

Применение онлайн-эмулятора Wokwi для разработки и тестирования прошивок микроконтроллеров

Болтовский Гавриил Александрович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Цель статьи – анализ возможностей эмулятора Wokwi для разработки и тестирования приложений на микроконтроллерах Arduino. Описаны функциональные возможности веб-версии и интеграции с средами разработки, такими как VS Code. В результате представлены ключевые преимущества использования Wokwi в образовательных и профессиональных задачах, включая моделирование и отладку в виртуальной среде.

Ключевые слова: разработка встраиваемых систем, Arduino, эмуляция

Using Wokwi online emulator for microcontroller firmware development and testing

Boltovsky Gavriil Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The aim of the article is to analyze the capabilities of the Wokwi emulator for developing and testing applications on Arduino microcontrollers. It describes the functionality of the web version and integration with development environments such as VS Code. As a result, the key benefits of using Wokwi in educational and professional tasks, including modeling and debugging in a virtual environment, are presented.

Keywords: embedded systems development, Arduino, emulation

1 Введение

1.1 Актуальность исследования

Современная разработка микроконтроллерных систем требует удобных инструментов для моделирования, тестирования и быстрой проверки работоспособности программного кода без необходимости использования физического оборудования. Wokwi предоставляет универсальную среду, позволяющую выполнять эмуляцию различных микроконтроллеров, что снижает затраты на оборудование и ускоряет процесс разработки. Актуальность темы обусловлена растущим интересом к цифровой трансформации и потребностью в эффективных средствах обучения и прототипирования в сфере встраиваемых систем.

1.2 Обзор исследований

Основные принципы разработки на платформе Arduino были подробно рассмотрены Ю. Абдулахи [1]. Вопросы установки интегрированной среды разработки (IDE), прошивки базовых программ, работы с монитором порта и построения графиков освещены в статье М. Fezari и А. Dahoud [2]. Исторический контекст популяризации бренда Arduino в России представлен в работе Шевченко В. Л. [3].

Вход в сообщество Arduino характеризуется низким порогом для начинающих разработчиков, что делает платформу доступной даже для новичков. Для упрощения процесса программирования существуют визуальные инструменты, такие как Ardublock и miniBloq, которые позволяют создавать код в виде блоков. Это расширяет аудиторию разработчиков, включая детей и людей без технической подготовки [4].

Среди разнообразия плат Arduino особое место занимает Arduino LilyPad, разработанная для использования в умной одежде. Историю её создания и особенности применения описывает Ф. В. Патюченко [5]. Одним из практических применений LilyPad является медицинская область: в частности, удалённое наблюдение за состоянием пациентов. Такой подход предложен в работе Н. С. Бочкарёва [6].

1.3 Цель исследования

Цель исследования – анализ возможностей эмулятора Wokwi для разработки и тестирования приложений на микроконтроллерах Arduino.

1.4 Постановка задачи

Задачей данного исследования является анализ функциональных возможностей эмулятора Wokwi, его интеграции с современными инструментами разработки и программирования, а также выявление преимуществ его использования для обучения и проектирования микроконтроллерных систем.

2 Методы исследования

Эмулятор Wokwi [7] предназначен для упрощения разработки и тестирования программного обеспечения для микроконтроллеров. Его использование позволяет разработчикам проводить отладку кода и проектирование схем без необходимости физического доступа к аппаратным компонентам. Это особенно важно на этапе прототипирования, когда окончательный выбор аппаратного обеспечения может ещё не быть сделан.

Одним из ключевых преимуществ эмулятора является возможность быстрого тестирования различных конфигураций устройств и подключения модулей. Кроме того, Wokwi предоставляет доступ к большому числу виртуальных компонентов, таких как сенсоры, дисплеи, приводы и логические элементы, что позволяет моделировать сложные устройства. Это упрощает образовательный процесс для студентов и инженеров-новичков, предоставляя

им инструмент для экспериментов и обучения без необходимости покупки дорогостоящего оборудования.

Использование эмулятора также снижает риски повреждения оборудования из-за ошибок в программировании или неверного подключения модулей.

Visual Studio Code (VS Code) – это современный текстовый редактор, который используется для разработки программного обеспечения [8]. Одной из сильных сторон VS Code является система расширений. Расширения добавляют новые функции, такие как поддержка дополнительных языков, инструменты для отладки, интеграция с системами контроля версий и взаимодействие с внешними платформами. Например, для работы с микроконтроллерами или эмуляторами, такими как Wokwi, расширения позволяют автоматизировать компиляцию, настройку среды и отладку, облегчая разработку.

Эмулятор Wokwi полезен для широкой аудитории, начиная от школьников, делающих первые шаги в программировании, и заканчивая опытными разработчиками встроенных систем благодаря тому, что может работать как веб-приложение и также запускаться локально.

3 Результаты и обсуждения

Для начала работы с онлайн-эмулятором Wokwi пользователю достаточно зарегистрироваться в системе. Сервис предоставляет возможность выбора одной из поддерживаемых платформ, таких как Arduino (Uno, Mega, Nano), ESP32, STM32 и Raspberry Pi Pico (рис. 1).

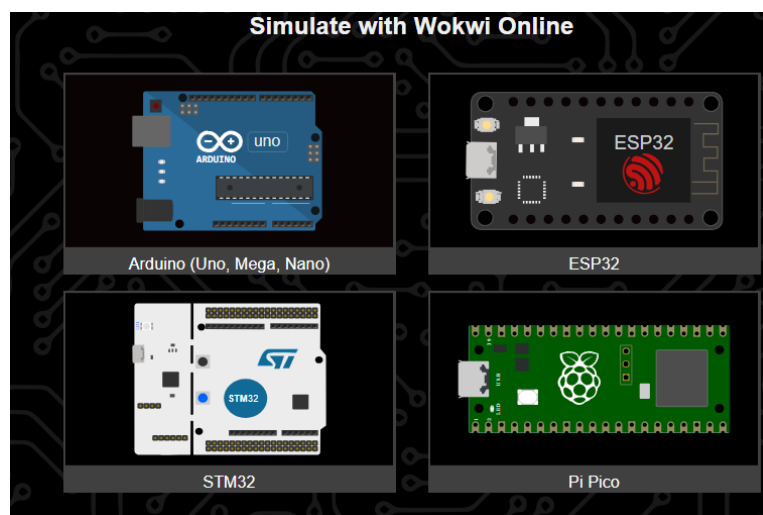


Рисунок 1 – Выбор платы для эмуляции

При выборе конкретной платформы отображаются рекомендуемые проекты и доступные шаблоны, облегчающие начальный этап разработки. Интерфейс среды представлен двумя рабочими областями: область для написания кода расположена слева, а окно для выполнения эмуляции – справа (рис. 2).

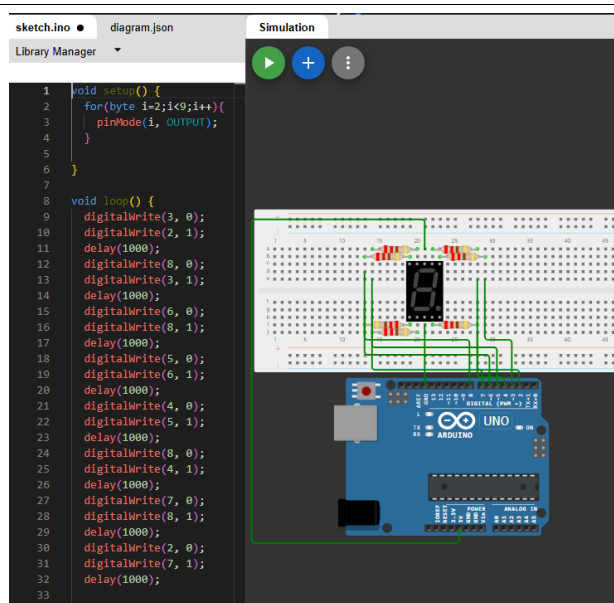


Рисунок 2 – Выбор платы для эмуляции

Эмулятор предлагает широкий набор элементов для создания виртуальных устройств, включая светодиоды, кнопки, резисторы, дисплеи (OLED, жидкокристаллические различных размеров), светодиодные матрицы, семисегментные индикаторы, драйверы, джойстики, ультразвуковые датчики, датчики температуры, фоторезисторы, акселерометры, модули реального времени (RTC), пьезоизлучатели, сервоприводы, шаговые двигатели, реле, карт-ридеры и логические вентили. Для каждого компонента доступна подробная документация, что значительно упрощает процесс разработки.

Среда разработки поддерживает установку дополнительных библиотек. Библиотеки можно добавлять как через встроенный менеджер, так и загружать в формате ZIP-архива.

Эмуляция работы устройства выполняется на удалённом сервере, однако возможны временные перебои в доступности сервиса из-за высокой нагрузки. Для решения этой проблемы разработчики предоставляют расширение для интеграции Wokwi с Visual Studio Code, позволяющее запускать эмуляцию локально на компьютере. Установка осуществляется в виде стандартного расширения для редактора.

Wokwi представляет собой исключительно эмулятор, способный исполнять только предварительно скомпилированный бинарный код. Для компиляции прошивки требуется использование сторонних инструментов. Одним из наиболее популярных решений является PlatformIO IDE [9], однако возможно применение и других аналогичных инструментов.

При настройке эмулятора нужно установить расширение в VS Code, а также получить лицензию на использование.

При создании проекта в среде PlatformIO IDE необходимо указать целевую платформу, для которой будет осуществляться разработка. Процесс написания программного кода имеет определённые отличия по сравнению с использованием Arduino IDE. В частности, функции должны быть объявлены

до их вызова, а в директивах препроцессора требуется подключать библиотеку «Arduino.h».

После завершения написания кода прошивка компилируется, в результате чего в директории «.pio» создаются временные файлы, включая «firmware.hex» и «firmware.elf». Файл «firmware.hex» содержит бинарный код прошивки, предназначенный для загрузки в микроконтроллер, тогда как файл «firmware.elf» включает отладочную информацию. Оба файла необходимо указать в конфигурационном файле «wokwi.toml» для дальнейшей работы с эмулятором:

```
[wokwi]
version = 1
firmware = '\.pio\build\nanoatmega328\firmware.hex'
elf = '\.pio\build\nanoatmega328\firmware.elf'
```

Для полноценной работы с эмулятором требуется также файл «diagram.json», в котором задаётся описание подключения модулей к выбранной плате.

Создание всех необходимых файлов позволяет пользователю полностью осуществлять отладку своей прошивки локально, непосредственно в среде разработки (рис. 3). Это обеспечивает разработчику независимость от доступности серверов эмулятора, что особенно важно при высокой нагрузке на облачную инфраструктуру.

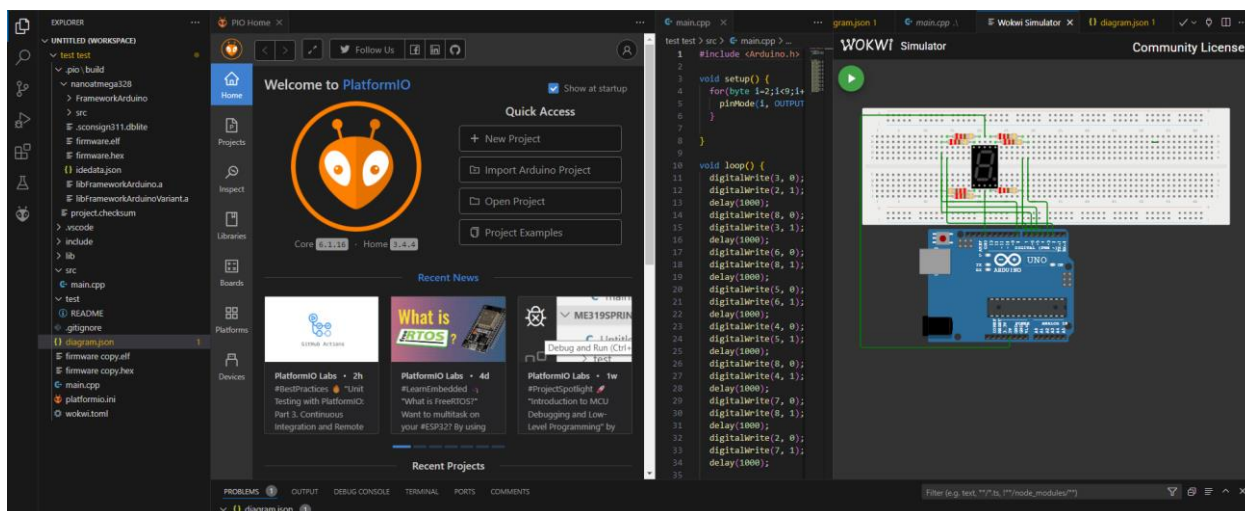


Рисунок 3 – Работа с эмулятором локально из VS Code

Переход на использование эмуляторов, таких как Wokwi, и более продвинутых инструментов разработки, таких как Visual Studio Code (VS Code), зависит от уровня задач, стоящих перед разработчиком, и их сложности.

Эмуляция становится особенно полезной в ситуациях, когда необходимо ускорить процесс разработки, снизить затраты на оборудование или протестировать сложные сценарии, недоступные в реальных условиях.

Переход с Arduino IDE на VS Code чаще всего является закономерным шагом для разработчиков, выходящих за рамки начального уровня. Arduino IDE удобна своей простотой и подходит для обучения или небольших проектов. Однако по мере увеличения сложности задач и необходимости работы с внешними библиотеками, интеграцией систем контроля версий, отладкой и тестированием, ограниченность функционала Arduino IDE становится очевидной.

3. Выводы

Таким образом, использование эмулятора Wokwi значительно упрощает процесс разработки и тестирования микроконтроллерных приложений, обеспечивая доступность и универсальность для образовательных и профессиональных задач.

Библиографический список

1. Yusuf A. The Working Principle Of An Arduino // ACADEMIA. 2014. С. 1-3. URL: <https://www.academia.edu/40296673> (дата обращения: 31.05.2021).
2. Mohamed F., Ali D. Integrated Development Environment “IDE” For Arduino // YA Badamasi 11th international conference. 2014. С. 1-2. URL: <https://www.researchgate.net/publication/328615543> (дата обращения: 31.05.2021).
3. Шевченко В.Л. Платформа Arduino как развитие познания людей в области электроники и программирования. // Человек и Вселенная, 2020. № 1 (99). С. 41-44. URL: <https://elibrary.ru/item.asp?id=42956009> (дата обращения: 31.05.2021).
4. Pratomo A. B., Perdana, R. S. Arduviz, a visual programming IDE for arduino. // International Conference on Data and Software Engineering (ICoDSE). 2017. С. 4-6.
5. Патюченко Ф.В. ARDUINO LILYPAD // Modern Science, 2020. С. 329-331. URL: <https://elibrary.ru/item.asp?id=42316004> (дата обращения: 31.05.2021).
6. Бочкарёв Н.С. Разработка системы удаленного мониторинга за состоянием пациента // Информация и образование: границы коммуникаций, 2017. С. 95-96.
7. Wokwi. URL: <https://wokwi.com/> (дата обращения: 28.01.2025).
8. Visual Studio Code. URL: <https://code.visualstudio.com/> (дата обращения: 28.01.2025).
9. PlatformIO. URL: <https://platformio.org/> (дата обращения: 28.01.2025).