

МИНИСТЕРСТВО ОБРАЗОВАНИЯ САМАРСКОЙ ОБЛАСТИ

структурное подразделение «Центр внешкольной работы»
государственного бюджетного общеобразовательного учреждения
лицей имени Героя Советского Союза П.И. Викулова городского округа Сызрань
Самарской области

УТВЕРЖДАЮ

Директор ГБОУ лицей г.Сызрани

_____ Лобачева Н.В.

Приказ № 203 от «26» июля 2024 г.

Программа принята решением
методического совета
Протокол № 16 от « 15 » июля 2024г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«ЮНЫЙ ПРОГРАММИСТ»

НАПРАВЛЕННОСТЬ: ТЕХНИЧЕСКАЯ

Возраст обучающихся: 7-13 лет

Срок реализации: 1 год

Разработчики: Ковалев С.С.,

Жугарева Л.Ю.

педагог дополнительного образования

г. Сызрань, 2024 г.

Оглавление	2
Пояснительная записка	4
Введение	4
Цель и задачи программы.....	5
Формы и методы обучения.....	7
Ожидаемые результаты	7
Критерии и способы определения результативности.....	8
Учебный план	9
Модуль 1.....	16
Модуль 2.....	20
Модуль 3.....	22
Ресурсное обеспечение программы.....	28
Учебно-методическое обеспечение образовательной программы.....	28
Материально-техническое обеспечение	28
Список литературы	29
Приложение 1. «Календарно-тематический план»	31
Приложение 2.	41
Приложение 3.	42

Краткая аннотация

Дополнительная общеобразовательная общеразвивающая программа «Юный программист» предназначена для учащихся с 7-13 лет, проявляющих интерес к знаниям информационных технологий.

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Юный программист» включает в себя 3 тематических модуля.

Обучение по данной программе направлено на приобретение учащимися знаний и привлечение их к современным технологиям телекоммуникаций.

Работа с различными языковыми конструкциями позволит школьникам развивать логическое и структурное мышление, комплексный подход при выполнении проектов и декомпозицию задач.

Нормативные основания для создания дополнительной общеобразовательной общеразвивающей программы:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Указ Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года»;
- Концепция развития дополнительного образования до 2030 года (утверждена распоряжением Правительства РФ от 31.03.2022 № 678-р);
- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства Российской Федерации от 29.05.2015 № 996-р);
- План мероприятий по реализации в 2021-2025 годах Стратегии развития воспитания в Российской Федерации на период до 2025 года (утвержден распоряжением Правительства Российской Федерации от 12.11.2020 № 2945-р);
- Приказ Министерства образования и науки Российской Федерации от 23.08.2017 № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;
- Приказ Министерства просвещения России от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказ Министерства просвещения РФ от 03.09.2019 № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;
- Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

• Стратегия социально-экономического развития Самарской области на период до 2030 года (утверждена распоряжением Правительства Самарской области от 12.07.2017 № 441).

Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242 «О направлении информации» (с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»);

Письмо министерства образования и науки Самарской области от 30.03.2020 № МО-16-09-01/434-ТУ (с «Методическими рекомендациями по подготовке дополнительных общеобразовательных общеразвивающих программ к прохождению процедуры экспертизы (добровольной сертификации) для последующего включения в реестр образовательных программ, включенных в систему ПФДО»).

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы дополнительной общеобразовательной общеразвивающей программы «Юный программист» техническая.

Актуальность программы обусловлена современной потребностью рынка в специалистах в области информационных технологий. Учитывается и междисциплинарность ИТ, что способствует интегрированию преподавания информатики, математики, физики, черчения, естественных наук с развитием инженерного мышления, через техническое творчество. Техническое творчество — мощный инструмент синтеза знаний, закладывающий прочные основы системного мышления. Предусмотрено приобретение навыков в области применения информационных технологий в робототехнике, виртуальной реальности, дизайне, геоинформационных системах, аэрокосмических технологиях и т.д.

Новизна программы заключается в новом решении проблем дополнительного образования и основана на комплексном подходе к подготовке ребенка к получению дальнейшего образования, развитию технических и интеллектуальных способностей через использование проектной и исследовательской технологий, подготовке личности «новой формации», готового к освоению информационных технологий и языкам программирования.

Отличительной особенностью программы является использование широкого спектра оборудования для приобретения практических навыков работы с ультрасовременными технологиями, такими как Internet of Things (IoT). Это технологическая концепция, согласно которой физические объекты и приборы оснащаются устройствами для обмена данными между собой и внешней средой. Считается, что такой подход открывает принципиально новые возможности в идентификации, измерении, сборе и обработке данных, невозможные без

применения данной технологии.

В данную программу введен региональный компонент. Ряд тем рассматривается на примере достижений науки и техники родного края (Самарская область).

Педагогическая целесообразность программы состоит в том, что введение в дополнительное образование образовательной программы «Юный программист» с использованием таких методов, как командная работа, поиск проблем и их практическое решение, анализ и обобщение опыта, подготовка исследовательских проектов и их защита, элементы соревнований и т.д., неизбежно изменит картину восприятия учащимися технических дисциплин, переводя их из разряда умозрительных в разряд прикладных. Применение детьми на практике теоретических знаний, полученных из области математики, физики, информатики, ведет к более глубокому пониманию основ, закрепляет полученные навыки, формируя образование в его наилучшем смысле. И с другой стороны, практическая работа с самым современным оборудованием данной области позволит учащимся в дальнейшем самостоятельно следовать тенденциям развития средств вычислительной техники, телекоммуникаций и веб-технологий. Таким образом, новое поколение теоретически окажется способным к совершению инновационного прорыва в современной науке и технике. Сформируется проектный подход и развивается командная работа юных «специалистов». Учениками приобретаются коммуникабельность, предсказательная аналитика и другие, что предоставит возможность в будущем стать успешными специалистами в любой области технологических разработок.

Цель – развитие интереса к информационным и телекоммуникационным технологиям, а также реализация творческих идей в области программирования в виде проектов различного уровня сложности.

Задачи:

Обучающие:

- ознакомить с методами программирования на языках, применяемых в современной вычислительной технике;
- дать первоначальные знания по работе в интегрированных средах разработки;
- навыкам конструирования сложных систем, управляемых микроконтроллерами и миникомпьютерами;
- сформировать общенаучные и технологические навыки программирования и проектирования.
- расширить, актуализировать знания о информационных технологиях.

Развивающие:

- развивать образное, техническое мышление;
- развивать умение работать в команде по предложенным инструкциям;
- развивать творческое мышление и воображение, умение излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений и информационного поиска;
- развивать навыки программирования, проектирования и эффективного использования электронного вычислительного оборудования;
- развивать внимательность, аккуратность и изобретательность при работе с техническими устройствами, создании электронных устройств и выполнении учебных проектов.

Воспитательные:

- воспитывать нравственные качества личности: настойчивость в достижении целей и продуктивно законченных результатов проектной деятельности, ответственность, дисциплинированность, трудолюбие;
- воспитывать коммуникативные качества;
- формировать творческое отношение к выполняемой работе;
- мотивировать учащихся к созданию собственных программных реализаций электронных устройств.

Возраст детей, участвующих в реализации программы: Программа «Юный программист» адресована обучающимся школьного возраста (7-13 лет). Данная возрастная категория характеризуется стремлением подростков овладеть различными умениями, способствует развитию чувства собственной умелости, компетентности и полноценности., что позволяет использовать в программе групповые и игровые технологии. Набор в группы осуществляется на добровольной основе, то есть принимаются все желающие заниматься.

Для детей важно разработать систему мотивации участия во всем, рейтинговая система. Они уже не маленькие, поэтому многое понимают, и готовы во всем вам помогать. В этом возрасте у них особенно развито желание лидерства. Они с удовольствием участвуют во всевозможных конкурсах и соревнованиях, понимают правила игры или идеи выступления, но могут быть обременены подростковыми комплексами и не сформированы окончательно. Этот возраст характеризует также стремление к сплочённости. Этим детям очень нравится быть командой, быть лучше всех.

Группы формируются из расчета 10-15 человек. Система набора в группы осуществляется по собственному желанию ребенка.

Сроки реализации: программа рассчитана на 1 год, объем за учебный год-108 часов

Формы обучения:

- занятие;
- лекция;
- создание и защита проектов;
- практические занятия.

Формы организации деятельности:

При изучении тем программа предусматривает использование фронтальной, индивидуальной и групповой формы учебной работы обучающихся:

- фронтальная форма - для изучения нового материала, информация подаётся всей группе из 10-15 человек;
- индивидуальная форма - самостоятельная работа учащихся, педагог может направлять процесс в нужную сторону;
- групповая форма помогает педагогу, сплотить группу общим делом, способствует качественному выполнению задания, для реализации проектной деятельности в малых группах (3-5 человека).

Занятия групп проводятся 2 раза в неделю по 2 и 1 часу, т.е. 3 часа в неделю (108 часов в год)

Планируемые результаты:

Личностные:

- формирование ответственного отношения к учению, готовности и способности, учащихся к самообразованию;
- развитие самостоятельности, личной ответственности за свои поступки;
- мотивация детей к познанию, творчеству, труду;
- формирование осознанного, уважительного и доброжелательного отношения к другому человеку;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе разных видов деятельности.

Метапредметные

Познавательные:

- самостоятельное создание алгоритмов познавательной деятельности для решения задач творческого и поискового характера;
- поиск нужной информации по заданной теме в источниках различного типа;
- умение давать определения, приводить доказательства, объяснять изученные положения на самостоятельно подобранных конкретных примерах;
- владение навыками исследовательской, проектной и социальной деятельности
- представлять информацию о проекте изделия и конечном результате

Регулятивные:

- объективное оценивание своих учебных достижений
- формирование умения понимать причины успеха/неуспеха своей деятельности;

Коммуникативные:

- формирование умения излагать свое мнение и аргументировать свою точку зрения
- готовность слушать собеседника и вести диалог
- не создавать конфликтов и находить выход из спорных ситуаций
- отвечать на вопросы по теме
- участвовать в парной, групповой, командной работе в процессе создания роботов
- уметь обосновывать свои предположения

Предметные:

Модульный принцип построения программы предполагает описание предметных результатов в каждом конкретном модуле.

Критерии оценки достижения планируемых результатов

Оценка достижения планируемых результатов освоения программы осуществляется по трем уровням: высокий (от 80 до 100% освоения программного материала), средний (от 51 до 79% освоения программного материала), низкий (менее 50% освоения программного материала). Оценочные материалы — пакет диагностических методик, позволяющих определить достижение учащимися планируемых результатов представлен в приложениях к программе.

Результат Высокий уровень освоения программы Учащиеся демонстрируют высокую заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание программы. На итоговом тестировании показывают отличное знание теоретического материала, практическое применение знаний воплощается в качественный продукт...

Средний уровень освоения программы Учащиеся демонстрируют достаточную заинтересованность в учебной, познавательной и творческой деятельности, составляющей содержание Программы. На итоговом тестировании показывают хорошее знание теоретического материала, практическое применение знаний воплощается в продукт, требующий незначительной доработки.

Низкий уровень освоения программы Учащиеся демонстрируют низкий уровень заинтересованности в учебной, познавательной и творческой деятельности, составляющей содержание программы. На итоговом тестировании показывают недостаточное знание теоретического материала, практическая работа не соответствует требованиям.

Формы подведения итогов

Для подведения итогов в программе используются учебно-исследовательские конференции, документальные формы подведения итогов реализации программы отражают достижения каждого обучающегося, к ним относятся: дневники достижений обучающихся, карты оценки результатов освоения программы, дневники педагогических наблюдений, портфолио обучающихся и т.д.

Фонд оценочных средств, методики и формы оценки учебных достижений:

- в течение курса предполагаются регулярные практики, на которых решение поставленной заранее известной задачи принимается в свободной форме (не обязательно предложенной преподавателем);
- по окончании курса учащиеся защищают творческий проект, требующий проявить знания и навыки по ключевым темам;
- полученные знания и навыки могут быть проверены на открытых конференциях, конкурсах и состязаниях, куда направляются наиболее успешные ученики.

Условия реализации программы.

Автоматизированное рабочее место учителя с выходом в Интернет;

Автоматизированное рабочее место учащегося с выходом в Интернет.

2. Учебный план

№	Наименование тем	Кол-во часов	Теория	Практика
1	Модуль «Основы программирования и алгоритмизации»	36	20	16
1.1	РАЗДЕЛ Введение в программирование	24	12	9
1.2	РАЗДЕЛ Программирование микроконтроллеров	15	8	7
2	Модуль «Программирование и разработка игр»	36	14	22
2.1	РАЗДЕЛ Разработка игр на Unity	14	8	6
2.2	РАЗДЕЛ Проектная	25	6	16

	деятельность			
3	Модуль «Веб-технологии и веб-разработка»	36	19	17
3.1	РАЗДЕЛ язык HTML и каскадная таблица стилей	23	15	8
3.2	РАЗДЕЛ Проектная деятельность	13	4	9
	Итого	108	53	55

Критерии оценки знаний, умений и навыков при освоении программы

Для того, чтобы оценить усвоение программы, в течении года используются следующие методы диагностики: тестирование, защита проектов, выполнение творческих заданий, участие в конкурсах, наблюдение.

По завершению учебного плана каждого модуля оценивание знаний проводится посредством тестирования, создания и защиты проектов.

Применяется 3-х балльная система оценки знаний, умений и навыков обучающихся (выделяется три уровня: ниже среднего, средний, выше среднего). Итоговая оценка результативности освоения программы проводится путём вычисления среднего показателя, основываясь на суммарной составляющей по итогам освоения 3-х модулей.

Уровень освоения программы ниже среднего – ребёнок овладел менее чем 50% предусмотренных знаний, умений и навыков, испытывает серьёзные затруднения при работе с учебным материалом; в состоянии выполнять лишь простейшие практические задания педагога.

Средний уровень освоения программы – объём усвоенных знаний, приобретённых умений и навыков составляет 50-70%: работает с учебным материалом с помощью педагога: в основном выполняет задания на основе образца; удовлетворительно владеет теоретической информацией по темам курса, умеет пользоваться литературой.

Уровень освоения программы выше среднего – обучающийся овладел на 70-100% предусмотренным программой учебным планом; работает с учебными материалами

самостоятельно, не испытывает особых трудностей; выполняет практические задания с элементами творчества; свободно владеет теоретической информацией по курсу; умеет применять полученную информацию на практике.

Формы контроля качества образовательного процесса

- наблюдение
- тестирование
- выполнение творческих заданий
- участие в конкурсах, викторинах в течение года
- защита проектов

3. Учебно-тематический план

№	Наименование тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.1	РАЗДЕЛ 1 Введение в программирование	24	12	10	Опрос
1.	Вводное занятие. Техника безопасности. Алгоритм и его формальное исполнение	2	1	1	Наблюдение, беседа
2.	Принципы разработки алгоритмов и программ	1	1	-	Беседа
3.	Изучение и прохождение Algotica Iterations	1	-	1	Тренинг, игра
4.	Кодирование основных типов алгоритмических структур	1	1	-	Наблюдение
5.	Линейный алгоритм, организация ветвления и алгоритмическая структура циклов	2	1	1	Наблюдение
6.	Блок-схемы и графы	1	1	-	Беседа
7.	Языки программирования	2	2	-	Беседа
8.	Операторы ветвления	1	1	-	Наблюдение
9.	Операторы повторений	2	1	1	Наблюдение
10.	Изучение и прохождение Kodu game lab	1	-	1	Тренинг, игра
11.	Функции и процедуры	2	1	1	Упражнение

12.	Массивы	1	1	-	Наблюдение
13.	Одномерные массивы	1	-	1	Упражнение
14.	Двумерные массивы	1	-	1	Упражнение
15.	Классы	2	1	1	Беседа
16.	Запись алгоритмов на языках программирования и решение задач	1	-	1	Тестирование
1.2	РАЗДЕЛ 2 Программирование микроконтроллеров	15	8	6	Опрос
1.	Что такое микроконтроллер?	2	1	1	Беседа
2.	Обзор языка программирования Arduino	1	1	-	Беседа
3.	Простейшие компоненты: кнопка, светодиод. Управление яркостью.	2	1	1	Наблюдение
4.	Сенсоры. Протоколы обмена данными, получение информации с сенсоров	1	1	-	Наблюдение
5.	Вывод информации. LCD - дисплей. Соединение с компьютером	2	1	1	Наблюдение
6.	Как сделать свой дом умным?	1	1	-	Беседа
7.	Центры управления	2	1	1	Наблюдение
8.	Датчики	1	-	1	Наблюдение
9.	Системы видеонаблюдения	1	-	1	Упражнение
10.	Готовые наборы, умная техника	1	1	-	Тестирование
Итого:		36	20	16	

2.1	РАЗДЕЛ 1	14	8	6	Опрос
------------	-----------------	-----------	----------	----------	--------------

	Разработка игр на Unity				
1.	2D и 3D моделирование	2	2	-	Беседа
2.	3D моделирование в Blender	1	-	1	Упражнение
3.	Работа с анимацией и текстурами	2	1	1	Беседа
4.	Создание текстур при помощи Gimp	1	-	1	Упражнение
5.	Звук и озвучивание	2	2	-	Наблюдение
6.	Работа со звуком в Audacity	1	-	1	Упражнение
7.	Редакторы кода и скрипты	2	2	-	Наблюдение
8.	MonoDevelop, Microsoft Visual Studio, Notepad ++	1	-	1	Беседа
9.	Межплатформенная среда разработки компьютерных игр Unity	2	1	1	Тестирование
2.2	РАЗДЕЛ 2 Проектная деятельность	25	6	16	Опрос
1.	Графика и обработка пошаговых событий	1	1	-	Беседа
2.	Проект: «Крестики-нолики»	2	-	2	Упражнение, защита проекта
3.	Проект: «Четыре в ряд»	1	-	1	Упражнение, защита проекта
4.	Синхронизация, движение, столкновение и анимация в реальном	1	1	-	Наблюдение,

	времени				
5.	Проект: «Space Invaders»	1	-	1	Упражнение, защита проекта
6.	Проект: «Тетрис»	2	-	2	Упражнение, защита проекта
7.	Переинженеринг	1	1	-	Беседа
8.	Планирование, анализ столкновений, физика, искусственный интеллект	1	1	-	Наблюдение
9.	Проект: «Бильярд»	1	-	1	Упражнение, защита проекта
10.	Проект: «Pac Man»	2	-	2	Упражнение, защита проекта
11.	Платформеры, Action/Adventure, RPG	1	1	-	Беседа
12.	Проект: «Never Alone»	2	-	2	Упражнение, защита проекта
13.	Проект: «Ori»	1	-	1	Упражнение, защита проекта
14.	Проект: «LIMBO»	2	-	2	Упражнение, защита проекта
15.	Проект: «Child of	1	-	1	Упражнение

	Light»				ие, защита проекта
16.	3D игры	1	1	-	Беседа
17.	Проект: «Cubium Dreams»	1	-	1	Тестирова ние, защита проекта
Итого:		36	14	22	

3.1	РАЗДЕЛ 1 язык HTML и каскадная таблица стилей	23	15	8	Опрос
1.	Передача информации	2	2	-	Беседа
2.	Базовые понятия компьютерных сетей	1	1	-	Беседа
3.	Локальные и глобальные сети. Конфигурирование компьютеров в локальной сети	2	1	1	Беседа
4.	Доменная система и протоколы передачи данных	1	1	-	Беседа
5.	Всемирная глобальная паутина. Язык разметки гипертекстовых страниц HTML: обзор возможностей	2	1	1	Беседа
6.	Структура HTML- документа	1	1	-	Наблюден ие
7.	Модульность и виды подходов к верстке	2	1	1	Наблюден ие
8.	Табличная верстка	1	1	-	Упражнен ие
9.	Верстка слоями	2	1	1	Упражнен ие
10.	Блочная верстка	1	1	-	Упражнен ие
11.	Верстка фреймами	2	1	1	Упражнен ие

12.	Адаптивная верстка	1	1	-	Беседа
13.	Возможности и методы HTML. Создание собственных веб-страниц и размещение их на сервере	2	-	2	Упражнение
14.	Валидность и кроссбраузерность	1	1	-	Упражнение
15.	Понятие каскадной таблицы стилей CSS, применение и написание	2	1	1	Тестирование
3.2	РАЗДЕЛ 2 Проектная деятельность	13	4	9	Опрос
1.	Целеполагание, целевая аудитория и функционал сайта (Разбор сайтов Самарской области)	1	1	-	Беседа
2.	Содержание и структура сайтов Самарской области	2	1	1	Беседа
3.	Дизайнерские идеи и решения. Лучшие сайты Самарской области	1	1	-	Наблюдение
4.	Создание персонального сайта	2	-	2	Упражнение
5.	Создание личного блога	1	-	1	Упражнение
6.	Создание тематического сайта	2	-	2	Упражнение
7.	Создание новостного сайта	1	-	1	Упражнение
8.	Создание интернет-магазина	2	-	2	Упражнение
9.	Размещение сайта в интернете	1	1	-	Тестирование, защита сайта
Итого:		36	19	17	

Итого за год	108	53	55	
---------------------	------------	-----------	-----------	--

4. Содержание программы

Модуль «Основы программирования и алгоритмизации»

Цель: знакомство с традиционными языковыми конструкциями, как в лекционной, так и в игровой формах, а также освоение функционирования и программирования интернет вещей.

Задачи:

Обучающие:

- изучить основы алгоритмизации;
 - закрепить знания, умения и навыки известные по школьному курсу информатики;
 - сформировать навыки программирования простейших программ;
- на основе полученных знаний о микроконтроллерах спроектировать подобие «умного дома».

Развивающие:

- развивать навыки программирования;
- развивать умение работать в команде по предложенным инструкциям;
- развивать навыки работы со схемами и графами.

Воспитательные:

- формирование творческого отношения к работе с умной техникой;
- воспитание ответственности в процессе создания проекта «Умный дом»

Предметные ожидаемые результаты

Обучающийся должен знать:

- основы программирования и простейшие компоненты

Обучающийся должен уметь:

- общаться с передовыми информационными технологиями и программным обеспечением;
- закладывать фундамент для дальнейшей успешной деятельности в сфере IT.

Обучающийся должен приобрести навык:

- управления микроконтроллерами

Содержание программы модуля

Раздел 1. Введение в программирование.

Тема 1. Вводное занятие. Инструктаж по ОТ и ТБ. Алгоритм и его формальное исполнение.

Теория: введение в программу. Инструктаж по технике безопасности. Понятие алгоритма.

Практика: понимание и чтение алгоритмов. Усвоение свойств алгоритмов. Приведение примеров.

Тема 2. Принципы разработки алгоритмов и программ.

Теория: виды алгоритмов. Примеры линейных алгоритмов и работа с ними. Связь понятий алгоритм и программа.

Тема 3. Изучение и прохождение Kodu game lab.

Практика: Kodu game lab представляет собой интегрированную среду разработки для обучения основам программирования и алгоритмизации.

Тема 4. Кодирование основных типов алгоритмических структур.

Теория: типы алгоритмических структур. Знакомство с ветвлением, повторением и подпрограммами.

Тема 5. Линейный алгоритм, организация ветвления и алгоритмическая структура циклов.

Теория: отличие линейных алгоритмов от разветвляющихся и повторяющихся.

Переход от линейного алгоритма к другим алгоритмическим структурам.

Практика: обучающиеся составляют простейшие линейные алгоритмы, а затем модифицируют их.

Тема 6. Блок-схемы и графы.

Теория: наглядное представление алгоритмов. Изучение принципов визуализации алгоритма в блок-схему или граф.

Тема 7. Языки программирования.

Теория: что такое язык программирования? Какими бывают языки программирования? Для чего служат различные языковые системы? Примеры.

Тема 8. Оператор ветвления.

Теория: оператор ветвления IF (ЕСЛИ), его языковая конструкция на различных языках. Применение ELSE (ИНАЧЕ). Примеры.

Тема 9. Операторы повторений.

Теория: циклы WHILE (ПОКА), FOR (ДЛЯ). Применение.

Практика: написание программы с использованием ветвления и повторения на различных языках (Pascal, Python, C++).

Тема 10. Изучение и прохождение CodeCombat.

Практика: CodeCombat самый захватывающий и бесплатный способ в игровой форме научиться реальному программированию. Управляя персонажем путем написания кода на нескольких возможных языках программирования, вы совершенствуетесь и постепенно сталкиваетесь с все более запутанными заданиями и тем самым познаете основные принципы программирования.

Тема 11. Функции и процедуры.

Теория: понятия функций и процедур. Написание подпрограмм на различных языках.

Применение. Дополнительные библиотеки и их подключение.

Практика: написание кода с использованием подпрограмм и подключением дополнительных библиотек.

Тема 12. Массивы.

Теория: понятие массива. Виды массивов. Примеры.

Тема 13. Одномерные массивы.

Практика: работа с одномерными массивами данных на практике. Составление программ с их применением.

Тема 14. Двумерные массивы.

Практика: работа с двумерными массивами данных на практике. Составление программ с их применением.

Тема 15. Классы.

Теория: понятие класс. Что можно описать с его помощью и как использовать в процессе программирования. Объектно-ориентированное программирование.

Практика: работа посредством объектно-ориентированного программирования. Создание классов.

Тема 16. Запись алгоритмов на языках программирования и решение задач.

Практика: решение задач с применением ранее изученного материала.

Раздел 2. Программирование микроконтроллеров. Тема 1. Что такое микроконтроллер?

Теория: понятие микроконтроллер. Для чего используется Arduino. Демонстрация возможностей наборов Матрешка Z.

Практика: сборка различных проектов согласно инструкции.

Тема 2. Обзор языка программирования Arduino.

Теория: Си-подобный язык программирования Arduino, представление простейших программ для управления микроконтроллером.

Тема 3. Простейшие компоненты: кнопка, светодиод. Управление яркостью.

Теория: что такое кнопка и диод, их представление в электротехнике.

Практика: сборка проекта «Светильник» при помощи набора Матрешка Z.

Тема 4. Сенсоры. Протоколы обмена данными, получение информации с сенсоров.

Теория: работа различных сенсоров. Примеры использования.

Тема 5. Вывод информации. LCD - дисплей. Соединение с компьютером.

Теория: портативный дисплей и вывод информации на него при помощи Arduino Uno и набора Матрешка Z.

Практика: реализация заданного проекта согласно инструкции Матрешка Z.

Тема 6. Как сделать свой дом умным?

Теория: как управлять температурой, влажностью, видеть что происходит у вас дома в ваше отсутствие и многое другое.

Тема 7. Центры управления.

Теория: что может являться центром управления умным домом. Обзор портативного компьютера Raspberry Pi из набора Малина.

Практика: работа с Raspberry pi из набора Малина.

Тема 8. Датчики.

Практика: работа с датчиками из набора Матрешка Z на базе Arduino UNO.

Тема 9. Системы видеонаблюдения.

Практика: установка системы видеонаблюдения и настройка управления через смартфон.

Тема 10. Готовые наборы, умная техника

Теория: изучение готовых наборов умного дома, а также технический обзор на тему: «Какую технику в наше время можно назвать умной».

Модуль «Программирование и разработка игр»

Цель: освоить работу и программирование самостоятельных игровых систем, а также изучить необходимое программное обеспечение в процессе создания игр.

Задачи:

Обучающие:

- научиться планировать деятельность и реализовать в командах простейшие игровые проекты;
- научиться программировать 2D игры, изучить основы искусственного интеллекта, физики, анимации, текстур и многого другого.

Развивающие:

- развивать навыки проектной деятельности

Воспитательные:

- формировать творческое отношение к выполняемой работе;
- мотивировать учащихся к созданию собственных программных реализаций.

Предметные ожидаемые результаты

Обучающийся должен знать:

- процесс создания игр в современном мире;

Обучающийся должен уметь:

- проектировать свои собственные игровые оболочки с помощью ранее полученных умений;

Обучающийся должен приобрести навык:

- работы над проектами.

Содержание программы модуля

Раздел 1. Разработка игр на Unity.

Тема 1. 2D и 3D моделирование.

Теория: моделирование. Различия 2D и 3D моделирования. Бесплатные программные средства.

Тема 2. 3D моделирование в Blender.

Практика: Blender – профессиональное свободное и открытое программное обеспечение для создания трёхмерной компьютерной графики, включающее в себя средства моделирования, скульптинга, анимации, симуляции, рендеринга, постобработки и монтажа видео со звуком, компоновки с помощью «узлов», а также создания 2D анимаций. Знакомство с интерфейсом программы. Создание простейших моделей.

Тема 3. Работа с анимацией и текстурами.

Теория: создание анимации и добавление готовых текстур инструментами Blender.

Практика: моделирование 3D-модели, добавление текстуры и анимирование по ключевым кадрам.

Тема 4. Создание текстур при помощи Gimp.

Практика: создание бесшовных текстур при помощи Gimp. GNU Image Manipulation Program или GIMP — свободно распространяемый растровый графический редактор, программа для создания и обработки растровой графики и частичной поддержкой работы с векторной графикой.

Тема 5. Звук и озвучивание.

Теория: откуда рождается звук? Бесплатные программные средства для создания и редактирования звуков.

Тема 6. Работа со звуком в Audacity.

Практика: Audacity — свободный многоплатформенный аудиоредактор звуковых файлов, ориентированный на работу с несколькими дорожками. Изучение программного интерфейса.

Тема 7. Редакторы кода и скрипты.

Теория: что такое скрипт и почему необходимы редакторы кода.

Тема 8. MonoDevelop, Microsoft Visual Studio, Notepad ++.

Практика: изучение на практике различных программных средств и оболочек для редактирования кода.

Тема 9. Межплатформенная среда разработки компьютерных игр Unity.

Теория: Unity. Как создавать игры для различных платформ с его помощью и почему для этого необходимы ранее изученные приложения в лице Blender, Gimp, Audacity и Notepad++.

Практика: интерфейс Unity. Добавление объектов и создание игровой сцены.

Раздел 2. Проектная деятельность.

Тема 1. Графика и обработка пошаговых событий.

Теория: обработка пошаговых игровых событий и типы игровой графики.

Тема 2. Проект: «Крестики-нолики».

Практика: разработка проекта под кодовым названием «Крестики-нолики». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 3. Проект: «Четыре в ряд».

Практика: разработка проекта под кодовым названием «Четыре в ряд». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 4. Синхронизация, движение, столкновение и анимация в реальном времени.

Теория: как реализовать многопоточность, реагировать на столкновения игровых объектов, запускать анимации в реальном времени.

Тема 5. Проект: «Space Invaders».

Практика: разработка проекта под кодовым названием «Space Invaders». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 6. Проект: «Тетрис».

Практика: разработка проекта под кодовым названием «Тетрис». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 7. Переинженеринг.

Теория: понятие переинженеринга. Как его избежать и чем оно опасно в рамках программирования и игростроения.

Тема 8. Планирование, анализ столкновений, физика, искусственный интеллект.

Теория: добавление искусственного интеллекта, физических свойств игровым объектам.

Тема 9. Проект: «Бильярд».

Практика: разработка проекта под кодовым названием «Бильярд». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 10. Проект: «Pac Man».

Практика: разработка проекта под кодовым названием «Pac Man». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 11. Платформеры, Action/Adventure, RPG.

Теория: изучение различных и распространенных игровых жанров, над элементами которых придется работать в процессе обучения.

Тема 12. Проект: «Never Alone».

Практика: разработка проекта под кодовым названием «Never Alone». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 13. Проект: «Ori».

Практика: разработка проекта под кодовым названием «Ori». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 14. Проект: «LIMBO».

Практика: разработка проекта под кодовым названием «LIMBO». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 15. Проект: «Child of Light».

Практика: разработка проекта под кодовым названием «Child of Light». В данном задании требуется воссоздать собственное видение или подобие известной игры.

Тема 16. 3D игры.

Теория: обзор различных геймдизайна и внутреннего игростроения в 2D- и 3D-форматах.

Тема 17. Проект: «Cubium Dreams».

Практика: разработка проекта под кодовым названием «Cubium Dreams». В данном задании требуется воссоздать собственное видение или подобие малоизвестной игры в 3D.

Модуль «Веб-технологии и веб-разработка»

Цель: знакомство с языком гипертекстовой разметки HTML и каскадной таблицей стилей CSS.

Задачи:

Обучающие:

- научиться конструировать и верстать простейшие структуры сайтов самостоятельно.
- познакомиться с программным обеспечением для работы с интернет ресурсами.

Развивающие:

-развивать навыки работы с интернет браузерами;

Воспитательные:

- формировать творческое отношение к выполняемой работе;
- мотивировать обучающихся к созданию собственных интернет ресурсов

Предметные ожидаемые результаты

Обучающийся должен иметь:

- целостное представление о сущности всемирной паутины;

Обучающийся должен уметь:

- ориентироваться в просторах интернета и находить необходимую информацию;

Обучающийся должен приобрести навык:

- создания собственных интернет ресурсов.

Содержание программы модуля

Раздел 1. Язык HTML и каскадная таблица стилей CSS. Тема 1. Передача информации.

Теория: передача информации на расстояния посредством компьютерных сетей.

Тема 2. Базовые понятия компьютерных сетей.

Теория: топология сети, протоколы передачи данных, адресация.

Тема 3. Локальные и глобальные сети. Конфигурирование компьютеров в локальной сети.

Теория: как объединить компьютеры в локальную сеть. Отличие локальных сетей от глобальных.

Практика: создание локальной сети на практике.

Тема 4. Доменная система и протоколы передачи данных.

Теория: протоколы TCP/IP и доменная система. Что означают всемирно известные .com, .ru, .edu и многое другое.

Тема 5. Всемирная глобальная паутина. Язык разметки гипертекстовых страниц HTML: обзор возможностей.

Теория: всемирная сеть интернет и язык гипертекстовой разметки веб-страниц HTML, каскадная таблица стилей CSS. Программное обеспечение для создания веб-страниц. Конструктор WIX.

Практика: работа с заголовками и добавление текста на HTML-страницу.

Тема 6. Структура HTML-документа.

Теория: рассмотрение структуры веб-страниц. Head, body, footer.

Тема 7. Модульность и виды подходов к верстке.

Теория: таблицы, слои, блоки, фреймы, модули и комбинирование различных подходов к верстке веб-страниц.

Практика: рукописная заготовка проекта своего сайта.

Тема 8. Табличная верстка.

Теория: обзор метода верстки веб-документов, при котором в качестве структурной основы для расположения текстовых и графических элементов документа используются таблицы.

Тема 9. Верстка слоями.

Теория: обзор метода верстки веб-документов, при котором в качестве структурной основы для расположения текстовых и графических элементов документа используются слои.

Практика: проектирование своего сайта.

Тема 10. Блочная верстка.

Теория: обзор метода верстки веб-документов, при котором в качестве структурной основы для расположения текстовых и графических элементов документа используются блоки.

Тема 11. Верстка фреймами.

Теория: обзор метода верстки веб-документов, при котором в качестве структурной основы для расположения текстовых и графических элементов документа используются фреймы.

Практика: создание своего сайта иными методами верстки и анализ полученных результатов.

Тема 12. Адаптивная верстка.

Теория: обзор метода верстки веб-документов, при котором в качестве структурной основы для расположения текстовых и графических элементов документа используются сочетания и различные комбинации традиционных подходов к верстке веб-страниц.

Тема 13. Возможности и методы HTML. Создание собственных веб-страниц и размещение их на сервере.

Практика: зная содержание и компоновку своего сайта применить полученные знания и сверстать его, используя комбинированный стиль.

Тема 14. Валидность и кроссбраузерность.

Теория: понятие валидности и кроссбраузерности. Соответствие сайтов определенным правилам и работа в разных браузерах и версиях.

Тема 15. Понятие каскадной таблицы стилей CSS, применение и написание.

Теория: CSS. Почему принято интегрировать на веб-страницу различные стили оформления из отдельного файла.

Практика: работа с каскадной таблицей стилей. Изменение оформления ранее созданного сайта.

Раздел 2. Проектная деятельность.

Тема 1. Целеполагание, целевая аудитория и функционал сайта. *Теория:* с чего лучше начать разработку своего нового проекта. **Тема 2.** Содержание и структура сайта.

Теория: как определиться со структурой сайта и его наполнением.

Практика: рукописные заготовки новых идей.

Тема 3. Дизайнерские идеи и решения. *Теория:* как оформить разные типы сайтов.

Тема 4. Создание персонального сайта.

Практика: верстка своей личной страницы в интернете.

Тема 5. Создание личного блога. *Практика:* верстка своего сайта-блога. **Тема 6.**

Создание тематического сайта.

Практика: верстка сайта соответствующего определенной тематике.

Тема 7. Создание новостного сайта. *Практика:* верстка проекта «новостной сайт».

Тема 8. Создание интернет-магазина.

Практика: верстка проекта «интернет-магазин»

Тема 9. Размещение сайта в интернете.

Теория: Как разметить свою страницу в интернете, чтобы она стала достоянием общественности, и любой желающий мог ее посетить. Как продвигать свой сайт в поисковых системах.

Критерии оценки знаний, умений и навыков при освоении программы.

Для того чтобы оценить усвоение программы, в течение года используются следующие методы диагностики: входная диагностика, промежуточная диагностика, итоговая диагностика. По завершению каждого модуля оценивание знаний проводится посредством интерактивного занятия. Применяется 3-х балльная система оценки знаний, умений и навыков обучающихся (выделяется три уровня: ниже среднего, средний, выше среднего). Итоговая оценка результативности освоения программы проводится путём вычисления среднего показателя, основываясь на суммарной составляющей по итогам освоения 3-х модулей.

Уровень освоения программы ниже среднего – ребёнок овладел менее чем 50% предусмотренных знаний, умений и навыков, учащиеся не знают значительной части материала, допускают существенные ошибки, с большими затруднениями выполняют практические задания.

Средний уровень освоения программы – объём усвоенных знаний, приобретённых умений и навыков составляет 50-70%; учащиеся должны знать основные блоки команд, уметь выполнять алгоритмы, описанные с использованием конструкций ветвления и повторения, грамотно и по существу излагать программный материал, не допуская существенных неточностей в ответе.

Уровень освоения программы выше среднего – учащийся овладел на 70-100% предусмотренным программой учебным планом; учащиеся должны знать правила техники безопасности при работе, грамотно излагать программный материал, знать основные блоки команд, уметь выполнять алгоритмы, описанные с использованием конструкций ветвления и повторения и уметь самостоятельно создавать и выполнять программы для решения алгоритмических задач в программе Scratch.

Для оценки уровня освоения дополнительной общеобразовательной программы проводится посредством **текущего контроля, промежуточной итоговой аттестации.**

Текущий контроль выявляет степень сформированности практических умений и навыков учащихся в выбранном ими виде деятельности. Текущий контроль может проводиться в форме наблюдения, индивидуальное собеседование, групповая беседа, опрос.

Текущий контроль осуществляется без фиксации результатов.

Промежуточная аттестация проводится с целью установления уровня (высокий, средний, ниже среднего) освоения отдельной части или всего объёма дополнительной общеобразовательной программы.

-высокий – программный материал усвоен учащимся полностью, учащийся имеет высокие достижения;

-средний – усвоение программы в полном объеме, при наличии несущественных

ошибок;

-ниже среднего – усвоение программы в неполном объеме, допускает существенные ошибки в теоретических и практических заданиях.

Формы промежуточной аттестации учащихся: тестирование, выполнение практического задания, защита проекта.

Итоговая аттестация – это промежуточная аттестация, которая проводится по завершению всего объёма дополнительной общеобразовательной программы, которая проводится в следующих формах: тестирование, доклад, защита творческих работ и проектов, итоговое мероприятие.

Способы определения результативности:

- Практические задания.
- Самостоятельные творческие, проектные работы,
- Выставки, конкурсы.
- Работы – участники конкурсов, выставок, документы – свидетельства, дипломы с выставок и т.д.)

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

	Раздел	Методические виды продукции (разработки игр, походов, экскурсий, конкурсов, бесед, конференций и т.д.)	Рекомендации по проведению лабораторных и практических работ, по постановке опытов или экспериментов и т.д.	Дидактический и лекционный материалы, тематика(или методики) опытнической или исследовательской работы и т.д.
Модуль «Основы программирования и алгоритмизации»				
	Введение в программирование	The Official Guide to Creating Your Own Video Games	Инструкция по технике безопасности в процессе работы за ноутбуками	Ноутбук – 10шт., ПО
Модуль «Программирование микроконтроллеров и разработка игр»				

	Программирование микроконтроллеров	The Internet of Things for Education: A Brief Guide	Инструкция по технике безопасности в процессе работы за ноутбуками	ПО Arduino IDE, The Internet of Things 5 шт, набор «Матрешка Z» -5 шт, Ноутбук – 10шт
	Разработка игр на Unity	Обучающий курс https://learn.unity.com/	Инструкция по технике безопасности в процессе работы за ноутбуками	ПО Unity 3D, Ноутбук – 10шт
	Проектная деятельность	Обучающий курс https://learn.unity.com/	Инструкция по технике безопасности в процессе работы за ноутбуками	Ноутбук – 10шт, ПО
Модуль «Веб-технологии и веб-разработка»				
	язык HTML и каскадная таблица стилей	Обучающий курс Learn to Code HTML & CSS	Инструкция по технике безопасности в процессе работы за ноутбуками	Ноутбук – 10шт, ПО
	Проектная деятельность	Обучающий курс Learn to Code HTML & CSS	Инструкция по технике безопасности в процессе работы за ноутбуками	Ноутбук – 10шт, ПО

5. Ресурсное обеспечение программы

Информационно-методическое обеспечение включает в себя перечень:

- дидактических игр, пособий, материалов;
- методической продукции по разделам программы;
- учебных и информационных ресурсов: учебно-методический комплекс (учебники, кассеты, рабочие тетради и т.п.); разработки из опыта работы педагога (сценарии, игры и т.д.).

Применяемые технологии и средства обучения и воспитания:

В образовательном процессе используются элементы педагогических технологий: технология развивающего обучения, коллективного взаимообучения, проектной деятельности, модульного обучения, игровые технологии, технология дифференцированного обучения, проблемно-поисковая технология и др.

Средства обучения

визуальные: таблицы, карты, натуральные объекты

аудиальные: радио, магнитофон, музыкальные инструменты

аудиовизуальные: фильмы, телевидение, видеосюжеты

Материально-техническое обеспечение

Занятия по программе проводятся на базе ГБОУ лицей г.Сызрани. Занятия организуются в кабинете №12 соответствующих требованиям СанПиН и техники безопасности.

В кабинетах имеется следующее учебное оборудование:

- мультимедийное оборудование;
- ноутбук;
- тренажеры.

6. СПИСОК ЛИТЕРАТУРЫ

Литература, используемая педагогом дополнительного образования

1. Блум Д. Изучаем Arduino: инструменты и методы технического [пер. с англ.]. — СПб.: БХВ-Петербург, 2017. — 336 с.
2. Джонс М. Х., Электроника — практический курс [пер. с англ.]. — М.: Техносфера, 2016. — 512 с.
3. Колисниченко Д.Н. PHP и MySQL. Разработка Web-приложений. — 5е изд., перераб. и доп.- СПб.: БХВ-Петербург, 2021. — 592с.
4. Кондратенко С.В. , Новиков Ю.В., — М.: Национальный Открытый университет "Интуит", 2018. — 407с.
5. Монк С., Програмируем Arduino. Основы работы со скетчами [пер. с англ.]. — СПб.: Питер, 2020. — 176 с.
6. Никсон Р., Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 [пер. с англ.]. — СПб.: Питер, 2020. — 688с.

7. Олифер В.Г., Олифер Н.А., Компьютерные сети. Принципы, технологии, протоколы. Учебник для вузов [пер. с англ.]. 5-ое изд. — СПб.: Питер, 2019. — 992 с.
8. Партыка Т. Л., Попов И. И., Периферийные устройства вычислительной техники: учеб. пособие / под ред. - 3-е изд., испр. и доп. — М. : ФОРУМ, 2020. — 432 с.
9. Робачевский А.М., Немнюгин С.А., Стесик О.Л., Операционная система UNIX. — 2-е изд., перераб. и доп. — СПб.. БХВ-Петербург, 2017. — 656 с.
- 10.Соммер У., Программирование микроконтроллерных плат Arduino/Freduino [пер. с нем.]. — СПб.: БХВ-Петербург, 2018. —244 с.
- 11.Таненбаум Э., Бос Х. ,Современные операционные системы [пер. с англ.]. 4- еизд.— СПб.: Питер, 2019. — 1120 с.
- 12.Хоровиц П., Хилл У. Искусство схемотехники [пер. с англ.]. — 7-е изд, пер. — М.: Бином, 2016. — 704 с.
- 13.Хоровиц П., Хилл У. ,Искусство схемотехники [пер. с англ.]. — 7-е изд, пер. — М.: Бином, 2016. — 704 с.
- 14.Язык программирования C++. Базовый курс, 5-е изд. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2017. – 1120 с. : ил. – Парал. тит. англ.

Литература, рекомендованная для чтения учащимся

1. Введение в практическую электронику (<https://universarium.org/course/738>).
2. Введение в программирование (C++) (<https://stepik.org/course/363/>)
3. Джереми Блум Изучаем Arduino. <http://radiohata.ru/arduino/162-dzheremiblum-izuchaem-arduino-instrumenty-i-metody-tehnicheskogo-volshebstva.html>
4. Знакомство с цифровой электроникой (<https://universarium.org/course/496>)
5. Основы HTML и CSS (<https://www.coursera.org/learn/snovy-html-i-css>).
6. Пятибратов А.П. , Гудыно Л.П., Кириченко А.А., Вычислительные системы, сети и телекоммуникации. Учебное пособие - 2019 - books.google.com
7. Строим роботов и другие устройства на Arduino. От светофора до 3Dпринтера (<https://www.coursera.org/learn/roboty-arduino>).
8. Таненбаум Э.С. Архитектура компьютера[пер. с англ.] — 2019
9. Энциклопедический словарь юного техника. - М., «Педагогика», 2020.

Дополнительные интернет-ресурсы

1. www.school.edu.ru/int
2. <http://www.int-edu.ru>
3. <http://www.samsung.com/ru/iotacademy/>
4. <http://www.samsung.com/ru/iotacademy/equip/>
5. <http://radiohata.ru/arduino/162-dzheremi-blumizuchaem-arduino-instrumenty-i-metody-tehnicheskogo-volshebstva.html>
6. <http://avidreaders.ru/book/arduino-i-raspberry-pi-v-proektah.html>

Литература, рекомендованная для чтения родителям

1. Владимир Войков., – М.: Фонд новых форм развития образования, 2017 –128 с.
2. Том Иго., Arduino, датчики и сети для связи устройств. СПб.: БХВ-Петербург, 2020. – 544с.
3. Хофман Михаэль. Микроконтроллеры для начинающих, СПб.: БХВ-Петербург, 2019. – 304с.

Дополнительная литература:

1. Arduino/Freduino, СПб.: БХВ-Петербург, 2019. – 256 с.
2. Программирование Ардуино. Режим доступа: [http://www.http://ard uino.ru/Reference](http://www.http://arduino.ru/Reference).
3. Теоретический материал по работе с датчиками компании «Амперка». – Режим доступа: <http://wiki.amperka.ru/>
4. Теоретический материал по аквариумистике. – Режим доступа: <http://akvariumnyerybki.ru/>

КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Дата проведения занятия	Время проведения занятия	Кол-во часов	Тема занятия	Форма занятия	Место проведения	Форма контроля
Модуль 1 «Основы программирования и алгоритмизации»							
1.			1	Вводное занятие. Техника безопасности. Алгоритм и его формальное исполнение	Вводное занятие	12 каб	Опрос
2.			1	Вводное занятие. Техника безопасности. Алгоритм и его формальное исполнение	Практикум	12 каб	Наблюдение, беседа
3.			1	Принципы разработки алгоритмов и программ	Вводное занятие	12 каб	беседа
4.			1	Изучение и прохождение Algotica Iterations	Практикум	12 каб	Тренинг, игра
5			1	Кодирование основных типов алгоритмических структур	Вводное занятие	12 каб	Наблюдение
6.			1	Линейный алгоритм,	Вводное	12 каб	Наблюдение

				организация ветвления и алгоритмическая структура циклов	занятие		
7.			1	Линейный алгоритм, организация ветвления и алгоритмическая структура циклов	Практикум	12 каб	Наблюдение
8.			1	Блок-схемы и графы	Вводное занятие	12 каб	Беседа
9.			1	Языки программирования	Вводное занятие	12 каб	Беседа
10.			1	Языки программирования	Вводное занятие	12 каб	Беседа
11.			1	Операторы ветвления	Вводное занятие	12 каб	Наблюдение
12.			1	Операторы повторений	Вводное занятие	12 каб	Наблюдение
13.			1	Операторы повторений	Практикум	12 каб	Наблюдение
14.			1	Изучение и прохождение Kodu game lab	Практикум	12 каб	Тренинг, игра
15.			1	Функции и процедуры	Вводное занятие	12 каб	Упражнение
16.			1	Функции и процедуры	Практикум	12 каб	Упражнение
17.			1	Массивы	Вводное занятие	12 каб	Наблюдение
18.			1	Одномерные массивы	Практикум	12 каб	Упражнение

19.			1	Двумерные массивы	Практикум	12 каб	Упражнение
20.			1	Классы	Вводное занятие	12 каб	Беседа
21.			1	Классы	Практикум	12 каб	Беседа
22.			1	Запись алгоритмов на языках программирования и решение задач	Вводное занятие	12 каб	Тестирование
23			1	Что такое микроконтроллер?	Вводное занятие	12 каб	Беседа
24			1	Что такое микроконтроллер?	Практикум	12 каб	Беседа
25			1	Обзор языка программирования Arduino	Вводное занятие	12 каб	Беседа
26			1	Простейшие компоненты: кнопка, светодиод. Управление яркостью.	Вводное занятие	12 каб	Наблюдение
27			1	Простейшие компоненты: кнопка, светодиод. Управление яркостью.	Практикум	12 каб	Наблюдение
28			1	Сенсоры. Протоколы обмена данными, получение информации с сенсоров	Вводное занятие	12 каб	Наблюдение

29			1	Вывод информации. LCD - дисплей. Соединение с компьютером	Вводное занятие	12 каб	Наблюдение
30			1	Вывод информации. LCD - дисплей. Соединение с компьютером	Практикум	12 каб	Наблюдение
31			1	Как сделать свой дом умным?	Вводное занятие	12 каб	Беседа
32			1	Центры управления	Вводное занятие	12 каб	Наблюдение
33			1	Центры управления	Практикум	12 каб	Наблюдение
34			1	Датчики	Практикум	12 каб	Наблюдение
35			1	Системы видеонаблюдения	Практикум	12 каб	Упражнение
36			1	Готовые наборы, умная техника	Вводное занятие	12 каб	Тестирование
Модуль 2 «Программирование и разработка игр»							
37			1	2D и 3D моделирование	Вводное занятие	12 каб	Беседа
38			1	2D и 3D моделирование	Вводное занятие	12 каб	Беседа
39			1	3D моделирование в Blender	Практикум	12 каб	Упражнение
40			1	Работа с анимацией и текстурами	Вводное занятие	12 каб	Беседа

41			1	Работа с анимацией и текстурами	Практикум	12 каб	Беседа
42			1	Создание текстур при помощи Gimp	Практикум	12 каб	Упражнение
43			1	Звук и озвучивание	Вводное занятие	12 каб	Наблюдение
44			1	Звук и озвучивание	Вводное занятие	12 каб	Наблюдение
45			1	Работа со звуком в Audacity	Практикум	12 каб	Упражнение
46			1	Редакторы кода и скрипты	Вводное занятие	12 каб	Наблюдение
47			1	Редакторы кода и скрипты	Вводное занятие	12 каб	Наблюдение
48			1	MonoDevelop, Microsoft Visual Studio, Notepad ++	Практикум	12 каб	Беседа
49			1	Межплатформенная среда разработки компьютерных игр Unity	Вводное занятие	12 каб	Тестировани
50			1	Межплатформенная среда разработки компьютерных игр Unity	Практикум	12 каб	Тестировани
51			1	Графика и обработка пошаговых событий	Вводное занятие	12 каб	Беседа
52			1	Проект: «Крестики-нолики»	Практикум	12 каб	Упражнение, защита проекта

53			1	Проект: «Крестики-нолики»	Практикум	12 каб	Упражнение, защита проекта
54			1	Проект: «Четыре в ряд»	Практикум	12 каб	Упражнение, защита проекта
55			1	Синхронизация, движение, столкновение и анимация в реальном времени	Вводное занятие	12 каб	Наблюдение
56			1	Проект: «Space Invaders»	Практикум	12 каб	Упражнение, защита проекта
57			1	Проект: «Тетрис»	Практикум	12 каб	Упражнение, защита проекта
58			1	Проект: «Тетрис»	Практикум	12 каб	Упражнение, защита проекта
59			1	Переинженеринг	Вводное занятие	12 каб	Беседа
60			1	Планирование, анализ столкновений, физика, искусственный интеллект	Вводное занятие	12 каб	Наблюдение
61			1	Проект: «Бильярд»	Практикум	12 каб	Упражнение, защита проекта
62			1	Проект: «Рас Ман»	Практикум	12 каб	Упражнение, защита проекта
63			1	Проект: «Рас Ман»	Практикум	12 каб	Упражнение, защита проекта
64			1	Платформеры, Action/Adventure, RPG	Вводное занятие	12 каб	Беседа

65			1	Проект: «Never Alone»	Практикум	12 каб	Упражнение, защита проекта
66			1	Проект: «Never Alone»	Практикум	12 каб	Упражнение, защита проекта
67			1	Проект: «Ori»	Практикум	12 каб	Упражнение, защита проекта
68			1	Проект: «LIMBO»	Практикум	12 каб	Упражнение, защита проекта
69			1	Проект: «LIMBO»	Практикум	12 каб	Упражнение, защита проекта
70			1	Проект: «Child of Light»	Практикум	12 каб	Упражнение, защита проекта
71			1	3D игры	Вводное занятие	12 каб	Беседа
72			1	Проект: «Cubium Dreams»	Практикум	12 каб	Тестирование, защита проекта
Модуль «Веб-технологии и веб-разработка»							
73			1	Передача информации	Практикум	12 каб	Беседа
74			1	Передача информации	Практикум	12 каб	Беседа
75			1	Базовые понятия компьютерных сетей	Практикум	12 каб	Беседа
76			1	Локальные и глобальные сети. Конфигурирование компьютеров в локальной сети	Вводное занятие	12 каб	Беседа

77			1	Локальные и глобальные сети. Конфигурирование компьютеров в локальной сети	Практикум	12 каб	Беседа
78			1	Доменная система и протоколы передачи данных	Практикум	12 каб	Беседа
79			1	Всемирная глобальная паутина. Язык разметки гипертекстовых страниц HTML: обзор возможностей	Вводное занятие	12 каб	Беседа
80			1	Всемирная глобальная паутина. Язык разметки гипертекстовых страниц HTML: обзор возможностей	Практикум	12 каб	Беседа
81			1	Структура HTML-документа	Вводное занятие	12 каб	Наблюдение
82			1	Модульность и виды подходов к верстке	Вводное занятие	12 каб	Наблюдение
83			1	Модульность и виды подходов к верстке	Практикум	12 каб	Наблюдение
84			1	Табличная верстка	Вводное занятие	12 каб	Упражнение
85			1	Верстка слоями	Вводное занятие	12 каб	Упражнение

86			1	Верстка слоями	Практикум	12 каб	Упражнение
87			1	Блочная верстка	Вводное занятие	12 каб	Упражнение
88			1	Верстка фреймами	Вводное занятие	12 каб	Упражнение
89			1	Верстка фреймами	Практикум	12 каб	Упражнение
90			1	Адаптивная верстка	Вводное занятие	12 каб	Беседа
91			1	Возможности и методы HTML. Создание собственных веб-страниц и размещение их на сервере	Практикум	12 каб	Упражнение
92			1	Возможности и методы HTML. Создание собственных веб-страниц и размещение их на сервере	Практикум	12 каб	Упражнение
93			1	Валидность и кроссбраузерность	Вводное занятие	12 каб	Упражнение
94			1	Понятие каскадной таблицы стилей CSS, применение и написание	Вводное занятие	12 каб	Упражнение
95			1	Понятие каскадной таблицы стилей	Практикум	12 каб	Тестирование

				CSS, применение и написание			
96			1	Целеполагание, целевая аудитория и функционал сайта (Разбор сайтов Самарской области)		12 каб	Беседа
97			1	Содержание и структура сайтов Самарской области	Вводное занятие	12 каб	Беседа
98			1	Содержание и структура сайтов Самарской области	Вводное занятие	12 каб	Беседа
99			1	Дизайнерские идеи и решения. Лучшие сайты Самарской области	Практикум	12 каб	Наблюдение
100			1	Создание персонального сайта	Вводное занятие	12 каб	Упражнение
101			1	Создание персонального сайта	Практикум	12 каб	Упражнение
102			1	Создание личного блога	Практикум	12 каб	Упражнение
103			1	Создание тематического сайта	Практикум	12 каб	Упражнение
104			1	Создание тематического сайта	Практикум	12 каб	Упражнение
105			1	Создание новостного сайта	Практикум	12 каб	Упражнение

106			1	Создание интернет-магазина	Практикум	12 каб	Упражнение
107			1	Создание интернет-магазина	Практикум	12 каб	Упражнение
108			1	Размещение сайта в интернете	Вводное занятие	12 каб	Тестирование, защита сайта

ИТОГО									
-------	--	--	--	--	--	--	--	--	--

- «Высокий» - высокие показатели изменений, знаний в течение учебного года (8-10 баллов);
- «Средний» - средний уровень подготовки и заинтересованности обучающихся (4-7 баллов);
- «Низкий» - низкие показатели изменений, знаний в течение учебного года (1-3 балла).

Образец проекта

Автор Ковалев С.С., педагог дополнительного образования.

Тема: «Разработка платформы перемещения планетохода по экзопланете»

Введение

Люди во все времена устремляли взгляд в небо. Космос всегда являлся загадкой для человека. Несмотря на современные достижения науки, мы до сих пор знаем лишь малую часть всего того, что может предоставить нам вселенная. Сейчас над исследованием космоса работает множество специалистов.

Для изучения космических объектов разрабатываются разные устройства, от телескопов до космических аппаратов. В данной работе разрабатывается платформа перемещения планетохода по поверхности экзопланеты.

На основании выбранной темы, была поставлена следующая цель и необходимые задачи для ее реализации.

Цель проекта: разработать концепцию прототипа исследовательского планетохода для высадки на поверхность экзопланет и создать его.

Задачи:

1. Планетоход должен быть приспособлен для передвижения по различным типам поверхностей.
2. На планетоходе должна быть предусмотрена возможность установки дополнительного оборудования.

Учитывая, что планетоходы могут обладать различной платформой перемещения, к ней имеются необходимые требования:

1. Обеспечение высокой проходимости в различных условиях (например, при вязком или сыпучем грунте).
2. Подвеска также должна иметь возможность преодолевать препятствия сложной формы: высокие подъёмы и склоны, или наоборот — спуски и провалы грунта.

Методы исследования: теоретический анализ литературы, индукция, дедукция, конкретизация, обобщение, классификация, эксперимент, сравнение, анализ полученных данных.

Теоретическая значимость работы: вовлечение обучающихся в тематику освоения космоса, стимулирование к изучению роботехники и программирования.

Практическая значимость работы: освоение моделирования будущего прототипа планетохода, что развивает логическое мышление, а так же визуализацию идеи. Освоение навыков сборки и программирования.

Глава I

Планетоход — это транспортное средство, предназначенное для передвижения по поверхности других планет, спутников или астероидов. Такие аппараты должны выдерживать перегрузки, существенные перепады температур и воздействие космической радиации, сохраняя работоспособность на протяжении всего срока эксплуатации. Планетоходы позволяют исследователям получать важнейшие научные сведения.

Основными факторами, определяющими возможность и эффективность использования планетоходов, являются условия поверхности планеты или спутника, строение, рельеф и механические свойства верхних слоев грунта.

Едва ли не главной задачей, которую приходится решать конструкторам планетоходов, является выбор типа движителя планетохода. Движитель — устройство, которое преобразует энергию двигателя либо другого внешнего источника в полезную работу, при помощи взаимодействий с окружающей средой, в результате которой происходит перемещение транспортного средства. Так, например, движитель автомобиля — колёса.

Выбранный тип движителя должен обеспечить максимально возможную проходимость по неорганизованной поверхности, информация о которой, как правило, отсутствует. Поскольку планетоходы вынуждены работать только в условиях бездорожья, главная функция их движителя должна выполняться в условиях взаимодействия движителя разнообразием грунтов и рельефа.

Можно выделить следующие типы движителей для планетохода:

1. Колесный движитель.

Колесные планетоходы обладают большим коэффициентом полезного действия движителя и широким набором типов грунтов, на которых возможна эксплуатация планетохода; обеспечивают меньшие динамические нагрузки при преодолении выступающих препятствий; обладают более простой конструкцией и имеют больший срок службы в сравнении с гусеничным движителем. При этом планетоходы с колесным движителем обладают примерно равной с гусеничными маневренностью и поворотливостью.

Колесные планетоходы обладают большим коэффициентом полезного действия движителя и широким набором типов грунтов, на которых возможна эксплуатация планетохода; обеспечивают возможность создания тягового усилия в ведущем и тормозном режимах работы; обеспечивают меньшие динамические нагрузки при преодолении выступающих препятствий, возможность прогнозирования недопустимого буксования, преодоление выступающих препятствий больших размеров; обладают более простой конструкцией; позволяют отключать часть приводов колес и тем самым выводить электропривод на оптимальный по КПД режим работы; При этом планетоходы с колесным движителем обладают примерно равной с гусеничными маневренностью и поворотливостью и обеспечивают примерно равный дорожный просвет.

2. Гусеничный движитель

Преимущества гусеничных планетоходов:(это надо записать попроще) более низкое удельное давление на грунт, более высокие показатели по запасу силы тяги на слабонесущих грунтах, более низкая нагрузка на катки, меньшая масса при равной проходимости. Характерным свойством гусеничного движителя является получение высоких тяговых показателей даже при малых значениях буксования. Причем тяговое усилие мало изменяется с ростом буксования.

Однако это качество имеет и отрицательную сторону - незначительное изменение внешних условий может привести к полному буксованию и застреванию машины. К тому же, практика работы земных машин показывает, что гусеничный движитель обладает таким существенным недостатком, как расклинка его камнями и сброс гусеницы.

3. Колесно-шагающий движитель

В основном данные типы работают в режиме колесного качения. На трудных участках, например, когда движение происходит на подъеме по сыпучему грунту, колесный движитель быстро зароется в грунт. Для того, чтобы этого не случилось, передние колеса, вращаясь, одновременно выдвигаются вперед, затем к ним подтягиваются остальные колеса (одновременно или по очереди). Проходимость машины с колесно-шагающим движителем значительно повышается. Известны схемы, при которых «шагание» колес происходит без остановки планетохода.

Все те немногочисленные устройства, которые реально перемещались по другим планетам колесные. Почти все небесные тела доступные нам для исследования имеют твердую поверхность с множеством относительно ровных участков. Там нет болот, зыбучих песков, леса и растительности, которые могли бы потребовать гусениц или шагающих движителей.

Колеса - очень экономичный вид движителя. Чтобы прокручивать гусеницы, нужна куда большая мощность. Даже поломка даже нескольких колес совсем не ставит под угрозу выполнение задачи, в отличие от порванной гусеницы. Сравнительно прост и привод колес от электромоторов, легко обеспечивать разворот.

Так же для управления скоростью моторов будет использоваться ШИМ сигнал. Цифровые выводы Arduino могут выдавать только два значения: логический 0 (LOW, низкий уровень) и логическую 1 (HIGH, высокий). На то они и цифровые. Но есть у Ардуино «особые» выводы, которые обозначаются PWM. Их иногда обозначают волнистой чертой "~" или обводят кружочками или ещё как-то выделяют среди прочих. PWM расшифровывается как Pulse-width modulation или широтно-импульсная модуляция, ШИМ.

Широтно-импульсно модулированный сигнал – это импульсный сигнал постоянной частоты, но переменной скважности (соотношение длительности импульса и периода его следования). Из-за того, что большинство физических процессов в природе имеют инерцию, то резкие перепады напряжения от 1 к 0 будут сглаживаться, принимая некоторое среднее значение. С помощью задания скважности можно менять среднее напряжение на выходе ШИМ.

Если скважность равняется 100%, то всё время на цифровом выходе Arduino будет напряжение логическая "1" или 5 вольт. Если задать скважность 50%, то половину времени на выходе будет логическая "1", а половину – логический "0", и среднее напряжение будет равняться 2,5 вольтам (рис.1.).

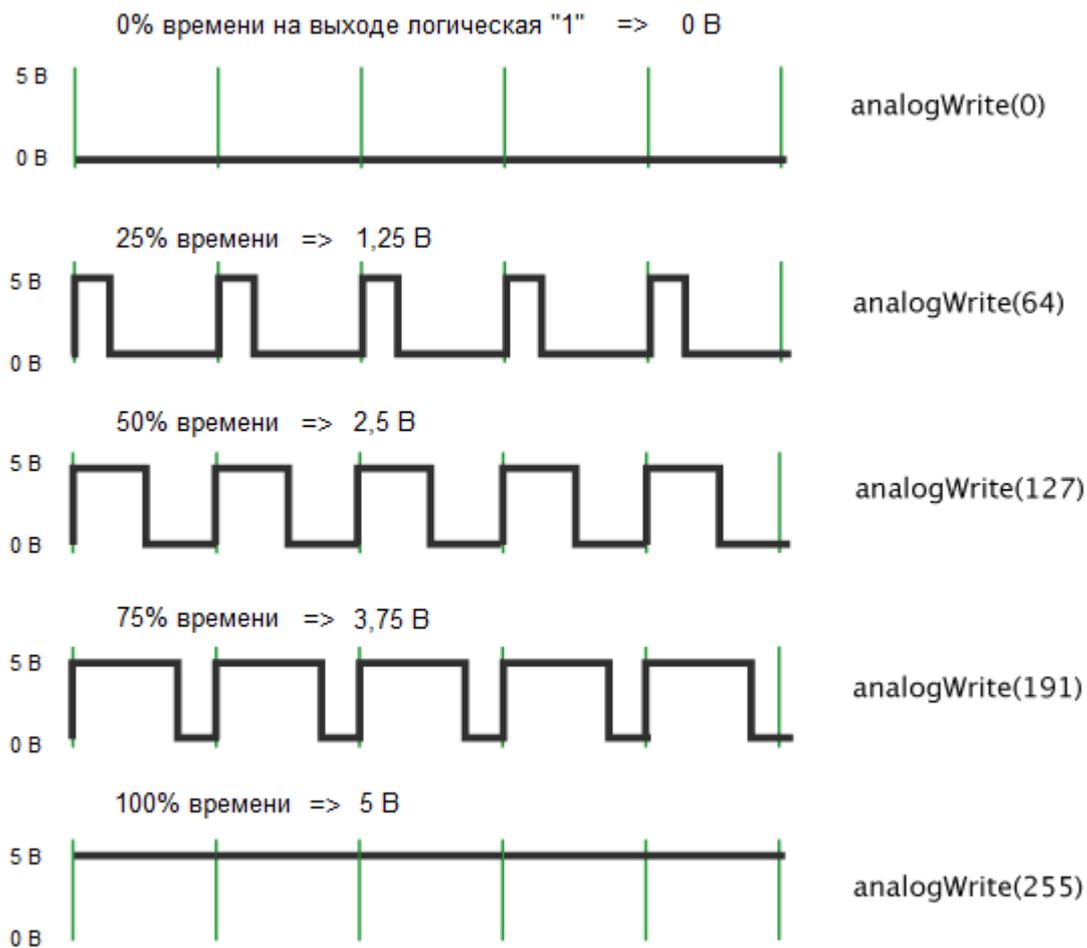


Рис.1. Принцип работы широтно-импульсной модуляции

Глава II

Для разработки и сборки макета планетохода используется наборы с различными конструктивными элементами, такие как TETRIX и ТЕХНОЛАБ. Данные наборы содержат металлические детали и крепежные элементы, позволяющие разрабатывать на их базе сложные и прочные механизмы для оценки эффективности собранной модели и обнаружении на раннем этапе сборки недочетов. Такой подход позволяет получить качественную модель для ее дальнейшего использования при разработке чертежа и дальнейшей сборки готового прототипа. При этом собранный макет позволяет в дальнейшем произвести доработку, посмотреть полученный результат и так же перенести улучшенный узел или деталь на основной прототип (рис. 2.).



Рис. 2. Металлические элементы в процессе сборки макета

Для организации контроля и управления планетоходом будет применяться входящий в состав набора ТЕХНОЛАБ программируемый контроллер, преемственный контроллерам Arduino. Данные контроллеры являются программно-аппаратными комплексами на базе платы ввода/вывода с наличием микроконтроллеров семейства AVR (Atmel), а также специальной среды программирования на основе языка C/C++.

Главным достоинством контроллеров Arduino является, что каждый контроллер семейства обладает унифицированным расположением портов и единым стандартом разъемов, используемой схемой электропитания, позволяющей использовать большинство распространенных аккумуляторных батарей, встроенными средствами программирования. Так же данные контроллеры обладают многообразием дополнительных плат расширения, обеспечивающие подключения к базовой плате различные привода, устройства связи и мультимедийные системы.

Программируемый контроллер является преемственным по отношению к оригинальной плате Arduino Mega 2560, но в отличие от неё обладает встроенной системой контроля заряда внешней

аккумуляторной батареи, интегрированной силовой частью для подключения внешних двигателей постоянного тока, встроенным модулем беспроводной связи Bluetooth, а также полной совместимостью с комплектующими набора (рис. 3.).

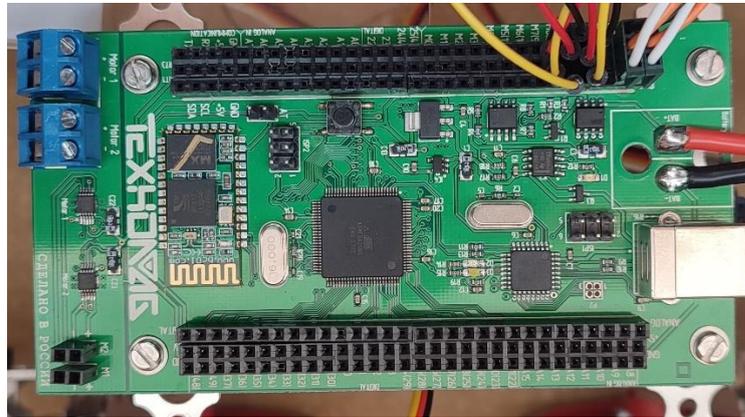


Рис. 3. Программируемый контроллер

Для организации перемещения планетохода будут использоваться следующие устройства: привода 2-Wire Motor 269 w, TETRIX 44298 и Motor Controller 29.

Привод 2-Wire Motor 269 w (рис. 4.) является двигателем коллекторным постоянного тока с редуктором, состоящим из металлических зубчатых передач. В комплект к двигателю входит набор винтов для его крепления и вал для передачи вращения от двигателя к исполнительному механизму.



Рис. 4. Привод 2-Wire Motor 269 w

Данный привод является наиболее распространённым в роботах, поскольку 2-Wire Motor 269 w обладает наиболее оптимальным соотношением технических характеристик и габаритов, а также может быть оснащён внешним инкрементным энкодером. При стандартном для комплектующих МЕХ питании в 7,2В данный привод развивает момент 0,97 Нм при скорости до 100 об/мин. Поскольку в процессе работы привод потребляет ток величиной до 2,6 А, для подключения его к программируемому контроллеру необходимо применять силовую схему. Так же в работе используется TETRIX 44298, данный мотор

оснащен двухпроводным двигателем главной передачи постоянного тока, встроенным в сервопривод (рис. 5.).



Рис. 5. TETRIX 44298

Устройство Motor Controller 29 (рис. 6.) представляет собой силовую схему для управления приводами на базе двигателей постоянного тока. С помощью Motor Controller 29 можно подключать к программному контроллеру привода с максимальным рабочим током не превышающим 3А. При подключении стандартного привода к программируемому контроллеру становится возможным управлять скоростью его вращения с помощью программируемого значения выходного ШИМ сигнала контроллера.



Рис. 6. Motor Controller 29

Для подключения между собой привода и драйвера двигателя необходимо соединить с помощью двухпроводного разъема.

На последнем прототипе планетохода применяется 4 привода с набора ТЕХНОЛАБ и 4 электродвигателя постоянного тока TETRIX 44298 со скобой крепления сервоприводов 40232 из набора TETRIX. Привода TETRIX 44298 находятся во внутренней части планетохода, привода 2-Wire Motor 269 располагаются во внешней части планетохода, т.к. колеса используемые из набора ТКХНОЛАБ имеют больший диаметр чем колеса из набора TETRIX. Привода разделены следующим образом, левая и правая часть не зависит друг от друга. Каждая сторона контролируется отдельным драйвером, моторы подключены

параллельно, что позволяет в случае неисправности одного из электродвигателей продолжить перемещение планетохода по поверхности экзопланеты.

В качестве источника питания выступает аккумуляторная батарея (рис. 7.) мощностью 7,2В.



Рис. 7. Аккумуляторная батарея

В качестве среды разработки будет применяться Arduino IDE (рис. 8.), данная IDE является интегрированной средой разработки для Windows, MacOS и Linux, разработанная на Си и С ++, предназначенная для создания и загрузки программ на Arduino-совместимые платы, а также на платы других производителей.

Программирование микроконтроллеров Arduino осуществляется на языке программирования С++. Этот язык является низкоуровневым, поэтому считается сложным и имеет высокий порог вхождения. Но для программирования Arduino используется упрощенная версия этого языка программирования. Так же для упрощения разработки прошивок существует множество функций, классов, методов и библиотек. Благодаря этому работать с этими микроконтроллерами очень удобно и легко.

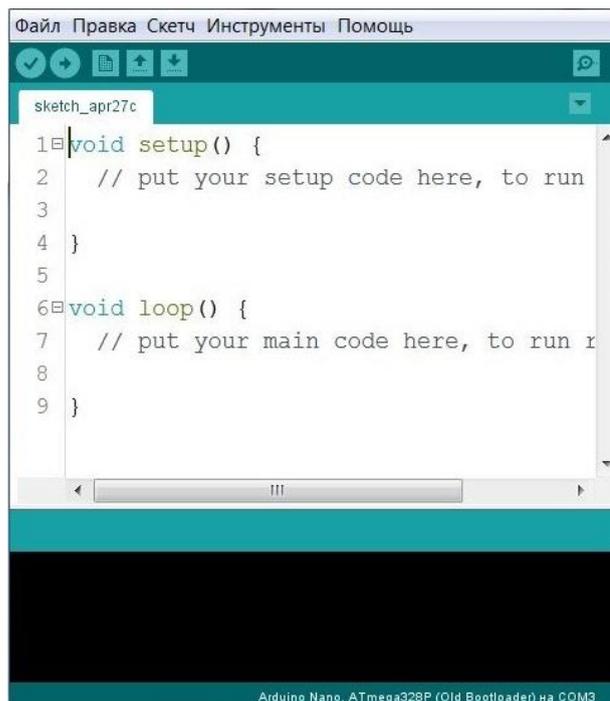


Рис. 8. Среда разработки Arduino IDE

В качестве основной идеи была взята концепция, разработанная в лаборатории реактивного движения NASA подвеска типа Rocker-Bogie. Bogie - система “балансир-тележка” (рис. 9.). Конструкция имеет массу достоинств — жесткая, без упругих элементов подвеска обеспечивает практически постоянный контакт с поверхностью всех шести колес и уменьшает наклон корпуса. Типовым решением также является поворот крайних колес, среднее поворачиваться не может.

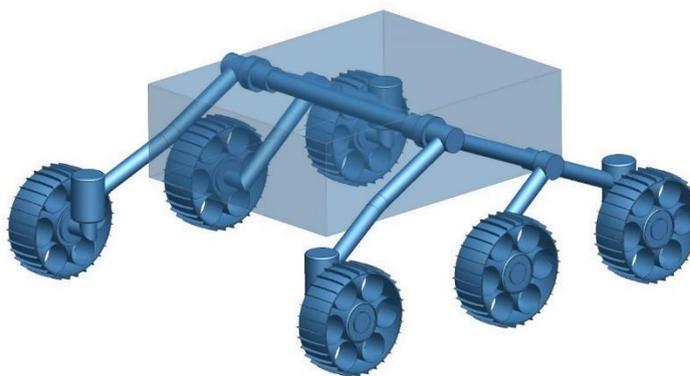


Рис. 9. Подвеска типа Rocker-Bogie

Подвеска типа Rocker-Bogie полностью не удовлетворяет дальнейшему варианту оснастки планетохода. В связи с тем, что в качестве дополнительного оборудования планируется установить устройство по забору грунта, необходимо иметь жесткий каркас.

Подвеска в конечном варианте получила изменение, путем добавления дополнительных рычагов на ведущей движущей части планетохода (рис. 10-11.). Благодаря этому сохраняется проходимость и сцепление максимального количества колес с поверхностью. Данный вариант улучшения позволяет

максимально исключить холостую работу одного из двигателей. При этом при преодолении препятствия снижается вероятность вибрационных воздействий на основное оборудование, что позволяет исключить возможные повреждения или смещения элементов.

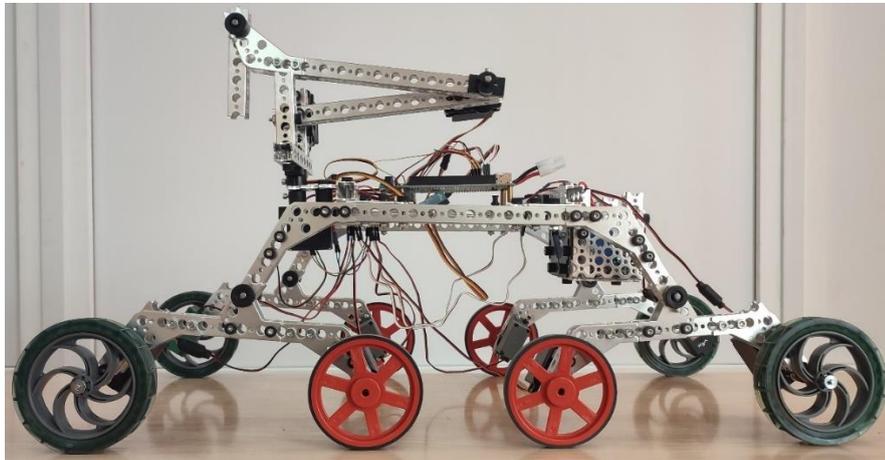


Рис. 10. Планетоход – вид сбоку

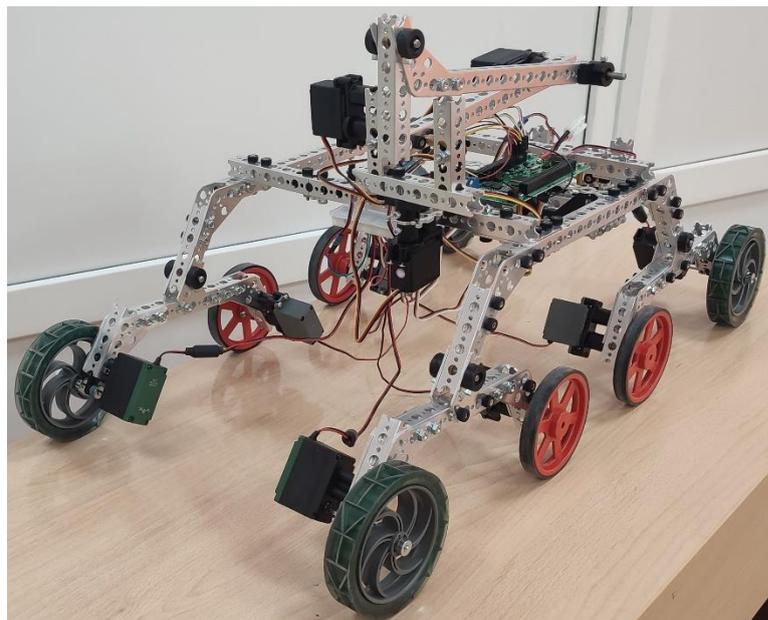


Рис. 11. Планетоход – общий вид

Заключение

В результате выполнения научной работы были достигнуты следующие результаты, сборка модели планетохода с подвеской обеспечивающая уменьшения вероятности возникновения вибраций, при этом сохраняя возможность преодоления различных препятствий. Так же были выполнены поставленные задачи:

1. Планетоход должен быть приспособлен для передвижения по различным типам поверхностей.
2. На планетоходе должна быть предусмотрена возможность установки дополнительного оборудования.
3. Обеспечение высокой проходимости в различных условиях (например, при вязком или сыпучем грунте).
4. Подвеска также должна иметь возможность преодолевать препятствия сложной формы: высокие подьёмы и склоны, или наоборот — спуски и провалы грунта.

При выполнении работы были получены знания по различным сферам необходимым для реализации проекта. Успешно повторены и закреплены навыки программирования микроконтроллеров, силовой части планетохода.

Список литературы

1. <http://www.c-cpp.ru/books/>
2. <http://wiki.amperka.ru/>
3. <https://habr.com/ru/company/timeweb/blog/645251/>
4. <https://ijisrt.com/wp-content/uploads/2017/05/Design-of-Rocker-Bogie-Mechanism-1.pdf>
5. <https://ijari.org/assets/papers/8/3/IJARI-ME-20-09-113.pdf>

