА. Такого значения тока недостаточно для большинства реле, соленоидов и двигателей. Короткие замыкания выводов Arduino или попытки подключить энергоемкие устройства могут повредить выходные транзисторы вывода или весь микроконтроллер ATmega.

Аналоговые входы.

Микроконтроллеры ATmega, используемые в Arduino. Содержат шести канальный аналого-цифровой преобразователь (АЦП). Разрешение преобразователя составляет 10 битов, что позволяет на выходе получать значения от 0 до 1023. Аналоговые входы могут использоваться как

цифровые выводы портов ввода/вывода, при этом они имеют номера от 14 до 19:

`pinMode (14, OUTPUT);`

`digitalWrite (14, HIGH);`

Для вывода, работавшего ранее как цифровой порт вывода, команда `analogRead` будет работать некорректно. В этом случае рекомендуется сконфигурировать его как аналоговый вход.

Структура программы.

Arduino программируется на языке `Wiring`, которого на самом деле не существует, как не существует и компилятора `Wiring` написанные на `Wiring` программы преобразуются в программу на языке `C/C++` и затем компилируются компилятором `AVR-GCC`. Фактически используется специализированный для микроконтроллеров `AVR` вариант `C/C++`.

Функции `setup()` и `loop()`

Базовая структура программы для `Arduino` состоит, по меньшей мере, из двух обязательных частей: функций `setup()` и `loop()`. Перед функцией `setup()` идет объявление переменных, подключение библиотек. Функция `setup()` запускается один раз после каждого включения питания или сброса платы `Arduino`. Она используется для инициализации переменных, установки режима работы портов и прочих подготовительных для основного цикла программы действий. Она обязательно должна быть включена в программу, даже если не выполняет никаких действий.

Функция `loop()` в бесконечном цикле последовательно исполняет команды, которые описаны в ее теле. Эта функция выполняется циклически, она выполняет основную работу.

Оператор `if` (условие) и операторы сравнения `==, !=, <, >`.

Оператор `if` используется в сочетании с операторами сравнения, он проверяет, достигнута ли истинность условия например, превышает ли входное значение заданное число. Формат оператора `if` следующий:

```
If (someVariable > 50) {
```

```
// выполнять действия
```

```
}
```

Программа проверяет, значение `someVariable` больше чем 50 или нет. Если да, то выполняются определенные действия. Говоря иначе, если выражение в круглых скобках истинно, выполняются операторы внутри фигурных скобок. Если нет, программа пропускает этот код.

Выражения, которые вычисляются внутри круглых скобок, могут состоять из одного или нескольких операторов.

Операторы сравнения:

```
x == y (x равно y);
```

```
x != y (x не равно y);
```

```
x < y (x меньше чем y);
```

```
x > y (x больше чем y);
```

```
x <= y (x меньше чем или равно y);
```

```
x >= y (x больше чем или равно y).
```

```
; (точка с запятой) semicolon.
```

```
; (точка с запятой) используется для обозначения конца оператора.
```

```
int a = 13;
```

```
{ } (фигурные скобки) { } (curly braces)
```

Фигурные скобки `{ }` - важный элемент языка программирования C.

Открывающая скобка `{` должна всегда сопровождаться закрывающей скобкой `}`. Это условие, известное как парность (симметричность) фигурных скобок.

Функции:

```
void Название функции (тип данных аргумента) {оператор(ы)};
```

Циклы:

```
while (логическое выражение) {оператор(ы)};
```

```
do {оператор(ы)} while (логическое выражение);
```

```
for (инициализация; условие окончания цикла; приращения цикла) {  
оператор(ы)};
```

Условные операторы:

If (логическое выражение) {оператор(ы)}.

Комментарии //(single line comment), /**/ (multi-line comment).

Комментарии - это строки в программе, которые используются для информирования вас самих или других о том, как работает программа. Они игнорируются компилятором и не занимают место в памяти микроконтроллера.

Есть два способа пометить строку как комментарий:

Однострочный комментарий - // ;

Многострочный комментарий - /* ... */ .

= (assignment) = оператор присваивания.

Присваивает переменной слева от оператора значение переменной или выражения, Переменная слева от оператора присваивания (=) должна быть способна сохранить присваиваемое значение. Если оно выходит за диапазон допустимых значений, то сохраненное значение будет не верно. Необходимо различать оператор присваивания (=) и оператор сравнения (== двойной знак равенства), который осуществляет проверку на равенство.

Функция pinMode.

Устанавливает режим работы заданного входа/выхода (pin) как входа или как выхода.

Синтаксис:

```
pinMode(pin, mode);
```

Параметры:

pin - номер входа/выхода (pin), который вы хотите установить;

mode - режим. Одно из двух значений: input или output устанавливает на вход или выход соответственно.

Пример:

```
Int ledPin =13; // Светодиод, подключенный к входу/выходу 13
```

```
void setup( )
```

```
{
```

```
pinMode(ledpin, OUTPUT); // устанавливает режим работы - выход  
}
```

Функция `analogRead()`.

Функция считывает значение с указанного аналогового входа. Большинство плат Arduino имеют 6 каналов (8 каналов у платы Mini и Nano, 16 - у Mega) с 10- битным аналого-цифровым преобразователем (АЦП). Напряжение, поданное на аналоговый вход (обычно от 0 до 5 вольт), будет преобразовано в значение от 0 до 1023 - это 1024 шага с разрешением 0,0049 вольт. Разброс напряжения и шаг может быть изменен функцией `analogReference()`. Считывание значения с аналогового входа занимает примерно 100 микросекунд (0,0001 сек), т.е. максимальная частота считывания приблизительно 10000 раз в секунду.

Синтаксис:

```
analogRead(pin);
```

Параметр: `pin` - номер порта аналогового входа, с которого будет производиться считывание: 0..5 для большинства плат, 0..7 для Mini и Nano и 0..15 для Mega..

Функция `millis ()`.

Возвращает количество миллисекунд с момента начала выполнения текущей программы на плате Arduino. Это количество сбрасывается на ноль вследствие переполнения значения приблизительно через 50 дней.

Параметров нет.

Возвращаемое значение - количество миллисекунд с момента начала выполнения программы (тип `unsigned long`).

Функция `min(x, ux)`.

Возвращает наименьшее из двух значений.

Параметры:

`x` - первое число, любой тип;

`y` - второе число, любой тип.

Возвращаемое значение возвращает меньшее из двух сравниваемых значений.

Пример использования функции:

```
senseVal = min(senseVal, 100);
```

```
// проверяем, если senseVal больше 100, то senseVal будет присвоено 100
```

Функция max(x, y).

Возвращает большее из двух значений.

Параметры:

x - первое число, любой тип;

y - второе число, любой тип.

Возвращаемое значение - возвращает большее из двух сравниваемых значений. Пример использования функции:

```
senseVal = max{senseVal, 20};
```

```
// проверяем, если senseVal меньше 20, то senseVal будет присвоено 20.
```

Функция max () зачастую используется для ограничения нижней границы значений переменной. Функцией min () ограничивают верхнюю границу переменной. В силу специфики реализации функции max () следует избегать использования других функций в качестве параметров. Например:

```
max (a--, 0); // может привести к некорректным результатам
```

```
a--;
```

```
max(a, 0); // так корректно
```

Функция sin(rad).

Возвращает синус угла, заданного в радианах в передаваемом параметре.

Результат функции всегда в диапазоне -1 ... 1.

Параметр: rad - угол в радианах (тип float).

Возвращаемое значение: синус угла (тип double).

Функция cos(rad).

Возвращает косинус угла, заданного в радианах в передаваемом параметре.

Результат функции всегда находится в диапазоне -1 ... 1.

Параметр: rad - угол в радианах (тип float).

Возвращаемое значение: косинус угла (тип double).

Функция tan(rad).

Возвращает тангенс угла, заданного в радианах в передаваемом параметре. Результат функции в диапазоне от минус бесконечности до плюс бесконечности.

Параметр: rad - угол в радианах (тип float).

Возвращаемое значение: тангенс угла (тип double).

Функция randomSeed(seed).

Функция randomSeed () инициализирует генератор псевдослучайных чисел. Генерируемая последовательность случайных чисел очень длинная, и всегда одна и та же. Точка в этой последовательности, с которой начинается генерация чисел, зависит от параметра seed. Параметр: seed - параметр, задающий начало выдачи псевдослучайных значений на последовательности (тип int, long).

Синтаксис:

```
random(max);
```

```
random(min, max);
```

Параметры:

min - нижняя граница случайных значений, включительно (опционально);

max - верхняя граница случайных значений, включительно.

Возвращаемое значение: случайное число между min и max - 1 (тип long).

Если при каждом запуске программы необходимо получать разные последовательности значений, генерируемых функцией random (), то необходимо инициализировать генератор псевдослучайных чисел со случайным параметром.

Например, можно использовать значение, отдаваемое функцией `analogRead()` с неподключенного порта входа/выхода. В некоторых случаях необходимо получать одинаковую последовательность при каждом запуске программы на Arduino. Тогда инициализировать генератор псевдослучайных чисел следует вызовом функции `randomSeed()` с фиксированным параметром.

Практические занятия.

Цель: Создать модель, имитирующую игру пламени свечи.

Для имитации пламени свечи мы будем использовать три ярких светодиода. Яркость свечения будет случайной и случайным будет момент изменения яркости. Случайное значение яркости будет меняться в некотором числовом диапазоне, то есть может принимать не только значение 0 или 1.

Поэтому для подключения светодиодов нужно использовать ШИМ-пины. (ШИМ – широтно- импульсная модуляция). Вспомним, что на плате Ардуино несколько контактов (пинов) могут использоваться как аналоговые выходы или ШИМ-пины, напряжение на которых может меняться в диапазоне от 0 до 255. Эти пины на плате обозначены знаком ~. На нашей плате это пины с номерами 3, 5, 6, 9, 10, 11. В случае, когда пин используется в режиме ШИМ, нужно вместо функции `digitalWrite(pin, HIGH)` или `digitalWrite(pin, LOW)` использовать функцию `analogWrite(pin, value)` где значение `value` может принимать значения от 0 до 255.

Микроконтроллер переводит число от 0 до 255 к напряжению от 0 до 5 В., Например, 85 — это $1 / 3$ от 255, т.е. $1 / 3$ от 5 В или 1,66 В. А для напряжения 2.5 В нужно задать значение `value = 127`.

Случайное число, как и в программе на языке Pascal, можно задать функцией `random(n)`. При этом случайным образом генерируется число в интервале от 0 до $n-1$ включительно.

Листинг скетча

```
// Объявляем переменные — три пина, к которым подключены  
светодиоды
```

```

int led1 = 9;
int led2 = 10;
int led3 = 11;
void setup() // Инициализация пинов
{
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}
void loop() // Бесконечный цикл выполнения программы
{
  analogWrite(led1, 56+random(200)); // Случайное значение яркости от 56
до 255
  analogWrite(led2, 56+random(200));
  analogWrite(led3, 56+random(200));
  delay(random(100)); // Случайное время задержки от 0 до 99
}

```

Возможное развитие проекта

Создать модель, имитирующую северное сияние.

Создать модель, имитирующую «мигалку» скорой помощи

Что нового?

Использование ШИМ-пинов

При использовании пинов в режиме ШИМ следует применять функцию `analogWrite()` вместо `digitalWrite()`. В качестве параметров указываются имя пина и значение на выходе, которое может меняться в интервале от 0 до 255

Использование генератора случайных чисел `random` для получения случайных значений яркости свечения и времени задержки `random (200)` генерирует случайное число от 0 до 199; `56 + random (200)` — случайное число от 56 до 255

Дальнейшие лабораторные работы будут находиться в разделе
Приложение

2.3 Анализ педагогического эксперимента

База проведения апробирования.

Некоторая часть данного факультативного курса была опробована на 6 учащихся 10 класса в «Гостищевской СОШ» Яковлевского района село Гостищево. Список детей указан ниже.

Ушакова Татьяна

Захаренко Наталья

Забаровский Вячеслав

Коровяковский Дмитрий

Рябчун Ирина

Клёнов Игорь

Пояснительная записка.

На сегодняшний день, когда автоматизация всех отраслей жизни человека лежит на плечах компьютеров и роботов, неудивительно что тенденция в Российских школах проводить всевозможные кружки, курсы и отдельные уроки по робототехнике приобретают всё большую популярность. Ведь не каждый курс может так серьёзно завлечь учащихся, привить им умение самостоятельно ставить цели, задачи, искать необходимую информацию для их решения, и в конечном счёте добиваться положительных результатов. Ведь курс «Основы робототехники» создан не только для того чтобы учащиеся могли самостоятельно строить и программировать собственных роботов, но и создаёт мотивацию для изучения таких дисциплин как: физика, информатика, английский. При изучении данного курса учащиеся учатся аналитически мыслить, творчески подходить к решению той или иной проблемы, а также укрепляет умение командной

работы. Ведь некоторые проблемы, с которыми сталкиваться учащиеся во время учебного процесса, они не в силах решить в одиночку.

Задачи, поставленные перед экспериментом:

Познавательные: Установление и укрепление связей, а также мотивации изучения таких дисциплин как физика, информатика, английский.

Образовательные: познакомить учащихся с платформой Arduino и средой программирования Arduino IDE, научить пользоваться и самостоятельно разрабатывать, и собирать различные устройства и роботов.

Развивающая задача: развитие творческой активности, самостоятельности в принятии оптимальных решений в различных ситуациях, развитие внимания, оперативной памяти, воображения, мышления (логического, комбинаторного, творческого).

Воспитывающая задача: воспитание ответственности, высокой культуры, дисциплины, коммуникативных способностей.

Ожидаемые результаты:

- 1) Приобретённые знания и умения:
- 2) Правила техники безопасности при работе с платой Arduino Uno и ПК
- 3) Знания основных элементов и возможностей платы, а также всех компонентов, которые можно использовать с её помощью.
- 4) Возможности и принцип действия большинства электронных приборов в доме и возможность их замены или дистанционного управления с помощью сделанного своими руками прибора на плате Arduino.
- 5) Получение первичных знаний и опыта программирования платы в программной среде Arduino IDE.

Данный факультативный курс имеет исключительно практическую ценность, и создан лишь для увлечения учащихся к возможностям платы. Метапредметные связи имеют только второстепенную важность. Каждый урок имеет под собой практическую ценность, и служит лишь для закрепления знаний, полученных в ходе изучения данного курса.

Самоанализ факультативного курса.

Данный курс рассчитан на изучение в течении одного учебного года, при условии, что занятия будут проводиться один раз в неделю. Поэтому весь курс опробовать в учебном заведении не представлялось возможным. По этой причине были выбраны самые интересные и простые лабораторные работы исключительно для изучения результатов, которые были получены на протяжении одного месяца.

После проведения лабораторной работы № 1 «Введение в Arduino» учащиеся изъявили желание в продолжении курса, поскольку отмечалось что возможности плат и того что с их помощью можно сделать были весьма и весьма обширны. К примеру, один из учащихся интересовался возможностью с помощью данной платы сделать полностью автоматическую систему освещения в доме. Поскольку в комплект вместе с платой Arduino идёт и всевозможные датчики (температуры, движения, инфракрасные приёмники и так далее), данный проект вполне возможно сделать самостоятельно. Этот и другие проекты осуществимы если иметь представление о возможностях, и умение программировать данные платы.

Следующая проведённая лабораторная работа ещё более укрепило увлечение учащихся в изучении данного курса. На этот раз они самостоятельно выполняли работу и программировали платы на выполнение простейших действий (в данном случае учащиеся самостоятельно устанавливали по своему усмотрению время свечения и затухания встроенного в плату светодиода). Также отмечается усиленная мотивация учащихся при изучении естественнонаучных дисциплин, о чём говорят учителя, с которыми проводилась беседа о актуальности данного курса.

Каждое занятие проходило в несколько этапов:

- 1) Приветствие;
- 2) Беседа с учащимися по постановке целей и задач;
- 3) Объяснение возможностей решения той или иной проблемы;

4) Самостоятельная работа учащихся по программированию и конструированию схемы для достижения поставленной темы;

5) Рефлексия.

В конце каждого занятия проводилась рефлексия: они делились впечатлениями друг с другом и педагогом своими достижениями, опытом, и объясняли то что узнали за пройденное занятие.

Из данного курса следует что:

По результатам опроса учащиеся всерьёз увлеклись изучением данного факультативного курса. Их привлекли почти «безграничные» возможности плат, простота их программирования и относительная дешевизна данной платформы;

Данные учащиеся улучшили свои знания по физике и информатике, о чём говорят проведённые тестирования, проводимые учителями-предметниками;

Лабораторные работы, учащиеся выполняли командой, поскольку все возникшие вопросы они решали, находясь за одним столом вместе. При невозможности самостоятельно решить проблему они обращались за помощью к педагогу. Также вне занятий учащиеся приносили самостоятельно разработанные скетчи и тестировали их, после чего, с под присмотром педагога, самостоятельно собирали и запускали разработанный ими проект.

Однако были и трудности, особенно при работе с монтажной платой. Учащиеся долго не могли понять по какому принципу они должны были соединять датчики и что к каким ножкам на плате подключать. Но по прошествии некоторого времени они научились решать эту проблему.

По результатам проведённого эксперимента можно сделать следующие выводы:

Данный курс полностью соответствует требованиям, которые ставят современные учебные заведения перед подобными курсами;

Актуальность данного курса была подтверждена, при проведении занятий учащиеся творчески подходили к выбору проектов, их разработке и планированию выполнения их реализации в реальной жизни;

За сравнительно небольшой промежуток времени учащиеся научились программировать и собирать схемы, чему способствует простота освоения программной среды Arduino IDE и аппаратной части Arduino Uno;

Возможность собирать почти любые конструкции и приборы на базе Arduino даёт полную свободу воображения, чем и воспользовались учащиеся составляя собственные схемы и разрабатывая проекты.

ЗАКЛЮЧЕНИЕ

Факультативные курсы - прекрасная возможность научить школьников размышлять и находить нужную информацию, решать сложные задачи, принимать решения, организовывать сотрудничество с одноклассниками и учителем. Ребёнок учится создавать идеи и воплощать их в жизнь, презентовать результаты своей работы. Введение учителем факультативных курсов по робототехнике в 10-11 классы на этапе обучения будет способствовать знакомству учеников с первыми шагами научной деятельности, их творческому и интеллектуальному развитию, научит организовывать и контролировать проект, тем самым развивая их гармонично и в ногу со временем.

Для того, чтобы внедрить факультативные курсы по робототехнике во внеурочную деятельность в школе нужно набрать группу учеников, которые будут увлечены и заинтересованы в данном элективном курсе. Основные понятия о факультативном курсе полностью описаны в параграфе 1.1.

Для того, чтобы что-то сконструировать нам необходимо знать основные данные о плате Arduino Uno. Информация о плате ней представлена в параграфе 1.2.

В параграфе 1.3 описываются и раскрываются методы обучения используемые в процессе преподавания робототехнике. Всего насчитывается пять методик (метод проектов, метод портфолио, метод взаимообучения, модульный метод и метод проблемного обучения.), которые раскрываются в данном параграфе. Эти методики раскрывают такие ученые как: Полат Е.С.; Фатеев И.А.; Дьяченко В.К.; Юцявичене П.А.; Оконь В.

Параграф 1.4 «Робототехника как средство формирования ключевых компетенций учащихся» выдвигает на первое место не информированность ученика, а способность организовывать свою работу. Запомнить и ответить - это накопление знаний; а применить свои знания и умения во вне учебной практической ситуации - это компетентность.

Особое место отведено практической деятельности, лабораторным и самостоятельным работам, применению приобретенных знаний и умений в реальных жизненных ситуациях. Все это можно воплотить на факультативных курсах по робототехнике, так как ученики могут проявить себя не только на курсах, но также и участвовать во всевозможных конкурсах по робототехнике, как на школьном уровне, так и в регионе.

Робототехника в школе представляет учащимся технологии 21 века, способствует развитию их коммуникативных способностей, развивает навыки взаимодействия, самостоятельности при принятии решений, раскрывает их творческий потенциал, навыки конструирования и развивает логическое мышление. Учащиеся лучше понимают, когда они что-либо самостоятельно создают или изобретают.

Внедрение факультативных курсов по робототехнике во внеурочное время школы помогает развитию коммуникативных навыков учащихся за счет активного взаимодействия детей в ходе групповой проектной деятельности. Чем разнообразнее образовательная среда, тем легче раскрыть индивидуальность личности ученика.

- 1) Робототехника всесторонне развивает личность учащегося:
- 2) развитие навыков конструирования, моделирования, элементарного программирования;
- 3) развитие логического мышления и мотивации к изучению наук;
- 4) формирование у учащихся целостного представления об окружающем мире;
- 5) ознакомление учащихся с основами конструирования и моделирования;
- 6) развитие способности творчески подходить к проблемным ситуациям;
- 7) развитие познавательного интереса и мышления учащихся;
- 8) овладение навыками начального технического конструирования и программирования.

Таким образом отмечаем следующие основные выводы:

Первое - анализ педагогической литературы показал, что метод проектов позволяет решить, мотивировать учащихся самостоятельно добывать, обрабатывать информацию, обмениваться ею, а также быстро и свободно ориентироваться в окружающем информационном пространстве. Метод проектов рассматривают как одну из личностно ориентированных технологий обучения, интегрирующую в себе проблемный подход, групповые методы, рефлексивные, презентативные, исследовательские, поисковые и прочие методики. Для того чтобы успешно организовать проектную деятельность учителю необходимо в полной мере изучить сущность метода проектов, этапы работы над ним, принципы, знать критерии оценки проектной деятельности и то, какие знания, умения и навыки формируются у участников такой деятельности.

Второе - эффективность обучения основам робототехники зависит от организации занятий, проводимых с применением следующих: объяснительно-иллюстративный, эвристический, проблемный, программированный, репродуктивный, поисковый, а также метод проектов. Обучение робототехнике - это отличный способ для подготовки детей к современной жизни, наполненной высокими технологиями, вовлекающий учащихся в процесс приобретения знаний и умений с помощью широкой исследовательской деятельности, базирующейся на комплексных, реальных вопросах и тщательно проработанных заданиях.

Третье - в дипломной работе определено место и роль робототехники в образовательном пространстве, обоснованы технологии, формы и методы обучения основам робототехники.

Таким образом, все задачи выполнены, цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бандурист В.Ю. Методический подход к разработке требований по созданию информационных средств обучения // // Проблемы информатизации образования: региональный аспект: Материалы V Всероссийской научно-практической конференции. – Чебоксары, 2007.
2. Беспалько В.П. Основы теории педагогических систем — Воронеж.: Изд-во ВГУ, 1997. - 304 с.
3. Бухаркина М.Ю., Полат Е.С., Моисеева М.В. Новые педагогические и информационные технологии в системе образования/под ред. Е. С. Полат. М.: Издательский центр «Академия», 2008. — 272 с.
4. Васильева З.И. История образования и педагогической мысли за рубежом и в России: учебное пособие для педагогических учебных заведений. Саратов: Издательский центр «Наука», 2009. – 143 с.
5. Гуляков, Е. Н. Новые педагогические технологии. М. Дрофа, 2006. – 328 с.
6. Дадонова А. В. Абушкин, Х. Х., Меж предметные связи в робототехнике как средство формирования ключевых компетенций учащихся //Учебный эксперимент в образовании. 2014. - 32-35 с.
7. Джереми Блум Год. Изучаем Arduino: инструменты и методы технического волшебства. БХВ-Петербург: 2015. – 334 с.
8. Дьяченко В.К. Организационная структура учебного процесса и ее развитие. М.: Просвещение, 1991. 192 с.
9. Злыгостев А.С. Информационно вспомогательный материал. Режим доступа: <http://etika-estetika.ru/sitemap/informat.shtml>.
10. Интернет энциклопедия. Определение «робототехника». Режим доступа: <https://ru.wikipedia.org/wiki/Робототехника>
11. Кравцова, Е.Е. Психология и педагогика. Краткий курс / Е.Е. Кравцова. - М.: Проспект, 2016. - 320 с.

12. Макаров И.М. Робототехника: история и перспективы / И.М.Макаров, Ю.И.Топчеев; Рос. акад. наук. – Москва.: МАИ, 2003. - 349 с.
13. Максимова В. Н. Межпредметные связи и совершенствование процесса обучения: книга для учителя. М.: Просвещение, 1984. - 143 с.
14. Нетесова О. С. Информатика и образование. – 2013. - 71-73 с.
15. Новый стандарт общего образования. Режим доступа: <http://www.standart.edu.ru>
16. Оконь В. Основы проблемного обучения М.: Просвещение, 1968. — 208 с.
17. Онлайн-магазин специализирующийся на микроконтроллерах. Режим доступа: <https://www.sunfounder.com/>
18. Официальный сайт платформы Arduino в России. Режим доступа <http://arduino.ru/>
19. Перельман Я. И. Занимательные задачи и опыты. - Хабаровск.: ВАП, 1994. – 452 с.
20. Петин В. Электроника. Проекты с использованием контроллера Arduino. СПб.: БХВ-Петербург, 2015. — 448 с.
21. Петин В. А., Биняковский А. А. Практическая энциклопедия Arduino. ДМК Пресс: 2017. – 152 с.
22. Ревич Ю. Занимательная электроника. Санкт-Петербург: БХВ-Петербург, 2007. – 576 с.
23. Рекомендации для пользователей Arduino. Режим доступа: <https://arduinomaster.ru/>
24. Сластенин, В.А. Педагогика: Учебник для студентов учреждений высш. проф. образования / В.А. Сластенин, И.Ф. Исаев, Е.Н. Шиянов. - М.: ИЦ Академия, 2012. - 608 с.
25. Старосельский В.И. Физика полупроводниковых. приборов микроэлектроники. Юрайт 2009. — 463 с.
26. Улли Соммер Программирование микроконтроллерных плат Arduino/Freduino. Санкт-Петербург: БХВ-Петербург, 2012. – 256 с.

27. Торо Карвинен, Киммо Карвинен, Вилле Валтокари Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi. М.: Вильямс 2015. – 445 с.
28. Фатеева И. А. Медиаобразование: теоретические основы и опыт реализации. Челябинск: Челяб. гос. ун-т, 2007. 270 с.
29. Факультативный курс //Большая советская энциклопедия /Сост. В. А. Юдин. - М.: Советская энциклопедия, 1985. – с. 573.
30. Юцявичене П. Теория и практика модульного обучения. Каунас, 1989. - 272с.

ПРИЛОЖЕНИЕ А

Лабораторная работа №5

Цель: Создать модель, имитирующую игру пламени свечи.

Краткая теория:

Для имитации пламени свечи мы будем использовать три ярких светодиода. Яркость свечения будет случайной и случайным будет момент изменения яркости. Случайное значение яркости будет меняться в некотором числовом диапазоне, то есть может принимать не только значение 0 или 1. Поэтому для подключения светодиодов нужно использовать ШИМ-пины. (ШИМ – широтно- импульсная модуляция).

Вспомним, что на плате Ардуино несколько контактов (пинов) могут использоваться как аналоговые выходы или ШИМ-пины, напряжение на которых может меняться в диапазоне от 0 до 255. Эти пины на плате обозначены знаком ~. На нашей плате это пины с номерами 3, 5, 6, 9, 10, 11. В случае, когда пин используется в режиме ШИМ, нужно вместо функции `digitalWrite (pin, HIGH)` или `digitalWrite (pin, LOW)` использовать функцию `analogWrite (pin, value)` где значение `value` может принимать значения от 0 до 255.

Микроконтроллер переводит число от 0 до 255 к напряжению от 0 до 5 В. Например, 85 — это 1 / 3 от

255, т.е. 1 / 3 от 5 В или 1,66 В. А для напряжения $\square 2.5$ В нужно задать значение `value = 127`. Случайное число, как и в программе на языке Pascal, можно задать функцией `random (n)`. При этом случайным образом генерируется число в интервале от 0 до $n-1$ включительно.

Соберите схему по указанному образцу (см. рисунок 7).

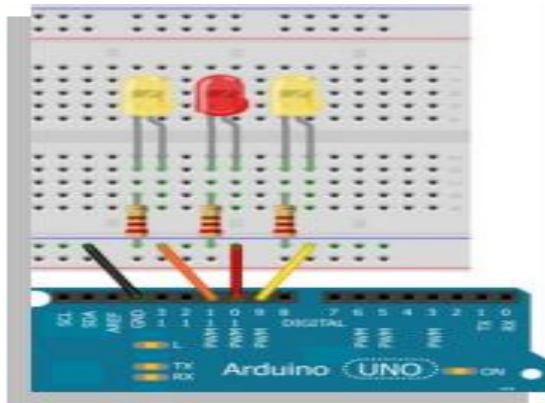


Рисунок 7 - схема подключения

// Объявляем переменные — три пина, к которым подключены
светодиоды

```
int led1 = 9;
```

```
int led2 = 10;
```

```
int led3 = 11;
```

```
void setup() // Инициализация пинов
```

```
{
```

```
pinMode(led1, OUTPUT);
```

```
pinMode(led2, OUTPUT);
```

```
pinMode(led3, OUTPUT);
```

```
}
```

```
void loop() // Бесконечный цикл выполнения программы
```

```
{
```

```
analogWrite(led1, 56+random(200)); // Случайное значение яркости от 56
```

до 255

```
analogWrite(led2, 56+random(200));
```

```
analogWrite(led3, 56+random(200));
```

```
delay(random(100)); // Случайное время задержки от 0 до 99
```

```
}
```

Задание для самостоятельного изучения:

Создать модель, имитирующую северное сияние.

Создать модель, имитирующую «мигалку» скорой помощи

Лабораторная работа №6

Работа с жидкокристаллическим экраном

Цель: ознакомиться с жидкокристаллическим экраном и научиться подключать к плате.

Краткая теория:

Жидкокристаллические индикаторы (ЖКИ, англ. LCD) являются удобным и недорогим средством для отображения данных ваших проектов. Символьный индикатор WH1602 позволяет выводить на экран 2 строки по 16 символов (размером 5×7 или 5×10 и дополнительная строка под курсор). Управляет работой дисплея контроллер.

В этом уроке подключим жидкокристаллический (LCD) экран к Arduino по последовательной шине данных i2c.

i2c - это аббревиатура слов Inter-Integrated-Circuit, интерфейсная шина для связи интегральных схем. Ее удобство заключается в том, что к ней можно подключать сразу несколько устройств, которые могут быть программно-адресованы по уникальному адресу.

Для управления ЖК-индикатором необходимо 6 или 10 выводов Arduino, в зависимости от того, выбран 4- или 8-битный режим обмена данными. Для сокращения требуемого числа выводов микроконтроллера можно работать в 4-битном режиме. В этом случае на выводах DB4–DB7 индикатора сначала будут передаваться старшие четыре бита данных/команды, затем – младшие четыре бита. Выводы DB0–DB3 останутся незадействованными.

В нашем эксперименте мы будем считывать данные с датчика температуры LM335, который мы рассмотрели в эксперименте 13, и выводить на экран ЖКИ значение температуры в Кельвинах и градусах Цельсия. Схема подключения датчика температуры и ЖКИ в 4-битном режиме к плате Arduino показана ниже (см. рисунок 8). Заметьте, что для питания ЖКИ нужен отдельный блок питания +5 В.

Приступим к написанию скетча. Функционал Arduino может быть расширен за счет использования библиотек. Библиотеки Arduino предоставляют дополнительную функциональность для использования в скетчах и сильно упрощают процесс написания программ. Ряд основных библиотек устанавливается вместе со средой Arduino IDE, а дополнительные, которых очень много, вы можете установить сами. При работе Arduino с ЖКИ-дисплеями на контроллере HD44780 будем использовать библиотеку LiquidCrystal. Для подключения библиотеки в начале скетча вставляем строку:

```
#include
```

Затем создаем переменную типа LiquidCrystal LiquidCrystal lcd(12, 11, 7, 6, 5, 4); где 12, 11, 7, 6, 5, 4 – номера контактов RS, E, D4, D5, D6, D7. В setup() запускаем функцию lcd.begin(), определяющую размерность индикатора, для установки курсора в определенную позицию – lcd.setCursor(), для вывода информации на экран дисплея – cd.print().

```
// Подключение библиотеки
```

```
#include <LiquidCrystal.h>
```

```
// инициализация с указанием контактов подключения
```

```
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
```

```
const int LM335=A0; // для подключения LM335
```

```
void setup() {
```

```
// установить размерность дисплея
```

```
lcd.begin(16, 2);
```

```
}
```

```
void loop()
```

```
{
```

```
double val = analogRead(LM335); // чтение
```

```
double voltage = val*5.0/1024; // перевод в вольты
```

```
// вывод значения в Кельвинах
```

```

lcd.setCursor(2,0);
lcd.print("Tk="); lcd.print(voltage*100); lcd.print("K");
double temp = voltage*100 - 273.15; // в градусы Цельсия
// вывод значения в градусах Цельсия
lcd.setCursor(2,1);
lcd.print("Tc="); lcd.print(temp); lcd.print("");
delay(1000); // пауза перед следующим измерением
}

```

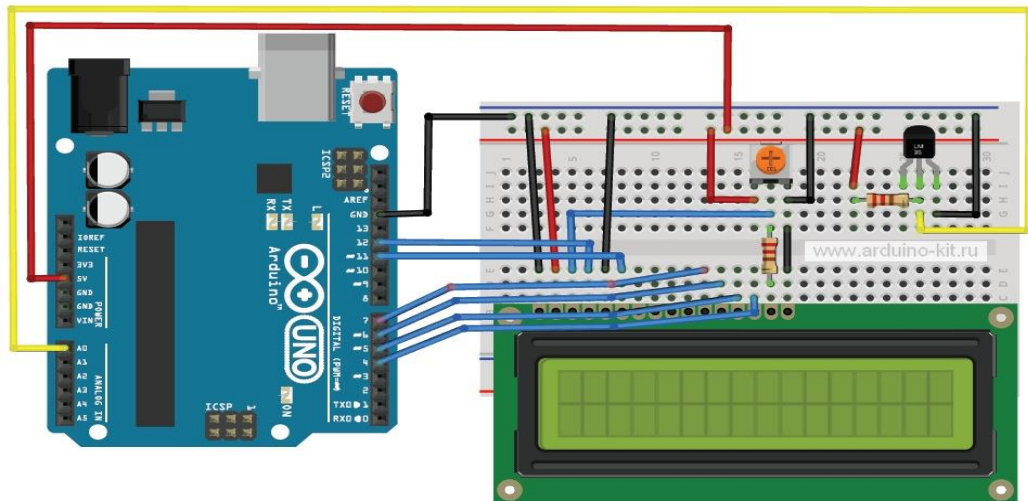


Рисунок 8 - Подключение жидкокристаллического экрана к плате Arduino.

Лабораторная работа №7

Передача данных в инфракрасном и ультразвуковом диапазонах.

Цель: Познакомиться с ультразвуковыми и инфракрасными датчиками расстояния и передачи данных, а так же возможностями применения их в создании роботизированных систем.

Arduino и датчики расстояния.

Рассмотрим работу Arduino с датчиками расстояния, которые являются неотъемлемой частью любого робота. Обратим внимание на ультразвуковые датчики HC-SR04 и инфракрасные датчики расстояния Sharp.

Ультразвуковые дальномеры HC-SR04.

Познакомимся с датчиками расстояния. Ультразвуковой дальномер HC-SR04 (см. рисунок 9) - это помещенные на одну плату приемник и передатчик ультразвукового сигнала. Кроме самих приемника и передатчика на плате находится еще и необходимая обвязка, чтобы сделать работу с этим датчиком простой и непринужденной.



Рисунок 9 - Датчик HC-SR04

Датчик обладает низким энергопотреблением, что также является немаловажным преимуществом в случае с мобильными роботами, не привязанными к розетке. Питается датчик HC-SR04 от 5 В. что тоже удобно при подключении его к Arduino.

Характеристики ультразвукового дальномера HC-SR04:

измеряемый диапазон - от 2-х до 500 см;

точность - 0.3 см;

угол обзора - $< 15^\circ$;

напряжение питания - 5 В.

Датчик имеет 4 вывода стандарта 2,54 мм:

VCC - питание +5 В;

65

Trig (T) - вывод входного сигнала;

Echo (R) - вывод выходного сигнала;

GND - земля.

Принцип работы ультразвукового дальномера HC-SR04.

В составе дальномера имеются два пьезоэлемента: один работает как излучатель сигнала, другой - как приемник. Излучатель генерирует сигнал, который, отразившись от препятствия, попадает на приемник. Измерив время, за которое сигнал проходит до объекта и обратно, можно оценить расстояние.

Последовательность действий следующая:

1. Подаем импульс продолжительностью 10 мкс на вывод Trig.
2. Внутри дальномера входной импульс преобразуется в 8 импульсов частотой 40 кГц и посылаются вперед через излучатель T (см. рис. 8.1).
3. Дойдя до препятствия, посланные импульсы отражаются и принимаются приемником R (см. рис. 8.1), в результате получаем выходной сигнал на выводе Echo.
4. Непосредственно на стороне контроллера переводим полученный сигнал в расстояние по формуле:
 - ширина импульса (мкс) / 58 = дистанция (см);
 - ширина импульса (мкс) / 148 = дистанция (дюйм).

Лабораторная работа №8

Тема: Работа сервопривода.

Цель: ознакомиться с сервоприводом Tower Pro и научиться подключать его к плате.

Краткая теория

Сервопривод (см. рисунок 10) – устройство, обеспечивающее преобразование сигнала в строго соответствующее этому сигналу перемещение (как правило, поворот) исполнительного устройства. Представляет собой прямоугольную коробку с мотором, схемой и редуктором внутри и выходным валом, который может поворачиваться на строго фиксированный угол, определяемый входным сигналом. Как правило, этот угол имеет предел в 60 в 180. Кроме этого, еще бывают сервоприводы и постоянного вращения.



Рисунок 10 – Сервопривод Tower Pro.

Сервопривод подключается с помощью трех проводов к управляющему устройству (драйверу или контроллеру) и источнику питания. Сервопривод управляется с помощью импульсов переменной длительности. Угол поворота определяется длительностью импульса, который подается по сигнальному проводу. Это называется широтно-импульсной модуляцией. Сервопривод ожидает импульса каждые 20 мс. Длительность импульса определяет, насколько далеко должен поворачиваться мотор. Например, импульс в 1,5 мс диктует мотору поворот в положение 90° (нейтральное положение). Когда сервопривод получает команду на перемещение, его управляющий орган перемещается в это положение и удерживает его. Если внешняя сила действует на сервопривод, когда он удерживает заданное положение,

сервопривод будет сопротивляться перемещению из этого положения. Максимальная величина силы, которую может выдерживать сервопривод, характеризует вращающий момент сервопривода. Однако сервопривод не навсегда удерживает свое положение, импульсы позиционирования должны повторяться, информируя сервопривод о сохранении положения. В нашем эксперименте мы будем управлять положением сервопривода с помощью потенциометра. Схема подключения сервопривода и потенциометра к плате Arduino (см рисунок 9).

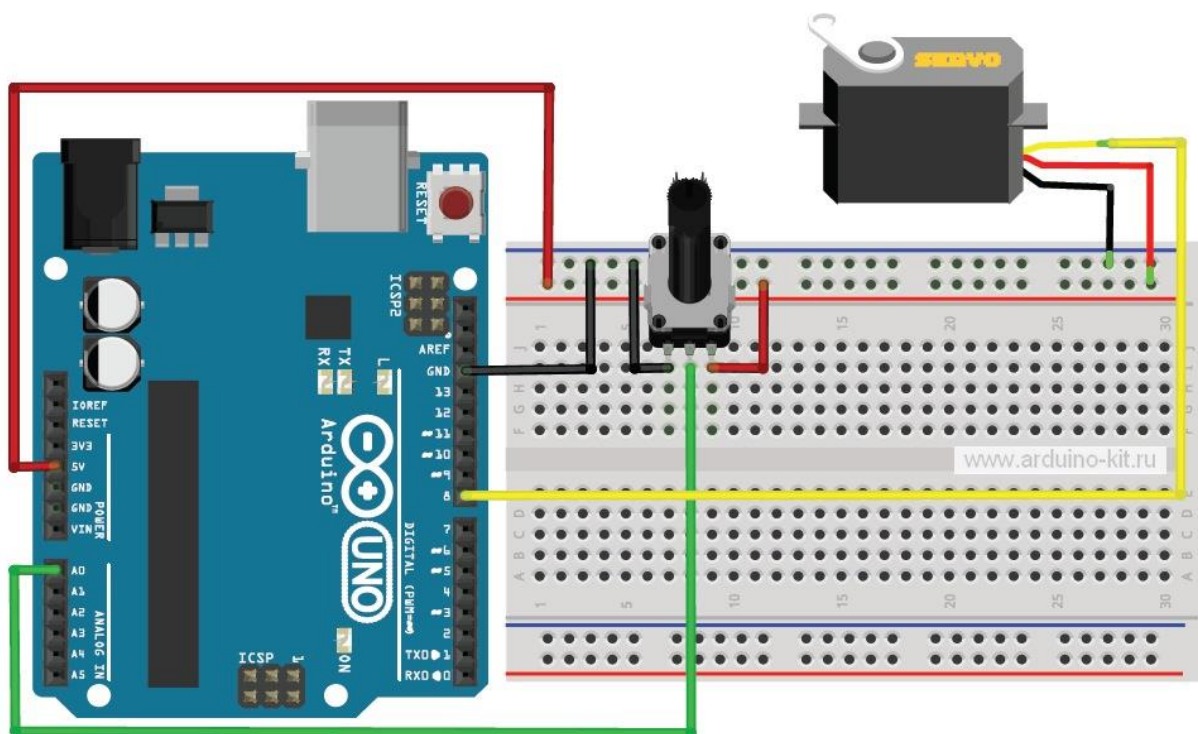


Рисунок 9 – Подключение сервопривода к плате Arduino.

Сервопривод подключается тремя проводами: питание (Vcc), «земля» (Gnd) и сигнальный (C). Питание – красный провод, он может быть подключен к +5 В внешнего источника питания, черны (или коричневый) провод – «земля» – подключается к GND-выводу Arduino GND, сигнальный (оранжевый/желтый/белый) провод подключается к цифровому выводу контроллера Arduino. Для питания сервопривода используем отдельный блок питания +5 В. Для управления сервоприводом в Arduino имеется стандартная

библиотека Servo. На платах, отличных от Mega, использование библиотеки отключает возможность применения analogWrite() (ШИМ) на пинах 9 и 10 (вне зависимости, подключены к этим пинам сервы или нет). На платах Mega до 12 сервоприводов могут использоваться без влияния на функциональность ШИМ, но использование от 12 до 23 сервомашинки отключит PWM ШИМ на пинах 11 и 12.

Аналоговые данные потенциометра (0–1023) масштабируем функцией map() в значения угла поворота сервопривода (0–180) и с помощью библиотечной функции servo.write(angle) даем сервоприводу команду для поворота. Скетч приведен в листинге 17.1.

```
#include <Servo.h> // подключение библиотеки Servo
Servo servo1;
const int pinServo=8; // Пин для подключения сервопривода
const int POT=0; // Аналоговый вход A0 для подключения
потенциометра
int valpot = 0; // переменная для хранения значения потенциометра
int angleServo = 0; // переменная для хранения угла поворота сервы
void setup()
{
// подключить переменную servo к выводу pinServo
servo1.attach(pinServo);
}
void loop()
{
valpot = analogRead(POT); // чтение данных потенциометра
// масштабируем значение к интервалу 0-180
angleServo=map(valpot,0,1023,0,180);
// поворот сервопривода на полученный угол
servo1.write(angleServo);
delay(15); // пауза для ожидания поворота сервопривода.
```


Лабораторная работа № 9

Собираем робота.

Цель: Спроектировать и собрать робота на гусеничном ходу, запрограммировать его на движение с помощью ИК ПДУ.

Оборудование и инструменты: набор гусениц Tamiya; перфорированная пластиковая панель с крепежом Tamiya 70098 Universal Plate Set; двойной редуктор Tamiya; МК Arduino Nano; среда программирования Arduino IDE, ПК.

Приступим к созданию робота. Наш робот представляет собой движущуюся гусеничную платформу с внешним управлением через ИК-пульт. Набор Tamiya 70098 Universal Plate Set (рис. 9.2) состоит из пластины со сквозными монтажными отверстиями, угловых скоб и крепежа, которые позволят смонтировать механические компоненты во множество позиций.

Двухмоторный редуктор Tamiya содержит два маленьких электродвигателя постоянного тока, которые вращают независимые 3-миллиметровые шестиугольные оси. Набор (рис. 9.3) позволяет собрать 4 различных варианта сдвоенных редукторов с разными передаточными числами: 12.7:1, 38:1, 115:1 и 344:1. которые могут быть легко заменены на более мощные моторы Pololu. Размеры корпуса редуктора 70x60x23 мм.

Драйвер двигателей L293D.

Нам необходимо реализовать следующие движения робота: движение вперед/назад два мотора крутятся в одну сторону; движение влево/вправо - моторы крутятся либо в одну сторону с разной скоростью, либо в разные стороны; движение на месте по кругу - моторы крутятся в разные стороны с одной скоростью; остановка - оба мотора не крутятся.

Для управления двигателями робота необходимо устройство, которое бы преобразовывало управляющие сигналы малой мощности в токи, достаточные для управления моторами. Такое устройство называют *драйвером двигателей*. Существует достаточно много самых различных схем для управления электродвигателями. Они различаются как мощностью, так и

элементной базой, на основе которой они выполнены. В нашем роботе использовался самый простой драйвер управления двигателями, выполненный на полностью готовой к работе микросхеме. Эта микросхема называется L293D и является одной из самых распространенных микросхем, предназначенных для этой цели.

Микросхема L293D содержит сразу два драйвера для управления электродвигателями небольшой мощности (четыре независимых канала, объединенных в две пары). Она имеет две пары входов для управляющих сигналов и две пары выходов для подключения электромоторов. Кроме того, у L293D есть два входа для включения каждого из драйверов. Эти входы используются для управления скоростью вращения электромоторов с помощью широтно-модулированного сигнала (ШИМ). Микросхема L293D обеспечивает разделение электропитания для микросхемы и для управляемых ею двигателей, что позволяет подключить электродвигатели с большим напряжением питания, чем у самой микросхемы. Разделение электропитания микросхем и электродвигателей может быть также необходимо для уменьшения помех, вызванных бросками напряжения, связанными с работой моторов. Принцип работы каждого из драйверов, входящих в состав микросхемы, идентичен, поэтому рассмотрим его на примере одного из них.

К выходам *OUTPUT1* и *OUTPUT2* подключим электромотор *MOTOR1*. На вход *ENABLE1*, включающий драйвер, подадим сигнал (соединим с положительным полюсом источника питания +5 В). Если при этом на входы *INPUT1* и *INPUT2* не подаются сигналы, то мотор вращаться не будет.

Если вход *INPUT1* соединить с положительным полюсом источника питания, а вход *INPUT2* - с отрицательным, то мотор начнет вращаться. Теперь попробуем соединить вход *INPUT1* с отрицательным полюсом источника мигания, а вход *INPUT2* - с положительным. Мотор начнет вращаться в другую сторону. Попробуем подать сигналы одного уровня сразу на оба управляющих входа *INPUT1* и *INPUT2* (соединить оба входа с

положительным полюсом источника питания или с отрицательным) - мотор вращаться не будет. Если мы уберем сигнал с входа *ENABLE1*, то при любых вариантах наличия сигналов на входах *INPUT1* и *INPUT2* мотор вращаться не будет. Эти входы используются для управления скоростью моторов с помощью ШИМ. Входы *ENABLE1* и *ENABLE2* отвечают за включение каждого из драйверов, входящих в состав микросхемы. Входы *INPUT1* и *INPUT2* управляют двигателем, подключенным к выходам *OUTPUT1* и *OUTPUT2*. Входы *INPUT3* и *INPUT4* управляют двигателем, подключенным к выходам *OUTPUT3* и *OUTPUT4*.

Контакт *Vs* соединяем с положительным полюсом источника электропитания двигателей или просто с положительным полюсом питания, если питание схемы и двигателей единое. Проще говоря, этот контакт отвечает за питание

электродвигателей. Контакт *Vss* соединяем с положительным полюсом источника питания. Этот контакт обеспечивает питание самой микросхемы. Четыре контакта *GND* соединяют с "землей" (общим проводом или отрицательным полюсом источника питания). Кроме того, с помощью этих контактов обычно обеспечивают теплоотвод от микросхемы, поэтому их лучше всего распаивать на достаточно широкую контактную площадку. Приступим к разработке скетча движений робота по командам (при получении кодов) с ИК-пульта. Процедура *loop* () проверяет, получен ли код с ИК-пульта. Если получен, вызывается процедура *ir_go*(). Процедура изменяет значения переменных в структуре текущего состояния моторов. На данной стадии, для отладки, будем выводить в монитор последовательного порта текущие действия робота. Рассмотрим назначение переменной *directions* структуры *POZ*. При выборе движения робота вперед или назад (клавишами \uparrow и \downarrow) по прямой и при увеличении/уменьшении скорости значение переменной *direction* равно значениям *poz1* и *poz2*.