Антипов Дмитрий Николаевич
магистр, педагог дополнительного образования
МАОУ ДО «Детский технопарк «Кванториум»
г. Чебоксары, Чувашская Республика

ПРИМЕНЕНИЕ ПЛАТФОРМЫ MAKECODE В ОБУЧЕНИИ ДЕТЕЙ ПРОГРАММИРОВАНИЮ

Аннотация: *в статье рассматривается методика применения программной среды Microsoft MakeCode (фреймворк PXT) в обучении детей программированию и робототехнике. Обобщен опыт работы с конструктором LEGO MINDSTORMS EV3 на базе развиваемого форка BrickCode, средой MakeCode Arcade и платами micro:bit. Представлен дифференцированный подход к выбору формата кода с учетом возраста учащихся: от визуальных блоков для младших школьников до прямой работы на Static TypeScript и Static Python для старшеклассников. Проанализированы функции аппаратного симулятора, встроенных инструментов диагностики ошибок и педагогический потенциал открытого исходного кода.*

Ключевые слова: *обучение программированию, образовательная робототехника, встраиваемые системы, Microsoft MakeCode, BrickCode, BBC micro:bit, блочное программирование, текстовое программирование, LEGO MINDSTORMS EV3, Static Python, Static TypeScript.*

Введение

Эффективная организация обучения детей программированию требует выбора среды разработки, объединяющей низкий порог входа для начинающих и достаточную глубину для решения инженерных задач. Традиционное визуальное программирование (Blockly или Scratch) исключает синтаксические ошибки, однако по мере усложнения проектов становится ограничивающим фактором. В то же время прямой переход на классические текстовые среды сопряжен с трудоемкой настройкой локальных компиляторов и резким возрастанием синтаксической нагрузки на обучающегося.

Content is licensed under the Creative Commons Attribution 4.0 license (CC-BY 4.0)

Оптимальным решением этой дидактической проблемы выступает открытая платформа Microsoft MakeCode – веб-ориентированная среда, созданная на базе фреймворка PXT (Programming Experience Toolkit) [1]. Среда объединяет блочный конструктор, строгие текстовые языки программирования и аппаратную симуляцию в рамках единого веб-интерфейса.

Архитектурные особенности, симулятор и диагностика ошибок

Основой MakeCode является бесшовная двусторонняя трансляция между тремя форматами представления алгоритма [1, 3]:

1. Визуальные блоки (Google Blockly): типизированные цветные блоки для освоения алгоритмических структур.

2. Static TypeScript: строго типизированное подмножество JavaScript, компилируемое непосредственно в машинный код микроконтроллера.

3. Static Python (MakeCode Python): статически типизированное подмножество Python, транслируемое через абстрактное синтаксическое дерево фреймворка PXT [3].

Изменение программы в любом из режимов мгновенно синхронизирует остальные представления. При этом боковая панель категорий в текстовом редакторе служит интерактивным справочником API, позволяющим быстро вставлять сигнатуры методов с подсказками типов аргументов.

Важнейшим компонентом платформы является интерактивный симулятор. Он запускает виртуальную модель устройства прямо в браузере, позволяя моделировать показания датчиков, проверять кинематику и выявлять логические ошибки до загрузки кода на физический контроллер. Режим пошагового выполнения и встроенный журнал данных формируют у школьников культуру осознанной отладки. Поддержка технологии PWA гарантирует автономную работу симулятора и редактора без постоянного подключения к сети Интернет [3].

В современных официальных версиях MakeCode (Arcade, micro:bit) также присутствует функция облачного ИИ-ассистента, который помогает анализировать ошибки компиляции и на доступном языке поясняет причину сбоя в коде [3].

Практические направления применения MakeCode

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)

Экосистема MakeCode охватывает три ключевых профиля подготовки:

1. Робототехника на базе LEGO MINDSTORMS EV3 (редактор BrickCode). Официальный редактор MakeCode для EV3 от Microsoft перестал обновляться, а штатное ПО EV3 Classroom не обеспечивает полноценного перехода к текстовому коду и лишено множества важных функциональных возможностей: в нем отсутствует подключение между контроллерами по Bluetooth, получение необработанных («сырых») значений с датчика цвета и ряда других сенсоров, а также поддержка датчиков прошлого поколения NXT. В связи с этим в учебном процессе используется BrickCode – активно развиваемый и регулярно обновляемый независимый форк [5]. В нем актуализирована кодовая база, исправлены системные ошибки, реализованы блоки для расширенной работы с периферией (включая сырые данные датчиков, Bluetooth-связь между модулями и поддержку сенсоров NXT).

В соревновательной робототехнике редактор позволяет:

- организовывать многозадачность через параллельные фоновые задачи;
- реализовывать алгоритмы управления движением (пропорциональные и ПИД-регуляторы следования по линии, выравнивание по гироскопу);
- применять модульный подход за счет создания пользовательских функций с аргументами и возвращаемыми значениями.

2. Разработка игр и алгоритмика (MakeCode Arcade). Редактор ориентирован на создание 2D-игр с пиксельной графикой (160×120 пикселей) [3]. Высокая личная вовлеченность детей позволяет глубоко изучить декартову систему координат, векторы скорости и ускорения, физику столкновения спрайтов и работу с двумерными массивами тайловых карт. Созданные проекты компилируются в бинарные файлы для загрузки на аппаратные консоли на базе микроконтроллеров RP2040, STM32 или SAMD51.

3. Встраиваемые системы и микроконтроллеры (BBC micro:bit). Редактор MakeCode for micro:bit является образцовым инструментом физического программирования для погружения школьников в мир схемотехники и встроенных систем [2]:

Content is licensed under the Creative Commons Attribution 4.0 license (CC-BY 4.0)

- интегрированная аппаратная периферия: наличие встроенной светодиодной матрицы 5×5 позволяет визуализировать текст, пиктограммы и выводить графики численных значений без подключения внешних экранов. Набортные датчики (акселерометр, магнитометр, микрофон, температурный сенсор) и емкостные контакты обеспечивают быстрый сбор телеметрии и управление жестами.

- беспроводная связь и Интернет вещей (IoT): встроенный радиомодуль и библиотека Radio позволяют без настройки сложных сетевых стеков организовать peer-to-peer радиообмен между платами. Это дает возможность изучать клиент-серверную архитектуру, создавать распределенные сети датчиков и беспроводные пульты управления для роботов. При этом симулятор MakeCode автоматически открывает второе виртуальное устройство для наглядной отладки сетевых пакетов.

- взаимодействие с внешними электронными компонентами: через краевой разъем (edge connector) и платы расширения обучающиеся подключают сервоприводы, реле, адресные светодиодные ленты и сторонние датчики, осваивая на практике работу с цифровыми и аналоговыми сигналами (АЦП), широтно-импульсной модуляцией (ШИМ) и стандартными шинами обмена данными (I2C, SPI).

Возрастная дифференциация при написании кода

Гибкость MakeCode и BrickCode позволяет дифференцировать процесс обучения в зависимости от возраста учащихся:

Младшие школьники (старт с блоков и постепенный переход). На начальном этапе дети концентрируются исключительно на алгоритмической логике, собирая программы из блоков без риска совершить синтаксическую опечатку. По мере взросления и усложнения задач они постепенно переходят на Static TypeScript или Static Python. Переход происходит без стресса, так как архитектура команд, симулятор и принципы взаимодействия с устройством остаются прежними.

Старшие школьники (прямой старт на текстовом коде). Подросткам среднего и старшего школьного возраста нет необходимости использовать блочный интерфейс – в силу возраста визуальные блоки воспринимаются ими как

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)

избыточный элемент, замедляющий разработку. Они начинают работу сразу в текстовом редакторе. Это дает им возможность писать компактный структурированный код, применять многомерные массивы, использовать возможности ООП и быстро управлять проектом без траты времени на развертывание сложных локальных IDE.

Преимущества открытого исходного кода (Open Source)

Распространение базового фреймворка PXT под свободной лицензией MIT гарантирует образовательным организациям технологический суверенитет [4]:

1. Автономность: возможность локального хостинга сред разработки внутри локальной сети и создание полностью независимых форков (проект BrickCode [5]).

2. Локализация: адаптация русскоязычной терминологии через открытые платформы перевода (Crowdin) в строгом соответствии с отечественными стандартами.

3. Расширяемость: возможность написания собственных библиотек (расширений) на TypeScript для подключения нестандартных датчиков и внешних модулей [3].

Заключение

Платформа Microsoft MakeCode и ее независимые ветки, такие как BrickCode, представляют собой целостную образовательную экосистему. Она обеспечивает непрерывный образовательный маршрут от наглядных визуальных блоков в младшем возрасте до профессиональной текстовой разработки на Static TypeScript и Static Python в старших классах. Встроенный симулятор, строгая типизация и открытость архитектуры делают данное ПО надежной базой для подготовки будущих инженеров и программистов встраиваемых систем.

Список литературы

1. Ball, T. Microsoft MakeCode: embedded programming for education, in blocks and TypeScript / T. Ball, J. Protzenko, J. Bishop, M. Moskal, J. de Halleux, M. Braun, S. Hodges, C. Riley / T. Ball // Proceedings of the 2019 ACM SIGPLAN Symposium on SPLASH-E. – 2019. – P. 7–12.

Content is licensed under the Creative Commons Attribution 4.0 license (CC-BY 4.0)

2. Devine, J. MakeCode and CODAL: intuitive and efficient embedded systems programming for education / J. Devine, J. Finney, P. de Halleux, M. Moskal, T. Ball, S. Hodges / J. Devine // Journal of Systems Architecture. – 2019. – Vol. 98. – P. 468–483.

3. Microsoft MakeCode Documentation. – URL: https://makecode.com (дата обращения: 17.11.2026). – Текст : электронный.

4. Microsoft Programming Experience Toolkit (PXT) // GitHub. – URL: https://github.com/microsoft/pxt (дата обращения: 17.11.2026). – Текст : электронный.

5. BrickCode (pxt-ev3-community): открытая среда программирования для LEGO MINDSTORMS EV3 на базе фреймворка PXT // GitHub. – URL: https://github.com/pxt-ev3-community/pxt-ev3 (дата обращения: 17.11.2026). – Текст : электронный.

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)