

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
1 Обзор робототехники в образовательных учреждениях	9
1.1 Роль робототехники в современном образовании.....	9
1.2 Введение в платформу Arduino	11
1.3 Преимущества использования платформы Arduino в образовательных учреждениях	14
2 Анализ аппаратных и программных средств Arduino	16
2.1 Устройство платформы и схема	16
2.2 Основные компоненты Arduino.....	16
2.2.1 Микроконтроллер	17
2.2.2 Кварцевый резонатор	18
2.2.3 Разъемы ввода/вывода.....	18
2.2.4 USB-порт.....	19
2.2.5 Разъем питания.....	20
2.2.6 Светодиоды.....	21
2.2.7 Кнопки.....	21
2.2.8 Шина I2C	21
2.3 Виды плат Arduino	23
2.4 Программное обеспечение для работы с Arduino	27
3 Разработка образовательных модулей на базе Arduino.....	29
3.1 Методические подходы к обучению робототехнике с использованием Arduino.....	29

3.2	Примеры робототехнических проектов для обучения.....	33
3.2.1	Мигающий светодиод.....	33
3.2.2	Робот, избегающий препятствий.....	34
3.2.3	Простой робот	35
3.2.4	Система охранной сигнализации	36
3.2.5	Робот с дистанционным управлением	37
3.2.6	Система автоматизации умного дома	38
3.2.7	Финальный проект.....	39
3.3	Создание учебных материалов и руководств.....	48
4	Оценка эффективности использования Arduino в образовательных учреждениях	52
4.1	Критерии оценки эффективности применения Arduino	52
4.2	Анализ результатов обучения робототехнике с использованием Arduino	53
4.3	Сравнение Arduino с альтернативными решениями для обучения робототехнике.....	54
5	Рекомендации по внедрению Arduino в образовательные программы	58
5.1	Технические требования и оборудование	58
5.2	Подготовка преподавателей и инструкторов.....	59
	ЗАКЛЮЧЕНИЕ	62
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	64
	ПРИЛОЖЕНИЕ А.....	66

ПРИЛОЖЕНИЕ Б.....	67
ПРИЛОЖЕНИЕ В	68
ПРИЛОЖЕНИЕ Г.....	73
ПРИЛОЖЕНИЕ Д.....	75

ВВЕДЕНИЕ

Работа "Применение Arduino для изучения робототехнических систем в образовательных учреждениях" описывает актуальность и значимость использования платформы Arduino в образовательных учреждениях для изучения робототехнических систем.

Современные технологии робототехники играют важную роль во многих отраслях, включая медицину, производство и транспорт. Это делает изучение робототехники и программирования важным для будущих специалистов во многих областях.

Arduino — это популярная открытая платформа для создания простых электронных устройств и робототехнических систем. Его открытый и доступный для пользователей подход делает его идеальным для использования в образовательных учреждениях для изучения робототехники и программирования.

Применение Arduino в образовательных учреждениях может помочь учащимся получить практические навыки в области робототехники и программирования. Ученики могут создавать и программировать простые робототехнические системы, которые могут выполнять различные задачи, такие как автоматическое управление домашней автоматикой, создание системы умного дома и многое другое.

Использование Arduino также способствует развитию креативности и улучшению способности решать задачи. Это происходит за счет того, что Arduino позволяет ученикам создавать и программировать свои собственные робототехнические системы, что может стимулировать их увлечение и интерес к технологиям.

В этой работе мы будем рассматривать примеры использования платформы Arduino в образовательных учреждениях и описывать процесс разработки робототехнической системы на базе Arduino. Мы также рассмотрим различные проблемы и вызовы, связанные с использованием Arduino в образовательных учреждениях, а также предложим рекомендации для преподавателей и студентов по эффективному использованию платформы Arduino для изучения робототехники и программирования.

1 Обзор робототехники в образовательных учреждениях

1.1 Роль робототехники в современном образовании

Робототехника играет важную роль в современном образовании, предоставляя уникальные возможности для учащихся познакомиться с технологиями и научиться программированию. Вот несколько способов, которыми робототехника влияет на образование:

- Развитие навыков программирования: Робототехника помогает учащимся научиться программированию, что является очень важным навыком в современном мире. С помощью программирования роботов, учащиеся могут понимать, как работают алгоритмы и как они могут управлять машиной.
- Привлечение внимания учащихся: Робототехника может быть очень занимательной для учащихся. Интерактивность и возможность создавать свои собственные проекты на роботах могут помочь учащимся стать более заинтересованными в науке и технологиях.
- Развитие креативности: Робототехника также может способствовать развитию креативности учащихся. Ученики могут создавать свои собственные дизайны для роботов и решать сложные задачи, что помогает развивать их умственные способности.
- Подготовка к будущей работе: Робототехника может помочь учащимся подготовиться к работе в будущем. Так как робототехника используется в различных отраслях, ученики, которые знакомы с этой технологией, могут иметь преимущество при поиске работы в таких областях, как инженерия, информационные технологии и автоматизация.

- Развитие социальных навыков: Робототехника может также способствовать развитию социальных навыков учащихся. Ученики могут работать в команде, чтобы создавать роботов и решать задачи, что помогает развивать их способности к сотрудничеству и коммуникации.

Кроме того, робототехника может помочь учащимся лучше понимать, как работают различные технологии, и как они могут быть применены в реальной жизни. Это может помочь учащимся стать более осведомленными и готовыми к принятию решений в будущем.

Робототехника также может иметь влияние на образовательные программы в целом. Она может стимулировать развитие новых курсов и программ, которые помогут учащимся получить более глубокое понимание науки, технологии и математики.

Наконец, робототехника может помочь сделать обучение более доступным и интересным для различных групп учащихся. Технологии, используемые в робототехнике, могут быть адаптированы для использования учениками с ограниченными возможностями, такими как слабовидящие или слабослышащие дети.

Таким образом, робототехника играет важную роль в современном образовании, предоставляя уникальные возможности для развития навыков программирования, креативности, социальных навыков и готовности к будущей работе. Она также может стимулировать развитие новых образовательных программ и сделать обучение более доступным и интересным для различных групп учащихся.

1.2 Введение в платформу Arduino

Arduino — это открытая платформа для создания электронных проектов, которая позволяет людям без особых знаний в области электроники создавать устройства, контроллеры и роботов. Она была создана в 2005 году в Италии и с тех пор стала одной из самых популярных платформ в мире.

Платформа Arduino состоит из двух основных компонентов: физического микроконтроллера и программного обеспечения, называемого Arduino IDE (интегрированная среда разработки). Микроконтроллер Arduino позволяет управлять электронными компонентами, такими как светодиоды, моторы и сенсоры, путем программирования его через IDE. Одной из самых популярных моделей является Arduino Uno (Рисунок 1).



Рисунок 1 – Arduino Uno

Одна из основных причин популярности Arduino заключается в том, что она очень проста в использовании. Для создания проекта вам не нужно иметь

большой опыт в программировании или электронике. Кроме того, платформа доступна по очень низкой цене, что делает ее доступной для всех.

Arduino имеет огромное сообщество пользователей по всему миру. Существует множество различных проектов, которые вы можете найти онлайн, и форумы, где вы можете общаться с другими пользователями и задавать вопросы.

Платформа Arduino также расширяема, что означает, что вы можете добавлять новые модули и датчики к вашему проекту, чтобы расширить его возможности. Это делает ее идеальной платформой для создания любых проектов, от простых светодиодных маяков до сложных роботов.

В целом, платформа Arduino представляет собой универсальный инструмент для создания электронных проектов любой сложности. Она проста в использовании, доступна по цене и расширяема, что делает ее идеальным выбором для начинающих и опытных пользователей.

Arduino в настоящее время является одной из самых популярных платформ в мире, и ее использование распространено в различных областях, таких как робототехника, автоматизация домашних устройств, IoT (интернет вещей), автомобильная и промышленная электроника и т. д.

Arduino основана на языке программирования C++ и имеет свою собственную библиотеку, которая упрощает создание проектов. Arduino IDE позволяет создавать и загружать программный код на микроконтроллер Arduino, а также отладку программы в режиме реального времени.

Arduino имеет несколько разновидностей, каждая из которых имеет свои особенности и характеристики. Однако все они используют одинаковые

основные принципы и позволяют создавать проекты с использованием микроконтроллера Arduino.

Кроме того, Arduino может работать с различными типами сенсоров и модулей, таких как сенсоры температуры, влажности, ускорения, гироскопы, GPS, Wi-Fi и т. д. Это позволяет создавать проекты, которые могут собирать и передавать данные в режиме реального времени.

В заключение, Arduino — это мощный инструмент, который позволяет создавать проекты любой сложности и применения. Она доступна по низкой цене, проста в использовании и расширяема, что делает ее идеальным выбором для создания электронных проектов для начинающих и опытных пользователей.

1.3 Преимущества использования платформы Arduino в образовательных учреждениях

Arduino — это простой и доступный инструмент для обучения электронике, программированию и созданию реальных устройств, который обладает рядом преимуществ для использования в образовательных учреждениях.

Одним из главных преимуществ является простота использования. Arduino основан на языке программирования C++, который является достаточно простым для понимания и изучения даже для новичков в программировании. Кроме того, эта платформа имеет дружелюбный пользовательский интерфейс, что делает его более доступным для студентов и преподавателей.

Arduino также относится к категории микроконтроллеров с открытым исходным кодом и доступна по низкой цене. Это позволяет образовательным учреждениям внедрять ее в учебный процесс, не прибегая к большим затратам на оборудование.

Она подходит для создания реальных устройств и электронных проектов, что делает его полезным для обучения практическим навыкам в различных областях, таких как робототехника, автоматизация, IoT и т.д.

Платформа совместима с множеством модулей и датчиков, которые могут быть использованы в различных учебных проектах. Это дает возможность преподавателям и студентам выбрать компоненты в соответствии с задачей, которую они пытаются решить.

Наконец, существует большое сообщество разработчиков и энтузиастов, которые активно работают с Arduino. Студенты и преподаватели могут

получить доступ к онлайн-ресурсам, форумам и сообществам, где они могут найти руководства, помощь и вдохновение для своих учебных проектов. Для знакомства с платформой можно использовать такие большие площадки как Arduino Playground [14], Hackster.io [15], Arduino Project Hub [16], Electronics-Lab [17] и Circuit Digest [18].

В целом, Arduino представляет собой полезный и доступный инструмент для обучения, который можно использовать для различных проектов и областей, что делает его привлекательным для образовательных учреждений.

2 Анализ аппаратных и программных средств Arduino

2.1 Устройство платформы и схема

Принципиальная схема Arduino Uno представлена в Приложении Б. Структурно на схеме можно увидеть все основные компоненты платформы. Давайте рассмотрим все компоненты более подробно.

2.2 Основные компоненты Arduino

Глава посвящена рассмотрению основных компонентов платформы Arduino. Эта информация будет полезна для людей, начинающих изучать робототехнику в образовательных учреждениях. При работе с Arduino важно понимать, как работают ее компоненты, как они взаимодействуют между собой и как это влияет на создание проектов на основе этой платформы. В данной главе мы рассмотрим различные компоненты Arduino, такие как микроконтроллеры, кристаллы генератора, разъемы USB и питания, светодиоды, кнопки, штырьки для подключения и шины I2C (Рисунок 2). Эта информация поможет читателям лучше понять устройство и функциональность Arduino и будет полезна для дальнейшей работы с этой платформой.

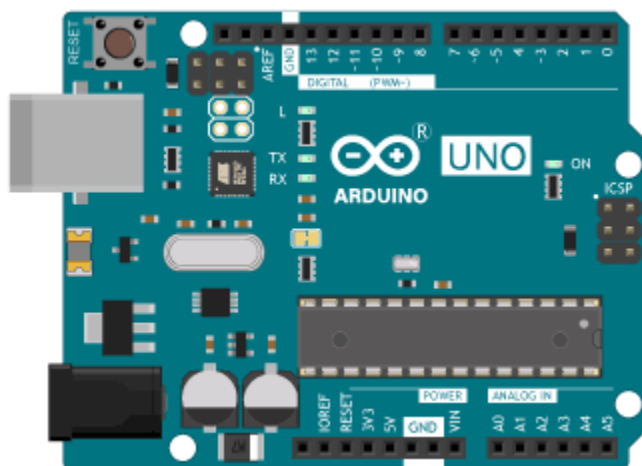


Рисунок 2 – Схематическое представление Arduino Uno R3

2.2.1 Микроконтроллер

Основным компонентом Arduino является микроконтроллер. Микроконтроллер [10] — это компьютер на одном чипе, который управляет функционированием Arduino. Он имеет процессор, память для хранения программ и данных, а также входы и выходы для подключения других устройств.

Наиболее распространенным микроконтроллером, используемым в Arduino, является ATmega328P (Рисунок 3). Он имеет 14 цифровых входов/выходов (digital I/O), 6 из которых могут использоваться как ШИМ-выходы (PWM) для управления яркостью светодиодов и скоростью моторов. Также он имеет 6 аналоговых входов (analog I/O) для чтения аналоговых сигналов, например, с датчиков температуры или освещенности.

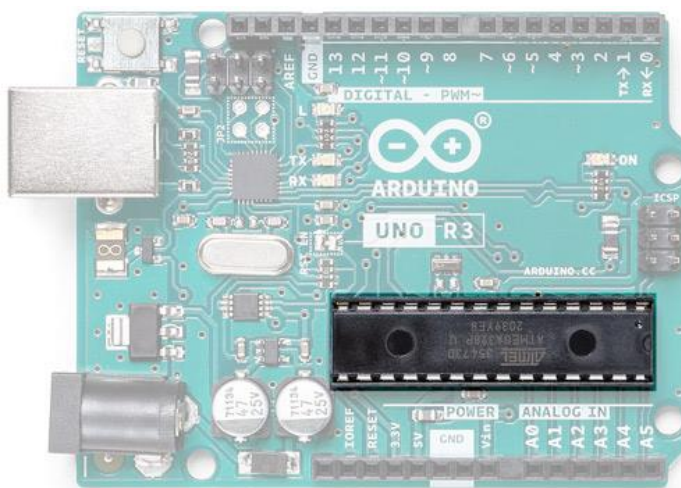


Рисунок 3 – Микроконтроллер АТМega328

2.2.2 Кварцевый резонатор

Кварцевый резонатор — это устройство, которое создает точный тактовый сигнал для микроконтроллера. Он позволяет микроконтроллеру точно знать, когда выполнять различные операции. Кварцевый резонатор на плате Arduino работает на частоте 16 МГц.

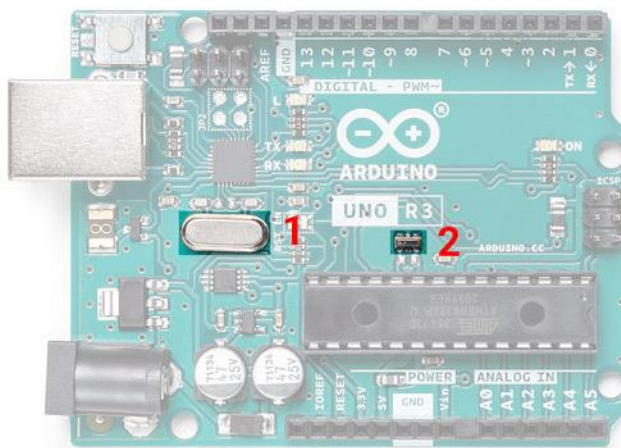


Рисунок 4 – Расположение кварцевого резонатора на плате

2.2.3 Разъемы ввода/вывода

Arduino имеет разъемы ввода/вывода для подключения датчиков, моторов, светодиодов и других устройств. Входы/выходы на плате могут быть цифровыми или аналоговыми. Цифровые входы могут быть либо высокими (HIGH), либо низкими (LOW), в зависимости от состояния подключенного устройства. Аналоговые входы позволяют измерять аналоговые сигналы, например, уровень освещенности.

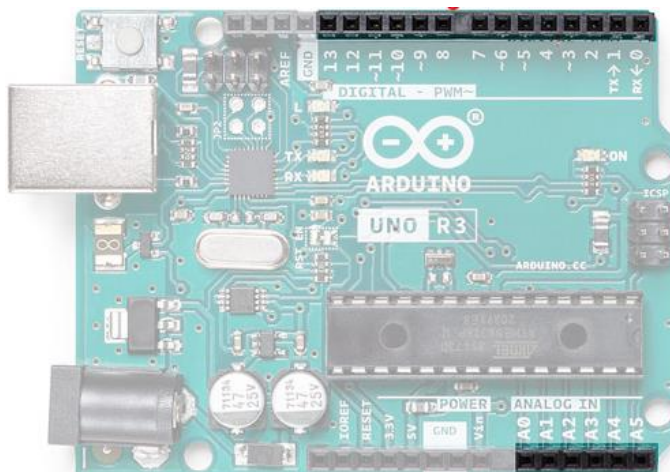


Рисунок 5 – Расположение разъемов на плате

2.2.4 USB-порт

Arduino имеет USB-порт для подключения к компьютеру. Это позволяет загружать программы на микроконтроллер и отладку кода. Кроме того, USB-порт позволяет Arduino использовать как HID-устройство, например, мышь или клавиатуру.

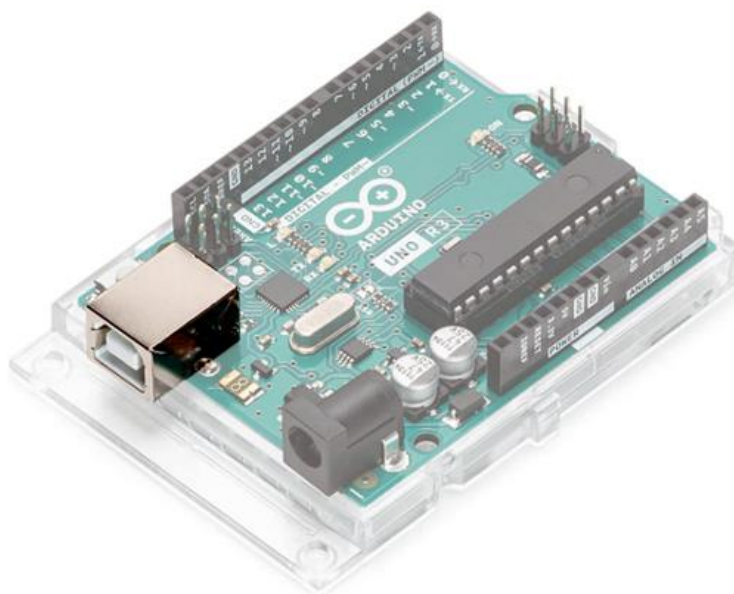


Рисунок 6 – Расположение USB-порта на плате

2.2.5 Разъем питания

Arduino имеет разъем для подключения источника питания. В большинстве случаев это может быть USB-порт компьютера, батареи или адаптер питания. Разъем питания на плате Arduino имеет обратную полярность, поэтому важно подключать источник питания правильно.

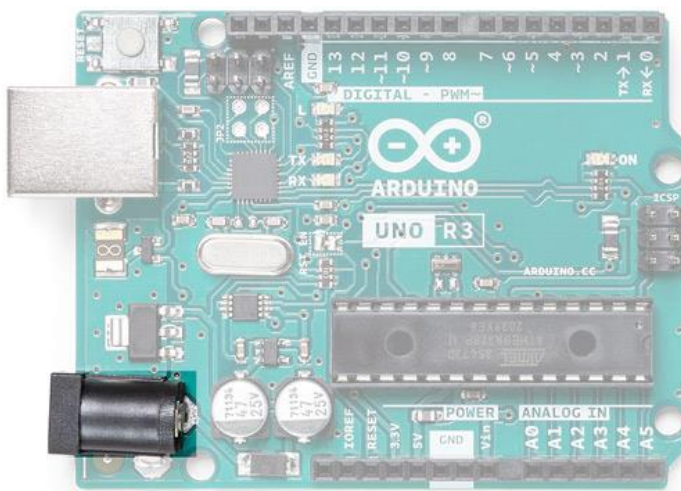


Рисунок 7 – Расположение разъема питания на плате

2.2.6 Светодиоды

Arduino имеет несколько светодиодов на плате, которые могут быть использованы для отладки программы или как индикаторы состояния. Например, светодиод на цифровом пине 13 может мигать в соответствии с программой, что помогает проверить, работает ли микроконтроллер и программа корректно.

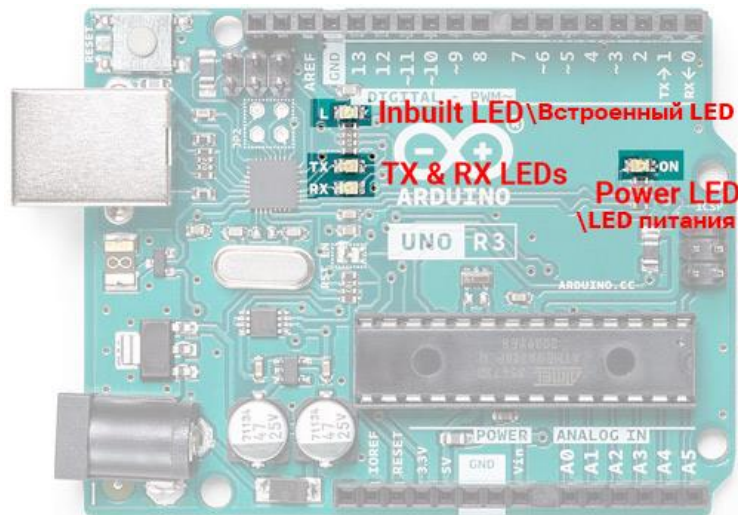


Рисунок 8 – Расположение светодиодов на плате

2.2.7 Кнопки

Arduino также имеет несколько кнопок на плате, которые можно использовать в качестве входов. Кнопки могут быть использованы для управления программой или для сброса микроконтроллера в начальное состояние.

2.2.8 Шина I2C

Шина I2C (Inter-Integrated Circuit) — это протокол коммуникации, который позволяет различным устройствам на плате Arduino обмениваться

данными. Шина I2C позволяет подключать различные датчики и другие устройства без использования множества проводов.

2.3 Виды плат Arduino

Платформа Arduino предоставляет различные варианты плат, каждая из которых имеет свои уникальные особенности и предназначена для определенных задач.

Существует несколько основных видов плат Arduino, которые мы рассмотрим ниже:

- Arduino Uno (Рисунок 1) — это одна из самых популярных и распространенных плат Arduino. Она оснащена микроконтроллером ATmega328P, имеет 14 цифровых входов/выходов, 6 аналоговых входов, разъем USB, разъем питания и другие компоненты. Arduino Uno является отличным выбором для начинающих, которые только начинают работать с Arduino, и для создания простых проектов.
- Arduino Mega (Рисунок 9) — это плата с большим количеством входов/выходов. Она оснащена микроконтроллером ATmega2560, имеет 54 цифровых входа/выхода, 16 аналоговых входов и другие компоненты. Arduino Mega отлично подходит для более сложных проектов, которые требуют большого количества входов/выходов.

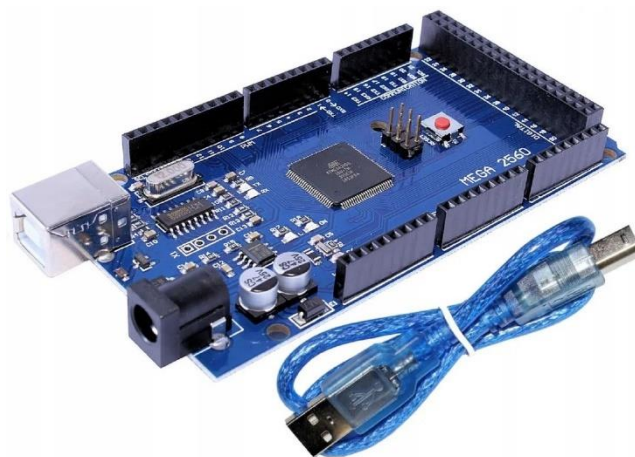


Рисунок 9 – Arduino Mega

- Arduino Nano (Рисунок 10) — это небольшая и компактная плата с микроконтроллером ATmega328P. Она имеет 14 цифровых входов/выходов, 8 аналоговых входов, разъем USB и другие компоненты. Arduino Nano является отличным выбором для создания небольших и компактных проектов.

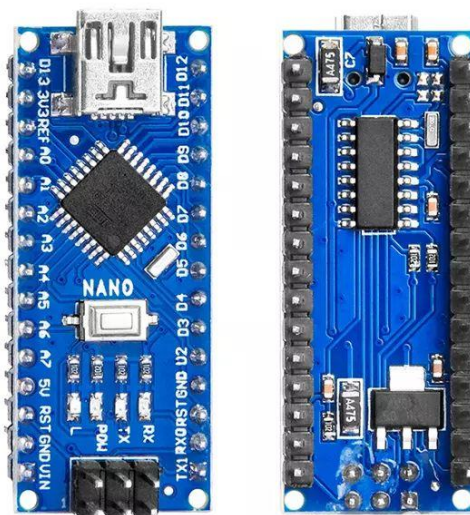


Рисунок 10 – Arduino Nano

- Arduino Due (Рисунок 11) — это плата с микроконтроллером SAM3X8E, который имеет большой объем памяти и более быстрый процессор, чем микроконтроллеры, используемые в других платах Arduino. Она имеет 54 цифровых входа/выхода, 12 аналоговых входов и другие компоненты. Arduino Due подходит для более сложных проектов, которые требуют большего объема памяти и быстродействия.



Рисунок 11 – Arduino Due

- Arduino Leonardo (Рисунок 12) — это плата с микроконтроллером ATmega32u4, который имеет встроенную поддержку USB. Она имеет 20 цифровых входов/выходов, 12 аналоговых входов и другие компоненты. Arduino Leonardo является отличным выбором для проектов, которые требуют поддержку USB, так как ее можно программировать через USB-порт и использовать как HID-устройство (Human Interface Device). Это значит, что вы можете создавать проекты, которые будут интерактивно

взаимодействовать с компьютером без необходимости использования дополнительного оборудования.



Рисунок 12 – Arduino Leonardo

В зависимости от проекта и задачи, которую вы хотите решить, можно выбрать наиболее подходящую плату Arduino. Выбор правильной платы поможет сэкономить время и усилия при разработке проекта и сделает процесс более эффективным и удобным. В следующей главе мы рассмотрим, как использовать эти платы для создания робототехнических проектов в образовательных учреждениях.

2.4 Программное обеспечение для работы с Arduino

Программное обеспечение является неотъемлемой частью работы с Arduino. Для программирования платы Arduino необходимо установить специальное программное обеспечение, называемое Arduino IDE (Integrated Development Environment). Это бесплатное кроссплатформенное приложение, доступное для Windows, MacOS и Linux, которое обеспечивает среду для написания, загрузки и отладки кода.

Arduino IDE имеет простой и понятный интерфейс, который позволяет быстро создавать программы на языке Arduino. В нем есть все необходимые инструменты для написания кода, включая подсветку синтаксиса, автодополнение и инструменты отладки.

Для начала работы с Arduino IDE необходимо скачать и установить его с официального сайта проекта. После установки необходимо выбрать правильную плату и порт для подключения платы. Для этого нужно выбрать в меню "Tools" плату и порт, которые вы используете.

Для загрузки кода на плату Arduino необходимо подключить ее к компьютеру с помощью USB-кабеля и выбрать в меню "File" пункт "Upload". После этого программа будет загружена на плату и начнет работать.

Arduino IDE также имеет множество библиотек и примеров кода, которые могут быть использованы для создания различных проектов на плате Arduino. Библиотеки позволяют использовать различные дополнительные функции и датчики, что расширяет возможности работы с платой.

Кроме Arduino IDE, существует множество альтернативных сред разработки, которые могут быть использованы для работы с платами Arduino. Некоторые из них предлагают более широкие возможности и дополнительные

функции, но они могут быть сложнее в использовании для начинающих разработчиков.

Кроме Arduino IDE, существует еще один инструмент, который может быть использован для разработки проектов на плате Arduino — это TinkerCad [11]. TinkerCad — это бесплатный веб-сервис, который позволяет создавать 3D-модели и электронные схемы, включая проекты на плате Arduino.

TinkerCad предоставляет пользователю простой и удобный интерфейс для создания проектов на плате Arduino, что делает его очень популярным среди начинающих разработчиков. TinkerCad позволяет создавать электрические схемы с помощью готовых элементов, которые могут быть выбраны из библиотеки или созданы самостоятельно.

Кроме того, TinkerCad имеет возможность симуляции [12], которая позволяет проверять работу электрических схем и проектов на плате Arduino в виртуальной среде. Это может быть очень полезно для отладки проектов и проверки их работы до того, как они будут загружены на реальную плату Arduino.

TinkerCad также имеет встроенный компилятор и загрузчик, что позволяет загружать скомпилированный код на плату Arduino без необходимости использования отдельного программного обеспечения. Это упрощает процесс разработки проектов и сокращает время, затрачиваемое на загрузку кода на плату.

В целом, TinkerCad — это удобный и легкий в использовании инструмент для создания проектов на плате Arduino. Он может быть особенно полезен для начинающих разработчиков и учеников, которые только начинают изучать робототехнику и электронику.

3 Разработка образовательных модулей на базе Arduino

3.1 Методические подходы к обучению робототехнике с использованием Arduino

Использование Arduino для обучения робототехнике имеет несколько преимуществ. Во-первых, Arduino — это доступная и недорогая плата, которая позволяет создавать множество различных проектов. Во-вторых, Arduino имеет большое сообщество разработчиков и пользователей, которые могут поделиться своими знаниями и опытом.

При разработке методических подходов к обучению робототехнике с использованием Arduino необходимо учитывать уровень подготовки учеников и их возраст. Для начинающих рекомендуется использовать простые проекты, которые помогут понять основы работы с платой Arduino и электроникой в целом. Для более продвинутых учеников можно предложить более сложные проекты, которые потребуют от них более глубокого понимания технологии и программирования.

Один из методов обучения робототехнике с использованием Arduino — это создание учебных модулей, которые могут использоваться в качестве дополнительных материалов для уроков. Такие модули могут включать в себя различные проекты на плате Arduino, инструкции по сборке и программированию проектов, а также теоретический материал, объясняющий принципы работы электронных компонентов.

Другой метод — это использование индивидуального подхода к ученикам. Каждый ученик может иметь свой уровень подготовки и интересы, поэтому рекомендуется предоставить им возможность самостоятельно выбирать проекты, которые они хотели бы реализовать на плате Arduino. Это

позволит ученикам углубить свои знания и навыки в области робототехники и электроники.

В целом, методические подходы к обучению робототехнике с использованием Arduino могут быть различными в зависимости от уровня подготовки учеников и их возраста. Главное — это создать интерес учеников к робототехнике и показать, что эта область может быть увлекательной и полезной для их будущей профессиональной деятельности.

Для эффективного обучения робототехнике с использованием Arduino необходимо использовать методические подходы, которые позволят студентам получить необходимые знания и навыки. Один из наиболее эффективных подходов — это разработка образовательной программы, состоящей из нескольких модулей, каждый из которых будет состоять из 4 занятий и завершаться практическим занятием.

Каждый модуль будет включать в себя теоретические и практические занятия, посвященные различным аспектам робототехники и программирования на Arduino. Студенты будут изучать основы электроники, программирования, механики и других важных тем.

После теоретического изучения каждой темы, студенты будут выполнять практические задания, которые позволят им закрепить полученные знания и научиться применять их на практике. Каждый модуль будет завершаться практическим занятием, на котором студенты будут создавать робототехническую систему с использованием Arduino и применять все полученные знания и навыки.

Такой подход к обучению позволит студентам получить полное представление о робототехнике и программировании на Arduino. Он поможет

студентам научиться создавать робототехнические системы, а также применять полученные знания и навыки в реальных проектах. Кроме того, такой подход к обучению поможет студентам развить свои творческие и инженерные способности, что очень важно для будущих профессионалов в области робототехники.

На основе подобранной литературы [1], [2], [3], [4] была разработана образовательная программа курса (Таблица 1).

Таблица 1 – Образовательная программа курса

Модуль	Урок	Название урока
1	1	Обзор Arduino и ее компонентов
	2	Настройка среды разработки
	3	Основные понятия электроники и схем
	4	Практическое занятие: Мигание светодиода
2	5	Введение в программирование и язык Arduino
	6	Переменные и типы данных
	7	Условные операторы и циклы
	8	Практическое занятие: Написание программы для управления светодиодом
3	9	Обзор датчиков препятствий
	10	Взаимодействие с инфракрасными датчиками
3	11	Взаимодействие с ультразвуковыми датчиками
	12	Практическое занятие: Создание робота, избегающего препятствий
4	13	Введение в разработку проекта
	14	Проектирование схемы и сборка компонентов

5	15	Написание кода
	16	Практическое занятие: Создание простого робота
	17	Обзор датчиков звука, движения, вибрации и света
	18	Взаимодействие с датчиками звука
6	19	Взаимодействие с датчиками движения и вибрации
	20	Практическое занятие: Создание системы охранной сигнализации
	21	Обзор IoT и его применения
6	22	Подключение Arduino к Интернету
	23	Отправка и получение данных через Интернет
6	24	Практическое занятие: Создание робота с дистанционным управлением
7	25	Введение в расширенные концепции IoT
	26	Настройка облачных платформ IoT
	27	Взаимодействие с веб-сервисами и API-интерфейсами
	28	Практическое занятие: Построение системы автоматизации умного дома
8	29	Обзор итогового проекта
	30	Проектирование и создание прототипа
	31	Написание кода
	32	Практическое занятие: Создание и тестирование финального проекта

3.2 Примеры робототехнических проектов для обучения

В рамках курса по Arduino и робототехнике студенты имеют возможность создавать разнообразные проекты, используя полученные знания и навыки. Ниже представлены примеры робототехнических проектов, которые можно реализовать в ходе курса:

3.2.1 Мигающий светодиод

Проект "Мигающий светодиод" [7] — это первый проект в курсе по робототехнике, который позволяет познакомиться с платформой Arduino и ее компонентами, а также основами программирования на языке Arduino.

В этом проекте ученик будет управлять светодиодом, который будет мигать с определенной задержкой. Для этого ученик подключает светодиод к платформе Arduino, используя резистор, чтобы не повредить светодиод. Затем ученик пишет программу на языке Arduino, используя функцию `delay()` для создания задержки между включением и выключением светодиода.

В процессе создания этого проекта ученик узнает, как подключать компоненты к платформе Arduino, как работать с резисторами и светодиодами, как писать простые программы на языке Arduino и как загружать программы на платформу Arduino с помощью среды разработки.

Этот проект - отличное введение в мир робототехники и позволяет начать с самых основных понятий. Робот, избегающий препятствий - в этом проекте студенты используют датчики препятствий, чтобы создать робота, который будет избегать препятствий на своем пути.

3.2.2 Робот, избегающий препятствий

Робот, избегающий препятствий [8] — это проект, в котором используются датчики препятствий для создания робота, который способен обнаруживать препятствия и избегать их.

Для этого проекта потребуются различные компоненты, включая Arduino, ультразвуковые датчики расстояния, моторы, колеса и батареи. Во время проектирования робота необходимо учитывать физические ограничения компонентов и создать механизм, который позволит роботу свободно передвигаться в окружающем пространстве.

В процессе программирования робота нужно определить правила, по которым он будет обнаруживать препятствия и реагировать на них. Например, если робот обнаруживает препятствие на своем пути, он должен остановиться, развернуться и попытаться продолжить движение в другом направлении.

Для создания робота, избегающего препятствий, можно использовать конструкцию с тремя колесами, где два колеса находятся сзади, а одно колесо находится спереди. Ультразвуковые датчики расстояния размещаются на передней части робота, чтобы он мог обнаруживать препятствия перед собой.

Для программирования робота можно использовать язык программирования Arduino, который позволяет легко взаимодействовать с различными компонентами и создавать логику поведения робота. После написания программы робот готов для тестирования.

Робот, избегающий препятствий, может быть использован в различных сферах, таких как автоматическая уборка, автоматическая доставка товаров или даже в медицинской сфере для мониторинга пациентов.

3.2.3 Простой робот

Проект "Простой робот" предполагает создание простого робота на платформе Arduino, который может двигаться в разные направления, используя два сервопривода и датчик расстояния.

Для этого проекта необходимо использовать следующие компоненты:

- Плата Arduino Uno
- Два сервопривода
- Датчик расстояния (ультразвуковой или инфракрасный)
- Провода и резисторы
- Аккумуляторная батарея и разъем для подключения

Основная идея проекта заключается в том, чтобы использовать сервоприводы для управления движением робота и датчик расстояния для определения расстояния до препятствий. Робот должен двигаться вперед, пока датчик не обнаружит препятствие, после чего робот должен повернуться в сторону, где нет препятствия, и продолжить движение.

Для начала работы над проектом, необходимо собрать робота, подключив компоненты к плате Arduino Uno. Затем, написать код на языке Arduino, который будет управлять движением робота. Для этого нужно определить функции движения вперед и поворота, используя сервоприводы, а также функцию определения расстояния до препятствия, используя датчик расстояния.

Примерный алгоритм работы робота, следующий:

1. Робот начинает движение вперед;

2. Датчик расстояния измеряет расстояние до препятствия;
3. Если расстояние больше 50 см, робот продолжает движение вперед;
4. Если расстояние меньше или равно 50 см, робот поворачивает влево;
5. После поворота, робот продолжает движение

Пример готового программного кода изложен в приложении А.

3.2.4 Система охранной сигнализации

Проект "Система охранной сигнализации" будет базироваться на датчиках звука и движения, а также на светодиодах и бужере. Робот будет слушать звуковые волны с помощью датчика звука и измерять движение с помощью датчика движения. Если будет замечено движение или превышен уровень звука, робот должен начать издавать сигнал тревоги.

В этом проекте необходимо будет создать программу, которая будет постоянно считывать значения датчиков звука и движения. Если уровень звука или уровень движения превысит определенный порог, то на светодиодах будет отображаться сигнал тревоги, а бужер будет издавать звуковой сигнал. После того, как сигнал будет прекращен, робот должен вернуться в начальное состояние.

Для проекта необходимо будет использовать:

- Arduino Uno;
- Датчик звука;
- Датчик движения;
- Светодиоды;

- Бuzzer.

Программа для этого проекта будет довольно простой. Она будет включать в себя:

- Настройку датчиков звука и движения;
- Установку пороговых значений для звука и движения;
- Считывание значений с датчиков;
- Управление светодиодами и бuzzerом.

В этом проекте основным вызовом будет являться правильная настройка пороговых значений для звука и движения, чтобы робот мог точно определять, когда нужно начинать издавать сигнал тревоги.

3.2.5 Робот с дистанционным управлением

"Робот с дистанционным управлением" — это проект, который позволяет управлять роботом с помощью дистанционного пульта. Этот проект включает в себя использование микроконтроллера, двигателей, приемника и передатчика радиосигналов.

Для создания этого робота необходимо сначала создать основу, на которую будут установлены все компоненты. Затем микроконтроллер должен быть подключен к двигателям, чтобы обеспечить движение робота. Приемник и передатчик радиосигналов должны быть установлены на роботе и дистанционном пульте соответственно.

После этого необходимо написать программное обеспечение для микроконтроллера, которое будет принимать сигналы от дистанционного пульта и управлять двигателями робота в соответствии с этими сигналами.

Когда робот готов, пользователь может управлять им, используя дистанционный пульт. Это позволяет управлять роботом на расстоянии и использовать его для различных целей, например, для исследования труднодоступных мест или для различных игр.

3.2.6 Система автоматизации умного дома

"Система автоматизации умного дома" — это проект, который может автоматизировать управление различными устройствами в доме, такими как освещение, кондиционеры, система отопления, замки на дверях и многое другое. Цель проекта — создание удобной и умной системы управления домом, которая может облегчить повседневную жизнь жильцов.

В проекте будет использоваться различное оборудование, такое как сенсоры движения, датчики температуры, умные выключатели и т.д. Все устройства будут подключены к центральному управляющему блоку, который будет отвечать за обработку информации и управление всеми подключенными устройствами.

Одной из главных функций системы автоматизации умного дома будет возможность управлять устройствами в доме с помощью мобильного приложения. Жильцы смогут использовать свои смартфоны или планшеты для контроля и управления всеми подключенными устройствами. Например, если жильцы выходят из дома, они могут с помощью приложения выключить все свет, закрыть замки на дверях, выключить кондиционер и т.д.

Кроме того, система автоматизации умного дома будет оснащена функцией "умного расписания", которая позволит жильцам настроить автоматическое включение и выключение устройств в соответствии с расписанием. Например, если жильцы хотят, чтобы свет в комнате

автоматически включался в определенное время каждый день, они могут настроить эту функцию.

3.2.7 Финальный проект

Финальный проект — это проект, который ученики выбирают сами. Он является наиболее сложным и продвинутым из всех описанных проектов.

В рамках финального проекта курса, я разработал инженерное решение под названием «Бегущая строка Wi-Fi», используя плату NodeMCU V3 ESP8266 Wi-Fi (Рисунок 13) с микроконтроллером и встроенным Wi-Fi адаптером. Эта плата, являющаяся аналогом Arduino, обладает большими возможностями и позволяет использовать среду разработки Arduino IDE.



Рисунок 13 – Плата NodeMcu V3

Для обеспечения питания проекта был выбран блок питания LRS-50 (Рисунок 14) с мощностью 50 Вт и выходным напряжением 5 Вольт. Этот блок

питания обеспечивает достаточную энергию для питания всей системы, включая микроконтроллер и светодиодную ленту.



Рисунок 14 – Блок питания LRS-50 50W

Самой важной частью проекта является адресная светодиодная лента WS2812b (Рисунок 15), состоящая из 360 светодиодов, установленных на 6-метровую ленту. Каждый светодиод обладает адресуемыми возможностями, что позволяет управлять отдельными светодиодами и создавать различные эффекты освещения.



Рисунок 15 – Адресная светодиодная лента WS2812b 60led ip33

Для расчета максимального потребления светодиодами нужно учесть количество светодиодов и их потребление. Предположим, у вас есть 360 светодиодов в вашем проекте.

Максимальное потребление светодиодами можно рассчитать следующим образом:

Пусть:

N - количество светодиодов

P_{led} - потребление одного светодиода в ваттах

Тогда максимальное потребление светодиодами выражается следующей формулой:

$$P_{max} = N * P_{led}, \quad (1)$$

В данном случае:

$$N = 360, \quad (2)$$

$$P_{led} = 0.15 \text{ Вт}, \quad (3)$$

Подставляя значения в формулу, получаем:

$$P_{max} = 360 * 0.15 \text{ Вт} = 54 \text{ Вт}, \quad (4)$$

Таким образом, максимальное потребление светодиодами в проекте составит 54 Вт.

Инженерное решение «Бегущая строка Wi-Fi» предлагает интересную комбинацию технологий, объединяющих микроконтроллеры, Wi-Fi-соединение и адресную светодиодную ленту. Это позволяет создать

впечатляющий визуальный эффект и управлять им с помощью мобильного устройства через Wi-Fi-соединение.

В первую очередь была разработана схема соединений (Рисунок 16).

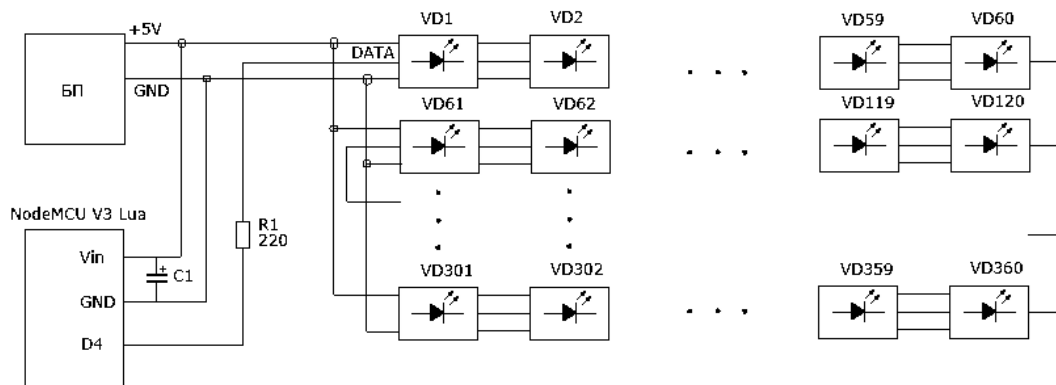


Рисунок 16 – Схема соединений финального проекта

Была составлена модель корпуса проекта и разделена на 4 части (Рисунок 17, Рисунок 18, Рисунок 19, Рисунок 20 и Рисунок 21) ввиду ограничения печати и разработан программный код «Бегущей строки Wi-Fi» ([Приложение В](#), [Приложение Г](#), [Приложение Д](#)). Корпус проекта был полностью распечатан на 3D-принтере (Рисунок 22).

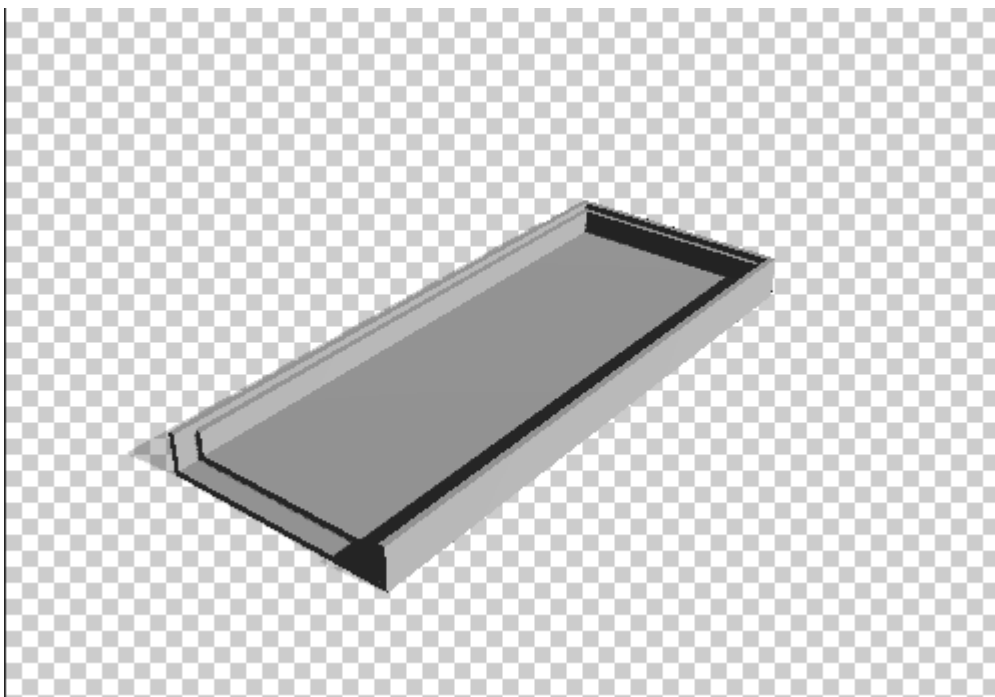


Рисунок 17 – Первая часть нижней части корпуса

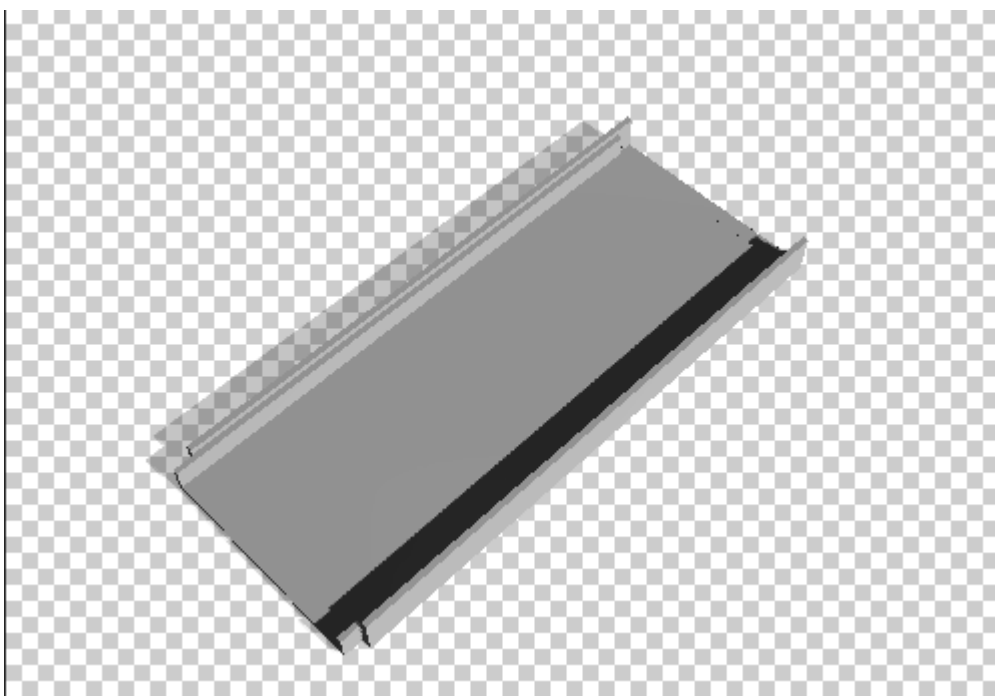


Рисунок 18 - Вторая часть нижней части корпуса

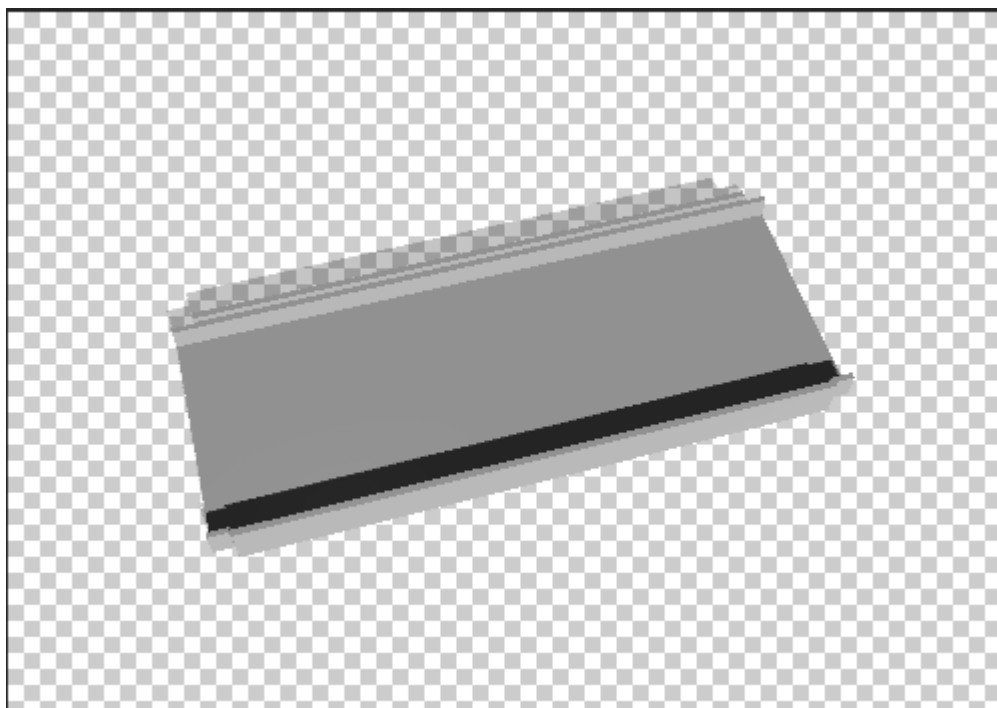


Рисунок 19 – Третья часть нижней части корпуса

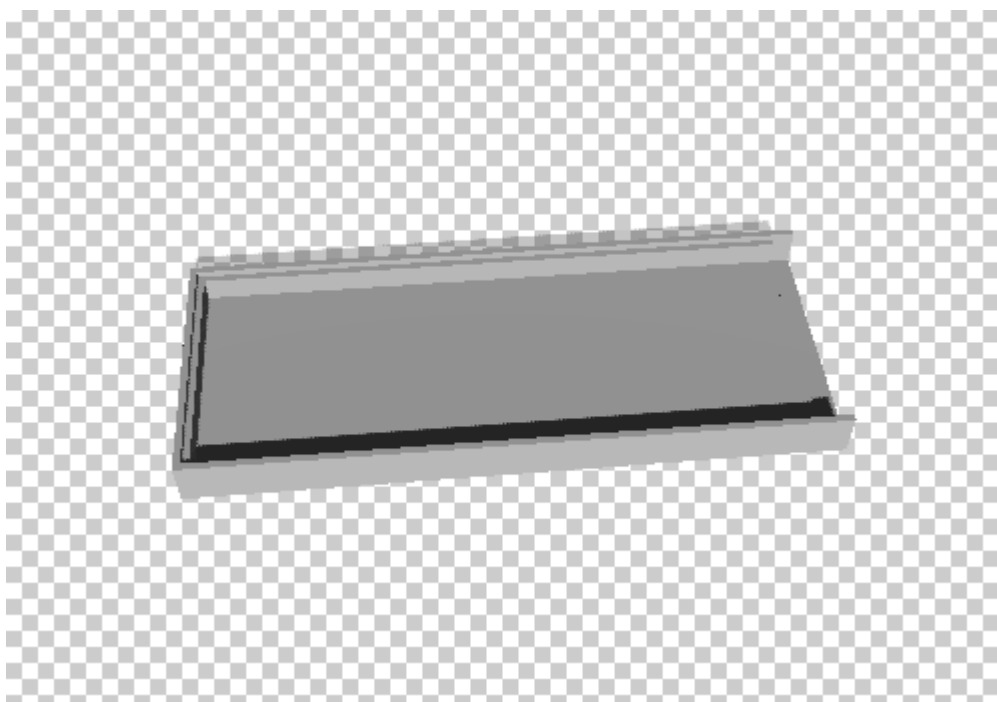


Рисунок 20 – Четвертая часть нижней части корпуса

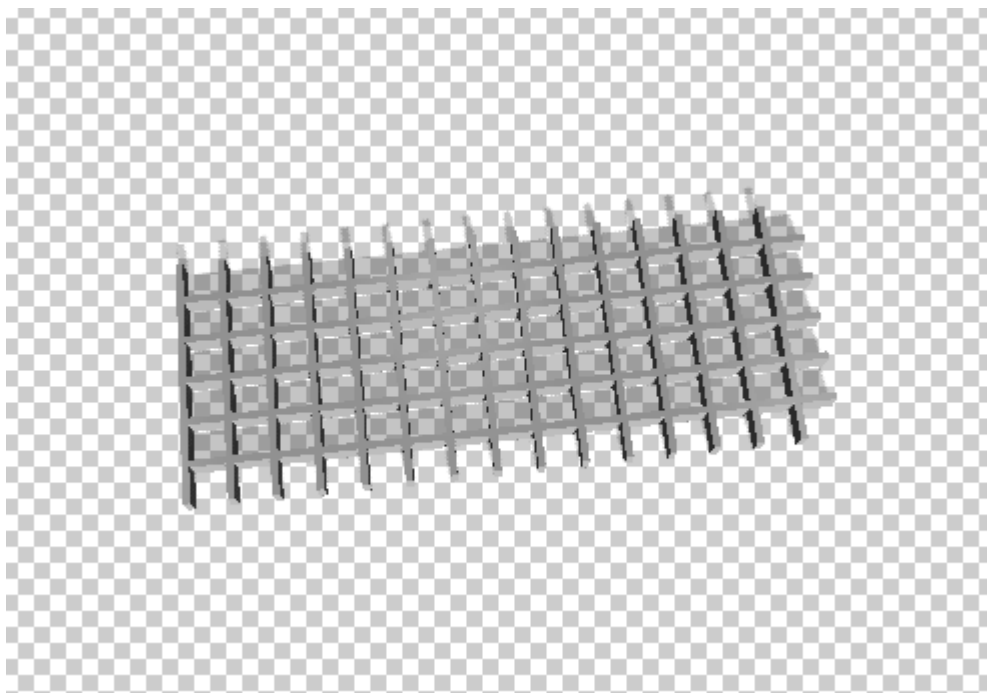


Рисунок 21 – Накладная часть корпуса



Рисунок 22 – Корпус устройства и его компоненты

Проект управляется через форму, которая создается платой (Рисунок 23). Простой интерфейс включает в себя ComboBox с двумя режимами: «Бегущая строка» и «Wi-Fi дисплей», а также строку для ввода текста, которая выводится на светодиодную ленту. Несколько TrackBar используются для

регулирования уровней RGB для текста и фона, а также скорости движения текста. В форме также присутствует информация для отладки, включая количество запусков и IP-адрес устройства.

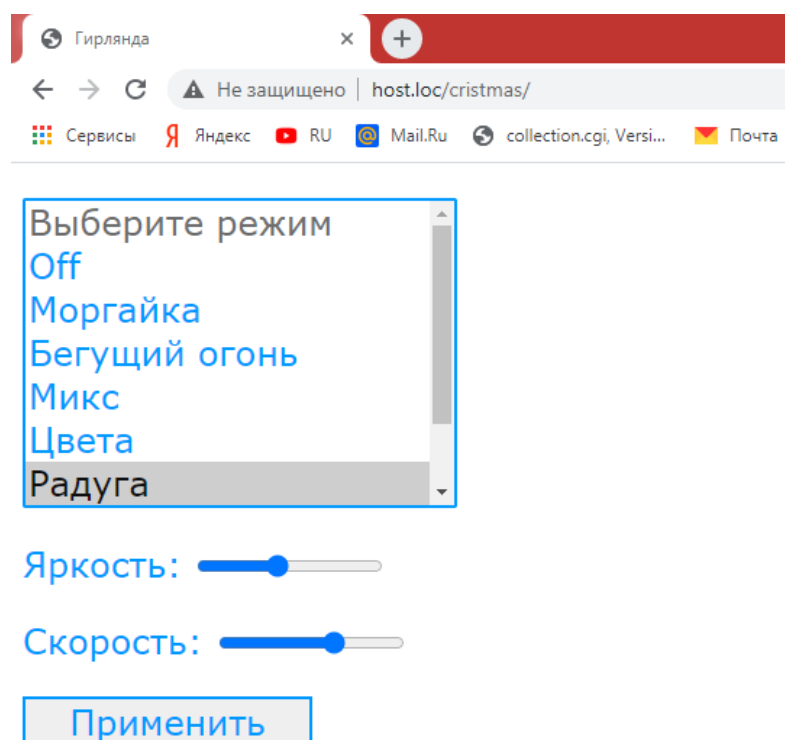


Рисунок 23 – Web-интерфейс проекта

Необходимо учитывать, что при первом подключении устройства оно работает в качестве точки доступа (с идентификатором сети NodeMCU). Изначально не предполагалось, чтобы у данной точки доступа был пароль, так как к ней мог подключиться любой пользователь. Однако, оказалось, что точка доступа может запомнить пароль от предыдущей сети, к которой она была подключена в качестве клиента, и использовать его для своей собственной защиты.

Чтобы подключиться к точке доступа NodeMCU, необходимо в браузере перейти по адресу 192.168.4.1 и выбрать вариант «Configure Wi-Fi». Затем

нужно ввести ssid и пароль вашей Wi-Fi сети, к которой вы хотите подключить устройство. Если данные введены верно, то плата подключится к указанной сети и перейдет в режим клиента, который сможет принимать UDP-пакеты и отображать их с помощью светодиодных лент.

Как только устройство подключится к Wi-Fi, его IP-адрес будет выведен на светодиодную ленту дважды. Этот адрес можно использовать в браузере, чтобы начать работу с режимом «Бегущая строка». Для работы в этом режиме достаточно ввести информацию, выбрать цвет текста, цвет фона и скорость движения текста. Режим довольно прост в использовании.

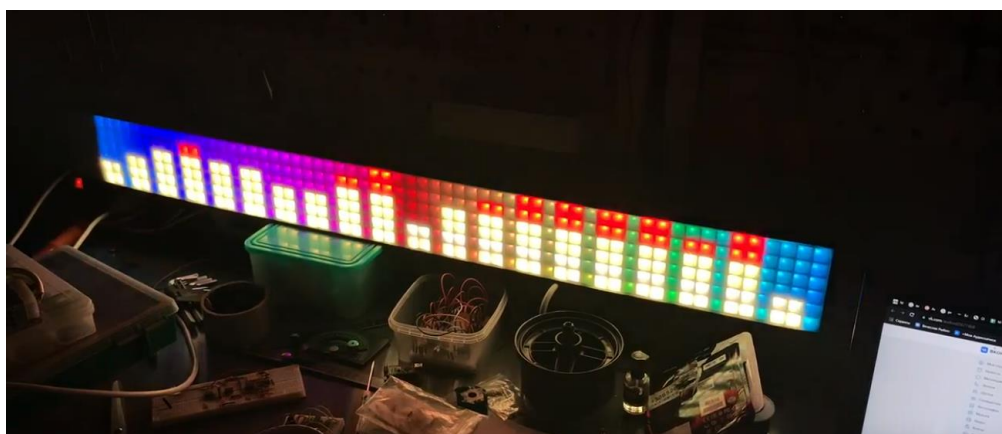


Рисунок 24 – Фотография готового финального проекта

3.3 Создание учебных материалов и руководств

Для каждого проекта было создано пошаговое руководство по сборке, программированию и тестированию робота, кроме финального. Руководства будут включать в себя детальные инструкции и фотографии для каждого шага процесса.

Кроме того, для каждого проекта были созданы учебные материалы, которые помогут пользователям понять технологии, используемые в проекте, и обучиться применению этих технологий в других проектах. Материалы включают в себя объяснения базовых концепций, диаграммы и примеры кода, чтобы помочь пользователям понять, как работает каждый проект.

В курсе есть несколько модулей, каждый из которых содержит важные уроки (Таблица 1), которые помогут ученикам получить знания и навыки в различных областях.

Модуль 1: Введение в платформу.

В этом модуле ученики получают общее представление об Arduino и ее компонентах, научатся настраивать среду разработки, понимать основные понятия электроники и схем, а также научатся мигать светодиодом. Эти уроки позволят ученикам понять базовые принципы работы Arduino и начать создавать свои проекты.

Модуль 2: Основы программирования.

В этом модуле ученики изучат язык программирования Arduino, научатся работать с переменными и типами данных, условными операторами и циклами, а также научатся управлять светодиодом с помощью программы. Эти уроки помогут ученикам освоить основы программирования, которые необходимы для создания более сложных проектов.

Модуль 3: Датчики препятствий.

В этом модуле ученики узнают об основных датчиках препятствий, таких как инфракрасные и ультразвуковые датчики, научатся работать с ними и создадут робота, который будет избегать препятствия. Эти уроки помогут ученикам понять, как работать с датчиками и создавать простые роботы.

Модуль 4: Первый проект.

В этом модуле ученики создадут свой первый проект, начиная с проектирования схемы и сборки компонентов и заканчивая написанием кода. Это практическое занятие поможет ученикам научиться применять свои знания для создания реальных проектов.

Модуль 5: Датчики звука, движения, вибрации и света.

В этом модуле ученики узнают о других типах датчиков, таких как датчики звука, движения, вибрации и света, научатся работать с ними и создадут систему охранной сигнализации. Эти уроки позволят ученикам расширить свои знания и научиться создавать более сложные проекты.

Модуль 6: Интернет вещей (IoT)

Этот модуль поможет студентам понять, как подключить Arduino к Интернету и отправлять данные через Интернет. Они изучат понятия IoT и применения этой технологии, а также научатся создавать робота с дистанционным управлением.

Модуль 7: Расширенный IoT

Этот модуль расширяет знания студентов в области IoT и позволяет им узнать о расширенных концепциях IoT. Они научатся настраивать облачные

платформы IoT, взаимодействовать с веб-сервисами и API-интерфейсами, а также создавать систему автоматизации умного дома.

Модуль 8: Итоговый проект

Этот модуль позволит студентам совместить все знания и навыки, полученные в предыдущих модулях, и создать свой собственный проект. Они научатся проектировать и создавать прототип, писать код и тестировать свой проект в действии.

В рамках моей деятельности преподавателя программирования в школе дополнительного образования я применял ряд навыков, которые необходимы для создания эффективных учебных материалов и руководств. Прежде всего, необходимо быть в курсе последних трендов и разработок в области программирования и технологий, чтобы создавать актуальные и релевантные учебные материалы. Кроме того, мой опыт работы как преподавателя программирования помогает мне лучше понимать потребности и уровень подготовки моих учеников, что позволяет мне создавать материалы, соответствующие их уровню и учитывать их потребности.

Для успешного создания учебных материалов также необходимы аналитические навыки. Хорошие аналитические способности позволяют разбивать материал на составные части, выделять ключевые понятия и разрабатывать структуру учебного курса. Креативность также не менее важна в процессе создания учебных материалов, поскольку это позволяет создавать увлекательные и интересные материалы для учеников. Это, в свою очередь, помогает им не только понимать материал, но и уделять ему больше внимания и времени.

Необходимость языковых навыков также важна для успешного создания учебных материалов. Хорошее знание языка, на котором преподается, позволяет легче и точнее выражать свои мысли и идеи в учебных материалах.

Относительно методов, которые я, как преподаватель, не рассматриваю, поскольку считаю их плохими, это может быть связано с тем, что я считаю их неэффективными или неадекватными для моей целевой аудитории. Некоторые из этих методов могут включать механическое запоминание и изучение только теории. Вместо этого, я предпочитаю использовать более интерактивные и практические методы обучения, которые помогают студентам лучше понимать материал и применять его на практике.

Таким образом, для создания эффективных учебных материалов и руководств в программировании, необходимо обладать знаниями технологии и опытом работы, а также хорошими аналитическими и креативными навыками. Важно также учитывать потребности и уровень подготовки учеников, чтобы создавать материалы, соответствующие их уровню и потребностям. Однако, не все методы обучения подходят для всех аудиторий, и необходимо выбирать те, которые будут наиболее эффективными для конкретной группы студентов.

В итоге были разработаны учебные материалы и руководства, которые составлены с использованием моих профессиональных навыков и опыта работы преподавателем программирования в школе дополнительного образования.

4 Оценка эффективности использования Arduino в образовательных учреждениях

4.1 Критерии оценки эффективности применения Arduino

Для оценки эффективности использования Arduino в образовательных учреждениях [9] были выбраны следующие критерии:

- **Уровень усвоения знаний.** Оценка степени понимания учениками теоретического материала по робототехнике с использованием Arduino.
- **Умение решать задачи.** Оценка умения учеников применять полученные знания на практике при создании робототехнических проектов с использованием Arduino.
- **Результаты работы группы.** Оценка того, насколько эффективно ученики работают в команде при создании робототехнических проектов с использованием Arduino.
- **Интерес учеников.** Оценка того, насколько ученики заинтересованы в обучении робототехнике с использованием Arduino и готовы самостоятельно изучать дополнительные материалы.
- **Вовлеченность учеников.** Оценка того, насколько ученики активно участвуют в занятиях и готовы задавать вопросы, обсуждать материал и вырабатывать собственное мнение.
- **Практическая применимость знаний.** Оценка того, насколько ученики готовы использовать полученные знания и навыки в реальной жизни для решения конкретных задач.

Выбранные критерии позволяют оценить не только уровень знаний, но и практические навыки, интерес и вовлеченность учеников, а также их готовность к применению полученных знаний в реальной жизни. Эти критерии будут использоваться для анализа результатов обучения робототехнике с использованием Arduino в следующем разделе.

4.2 Анализ результатов обучения робототехнике с использованием Arduino

В рамках исследования были проведены мастер-классы по робототехнике на основе разработанных методических материалов с использованием платформы Arduino. Ученики проявили большой интерес к теме и активно участвовали в занятиях. Результаты тестирования учеников показали, что они оценили полученные знания и навыки на высокий уровень.

Ученики отметили (Таблица 2), что получили ценный опыт работы с платформой Arduino и научились создавать и программировать различные устройства. Они также выразили желание продолжить обучение робототехнике с использованием платформы Arduino.

Таким образом, результаты мастер-классов свидетельствуют о том, что использование методических материалов на основе платформы Arduino является эффективным средством для обучения робототехнике в школе. Это подтверждает важность интеграции новых технологий в образовательный процесс и необходимость разработки соответствующих методических материалов.

Таблица 2 – Результаты опроса учеников

Учен ик	Уровен ь усвоен ия знаний	Умен ие решат ь задач и	Результа ты работы группы	Интерес с ученик ов	Вовлеченно сть учеников	Практичес кая применимо сть
Учени к 1	9	8	9	9	9	8
Учени к 2	8	7	8	8	8	7
Учени к 3	9	9	9	7	8	9
Учени к 4	7	8	8	9	9	7
Учени к 5	8	7	9	7	8	8

4.3 Сравнение Arduino с альтернативными решениями для обучения робототехнике

Различные платформы для обучения робототехнике доступны на рынке, и Arduino является только одним из них. В данном разделе мы проведем сравнительный анализ Arduino с альтернативными платформами для обучения робототехнике.

Одной из наиболее популярных платформ является Lego Mindstorms. Эта система предлагает студентам создавать роботов из конструктора Lego, используя программное обеспечение, которое позволяет программировать действия робота. Lego Mindstorms также имеет собственную операционную систему, которая может быть загружена на робота.

Другой популярной платформой является Raspberry Pi. Она предоставляет учащимся больше возможностей для программирования и настройки, чем Arduino. Raspberry Pi может использоваться для разработки программного обеспечения, контроля устройств и многих других приложений, которые требуют больше вычислительной мощности, чем Arduino.

Сравнивая Arduino с Lego Mindstorms и Raspberry Pi, можно сделать вывод, что каждая из этих платформ имеет свои преимущества и недостатки. Arduino ориентирована на простоту использования и доступность, поэтому она является лучшим выбором для начинающих. Кроме того, Arduino более дешева и имеет большую гибкость в работе с различными датчиками и актуаторами.

Lego Mindstorms подходит для тех, кто хочет создавать роботов из конструктора Lego, а не только программировать их. Она также имеет более продвинутые функции, такие как встроенная операционная система.

Raspberry Pi наиболее подходит для тех, кто имеет опыт программирования и хочет создавать более сложные проекты с использованием дополнительных компонентов.

Помимо Lego Mindstorms и Raspberry Pi, на рынке существует множество других альтернативных решений для обучения робототехнике,

которые также могут использоваться в образовательных учреждениях. Рассмотрим некоторые из них.

Первым альтернативным решением является VEX Robotics. Эта система включает в себя множество компонентов, в том числе контроллер, моторы, датчики и конструкционные элементы, которые могут быть использованы для создания широкого спектра роботов. VEX Robotics также предоставляет удобную среду программирования, которая может быть использована для написания кода на нескольких языках, включая C++, Python и RobotC. Однако, стоимость этой системы значительно выше, чем у Arduino, что может быть фактором, ограничивающим ее использование в образовательных учреждениях.

Еще одним альтернативным решением является mBot, созданный компанией Makeblock. Это недорогая система, которая включает в себя контроллер, моторы, датчики и конструкционные элементы, а также блок программирования, основанный на Scratch. mBot предоставляет удобную среду для обучения робототехнике и программированию, однако, она имеет ограниченные возможности по сравнению с Arduino и может ограничивать разнообразие проектов, которые могут быть созданы учениками.

Также стоит отметить, что Arduino обладает большим сообществом разработчиков и пользователей, что позволяет быстро получить помощь и поддержку в случае возникновения проблем. Кроме того, на рынке существует множество готовых проектов и библиотек для Arduino, что делает ее использование более удобным для обучения робототехнике.

В целом, Arduino является эффективным и доступным решением для обучения робототехнике в образовательных учреждениях. Она обладает широкими возможностями для создания разнообразных проектов и

предоставляет удобную среду программирования, что позволяет ученикам быстро и эффективно осваивать материал. Сравнение с другими альтернативными решениями показывает, что Arduino является лучшим решением.

5 Рекомендации по внедрению Arduino в образовательные программы

5.1 Технические требования и оборудование

Внедрение Arduino в образовательные программы требует наличия определенного технического оборудования и соответствующих требований к нему. Для использования Arduino необходимо наличие персонального компьютера с операционной системой Windows, Mac OS или Linux, а также установленного на нем интегрированной среды разработки Arduino (Arduino IDE).

Кроме того, для работы с Arduino необходимо наличие следующего оборудования:

- Плата Arduino с соответствующим микроконтроллером
- USB-кабель для подключения платы к компьютеру
- Электронные компоненты для создания проектов (например, светодиоды, резисторы, датчики)
- Блок питания для питания платы и электронных компонентов
- Периферийные устройства (например, дисплей, клавиатура, мышь) для удобной работы с компьютером

Также стоит учесть, что для работы с Arduino могут потребоваться дополнительные модули расширения, такие как модули для беспроводной связи или шилды для управления моторами. Однако, в зависимости от конкретных потребностей и задач, список необходимого оборудования может быть изменен.

Важно также учитывать, что при выборе оборудования следует учитывать возраст и уровень подготовки учеников, а также их потребности и цели обучения. Все оборудование должно быть безопасным для использования и соответствовать стандартам безопасности, чтобы предотвратить возможные несчастные случаи во время занятий.

5.2 Подготовка преподавателей и инструкторов

Чтобы обучение на Arduino было эффективным, необходимо, чтобы преподаватели и инструкторы имели достаточные знания и опыт в работе с платформой.

Подготовка преподавателей и инструкторов должна включать в себя не только обучение использованию платформы Arduino, но и методик преподавания и руководства учениками. Некоторые преподаватели и инструкторы могут иметь опыт работы с Arduino, но не обладать достаточными педагогическими навыками для эффективного обучения. Поэтому, помимо технического обучения, важно также организовать тренинги и семинары, нацеленные на развитие педагогических навыков.

Кроме того, необходимо создать инфраструктуру для постоянного обучения преподавателей и инструкторов, так как платформа Arduino постоянно развивается и обновляется. Например, можно организовать вебинары, конференции и мастер-классы для обмена опытом и получения новых знаний.

Конкретные шаги подготовки преподавателей для работы с Arduino могут включать в себя:

1. Ознакомление с основами Arduino и программирования на языке Arduino. Преподаватель должен уметь самостоятельно работать с

платформой Arduino, программировать на языке Arduino и знать основные функции и возможности платформы.

2. Изучение методических материалов и учебных пособий по работе с Arduino в образовательном процессе. Преподаватель должен знать как использовать Arduino для обучения робототехнике и электронике.
3. Разработка учебных программ и планов для работы с Arduino. Преподаватель должен определить, какие темы и задачи будут включены в учебный план, какие материалы и оборудование будут использоваться, и какие будут критерии оценки успеваемости студентов.
4. Организация мастер-классов и тренингов для преподавателей. Это поможет обмениваться опытом работы с Arduino и обсудить наиболее эффективные методы обучения и использования платформы.
5. Проведение практических занятий и проектов с учениками, чтобы преподаватель получил практический опыт работы с Arduino в роли инструктора. Это поможет преподавателю лучше понимать потребности учеников и настроить учебный процесс в соответствии с этими потребностями.
6. Оценка эффективности обучения и постоянное обновление методических материалов и учебных программ в соответствии с результатами оценки.

В целом, подготовка преподавателей и инструкторов является важным шагом в успешном внедрении Arduino в образовательные программы. Без достаточного уровня подготовки преподавателей и инструкторов, обучение на Arduino может оказаться неэффективным, и ученики могут не получить

полной пользы от использования этой платформы в образовательном процессе.

ЗАКЛЮЧЕНИЕ

В ходе данного исследования была рассмотрена эффективность использования Arduino для обучения робототехнике в образовательных учреждениях.

Кроме того, были разработаны методические материалы и проведены мастер-классы для учеников по изучению робототехники с использованием Arduino. Результаты показали, что ученики оценили полученные знания высоко и были заинтересованы в продолжении обучения. Также была проведена оценка результатов обучения учеников, которая показала высокий уровень усвоения знаний, умения решать задачи, результаты работы в команде, интерес и вовлеченность учеников, а также практическую применимость знаний.

Также был проведен анализ аппаратных и программных средств для обучения робототехнике с использованием Arduino. Было выявлено, что Arduino является эффективным и доступным средством для обучения робототехнике, благодаря своей простоте использования и широкому сообществу разработчиков, которое обеспечивает постоянную поддержку и развитие платформы.

Было проведено сравнение Arduino с альтернативными решениями для обучения робототехнике, которое показало преимущества использования Arduino: доступность, простота использования, гибкость, возможность расширения и настройки, а также большое сообщество разработчиков и пользователей.

Для успешного внедрения Arduino в образовательные программы были предложены рекомендации по техническим требованиям и оборудованию,

разработке учебных планов и программ, подготовке преподавателей и инструкторов, а также взаимодействию с внешними организациями и партнерами.

В результате исследования можно сделать вывод о том, что применение Arduino для изучения робототехнических систем в образовательных учреждениях имеет большой потенциал и может быть эффективным инструментом для развития творческого мышления, умения решать задачи, развития компетенций в области технологий и программирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Андреев, А. В. Основы работы с Arduino: практическое руководство / А. В. Андреев. – М.: ДМК Пресс, 2017. – 280 с.
2. Бондаренко, В. Г. Программирование на платформе Arduino / В. Г. Бондаренко, Д. В. Шабанов. – М.: Техносфера, 2018. – 352 с.
3. Васильев, А. С. Основы робототехники: Учебное пособие / А. С. Васильев. – СПб.: БХВ-Петербург, 2016. – 400 с.
4. Гуреев, А. М. Робототехника: от новичка к профессионалу / А. М. Гуреев, Л. Н. Рогов. – М.: ДМК Пресс, 2017. – 550 с.
5. Долгов, И. В. Робототехника в образовании: учебник и практикум / И. В. Долгов, Е. В. Беляева. – М.: Юрайт, 2020. – 375 с.
6. Иванов, А. П. Применение Arduino в образовательных учреждениях: методические рекомендации / А. П. Иванов. – М.: Инфра-М, 2021. – 180 с.
7. Кузнецов, В. М. Arduino для начинающих: базовые принципы и практические примеры / В. М. Кузнецов, И. В. Петров. – М.: Ленанд, 2019. – 240 с.
8. Макаров, О. Н. Использование Arduino в проектах робототехники: учебное пособие / О. Н. Макаров, В. Ю. Лаврентьев. – М.: Радио и связь, 2020. – 320 с.
9. Романов, С. И. Робототехнические системы на базе Arduino: научно-методическое пособие / С. И. Романов, Д. В. Савельев. – М.: Радио и связь, 2019. – 264 с.
10. Смирнов, А. В. Микроконтроллеры Arduino: принципы работы и применение в робототехнике / А. В. Смирнов. – М.: Издательство Ленанд, 2018. – 400 с.
11. TinkerCAD. Autodesk, Inc. 2011. <https://www.tinkercad.com/>
12. Tinkercad Circuits. Autodesk, Inc. 2012. <https://www.tinkercad.com/circuits>
13. Arduino. Arduino Team. 2005. <https://www.arduino.cc/>

14. Arduino Playground. Arduino Community. 2007.
<https://playground.arduino.cc/>
15. Hackster.io. Avnet, Inc. 2013. <https://www.hackster.io/>
16. Arduino Project Hub. Arduino. 2018. <https://create.arduino.cc/projecthub/>
17. Electronics-Lab. Electronics-Lab.com. 2008. <https://www.electronics-lab.com/>
18. Circuit Digest. Circuit Digest. 2016. <https://circuitdigest.com/>

ПРИЛОЖЕНИЕ А

Листинг 1 – Пример кода для движения робота

```
#include <Servo.h>

Servo leftMotor;
Servo rightMotor;

void setup() {
    leftMotor.attach(9);
    rightMotor.attach(10);
}

void moveForward() {
    leftMotor.write(180);
    rightMotor.write(0);
}

void turnLeft() {
    leftMotor.write(0);
    rightMotor.write(0);
}

void turnRight() {
    leftMotor.write(180);
    rightMotor.write(180);
}

void stop() {
    leftMotor.write(90);
    rightMotor.write(90);
}

void loop() {
    // определение расстояния до препятствия
    int distance = analogRead(A0);

    if (distance > 50) {
        moveForward();
    } else {
        turnLeft();
        delay(500);
    }
}
```

ПРИЛОЖЕНИЕ Б

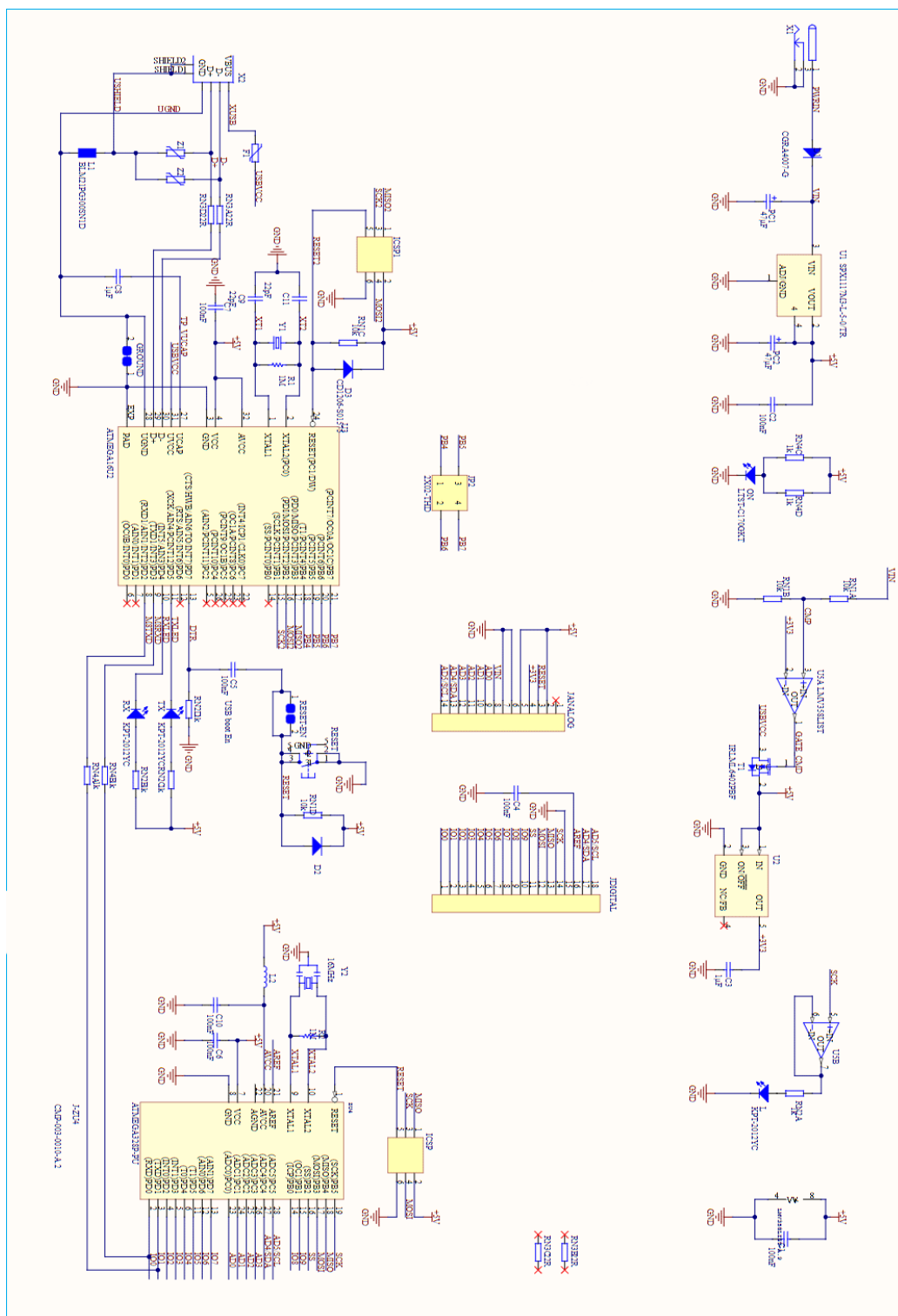


Рисунок 25 – Принципиальная схема платы Arduino Uno rev.3

ПРИЛОЖЕНИЕ В

Листинг 2 – Пример кода для финального проекта

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <Adafruit_NeoPixel.h>
#include <WiFiUdp.h>
#include <ESP_EEPROM.h>
#include <WiFiManager.h>
//https://github.com/tzapu/WiFiManager WiFi
Configuration Magic

/* Установите здесь свои SSID и пароль */
const char* ssid = "NodeMCU"; // SSID
const char* password = "12345678"; // пароль

/* Настройки IP адреса */
//IPAddress local_ip(192,168,1,1);
//IPAddress gateway(192,168,1,1);
//IPAddress subnet(255,255,255,0);
ESP8266WebServer server(80);

WiFiUDP Udp;
unsigned int localUdpPort = 65506; // local
port to listen on

//char replyPacket[] = "Hi there! Got the
message :-)"; // a reply string to send back
int packet_cnt = 0;
char incomingPacket[1090]; // buffer for
incoming packets

// Neopixel Config
#define NeoPIN D2
#define NUM_LEDS 360
int brightness = 150;
Adafruit_NeoPixel strip =
Adafruit_NeoPixel(NUM_LEDS, NeoPIN, NEO_RGB +
NEO_KHZ800);

#define LED1pin 2
#define Running_Text 1
#define WiFi_Display 2

struct MyEEPROMStruct {
    char Running_String[256]="Hello"; //
Текст бегущей строки
    uint8_t State = Running_Text; //
Режим работы дисплея. 1 - бегущая строка, 2 -
Wifi display

    uint8_t Speed = 7; // Скорость
бегущей строки
    uint8_t Color_Red = 0; // Три байта
для задания цвета букв
    uint8_t Color_Green = 20;
    uint8_t Color_Blue = 0;
    uint8_t Back_Color_Red = 0; // Три
байта для задания цвета фона
    uint8_t Back_Color_Green = 0;
    uint8_t Back_Color_Blue = 20;
    uint8_t Starts = 0;
} eepromVar1;

char newString[256] = "Hello";
#include "RunningStringFuncs_new_new.h"

uint8_t PrevState = Running_Text; //
Предыдущий режим работы дисплея. 1 - бегущая
строка, 2 - Wifi display
uint32_t ms, msl = 0;
int j, k = 0;

void setup()
{
    Serial.begin(115200);
    pinMode(LED1pin, OUTPUT);
    pinMode(NeoPIN, OUTPUT);

    //WiFi.softAP(ssid, password);
    //WiFi.softAPConfig(local_ip, gateway,
    subnet);
    WiFiManager wifiManager;
    wifiManager.autoConnect(ssid, password);

    EEPROM.begin(sizeof(MyEEPROMStruct));

    if(EEPROM.percentUsed()>=0) {
        EEPROM.get(0, eepromVar1);

        Serial.println("EEPROM has data from a
previous run.");
        Serial.print(EEPROM.percentUsed());
        Serial.println("% of ESP flash space
currently used");
    }
}
```

```

    // Увеличиваем счетчик стартов для
    контроля перезапусков
    eepromVar1.Starts += 1;
    Serial.print("Starts: ");
    Serial.println(eepromVar1.Starts);
} else {
    Serial.println("EEPROM size changed -
EEPROM data zeroed - commit() to make
permanent");
    // set the EEPROM data ready - для
    записи значений по умолчанию в первый запуск.

}
EEPROM.put(0, eepromVar1);
// write the data to EEPROM
boolean ok = EEPROM.commit();
Serial.println((ok) ? "Commit OK" : "Commit
failed");

server.on("/", HTTP_GET, handleRoot);
// Call the 'handleRoot' function when a
client requests URI "/"
server.on("/set", HTTP_POST, handleSet); //
Call the 'handleSet' function when a POST
request is made to URI "/set"
server.onNotFound(handleNotFound);
// When a client requests an unknown URI
(i.e. something other than "/"), call
function "handleNotFound"

server.begin();
Serial.println("HTTP server started");

// Start UDP server
Udp.begin(localUdpPort);
Serial.printf("Now listening at IP %s, UDP
port %d\n",
WiFi.localIP().toString().c_str(),
localUdpPort);
// start the strip and blank it out
strip.begin();
strip.show();

strncpy(newString,
WiFi.localIP().toString().c_str(), 255);
LedsPrintString();
LedsPrintString();
}

void loop(void){
    ms = millis();
    yield();

```

```

    if (eepromVar1.State == Running_Text)
    // Если выбран режим бегущей строки
    {
        // setting whole strip to the given
        color
        for(int i=0; i < NUM_LEDS; i++)
        {
            strip.setPixelColor(i, strip.Color(
eepromVar1.Back_Color_Green,
eepromVar1.Back_Color_Red,
eepromVar1.Back_Color_Blue ) );
        }
        // тут вызывается MakeNormalString и
        прочее...

        Make_Normal_String();
        LedsPrintString();

        /*// формируем бегущий пиксель
        j++;
        if (j > (25 - eepromVar1.Speed)){
            j = 0;
            k++;
            if (k > 359) {
                k =0;
            }
        }
        strip.setPixelColor(k, strip.Color(
0, 0, 0 ) );*/

        // Выводим в ленту
        //strip.show();
    }

    else if (eepromVar1.State == WiFi_Display){
    // Если выбран режим WiFi display
        server.handleClient();
    // Listen for HTTP requests from clients
        int packetSize = Udp.parsePacket();

        if (packetSize){
            // receive incoming UDP packets
            //Serial.printf("Received %d bytes from
%s, port %d\n", packetSize,
Udp.remoteIP().toString().c_str(),
Udp.remotePort());
            int len = Udp.read(incomingPacket,
1090);
            if (len > 0)
            {
                packet_cnt++;
                incomingPacket[len] = 0;
            }
        }
    }
}

```

```

    if (packet_cnt%40 < 20){
        digitalWrite(LED1pin, HIGH);
    }
    else{
        digitalWrite(LED1pin, LOW);
        // Serial.printf("Received %d bytes
from %, port %d\n", packetSize,
Udp.remoteIP().toString().c_str(),
Udp.remotePort());
    }

    }

    // Serial.printf("UDP packet contents:
%s\n", incomingPacket);

    // send back a reply, to the IP
address and port we got the packet from
    // Udp.beginPacket(Udp.remoteIP(),
Udp.remotePort());
    // Udp.write(replyPacket);
    // Udp.endPacket();

    // setting whole strip to the given
color
    for(int i=0; i < NUM_LEDS; i++) {
        j = i*3 + 6; // пропускаем заголовок
TPM2.net пакета 6 байт
        strip.setPixelColor(i, strip.Color(
incomingPacket[j+1], incomingPacket[j],
incomingPacket[j+2] ) );
    }
    // init
    yield();
    strip.show();

    }

    // Если нет пакетов UDP, все равно
периодически выводим данные на ленту, для
поддержания внешнего watchdog в спящем
состоянии
    else {
        if( ( ms - ms1 ) > 40 || ms < ms1 ){
            ms1 = ms;
            strip.show();
        }
    }

    }

}

```

```

void handleRoot() {
    // When URI / is requested, send a web page
with a button to toggle the LED

    String message = "<!DOCTYPE html
lang=\"ru-RU\" > <html>\n";
    message += "<head><meta charset=\"windows-
1251\" name=\"viewport\"
content=\"width=device-width, initial-
scale=1.0, user-scalable=no\">\n";
    message += "<title>LED Control</title>\n";
    message += "<style>html { font-family:
Helvetica; display: inline-block; margin: 0px
auto; text-align: center;}\n";
    message += "body{margin-top: 50px;} h1
{color: #444444;margin: 50px auto 30px;} h3
{color: #444444;margin-bottom: 50px;}\n";
    //message += ".superbutton {display:
block;width: 80px;background-color:
#1abc9c;border: none;color: white;padding:
13px 30px;text-decoration: none;font-size:
25px;margin: 0px auto 35px;cursor:
pointer;border-radius: 4px;}\n";
    message += ".button-set
{width:150px;height:40px;border-
radius:20px;background:#459DE5;color:#fff;fon
t-size:20px;cursor:pointer;}\n";
    message += ".button-set:hover
{background:#358DE5;}\n";
    //message += ".button-on:active {background-
color: #16a085;}\n";
    //message += ".button-off {background-color:
#34495e;}\n";
    //message += ".button-off:active
{background-color: #2c3e50;}\n";
    message += "p {font-size: 14px;color:
#000;margin-bottom: 10px;}\n";
    message += "</style>\n";
    message += "</head>\n";
    message += "<body>\n";
    message += "<h1>ESP8266 LED Display</h1>\n";
    //message += "<h3>Using WiFi Access Point
Mode</h3>\n";

    message += "<form action=\"/set\"
method=\"POST\">";

    message += "<p>Режим: <select
name=\"state\">";
    if (eepromVar1.State == Running_Text) {
        message += "<option selected
value=\"1\"> Берущая строка </option>";

```



```

if ((eepromVar1.State == Running_Text) &&
    (PrevState == EEPROMVar1.State)){ //
Считываем параметры бегущей строки только в
нужном режиме, чтобы значения не потерялись.
    // Заносим в переменную текст строки
    strncpy(eepromVar1.Running_String,
server.arg("r_string").c_str(), 255);
    EEPROMVar1.Speed =
server.arg("speed").toInt();
    EEPROMVar1.Color_Red =
server.arg("c_red").toInt();
    EEPROMVar1.Color_Green =
server.arg("c_green").toInt();
    EEPROMVar1.Color_Blue =
server.arg("c_blue").toInt();
    EEPROMVar1.Back_Color_Red =
server.arg("b_red").toInt();
    EEPROMVar1.Back_Color_Green =
server.arg("b_green").toInt();
    EEPROMVar1.Back_Color_Blue =
server.arg("b_blue").toInt();
}

// write the data to EEPROM
EEPROM.begin(sizeof(MyEEPROMStruct));
EEPROM.put(0, EEPROMVar1);
boolean ok = EEPROM.commit();
Serial.println((ok) ? "Commit OK" : "Commit
failed");

// Выводим в Serial port для диагностики

Serial.print("State: ");
Serial.println(EEPROMVar1.State);
Serial.print("Text: ");
Serial.println(EEPROMVar1.Running_String);
Serial.print("Speed: ");
Serial.println(EEPROMVar1.Speed);
Serial.print("Red: ");
Serial.println(EEPROMVar1.Color_Red);
Serial.print("Green: ");
Serial.println(EEPROMVar1.Color_Green);
Serial.print("Blue: ");
Serial.println(EEPROMVar1.Color_Blue);
Serial.print("Back_Red: ");
Serial.println(EEPROMVar1.Back_Color_Red);
Serial.print("Back_Green: ");
Serial.println(EEPROMVar1.Back_Color_Green);
Serial.print("Back_Blue: ");
Serial.println(EEPROMVar1.Back_Color_Blue);

handleRoot(); // обновляем страницу
}

void handleNotFound() {
    server.send(404, "text/plain", "404: Not
found"); // Send HTTP status 404 (Not Found)
when there's no handler for the URI in the
request
}

```

ПРИЛОЖЕНИЕ Г

Листинг 3 – Пример кода для финального робота

```
char A[] = { 0,0,0, 1,1,1, 1,0,1, 1,1,1, 1,0,1, 1,0,1 };
char B[] = { 0,0,0,0, 1,1,1,0, 1,0,1,0, 1,1,1,1, 1,0,0,1, 1,1,1,1 };
char C[] = { 0,0,0, 1,1,1, 1,0,0, 1,0,0, 1,0,0, 1,1,1 };
char D[] = { 0,0,0,0, 1,1,1,0, 1,0,0,1, 1,0,0,1, 1,0,0,1, 1,1,1,0 };
char E[] = { 0,0,0, 1,1,1, 1,0,0, 1,1,0, 1,0,0, 1,1,1 };
char F[] = { 0,0,0, 1,1,1, 1,0,0, 1,1,1, 1,0,0, 1,0,0 };
char G[] = { 0,0,0,0, 1,1,1,1, 1,0,0,0, 1,0,1,1, 1,0,0,1, 1,1,1,1 };
char H[] = { 0,0,0, 1,0,1, 1,0,1, 1,1,1, 1,0,1, 1,0,1 };
char I[] = { 0,0,0, 1,1,1, 0,1,0, 0,1,0, 0,1,0, 1,1,1 };
char J[] = { 0,0,0,0, 1,1,1,1, 0,0,1,0, 0,0,1,0, 1,0,1,0, 1,1,0,0 };
char K[] = { 0,0,0,0, 1,0,0,1, 1,0,1,0, 1,1,0,0, 1,0,1,0, 1,0,0,1 };
char L[] = { 0,0,0, 1,0,0, 1,0,0, 1,0,0, 1,0,0, 1,1,1 };
char M[] = { 0,0,0,0,0, 1,0,0,0,1, 1,1,0,1,1, 1,0,1,0,1, 1,0,0,0,1, 1,0,0,0,1 };
char N[] = { 0,0,0,0, 1,0,0,1, 1,1,0,1, 1,0,1,1, 1,0,0,1, 1,0,0,1 };
char O[] = { 0,0,0, 1,1,1, 1,0,1, 1,0,1, 1,0,1, 1,1,1 };
char P[] = { 0,0,0, 1,1,1, 1,0,1, 1,1,1, 1,0,0, 1,0,0 };
char Q[] = { 0,0,0,0, 1,1,1,0, 1,0,1,0, 1,0,1,0, 1,0,1,0, 1,1,1,1 };
char R[] = { 0,0,0, 1,1,1, 1,0,1, 1,1,0, 1,0,1, 1,0,1 };
char S[] = { 0,0,0, 0,1,1, 1,0,0, 1,1,1, 0,0,1, 1,1,0 };
char T[] = { 0,0,0, 1,1,1, 0,1,0, 0,1,0, 0,1,0, 0,1,0 };
char U[] = { 0,0,0, 1,0,1, 1,0,1, 1,0,1, 1,0,1, 1,1,1 };
char V[] = { 0,0,0,0,0, 1,0,0,0,1, 1,0,0,0,1, 0,1,0,1,0, 0,1,0,1,0, 0,0,1,0,0 };
char W[] = { 0,0,0,0,0,0,0, 1,0,0,0,0,0,1, 1,0,0,0,0,0,1, 0,1,0,1,0,1,0, 0,1,0,1,0,1,0,
0,0,1,0,1,0,0 };
char X[] = { 0,0,0, 1,0,1, 1,0,1, 0,1,0, 1,0,1, 1,0,1 };
char Y[] = { 0,0,0, 1,0,1, 1,0,1, 1,1,1, 0,1,0, 0,1,0 };
char Z[] = { 0,0,0, 1,1,1, 0,0,1, 0,1,0, 1,0,0, 1,1,1 };
char rus_B[] = { 0,0,0,0, 1,1,1,1, 1,0,0,0, 1,1,1,1, 1,0,0,1, 1,1,1,1 };
char rus_G[] = { 0,0,0, 1,1,1, 1,0,0, 1,0,0, 1,0,0, 1,0,0 };
char rus_D[] = { 0,0,0,0,0, 0,0,1,1,0, 0,1,0,1,0, 0,1,0,1,0, 0,1,0,1,0, 1,1,1,1,1 };
char rus_J[] = { 0,0,0,0,0, 1,0,1,0,1, 1,0,1,0,1, 0,1,1,1,0, 1,0,1,0,1, 1,0,1,0,1 };
char rus_Z[] = { 0,0,0,0, 1,1,1,0, 0,0,1,0, 1,1,1,1, 0,0,0,1, 1,1,1,1 };
char rus_I[] = { 0,0,0,0, 1,0,0,1, 1,0,0,1, 1,0,1,1, 1,1,0,1, 1,0,0,1 };
char rus_sI[] = { 0,1,1,0, 1,0,0,1, 1,0,0,1, 1,0,1,1, 1,1,0,1, 1,0,0,1 };
char rus_L[] = { 0,0,0, 0,1,1, 1,0,1, 1,0,1, 1,0,1, 1,0,1 };
char rus_P[] = { 0,0,0, 1,1,1, 1,0,1, 1,0,1, 1,0,1, 1,0,1 };
char rus_U[] = { 0,0,0, 1,0,1, 1,0,1, 1,1,1, 0,0,1, 1,1,0 };
char rus_F[] = { 0,0,0,0,0, 0,0,1,0,0, 1,1,1,1,1, 1,0,1,0,1, 1,1,1,1,1, 0,0,1,0,0 };
char rus_TS[] = { 0,0,0,0, 1,0,1,0, 1,0,1,0, 1,0,1,0, 1,0,1,0, 1,1,1,1 };
char rus_CH[] = { 0,0,0, 1,0,1, 1,0,1, 1,1,1, 0,0,1, 0,0,1 };
char rus_SH[] = { 0,0,0,0,0, 1,0,1,0,1, 1,0,1,0,1, 1,0,1,0,1, 1,0,1,0,1, 1,1,1,1,1 };
char rus_SHT[] = { 0,0,0,0,0,0, 1,0,1,0,1,0, 1,0,1,0,1,0, 1,0,1,0,1,0, 1,0,1,0,1,0, 1,1,1,1,1,1,1
};
char rus_TZ[] = { 0,0,0,0, 1,1,0,0, 0,1,0,0, 0,1,1,1, 0,1,0,1, 0,1,1,1 };
char rus_YI[] = { 0,0,0,0,0, 1,0,0,0,1, 1,0,0,0,1, 1,1,1,0,1, 1,0,1,0,1, 1,1,1,0,1 };
char rus_MZ[] = { 0,0,0, 1,0,0, 1,0,0, 1,1,1, 1,0,1, 1,1,1 };
char rus_IY[] = { 0,0,0, 1,1,0, 0,0,1, 0,1,1, 0,0,1, 1,1,0 };
char rus_YOU[] = { 0,0,0,0,0, 1,0,1,1,1, 1,0,1,0,1, 1,1,1,0,1, 1,0,1,0,1, 1,0,1,1,1 };
char rus_YA[] = { 0,0,0, 1,1,1, 1,0,1, 0,1,1, 1,0,1, 1,0,1 };
char one[] = { 0,0, 1,1, 0,1, 0,1, 0,1, 0,1 };
char two[] = { 0,0,0, 1,1,1, 0,0,1, 1,1,1, 1,0,0, 1,1,1 };
char three[] = { 0,0,0, 1,1,1, 0,0,1, 1,1,1, 0,0,1, 1,1,1 };
char four[] = { 0,0,0, 1,0,1, 1,0,1, 1,1,1, 0,0,1, 0,0,1 };
char five[] = { 0,0,0, 1,1,1, 1,0,0, 1,1,1, 0,0,1, 1,1,1 };
```

```

char six[] = { 0,0,0, 1,1,1, 1,0,0, 1,1,1, 1,0,1, 1,1,1 };
char seven[] = { 0,0,0, 1,1,1, 0,0,1, 0,0,1, 0,0,1, 0,0,1 };
char eight[] = { 0,0,0, 1,1,1, 1,0,1, 1,1,1, 1,0,1, 1,1,1 };
char nine[] = { 0,0,0, 1,1,1, 1,0,1, 1,1,1, 0,0,1, 1,1,1 };
char zero[] = { 0,0,0, 1,1,1, 1,0,1, 1,0,1, 1,0,1, 1,1,1 };
char error[] = { 1,1,1,1,1,1,1, 1,0,0,1,0,0,0, 1,1,0,1,0,1,0, 1,0,1,1,1,0,0, 1,0,1,0,1,0,0,
1,1,1,0,1,1,0 };
char space[] = {0,0,0,0,0,0};
char big_space[] = {0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0};
char dot[] = { 0, 0, 0, 0, 0, 1 };
char comma[] = { 0,0, 0,0, 0,0, 0,0, 0,1, 1,0 };
char exclamation_mark[] = { 0, 1, 1, 1, 0, 1 };
char question_mark[] = { 0,0,0, 1,1,1, 0,0,1, 0,1,1, 0,0,0, 0,1,0 };
char hyphen[] = { 0,0,0, 0,0,0, 0,0,0, 1,1,1, 0,0,0, 0,0,0 };
char underlining[] = { 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 1,1,1,1,1 };
char round_bracket_open[] = { 0,0, 0,1, 1,0, 1,0, 1,0, 0,1 };
char round_bracket_close[] = { 0,0, 1,0, 0,1, 0,1, 0,1, 1,0 };
char slash[] = { 0,0,0, 0,0,1, 0,0,1, 0,1,0, 0,1,0, 1,0,0 };
char backslash[] = { 0,0,0, 1,0,0, 1,0,0, 0,1,0, 0,1,0, 0,0,1 };
char quotation_mark[] = { 1,0,1, 1,0,1, 0,0,0, 0,0,0, 0,0,0, 0,0,0 };
char grave[] = { 1, 1, 0, 0, 0, 0 };
char colon[] = { 0, 1, 0, 0, 0, 1 };
char semicolon[] = { 0,0, 0,1, 0,0, 0,0, 0,1, 1,0 };
char angle_bracket_open[] = { 0,0,0, 0,0,1, 0,1,0, 1,0,0, 0,1,0, 0,0,1 };
char angle_bracket_close[] = { 0,0,0, 1,0,0, 0,1,0, 0,0,1, 0,1,0, 1,0,0 };
char vertical_bar[] = { 0, 1, 1, 1, 1, 1 };
char tilde[] = { 0,0,0,0, 0,0,0,0, 1,1,0,1, 1,0,1,1, 0,0,0,0, 0,0,0,0 };
char equals[] = { 0,0,0, 0,0,0, 1,1,1, 0,0,0, 1,1,1, 0,0,0 };
char square_bracket_open[] = { 0,0, 1,1, 1,0, 1,0, 1,0, 1,1 };
char square_bracket_close[] = { 0,0, 1,1, 0,1, 0,1, 0,1, 1,1 };
char star[] = { 0,0,0, 0,0,0, 1,0,1, 0,1,0, 1,0,1, 0,0,0 };
char plus[] = {0,0,0, 0,0,0, 0,1,0, 1,1,1, 0,1,0, 0,0,0};
char degree[] = {1,1,1, 1,0,1, 1,1,1, 0,0,0, 0,0,0, 0,0,0};

```

ПРИЛОЖЕНИЕ Д

Листинг 4 – Пример кода для финального робота

```
#pragma once
#include "Symbols.h"

void ToString(char* symbol, size_t size);

void Make_Normal_String()
{
    strncpy(newString, "", 255);
    for (size_t i = 0; i <
        strlen(eepromVar1.Running_String); i++)
    {
        newString[i] =
eepromVar1.Running_String[i];
        if
(eepromVar1.Running_String[i] >= 'À' &&
eepromVar1.Running_String[i] <= 'İ')
            newString[i] +=
'À' - 'à';
        else if
(eepromVar1.Running_String[i] >= 'Ð' &&
eepromVar1.Running_String[i] <= 'Ÿ')
            newString[i] +=
'D' - 'ð';
        else if
(eepromVar1.Running_String[i] >= 'a' &&
eepromVar1.Running_String[i] <= 'z')
            newString[i] +=
'A' - 'a';
        else if
(eepromVar1.Running_String[i] == ',' ||
eepromVar1.Running_String[i] == '"')
            newString[i] =
'E';
        switch (newString[i]) {
            case ('À'):
                newString[i] =
'A';
                break;
            case ('Ä'):
                newString[i] =
'B';
                break;
            case ('Å'):
                newString[i] =
'E';
                break;
            case ('Ê'):
                newString[i] =
'K';
                break;
            case ('İ'):
                newString[i] =
'M';
                break;
            case ('ı'):
                newString[i] =
'H';
                break;
            case ('î'):
                newString[i] =
'O';
                break;
            case ('ï'):
                newString[i] =
'P';
                break;
            case ('Ñ'):
                newString[i] =
'C';
                break;
            case ('Ö'):
                newString[i] =
'T';
                break;
            case ('Õ'):
                newString[i] =
'X';
                break;
            default:
                break;
        }
    }
}

void LedsPrintString()
{
    for (size_t i = 0; i <
        strlen(newString); i++)
    {
        switch (newString[i])
        {
            case ('A'):
            {
                newString[i] =
                break;
            }
        }
    }
}
```



```

        ToString(A,
sizeof(A) / 6);
        break;
    }
    case('B'):
    {
        ToString(B,
sizeof(B) / 6);
        break;
    }
    case('C'):
    {
        ToString(C,
sizeof(C) / 6);
        break;
    }
    case('D'):
    {
        ToString(D,
sizeof(D) / 6);
        break;
    }
    case('E'):
    {
        ToString(E,
sizeof(E) / 6);
        break;
    }
    case('F'):
    {
        ToString(F,
sizeof(F) / 6);
        break;
    }
    case('G'):
    {
        ToString(G,
sizeof(G) / 6);
        break;
    }
    case('H'):
    {
        ToString(H,
sizeof(H) / 6);
        break;
    }
    case('I'):
    {
        ToString(I,
sizeof(I) / 6);
        break;
    }
    case('J'):
    {
        ToString(J,
sizeof(J) / 6);
        break;
    }
    case('K'):
    {
        ToString(K,
sizeof(K) / 6);
        break;
    }
    case('L'):
    {
        ToString(L,
sizeof(L) / 6);
        break;
    }
    case('M'):
    {
        ToString(M,
sizeof(M) / 6);
        break;
    }
    case('N'):
    {
        ToString(N,
sizeof(N) / 6);
        break;
    }
    case('O'):
    {
        ToString(O,
sizeof(O) / 6);
        break;
    }
    case('P'):
    {
        ToString(P,
sizeof(P) / 6);
        break;
    }
    case('Q'):
    {
        ToString(Q,
sizeof(Q) / 6);
        break;
    }
    case('R'):
    {
        ToString(R,
sizeof(R) / 6);
        break;
    }

```

```

        case('S'):
        {
            ToString(S,
sizeof(S) / 6);
            break;
        }
        case('T'):
        {
            ToString(T,
sizeof(T) / 6);
            break;
        }
        case('U'):
        {
            ToString(U,
sizeof(U) / 6);
            break;
        }
        case('V'):
        {
            ToString(V,
sizeof(V) / 6);
            break;
        }
        case('W'):
        {
            ToString(W,
sizeof(W) / 6);
            break;
        }
        case('X'):
        {
            ToString(X,
sizeof(X) / 6);
            break;
        }
        case('Y'):
        {
            ToString(Y,
sizeof(Y) / 6);
            break;
        }
        case('Z'):
        {
            ToString(Z,
sizeof(Z) / 6);
            break;
        }
        case('Á'):
        {
            ToString(rus_B,
sizeof(rus_B) / 6);
            break;
        }
    }
    case('Ä'):
    {
        ToString(rus_G,
sizeof(rus_G) / 6);
        break;
    }
    case('Å'):
    {
        ToString(rus_D,
sizeof(rus_D) / 6);
        break;
    }
    case('Æ'):
    {
        ToString(rus_J,
sizeof(rus_J) / 6);
        break;
    }
    case('Ç'):
    {
        ToString(rus_Z,
sizeof(rus_Z) / 6);
        break;
    }
    case('È'):
    {
        ToString(rus_I,
sizeof(rus_I) / 6);
        break;
    }
    case('É'):
    {
        ToString(rus_sI,
sizeof(rus_sI) / 6);
        break;
    }
    case('Ê'):
    {
        ToString(rus_L,
sizeof(rus_L) / 6);
        break;
    }
    case('Ë'):
    {
        ToString(rus_P,
sizeof(rus_P) / 6);
        break;
    }
    case('Ó'):
    {
        ToString(rus_U,
sizeof(rus_U) / 6);
        break;
    }
}

```

```

        break;
    }
    case('Ö'):
    {
        ToString(rus_F,
sizeof(rus_F) / 6);
        break;
    }
    case('Û'):
    {
        ToString(rus_TS,
sizeof(rus_TS) / 6);
        break;
    }
    case('×'):
    {
        ToString(rus_CH,
sizeof(rus_CH) / 6);
        break;
    }
    case('Ø'):
    {
        ToString(rus_SH,
sizeof(rus_SH) / 6);
        break;
    }
    case('Ü'):
    {
        ToString(rus_SHT, sizeof(rus_SHT) /
6);
        break;
    }
    case('Ů'):
    {
        ToString(rus_TZ,
sizeof(rus_TZ) / 6);
        break;
    }
    case('Ű'):
    {
        ToString(rus_YI,
sizeof(rus_YI) / 6);
        break;
    }
    case('Ů'):
    {
        ToString(rus_MZ,
sizeof(rus_MZ) / 6);
        break;
    }
    case('Ÿ'):

```

```

    {
        ToString(rus_IY,
sizeof(rus_IY) / 6);
        break;
    }
    case('Ð'):
    {
        ToString(rus_YOU, sizeof(rus_YOU) /
6);
        break;
    }
    case('ß'):
    {
        ToString(rus_YA,
sizeof(rus_YA) / 6);
        break;
    }
    case(' '):
    {
        ToString(big_space,
sizeof(big_space) / 6);
        break;
    }
    case('1'):
    {
        ToString(one,
sizeof(one) / 6);
        break;
    }
    case('2'):
    {
        ToString(two,
sizeof(two) / 6);
        break;
    }
    case('3'):
    {
        ToString(three,
sizeof(three) / 6);
        break;
    }
    case('4'):
    {
        ToString(four,
sizeof(four) / 6);
        break;
    }
    case('5'):
    {
        ToString(five,
sizeof(five) / 6);

```

```

        break;
    }
    case('6'):
    {
        ToString(six,
sizeof(six) / 6);
        break;
    }
    case('7'):
    {
        ToString(seven,
sizeof(seven) / 6);
        break;
    }
    case('8'):
    {
        ToString(eight,
sizeof(eight) / 6);
        break;
    }
    case('9'):
    {
        ToString(nine,
sizeof(nine) / 6);
        break;
    }
    case('0'):
    {
        ToString(zero,
sizeof(zero) / 6);
        break;
    }
    case(' . '):
    {
        ToString(dot,
sizeof(dot) / 6);
        break;
    }
    case(' , '):
    {
        ToString(comma,
sizeof(comma) / 6);
        break;
    }
    case(' ! '):
    {
        ToString(exclamation_mark,
sizeof(exclamation_mark) / 6);
        break;
    }
    case(' ? '):
    {

```

```

        ToString(question_mark,
sizeof(question_mark) / 6);
        break;
    }
    case(' - '):
    {
        ToString(hyphen,
sizeof(hyphen) / 6);
        break;
    }
    case(' _ '):
    {
        ToString(underlining,
sizeof(underlining) / 6);
        break;
    }
    case(' ( '):
    {
        ToString(round_bracket_open,
sizeof(round_bracket_open) / 6);
        break;
    }
    case(' ) '):
    {
        ToString(round_bracket_close,
sizeof(round_bracket_close) / 6);
        break;
    }
    case(' / '):
    {
        ToString(slash,
sizeof(slash) / 6);
        break;
    }
    case(' \ \ '):
    {
        ToString(backslash,
sizeof(backslash) / 6);
        break;
    }
    case(' " '):
    {
        ToString(quotation_mark,
sizeof(quotation_mark) / 6);
        break;
    }
    case(' \ ' '):
    {

```

```

        {
            ToString(grave,
sizeof(grave) / 6);
            break;
        }
        case (':'):
        {
            ToString(colon,
sizeof(colon) / 6);
            break;
        }
        case (';'):
        {
            ToString(semicolon,
sizeof(semicolon) / 6);
            break;
        }
        case ('<'):
        {
            ToString(angle_bracket_open,
sizeof(angle_bracket_open) / 6);
            break;
        }
        case ('>'):
        {
            ToString(angle_bracket_close,
sizeof(angle_bracket_close) / 6);
            break;
        }
        case ('|'):
        {
            ToString(vertical_bar,
sizeof(vertical_bar) / 6);
            break;
        }
        case ('~'):
        {
            ToString(tilde,
sizeof(tilde) / 6);
            break;
        }
        case ('['):
        {
            ToString(square_bracket_open,
sizeof(square_bracket_open) / 6);
            break;
        }
        case (']'):

```

```

        {
            ToString(square_bracket_close,
sizeof(square_bracket_close) / 6);
            break;
        }
        case ('='):
        {
            ToString(equals,
sizeof(equals) / 6);
            break;
        }
        case ('*'):
        {
            ToString(star,
sizeof(star) / 6);
            break;
        }
        case ('°'):
        {
            ToString(degree,
sizeof(degree) / 6);
            break;
        }
        case ('+'):
        {
            ToString(plus,
sizeof(plus) / 6);
            break;
        }
        default:
        {
            ToString(error,
sizeof(error) / 6);
            break;
        }
        ToString(space, 1);
    }
    for (size_t i = 0; i < 60; i++)
    {
        ToString(space, 1);
    }
}

void SlideLeft()
{
    for (int i = 0; i < 6; i++)
    {
        if (i % 2 == 0)
        {
            for (int j = 0;
j < 59; j++)

```

```

        {
            int k = i*60 + j;

            strip.setPixelColor(k,
strip.getPixelColor(k + 1));
        }
        else
        {
            for (int j = 1;
j <= 59; j++)
            {
                int k = (i+1)*60 - j;

                strip.setPixelColor(k,
strip.getPixelColor(k - 1));
            }
        }
    }

void ToString(char* symbol, size_t size)
{
    // 360 elements in matrix!

    for (int j = 0; j < size; j++)
    {
        server.handleClient();
        // Listen for HTTP requests from clients
        SlideLeft();
        for (int i = 0; i < 6;
i++)
        {
            if (i % 2)
            {
                if
(symbol[j + i * size] == 1)
                    {
                        strip.setPixelColor(i*60,
eepromVar1.Color_Green, eepromVar1.Color_Red,
eepromVar1.Color_Blue);
                    }
                else
                {
                    strip.setPixelColor(i*60,
eepromVar1.Back_Color_Green,
eepromVar1.Back_Color_Red,
eepromVar1.Back_Color_Blue);
                }
            }
            else
            {
                if
(symbol[j + i * size] == 1)
                    {
                        strip.setPixelColor((i+1)*60 - 1,
eepromVar1.Color_Green, eepromVar1.Color_Red,
eepromVar1.Color_Blue);
                    }
                else
                {
                    strip.setPixelColor((i+1)*60 - 1,
eepromVar1.Back_Color_Green,
eepromVar1.Back_Color_Red,
eepromVar1.Back_Color_Blue);
                }
            }
        }
        delay(8 * (16 -
eepromVar1.Speed));
        strip.show();
    }
}

```