

Государственное бюджетное общеобразовательное учреждение средняя общеобразовательная школа им. В.С. Чекмасова с. Большое Микушкино муниципального района Исаклинский Самарской области

Тема: Основы программирования проекта «Умный дом»
на платформе Arduino.

Индивидуальный проект
обучающегося 9 класса
Семина Алексея

Руководитель: Верязов
Дмитрий Михайлович,
учитель технологии

2022-2023 учебный год

Содержание:

1. Введение.....	3 стр.
2. Актуальность.....	3 стр.
3. Цель и задачи.....	3 стр.
4. Устойчивость и распространение.....	5 стр.
5. Часть 1.....	6 стр.
1) Искусство создания роботов.....	6 стр.
2) История робототехники.....	6 стр.
3) Робототехника в современном мире. Её главные цели и задачи. Применение робототехники.....	8 стр.
4) Важность изучения робототехники.....	11 стр.
6. Часть 2.....	12 стр.
5) Ардуино.....	12 стр.
6) Внешний вид.....	12 стр.
7) Программирование.....	15 стр.
I. 1 этап.....	15 стр.
II. 2 этап.....	16 стр.
III. 3 этап.....	18 стр.
IV. 4 этап.....	19 стр.
7. Вывод.....	22 стр.
8. Использованная литература.....	23 стр.
9. Приложение.....	24 стр.

Введение:

Тема: Основы робототехники на Arduino.

Актуальность: В современном мире практически каждая техническая специальность связана с робототехникой. Мир роботизируется, множество людских задач выполняют машины. Из-за этого в будущем ученики, обучающиеся на технических профилях, столкнутся с робототехникой на следующем этапе обучения. Поэтому знакомство с основами робототехники даёт возможность учащимся более качественно подготовиться к следующему шагу в жизни, научится начальному уровню программирования, сборке различных простейших роботов, углубить свои знания в физике по теме «Электричество».

Проблема: К сожалению, в настоящее время в нашем городе нет дополнительных курсов по изучению Arduino. Поэтому моё обучение является самостоятельным. Сложность сборки и программирования являются главными проблемами в моей работе.

Цель: Научиться основам робототехники на Arduino. Создать робота и запрограммировать его на определённую задачу.

Объект: Электроника, механика, информатика, робототехника

Продукт: Робот, презентация.

Задачи:

1. Знакомство с робототехникой (где применяется, что в себя включает)
2. Знакомство с Arduino
3. Пошаговая сборка робота
4. Пошаговое программирование робота
5. Подведение итогов
6. Создание презентации, обобщение всех этапов работы

Результаты решения данных задач приведены в таблице (таблица 1).

Таблица 1. План мероприятий.

Номер	Мероприятие	Сроки	Ожидаемый результат
1.	Знакомство с робототехникой и Arduino. Сбор информации для написания реферата	Октябрь-25 ноября	Достаточное кол-во информации для обработки и создания первых глав печатной работы
2.	Анализ найденной информации. Написание первых глав печатной работы	25 ноября – 20 декабря	Первые, вводные главы реферата(печатной работы)
3.	Сборка робота по инструкции	20 декабря – 1 января	Собранный и полностью функционирующий робот
4.	Точное описание своих действий. Написание ещё одной главы печатной работы «Сборка робота»	1 января – 7 января	Глава печатной работы на тему «Сборка робота»
5.	Программирование робота	7 января – 1 февраля	Полностью готовый для проекта робот.
6.	Написание ещё одной главы на тему «Программирование робота»	1 февраля – 7 февраля	Глава печатной работы на тему «Программирование робота»
7.	Подведение промежуточных итогов. Закончить печатную работу	7 февраля – 24 февраля	Полностью законченная печатная работа
8.	Создание презентации. Обобщение всей работы. Проверка проекта с учителем	24 февраля – 10 марта	Полностью законченный проект, проверенный учителем и готовый к показу перед комиссией

Устойчивость и распространение

Мой проект направлен на популяризацию робототехники, ведь данная наука одна из самых быстро развивающихся в современном мире. Он будет интересен людям разных возрастов. Почти каждый человек хочет сделать что-то классное своими руками. Освоение робототехники на Arduino позволит людям всех возрастов наконец осуществить это желание. Вы интересно проведёте время и сможете создать множество неординарных и забавных роботов, а более серьёзное погружение сможет вас привести к достижению неожиданных для вас высот в любой сфере деятельности!

Arduino одна из самых дешёвых марок в этой области, поэтому люди разных финансовых положений смогут позволить себе данный электронный «конструктор». Самое главное учиться всеми возможными способами, не унывать после неудач, не потерять мотивацию, и тогда у каждого из нас получится стать настоящим робототехником!

1 Часть

Искусство создания роботов

Робототехника — наука, направление прикладной науки, занимающееся проектированием, производством и применением автоматизированных технических систем — роботов. Робот — это программируемое механическое устройство, способное действовать без помощи человека.

В последнее время технический прогресс в современном мире растёт, робототехника становится одной из главных наук. Эта сфера уже достигла больших высот. Техническая революция привела к тому, что роботы уже заняли свою нишу в жизни человека, как в промышленности, так и в быту. Всё больше и больше предприятий автоматизируются по последнему слову техники.

Робототехника - это не самостоятельная отрасль, она объединяет все последние достижения технических, естественных наук и информационных технологий. В робототехнике соединяются механика, система управления и искусственный интеллект, поэтому она является важнейшим направлением научно-технического прогресса. В ней сливаются такие дисциплины как электроника, механика, кибернетика, телемеханика, мехатроника, информатика, радиотехника, электротехника, в результате робототехник, в отличие от узкого специалиста, должен обладать широким кругозором и системным мышлением.

История робототехники

Робототехника и её элементы встречались ещё в древнейшие времена. Рассмотрим поэтапное развитие робототехники:

- Древний мир

Еще до нашей эры Архимед создал механизм «коготь», который опрокидывал римские осадные суда. Герон Александрийский смастерили самоходную тележку, что передвигалась по заданной траектории с помощью системы из тросов и колышков. Деревянный голубь Архита из Тарента запускался в воздух паровой катапультой и мог пролететь до 200 метров.

Изобретения древнейших времен были примитивными, они работали с помощью воды, пара, противовесов, зубчатых колес и рычагов, а в Китае — еще и ртути, и взрывов пороха. Но именно учёные-изобретатели тех времён связали роботостроение с математическими методами.

- Средневековье и эпоха Возрождения

Богослов Альберт Великий, если верить легенде, смастерили андроида-служанку и механическую голову, которая могла разговаривать. Часовщики, как европейские, так и русские, создавали автоматы, в которых фигурки животных, людей и ангелов разыгрывали целые представления.

Но главной и самой запоминающейся фигурой в истории робототехники эпохи Возрождения, конечно, стал Леонардо Да Винчи. Судя по документам, обнаруженным в 1950-е годы, легендарный изобретатель разработал чертеж человекоподобного робота в 1495 году. В схемах был изображен каркас робота, который был запрограммирован выполнять человеческие движения. Он обладал анатомически правильной моделью челюсти и умел садиться, двигать руками и шеей. Записи гласили, что поверх каркаса должна быть надета рыцарская броня. Этот робот-рыцарь являлся первым представителем андроидов.

Это самый длинный период в истории робототехники. В средние века и позже знания тщательно документировались, поэтому до наших дней дошло множество чертежей и описаний. Тогда появились более эффективные пружинный и маятниковый механизмы, а размеры автоматов уменьшились. Эта тенденция сохранилась и в тоже время: каждое новое поколение становится машин меньше, энергию расходует экономнее и работает дольше.

- Новое время

В этот период мастера явили миру поразительные плоды инженерной мысли. Механическая утка Вокансона клевала зерно и даже испражнялась. Андроиды работы Пьера-Жака и Анри Дро не просто двигались, а писали, рисовали и играли музыку.

- XIX век

В позапрошлом веке появился ткацкий станок с перфокартами. Первый шаг к автоматизации промышленности открыл невиданные ранее перспективы робототехники. Электричество дало толчок машиностроению и способствовало появлению первых роботов, в том числе андроидов. Последние с тех пор будоражат умы творческих людей: писателей, режиссеров, художников, которые порой выдают ценные идеи вроде трех законов Азимова.

- Отдельная наука

Только в 20 веке робототехника стала отдельной наукой. Сначала чешский писатель Карел Чапек представил публике пьесу под названием «Р. У. Р.» («Россумские Универсальные Роботы»), откуда и взяло начало слово «робот». Термин «робототехника» впервые был упомянут американским писателем А. Азимовым в 1941 году в научно-фантастическом рассказе «Лжец», повествующем о проблемах позитронных роботов. С этого началась история современной робототехники.

Робототехника в современном мире. Её главные цели и задачи.

Применение робототехники.

Законы робототехники предписывают машинам следующее:

1. Робот не может навредить человеку или позволить нанести ему вред в результате бездействия.
2. Робот обязан подчиняться командам человека за исключением тех, что не соответствуют Первому закону.
3. Робот должен следить за собственной сохранностью, если это не идет вразрез с Первым или Вторым законом.

Применение

Еще в 2007 году основатель компании Microsoft – Билл Гейтс, выпустил статью «Робот в каждом доме», где он размышлял о потенциале роботов и их роли в выполнении бытовых задач и обучении человека.

Основная цель создания роботов – это помочь человеку в выполнении тяжелой, опасной или монотонной работы. Устройства должны освободить человека от неквалифицированного труда. Это позволит человеку

сосредоточиться на разработке уникальных проектов и выполнении задач, где требуется не только точность, но и интуиция, индивидуальный вкус и опыт, а работу — помочь достичь максимальных показателей эффективности и производительности.

В современном мире роботы присутствуют почти в каждой сфере деятельности. Существуют разные системы классификации.

Основная классификация роботов.

1. Классификация роботов по типу управления:

1) Автономные — совершают работу без вмешательства человека. Запрограммированный цикл их действий в зависимости от обстоятельств изменяется. Автономными роботами являются: роботы-пылесосы, газонокосилки, мойщики окон и т. д.

2) Полуавтономные — выполняют задачу без вмешательства человека, по заранее определённому алгоритму.

Хорошим примером полуавтономного робота является сложный подводный робот. Человек контролирует основные движения робота. И в это время бортовой процессор измеряет и реагирует на подводные токи. Дополнительно бортовые датчики могут отслеживать температуру воды, давление и многое другое. Если робот теряет связь с поверхностью, то включается автономная программа и поднимает подводного робота на поверхность.

3) Управляемые — контролируются человеком непосредственно дистанционно. Самыми популярными примерами управляемых роботов являются, радиоуправляемые роботы, манипуляторы и т. д.

2. Классификация по типу позиционирования:

1) Стационарные — монтированные в фундамент, к несущим стенам или потолку по отношению к обслуживаемому оборудованию.

2) Передвижные — способные перемещаться в пространстве с помощью шасси, либо по ограниченной траектории по рельсам или индуктивным и оптическим трассам.

3. Классификация по типу назначения:

1) Промышленные — участвующие в производственном процессе изготовления изделий и деталей. Данный тип роботов находит свое применение, главным образом, на промышленных производствах и в научных лабораториях. Чаще всего под понятием «Промышленный робот» подразумевается система автоматических управляемых при помощи специальных программ манипуляторов, которая предназначена для выполнения операций, связанных с перемещениями различных предметов.

2) Бытовые — предназначены для помощи человеку повседневной жизни. Сейчас бытовые роботы не слишком распространены, однако есть все основания предполагать широкое их распространение в ближайшем будущем. На данном этапе бытовые роботы — чаще всего предназначены для развлекательных целей, но всё большую популярность набирают роботы-уборщики, по своей сути — автоматические пылесосы, способные самостоятельно прибраться в квартире и вернуться на место для подзарядки без участия человека.

3) Медицинские — предназначенные для работы в медицинской отрасли, облегчающие труд врачам и помогающие исключить человеческий фактор.

4) Военные — предназначенные для ведения боевых действий и обороны различных стран мира. К ним можно отнести различные противовоздушные системы, сапёры, военные беспилотники.

4. Классификация по способу передвижения:

1) Подземные — соответственно перемещающиеся под землёй. Это могут быть исследовательские дроны.

2) Подводные — перемещающиеся под водой. Это могут быть подводный батискаф или торпеда.

3) Надводные — перемещающиеся над водой. Это могут быть лодки или катера.

4) Наземные — передвигающиеся по суще. Это самоходные машины на гусеничном или колёсном ходу. Некоторые модели перемещаются при помощи механических ног.

5) Летательные — перемещающиеся по воздуху над землёй. Это так называемые беспилотники и квадрокоптеры.

Это основная классификация роботов, но далеко не вся. Активно развивающаяся робототехника позволяет конструировать огромное множество различных роботов, поэтому просто невозможно перечислить их всех.

Важность изучения робототехники

Как мы уже поняли робототехника — одно из самых популярных, развивающихся направлений как научного изучения, так и практического. Именно она является важнейшей составляющей научно-технической революции настоящего времени. Чем же роботы так важны?

Человек, ограниченный в своих физических возможностях может очень мало. Ещё с древнейших времён люди начали использовать разные приспособления для того, чтобы решать те или иные проблемы. И вот современнейшее приспособление — робот. В предыдущей главе описаны виды роботов и способы их применения.

Но также существует смысл её изучения для каждого отдельно взятого человека. Она подойдёт людям всех возрастов. У младших поколений робототехника развивает логику, структурированное мышление, даёт возможность творить, построить что-то своими руками, что вызывает немалый интерес! Старшее поколение может создать с помощью робототехники различные полезные устройства, например, разработать аналог системы умный дом.

Робототехника — это интересное, востребованное, а главное полезное занятие, способное развить в человеке новые качества, а растущая популярность открывает множество возможностей в современном мире.

2 Часть

Ардуино

Arduino — молодая торговая марка аппаратно-программных средств для построения и прототипирования простых систем, моделей и экспериментов в области электроники, автоматики, автоматизации процессов и робототехники. Она обладает аппаратной и программной частями для простой разработки электроники. Аппаратная часть включает в себя большое количество видов плат Arduino со встроенными программируемыми микроконтроллерами, а также дополнительные модули. Программная часть состоит из среды разработки (программы для написания скетчей и прошивки микроконтроллеров Arduino), упрощенного языка программирования, огромного множества готовых функций и библиотек. История её начинается с начала 2000-х. С помощью Arduino каждый сможет создать своего робота.

Все Arduino платы обладают своими личными характеристиками. В своём проекте я использовал Arduino Uno. Большинство программистов начинают именно с платы UNO. Контроллер Uno является самым наилучшим вариантом для начала работы с платформой: она имеет удобный размер, достаточно доступна из-за массового выпуска всевозможных клонов, под нее написано огромное количество бесплатных уроков и скетчей.

Внешний вид

Продукт моего проекта — работающий робот, ездащий при помощи пульта управления, своего рода радиоуправляемая машинка. Строение её можно сравнить со строением человека, а именно выделить основные части — «органы».

«Скелетом» моего робота являются две пластиковые пластины с определёнными отверстиями (большинство отверстий не являются заводскими), для пропускания проводов. Данная конструкция очень удобна и надёжна, к ней можно без всяких проблем прикрепить все «внутренности» робота: платы, датчик для приёма инфракрасного излучения, моторы, батарейные отсеки. Для крепления пластин я использовал обычный железный

конструктор, а элементы робота я зафиксировал вручную с помощью маленьких болтов и гаек.

Arduino удобен и в плане подключения. Из всей системы припаять пришлось только питательные элементы. К входам/выходам шли специальные провода с двумя типами насадок, которые в народе называют «папа»/«мама».

Главной частью устройства является центральная плата – Arduino Uno. Она служит «мозгом» всей системы. Именно здесь обрабатывается программа, и отходят сигналы к другим различным частям робота. Плата имеет на борту 6 аналоговых и 14 цифровых входов/выводов общего назначения, кварцевый генератор на 16 МГц, два разъема: силовой и USB, разъем ISCP для внутрисхемного программирования и кнопку горячей перезагрузки устройства. Весь процесс, работающей программы зависит от микроконтроллера, для Arduino Uno стандартным является микроконтроллер ATmega. Принцип его работы прост, есть сигнал - запускай, нет сигнала – отключай.

«Сердце» робота - это, конечно, батареи. В моём устройстве два источника питания, закреплённые в специальных резервуарах – батарейных отсеках. Первая батарея – это крона, она используется для питания моторов, поэтому подключена к драйверу двигателей. Второй источник питания – это четыре обычных батареи каждая на 1,5 вольта, поэтому они идеально питают основную плату Arduino Uno.

Всеми моторами управляет специальный драйвер L298N Motor Drive. Он как «спинной мозг» управляет моторами - «конечностями». Состоит драйвер из двух своеобразных мостов, один для выхода A, второй для выхода B. Такие конструкции часто применяются в электронике и служат для изменения вращения двигателем. Мосты содержат четыре транзистора (ключа) с двигателем в центре, образуя Н-подобную компоновку. Принцип работы прост, при одновременном закрытии двух отдельных транзисторов изменяется полярность напряжения, приложенного к двигателю. Это позволяет изменять направление вращения двигателя.

У модуля L298N, есть два контакта питания на напряжение от 5В до 35В и от 4,5В до 5,5В. Падение напряжения драйвера L298N составляет около 2В, а мои двигатели используют 6В. Простейшая математика и физика дают нам понять, что, чтобы использовать моторы на максимальной мощности питание должно быть не менее 8В. Именно для этого я дополнительно запитал их кроной на 9В. Поскольку напряжение батареи больше, чем совместное падение на драйвере и напряжение на моторах, я смог использовать двигатели на полной мощности.

В сборке различных роботов Arduino применяется огромное количество различных датчиков. С помощью них робототехники могут делать различные измерения (температура, расстояние, кол-во оборотов и т.д.), принимать различные сигналы, считывать параметры всего мира! Датчики, как органы чувств, позволяют технике ориентироваться с окружающим миром и взаимодействовать с ним, почти как человек.

В моём проекте я использовал лишь один датчик, а именно приёмник инфракрасного излучения. Главная особенность этого приёмника в том, что он принимает сигналы лишь на заданной частоте, примерно 38Гц, что позволяет ему не игнорировать остальные сигналы. Датчик имеет три вывода (три ноги). Одна нога ведёт к микроконтроллеру, две другие обеспечивают питание, соответственно земля и фаза.

К данному приёмнику прилагается специальный ИК-пульт. На самом деле для управления могут использоваться, например, телевизионный, но для большей эффективности лучше приобрести данное устройство вместе с приёмником. Принцип его работы такой же, как и у других пультов, посылается инфракрасный сигнал, а датчик его регистрирует с помощью тепловых и фотоэлектрических приемников и специальных фотоматериалов.

Это все части моего робота. Как уже говорилось он состоит из двух «этажей» для большей удобности. На нижней пластине я разместил питание и двигательную часть конструкции: два мотора, дополнительное колесо и драйвер двигателей. В верхней части располагаются основная плата и датчик

для приёма инфракрасного излучения. Вторая плата сверху очень удобна, в дальнейшем на ней можно будет разместить другие датчики.

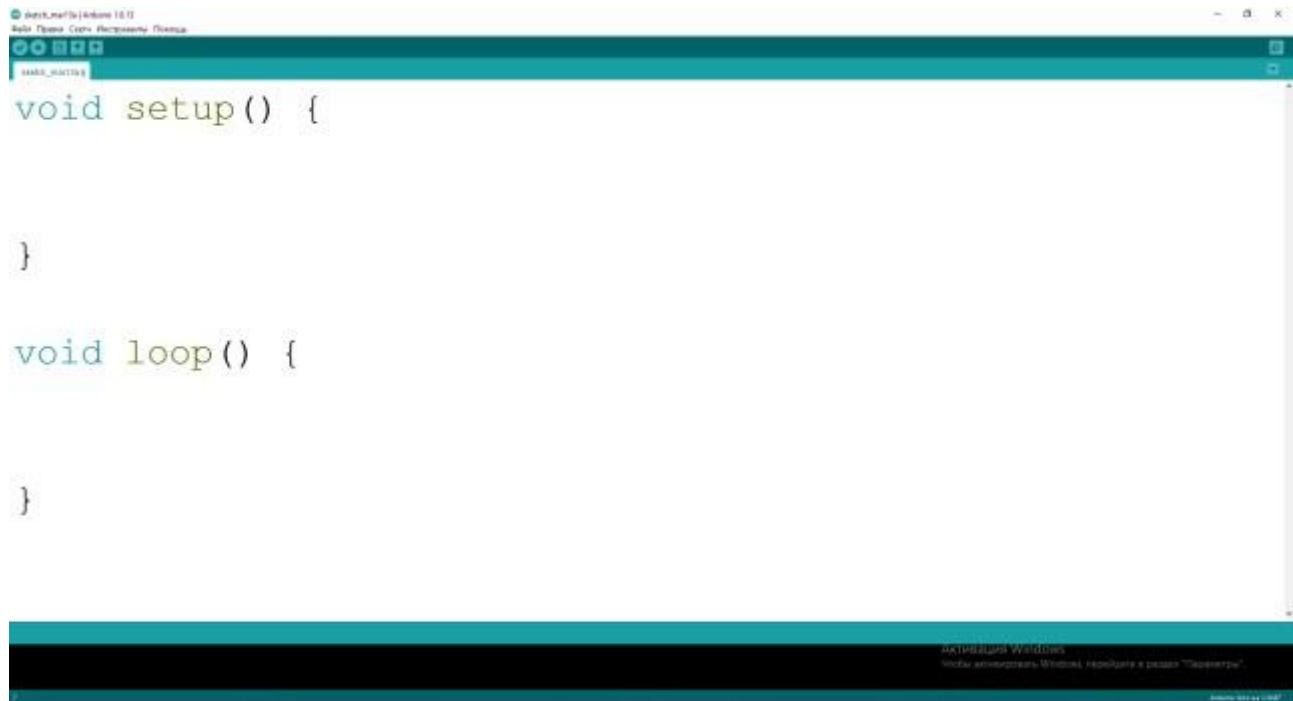
Программирование

Самая сложная, но и самая интересная часть – это написание программы. На эту часть работы ушло больше всего времени, поскольку я ещё неопытный программист. Код строился медленно методом проб и ошибок, а информация с интернета зачастую для меня была непонятна. Чтобы было проще освоится я разбил написание программы на отдельные части.

1 этап

Первым этапом для меня стало изучение программной платформы Arduino. При первоначальном запуске на поле, где мы в дальнейшем будем писать программу, мы видим только две функции void setup() и void loop(), ограниченных фигурными скобками.

Рисунок 1. Начальная среда разработки



```
void setup() {  
  
}  
  
void loop() {  
  
}
```

Сразу же хочется отметить, что фигурными скобками отделяются все структуры программы, циклы, функции, различные инструкции. void setup() дает нашему коду возможность поучаствовать в инициализации системы.

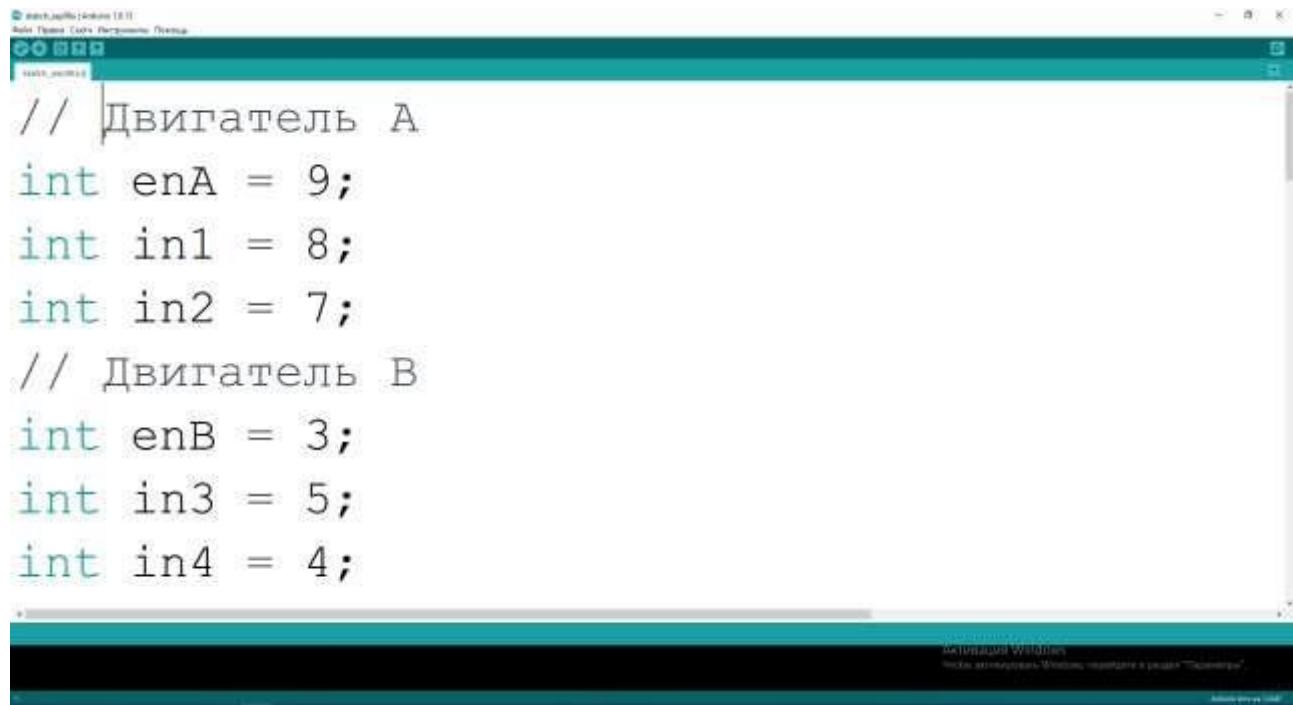
Проще говоря, здесь мы пропишем программы, которые выполняются только один раз – при запуске.

`void loop()` используется в качестве непрерывного цикла. Сюда помещается вся основная программа, то есть здесь прописываются все действия, которые мы хотим получить от робота.

2 этап

Вторым этапом в написании целостной программы для меня стал запуск двигателей. Главным в написании программы для любого устройства является процесс связывания кода с роботом, в нашем случае он происходит перед началом функции `void setup()`.

Рисунок 2. Объявление портов моторов



The screenshot shows the Arduino IDE interface with a code editor containing the following C++ code:

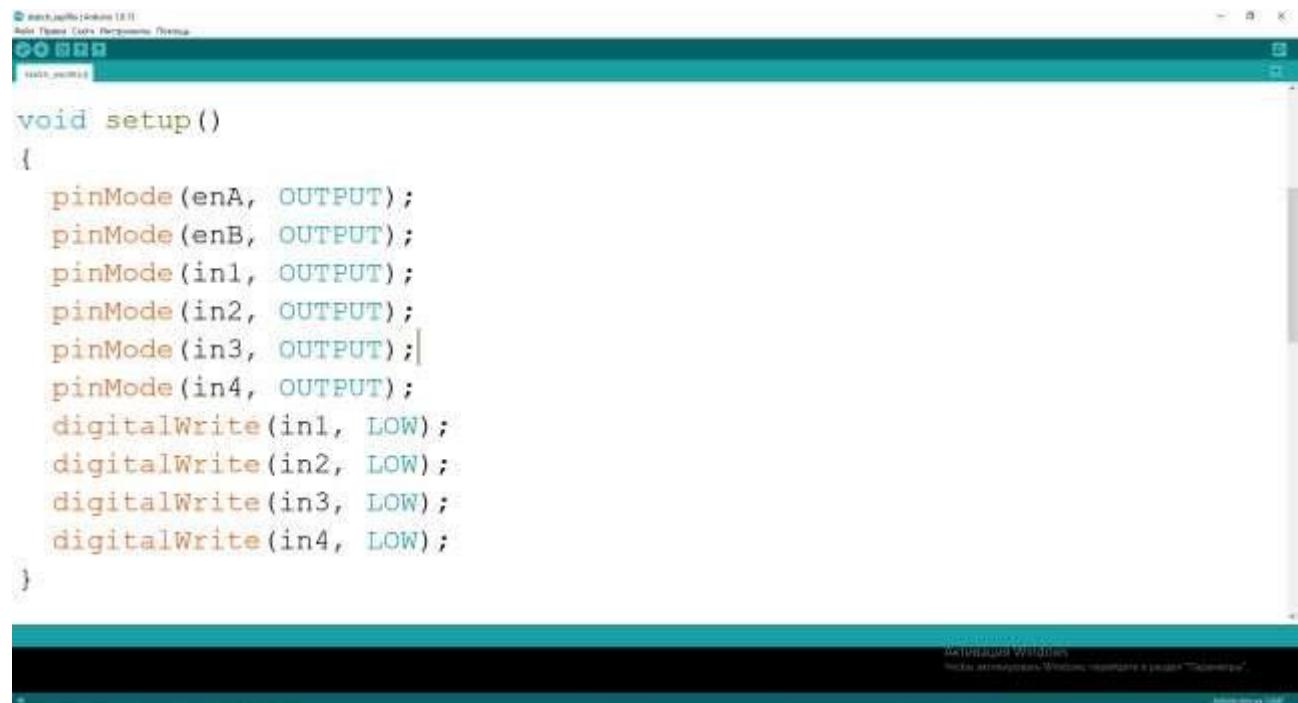
```
// Двигатель А
int enA = 9;
int in1 = 8;
int in2 = 7;
// Двигатель В
int enB = 3;
int in3 = 5;
int in4 = 4;
```

The code is written in a light blue font on a white background. The Arduino IDE interface includes a toolbar at the top, a status bar at the bottom, and a terminal window below the code editor.

Мы видим чёткое разделение на два мотора. `int enA` и `int enB` — это порты, к которым подключён драйвер моторов к основной плате Arduino, все остальные порты обозначают подключение двигателей к самому драйверу.

Следующая часть — это включение данных портов. Она происходит в `void setup()`.

Рисунок 3. Void setup()

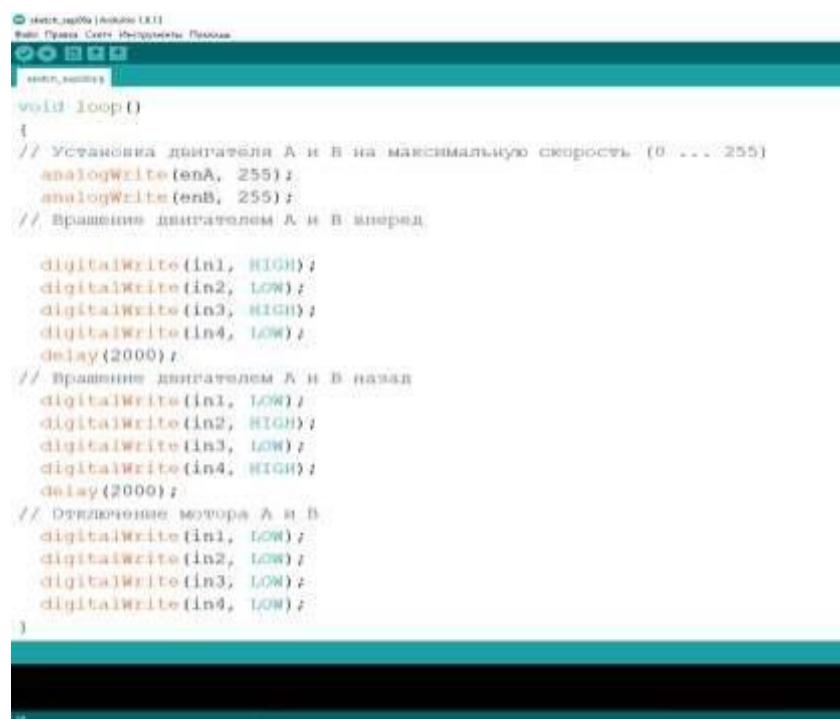


```
void setup()
{
    pinMode(enA, OUTPUT);
    pinMode(enB, OUTPUT);
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}
```

С помощью команды `pinMode` мы открываем порты, а с помощью `digitalWrite` мы подаём или отключаем напряжение (соответственно `HIGH` и `LOW`) на клеммах двигателей. В самом начале нам необходимо выставить напряжение, равное нулю.

Ну и наконец, основная часть программы, которая прописывается в `void loop()`.

Рисунок 4. Void loop()



```
void loop()
{
    // Установка двигателя А и В на максимальную скорость (0 ... 255)
    analogWrite(enA, 255);
    analogWrite(enB, 255);
    // Вращение двигателем А и В вперед
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    delay(2000);
    // Вращение двигателем А и В назад
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    delay(2000);
    // Отключение мотора А и В
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}
```

В начале функции прописана команда `analogWright` (название мотора, число). Она требуется для того, чтобы задать скорость вращения двигателей, в данном случае стоит максимальное значение 255.

Последующие команды уже известны, мы начинаем подавать напряжение, и робот начинает двигаться. Стоит заметить, чтобы заставить колёса крутиться в обратную сторону, нам требуется поменять полярность на клеммах мотора, то есть подать напряжение зеркально. Команда `delay(...)` производит остановку, на заданное количество миллисекунд.

Соответственно, данная программа позволит роботу сначала поехать вперёд, остановиться и поехать назад, снова остановиться.

Также на основе этой программы можно увидеть основной алгоритм написания различных кодов.

3 этап

Этот этап включает настройку ИК-пульта. Поскольку робот будет управляться пультом, необходимо запрограммировать кнопки, в моём случае это стрелочки. Каждая кнопка имеет собственный код, который улавливает датчик. Чтобы прописать код для каждой кнопки, нужно узнать данные коды.

Для этого я использовал маленькую программу.

Рисунок 5. Ищем коды кнопок

```
#include "IRremote.h"

IRrecv irrecv(11); // указываем вывод, к которому подключен приемник

decode_results results;

void setup() {
  Serial.begin(9600); // выставляем скорость COM порта
  irrecv.enableIRIn(); // запускаем прием
}

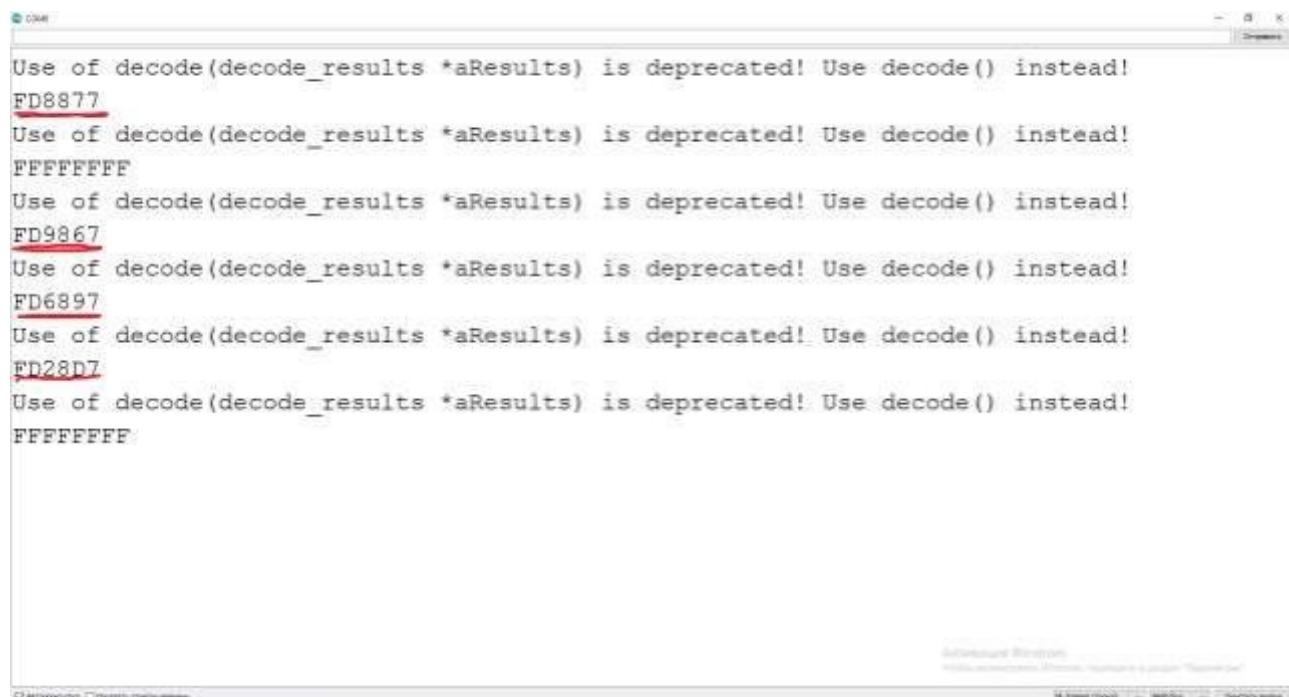
void loop() {
  if ( irrecv.decode( &results ) ) { // если данные пришли
    Serial.println( results.value, HEX ); // печатаем данные
    irrecv.resume(); // принимаем следующую команду
}
```

Это стандартная программа для данной процедуры. Чтобы её реализовать приходится использовать библиотеку. Библиотека — это набор готовых функций, классов и объектов для решения каких-то задач, которые были прописаны другими программистами для облегчения работы. Стандартно библиотеки объявляются в самом начале программы, с помощью `#include`.

Я использовал специальную библиотеку `IRremote.h`, предназначенную для инфракрасного датчика. С её помощью можно указать порт, выставить скорость этого порта, и вывести, нужные мне коды на монитор порта.

После написание загружаем программу в робота и открываем в инструментах монитор порта. Теперь нажимаем на кнопки, которые нам нужны, на мониторе порта будут появляться коды кнопок.

Рисунок 6. Монитор порта



The screenshot shows a terminal window with the title 'Сервис' (Service). The window contains a list of infrared codes captured from a remote control. Each line shows a warning about the use of `decode_results` being deprecated, followed by a hex code and a series of 'FFF' characters. The codes are: FD8877, FFFFFFFF, FD9867, FFFFFFFF, FD6897, FFFFFFFF, and FD28D7. The terminal window has a standard Windows-style title bar and status bar at the bottom.

```
Use of decode(decode_results *aResults) is deprecated! Use decode() instead!
FD8877
Use of decode(decode_results *aResults) is deprecated! Use decode() instead!
FFFFFF
Use of decode(decode_results *aResults) is deprecated! Use decode() instead!
FD9867
Use of decode(decode_results *aResults) is deprecated! Use decode() instead!
FD6897
Use of decode(decode_results *aResults) is deprecated! Use decode() instead!
FD28D7
Use of decode(decode_results *aResults) is deprecated! Use decode() instead!
FFFFFF
```

Таким образом кнопка вперёд имеет код FD8877, кнопка назад FD9867, кнопка вправо FD6897, кнопка влево FD28D7.

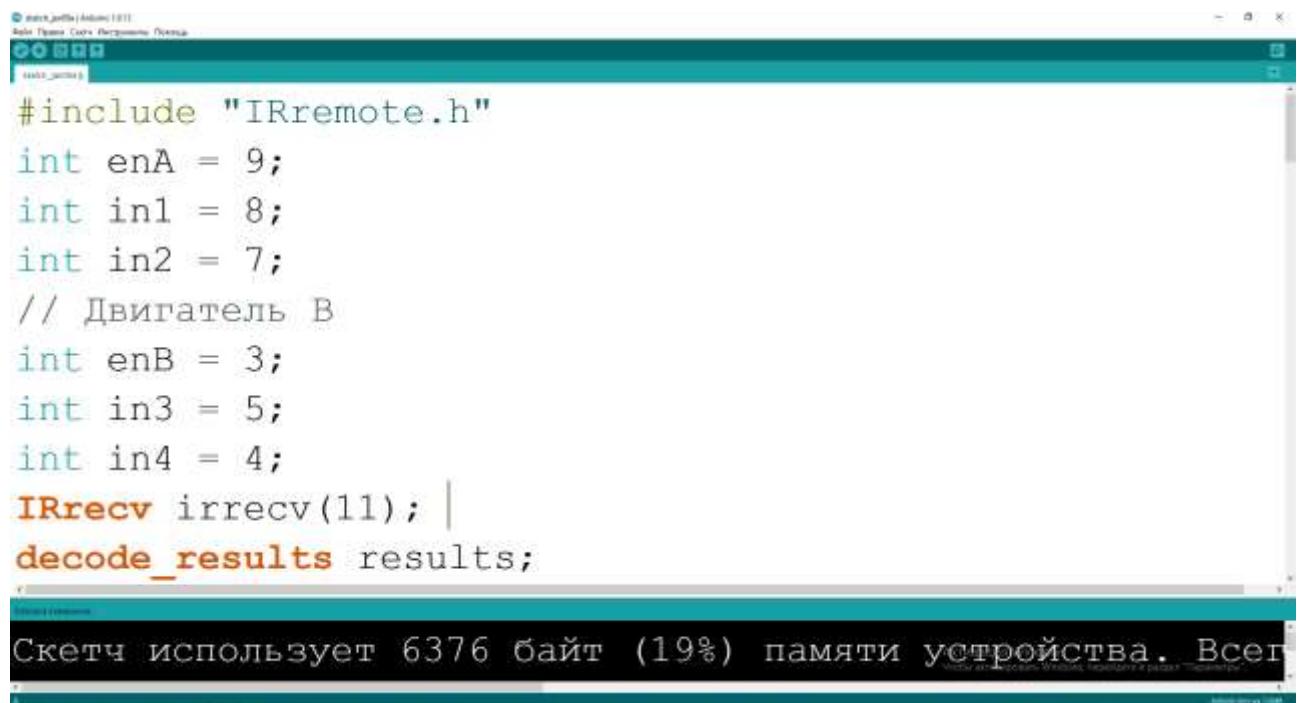
4 этап

Это последний этап. В нём необходимо связать предыдущие этапы и получить полностью работающий код.

Стоит признать, что этот этап самый сложный, получилось всё далеко не с первого раза.

В самом начале, необходимо также прописать библиотеку, обозначить порты, прописать строчку decode_results results (требуется для распознавания приёмником сигнала).

Рисунок 7. Начало программы

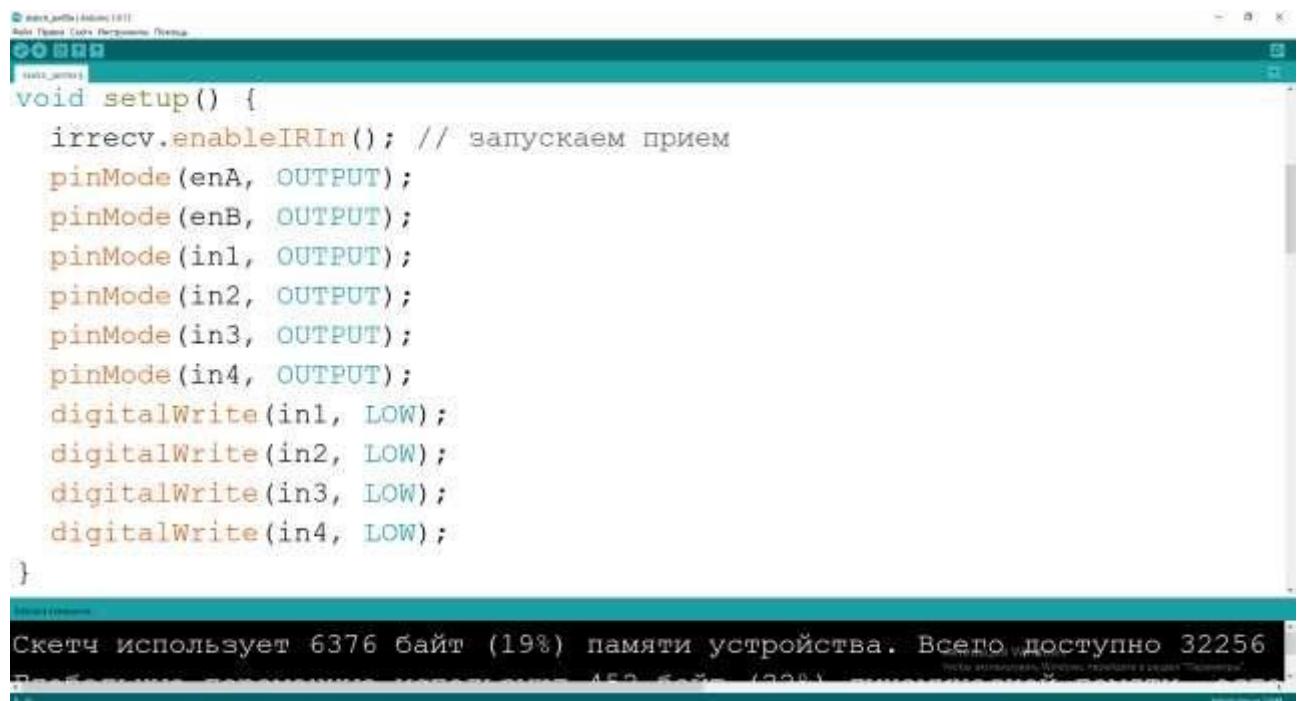


```
#include "IRremote.h"
int enA = 9;
int in1 = 8;
int in2 = 7;
// Двигатель В
int enB = 3;
int in3 = 5;
int in4 = 4;
IRrecv irrecv(11);
decode_results results;
```

Скетч использует 6376 байт (19%) памяти устройства. Всего доступно 32256

В void setup() пишем строчку начала приёма из прошлой программы irrecv.enableIRIn(), а также открываем порты моторов и подаём на них нулевое напряжение.

Рисунок 8. Открываем порты/запускаем приём



```
void setup() {
  irrecv.enableIRIn(); // запускаем прием
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  digitalWrite(in3, LOW);
  digitalWrite(in4, LOW);
}
```

Скетч использует 6376 байт (19%) памяти устройства. Всего доступно 32256

Первая строка содержит условный оператор if(если). С помощью этой команды проверяется наличие сигнала. Если сигнал есть программа идёт дальше, если нет, то ничего не происходит.

Основная программа строится с помощью конструкции switch(...) case. Она заменяет несколько условных операторов if. В моей программе цикл switch case сверяет получаемый датчиком сигнал с найденными мной кодами кнопок. В case записаны функции робота при данном сигнале.

Мы нажимаем на кнопку, и робот будет выполнять данное действие до тех пор, пока мы не нажмём на другую кнопку.

Рисунок 9. Основная программа



```
void loop() {
    if ( irrecv.decode( &results ) ) { // если данные пришли
        switch ( results.value ) {
            case 0x00000000:
                analogWrite(enR, 255);
                analogWrite(enL, 255);
                digitalWrite(im1, HIGH);
                digitalWrite(im2, LOW);
                digitalWrite(im3, HIGH);
                digitalWrite(im4, LOW);
                break;
            case 0x00000001:
                analogWrite(enR, 255);
                analogWrite(enL, 255);
                digitalWrite(im1, LOW);
                digitalWrite(im2, HIGH);
                digitalWrite(im3, LOW);
                digitalWrite(im4, HIGH);
                break;
            case 0x00000002:
                analogWrite(enR, 255);
                analogWrite(enL, 255);
                digitalWrite(im1, LOW);
                digitalWrite(im2, HIGH);
                digitalWrite(im3, HIGH);
                digitalWrite(im4, LOW);
                break;
            case 0x00000003:
                analogWrite(enR, 255);
                analogWrite(enL, 255);
                digitalWrite(im1, HIGH);
                digitalWrite(im2, HIGH);
                digitalWrite(im3, HIGH);
                digitalWrite(im4, HIGH);
                break;
        }
    }
}
```

В конце программы мы снова принимает сигнал и с помощью цикла loop всё повторяется беспрерывно.

Рисунок 10. Конец программы



The screenshot shows the Arduino IDE interface with the code for the robot program. The code is as follows:

```
analogWrite(enA, 255);
analogWrite(enB, 255);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
break;
case OmEGO2007:
analogWrite(enA, 255);
analogWrite(enB, 255);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
break;
case OmEGO857:
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, HIGH);
break;
}
// отправка команды

```

Below the code, the Arduino IDE status bar displays memory usage information:

Скетч использует 6376 байт (19%) памяти устройств. Всего доступно 32256 байт.
Глобальные переменные используют 452 байт (22%) динамической памяти, оставшись 1576 байт для локальных переменных. Максимум: 2048 байт.

Это и есть вся программа (Приложение 1) для моей импровизированной радиоуправляемой машинки.

Вывод

В ходе выполнения проекта мною были решены все планируемые задачи и в итоге достигнута основная цель: научиться основам робототехники на Arduino. Я создал робота и запрограммировал его на определённую задачу.

На изучение и создание проекта мне потребовалось 5 месяцев. На основе действующей модели можно создавать роботов, которые могут выполнять задачи в различных сферах деятельности человека.

Надеюсь, мой проект будет интересен для всех, кто хочет начать изобретать и станет примером в начальном изучении Arduino.

Использованная литература:

МАЙНИНГ КРИПТОВАЛЮТЫ [Электронный ресурс]. -Режим доступа <https://yandex.ru/turbo/mining-cryptocurrency.ru/s/robototekhnika/>, свободный - (17.11.2020)

AI-NEWS.ru [Электронный ресурс]. -Режим доступа https://ai-news.ru/2020/10/istoriya_robototekhniki_kak_vyglyadeli_samyе_pervye_robory.html, свободный - (17.11.2020)

Старт в науке [Электронный ресурс]. -Режим доступа <https://school-science.ru/6/4/38146>, свободный - (07.12.2020)

MENTAMORE [Электронный ресурс]. -Режим доступа <https://mentamore.com/robototekhnika/robory-v-povsednevnoj-zhizni-cheloveka.html>, свободный - (02.01.2021)

IOT.ru [Электронный ресурс]. -Режим доступа <https://iot.ru/wiki/robototekhnika>, свободный - (28.12.2020)

СЕРВОМЕХАНИЗМЫ [Электронный ресурс]. -Режим доступа <http://www.servomh.ru/stati/robototekhnika-istoriya-osnovnye-zadachi-robototekhniki>, свободный - (02.01.2021)

ERA BOT [Электронный ресурс]. -Режим доступа <https://erabot.ru/klassifikatsiya-robotov/>, свободный - (02.01.2021)

PROROBOT [Электронный ресурс]. -Режим доступа <https://www.prorobot.ru/slovarik/vidi-robotov.php>, свободный - (20.01.2021)

RobotChip [Электронный ресурс]. -Режим доступа <https://robotchip.ru/obzor-drayvera-motora-na-l298n/>, свободный - (13.02.2021)

Вольтик [Электронный ресурс]. -Режим доступа <https://voltiq.ru/wiki/arduino-uno-review/>, свободный - (13.02.2021)

ROBOTCLASS [Электронный ресурс]. -Режим доступа <https://robotclass.ru/tutorials/arduino-ir-remote-control/>, свободный - (14.02.2021)

Элекс [Электронный ресурс]. -Режим доступа https://led-displays.ru/ir_remote_theory.html, свободный - (14.02.2021)

```
#include "IRremote.h"

int enA = 9;
int in1 = 8;
int in2 = 7;
int enB = 3;
int in3 = 5;
int in4 = 4;
IRrecv irrecv(11);
decode_results results;
void setup() {
irrecv.enableIRIn();
pinMode(enA, OUTPUT);
pinMode(enB, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(in3, OUTPUT);
pinMode(in4, OUTPUT);
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
}
void loop() {
if ( irrecv.decode( &results ) ) {
switch ( results.value ) {
case 0xFD8877:
analogWrite(enA, 255);
analogWrite(enB, 255);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
break;
case 0xFD9867:
```

```
analogWrite(enA, 255);
analogWrite(enB, 255);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
break;
case 0xFD6897:
analogWrite(enA, 255);
analogWrite(enB, 255);
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
break;
case 0xFD28D7:
analogWrite(enA, 255);
analogWrite(enB, 255);
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
break;
case 0xFDA857:
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
break;
}
irrecv.resume();
}
}
```