

УДК 378

**С. Д. Коткин**

**S. D. Kotkin**

Коткин Сергей Дмитриевич, к. п. н., доцент кафедры ИОТД, Кузбасский гуманитарно-педагогический институт Кемеровского государственного университета, г. Новокузнецк, Россия.

Kotkin Sergey Dmitrievich, Ph.D., Associate Professor, Kuzbass Humanitarian Pedagogical Institute Kemerovo State University, Novokuznetsk, Russia.

## **ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ TINKERCAD В ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ МИКРОКОНТРОЛЛЕРОВ ARDUINO**

### **PECULIARITIES OF USING TINKERCAD IN TEACHING PROGRAMMING OF ARDUINO MICROCONTROLLERS**

**Аннотация.** В статье рассматривается организация программирования микроконтроллеров на платформе Arduino. Даются предложения по использованию в учебном процессе таких инструментальных средств как язык программирования Scratch и эмулятор электронных схем Tinkercad.

**Annotation.** The article discusses the organization of programming microcontrollers on the Arduino platform. Suggestions are made on the use in the educational process of such tools as the Scratch programming language and the electronic circuit emulator Tinkercad.

**Ключевые слова:** микроконтроллеры, Arduino, Scratch, Tinkercad, моделирование электронных схем.

**Keywords:** microcontrollers, Arduino, Scratch, Tinkercad, circuit simulators.

Одним из вех на пути развития вычислительной техники стало появление в 70-х годах XX века микропроцессоров. До их создания процессоры в вычислительных системах строились из универсальных микросхем, реализующих вентили, регистры, сумматоры, шифраторы и дешифраторы, а также прочие базовые компоненты цифровых систем. Микропроцессоры же представляют собой специализированные микросхемы для реализации вычислений и содержат арифметико-логическое устройство, регистры, контроллеры шины данных и системной шины. Первые микропроцессоры применялись в малопроизводительных вычислительных устройствах, таких как микрокалькуляторы. Однако вскоре микропроцессоры стали применяться в персональных компьютерах и даже в суперЭВМ. Современные микропроцессоры далеко ушли по своей структуре от первых процессоров, однако их основное назначение осталось прежним – выполнение вычислений и управление вычислительными системами. При этом с различными устройствами процессоры взаимодействуют опосредованно через их контроллеры, задачей которых является согласование физических и логических интерфейсов шин процессора с интерфейсами этих устройств. Микроконтроллеры изначально проектировались для применения в системах управления, поэтому по сравнению с микропроцессорами, они дополнительно содержат компоненты, позволяющие взаимодействовать с различными электронными устройствами напрямую, такие как цифровые порты, аналого-цифровые и цифроаналоговые преобразователи и интерфейсы ввода-вывода. Зачастую в микроконтроллере размещается и оперативная и электрически перепрограммируемая память для хранения программ [1, 2].

Несмотря на то, что микроконтроллеры содержат значительное число встроенных компонентов, для их полноценной работы требуются и внешние компоненты, такие как источник питания, включающий схему стабилизации напряжения, кварцевый резонатор для генерации тактовых импульсов, дополнительные контроллеры шин и прочие компоненты. Одной из платформ, содержащей помимо микроконтроллера и другие компоненты, позволяющие значительно упростить создание электронных систем на базе микроконтроллера является Arduino. Эта платформа представляет собой три семейства плат с микроконтроллером: Arduino Nano, Arduino MKR и Arduino Classic Family, различающихся своим форм-фактором, при этом для последних двух семейств также выпускаются специальные платы расширения (shields), содержащие дополнительные устройства, управляющиеся микроконтроллером. Сами платы выполнены с использованием микроконтроллеров ATmega328P, ATmega32u4, ATmega2560, Cortex-M0 32-bit SAMD21 и др. [3-8]. Кроме того, на платах содержатся схемы подачи питания, компоненты для подключения микроконтроллеров к компьютеру по USB, а также колодки с гнездами для подключения к цифровым и аналоговым выводам микроконтроллера различных устройств, подачи на них напряжения питания, опорного напряжения для цифроаналогового преобразования, и подключения устройств по поддерживаемым микроконтроллером шинам. Платы расширения выполнены таким образом, что их можно подключать к плате с микроконтроллером с помощью разъемов, размещенных непосредственно на этих платах. При этом существует возможность подключать платы расширения одна к другой, добавляя, таким образом, нужные аппаратные возможности к базовой функциональности микроконтроллера. Существуют платы расширения для управления обычными и шаговыми двигателями, для подключения к сети Ethernet, для управления мощными устройствами с помощью реле и др.

Разработчики платформы Arduino выпускают оборудование под открытой лицензией, то есть другие производители имеют право выпускать платы с микроконтроллерами и платы расширения по тем же самым схемам, что и производители оригинальных плат. Но сторонние производители не имеют права использовать при этом название Arduino. Таким образом, помимо производства Arduino в Италии существуют производители совместимых плат, в том числе в России или в Китае, что является важным в условиях происходящего сейчас санкционного давления со стороны некоторых стран. Кроме того, благодаря свободной лицензии производители разрабатывают и выпускают разнообразные новые платы расширения для самых различных проектов.

Помимо аппаратной части, в состав платформы Arduino входит и программная часть, включающая в себя интегрированную среду разработки для всех плат Arduino.

Самой популярной платой с микропроцессором из платформы Arduino сейчас является Arduino Uno R3. Эта плата построена на базе микроконтроллера ATmega328P. Этот микроконтроллер имеет ядро, построенное на основе гарвардской архитектуры (хранилища данных и команд в отличие от фон-неймановской архитектуры в ней разделены) с RISC набором команд, то есть набор инструкций процессора небольшой, что уменьшает его сложность, позволяет снизить энергопотребление и выполнять все инструкции за небольшое количество тактов процессора.

Данный микроконтроллер содержит 2 КБ ОЗУ, 1 КБ EEPROM (перепрограммируемая память), 32 КБ Flash-памяти (разновидность EEPROM, где стирание данных происходит крупными блоками), два восьмибитных и один шестнадцатитбитный таймер, шесть выходов с поддержкой широтно-импульсной модуляции (ШИМ), 8 аналоговых входов с десятиразрядным АЦП, 23 цифровых входа/выхода, интерфейс USART (синхронный последовательный интерфейс), интерфейс I2C (последовательный интерфейс, предназначенный для подключения нескольких устройств к одной шине) и интерфейс SPI (синхронный последовательный интерфейс, использующий для передачи данных четырехпроводную шину) [9].

Помимо выводов микроконтроллера, Arduino Uno содержит также разъем для подключения к компьютеру по USB, с помощью которого микроконтроллер можно программировать, обмениваться с ним данными, а также подавать на него напряжение питания, отдельный разъем для подключения источника питания, кнопку Reset, предназначенную для сброса микроконтроллера, светодиод, сигнализирующий о подаче напряжения питания, а также светодиод, подключенный к тринадцатому выходу микроконтроллера, с помощью которого можно, например, организовать простейшую индикацию режимов работы создаваемого устройства.

Несмотря на невысокую производительность из-за небольшой тактовой частоты 16 МГц и небольшой объем встроенной памяти, с использованием Arduino Uno R3 можно выполнять самые различные проекты: от простейших устройств, переключающих светодиоды в зависимости от освещенности или уровня шума до экранов на основе адресуемых RGB-светодиодов, контроллеров умного дома, управляемых по Wi-Fi или Ethernet, MIDI-контроллеров и прочих устройств, имеющих практическое применение [10].

Для программирования Arduino разработчики предложили использовать язык, разработанный на основе языков C и C++. В этом языке была упрощена система примитивных типов (разработчики оставили только несколько из предлагаемых языками C/C++), изменился состав стандартных библиотек, в частности, удалена работа с динамической памятью, а из языка C++ взяты только основные возможности объектно-ориентированного программирования, не включающие перезагрузку операций, обобщенное программирование, пространства имен и другие его средства. В то же время в состав ядра языка вошли специфичные функции и константы для управления микроконтроллером, например, для подачи на вывод микроконтроллеров сигналов бинарных уровней, функции для формирования ШИМ-сигналов, функции чтения часов реального времени, задержки выполнения программы, математические (модуль, квадратный корень, возведение в степень и т.п.) и тригонометрические (синус, косинус, тангенс) функции, функции для манипуляции отдельными битами чисел, функции для работы с прерываниями и функции для проверки свойств символов в расширенной восьмибитовой кодировке ASCII [11]. Однако, несмотря на то, что используемый в Arduino язык сильно проще, чем C++ и, даже, C, обучение программированию на нем требует обучению синтаксису языка, в том числе названий управляющих конструкций, являющихся словами английского языка, освоению особенностей записи условий, параметров циклов, пониманию правил выбора типов данных и тому подобных деталей, что требует приложения значительных усилий уже в начальный период обучения.

Одним из способов облегчить обучение программированию является использование языка Scratch. Он представляет собой язык программирования, в котором программа составляется из блоков, соединяющихся друг с другом по определенным правилам, что позволяет уже на этапе создания программы предотвратить множество синтаксических ошибок. В классической версии Scratch программа выполняет анимацию персонажей, а блоки, из которой она состоит, делятся на десять групп в зависимости от их предназначения: изменения потока управления, движения персонажа, изменения внешнего вида, управления звуками, обработки событий, арифметических и логических операторов, блоков для манипуляции переменными, прочих блоков и блоков из расширений. При этом многие блоки могут быть параметризированы как литеральными значениями, так и другими блоками.

Применение языка Scratch сильно упрощает знакомство с программированием, особенно для детей, так как при создании программ они могут видеть имеющиеся возможности языка, оперируют элементами программы, название которых записано на русском языке, и при попытке создать синтаксически неверную конструкцию немедленно получают обратную связь [12, 13]. Кроме того, пошаговое выполнение программы позволяет обучающемуся интериоризировать модель исполнителя.

Существуют и специальные адаптации Scratch для Arduino, например, Scratch for Arduino (S4A) [14] или mBlock [15]. Однако эти среды разработки обладают одним существенным недостатком: они не позволяют исполнять программы в пошаговом режиме, тем самым не давая ученику получить представление о влиянии той или иной команды на работу программы в конкретном контексте.

Средой программирования в Scratch, преодолевающей этот недостаток является онлайн-платформа Tinkercad [16]. Изначально эта платформа разрабатывалась как онлайн-трехмерный редактор, однако в дальнейшем, после её покупки компанией Autodesk, в неё помимо прочего был добавлен эмулятор электронных схем, способный эмулировать работу микроконтроллеров Arduino и позволяющий создавать программный код для него как на языке Scratch, так и на языке C++.

Tinkercad помимо микроконтроллеров позволяет добавлять в схемы множество других компонентов: макетные платы, резисторы, конденсаторы, катушки индуктивности, транзисторы, различные логические элементы, элементы питания, переключатели, датчики, индикаторы, исполнительные механизмы и прочее. При этом работу схемы можно проверить непосредственно в эмуляторе, не собирая реальную схему.

Реализация Scratch в Tinkercad несколько отличается от эталонной: поскольку в Tinkercad с помощью Scratch программируются не анимированные персонажи, а микроконтроллеры, то и состав категории, на которые разбиты блоки, несколько иной. Во-первых, это блоки выхода, с помощью которых можно управлять и отдельными выводами Arduino и различными типами устройств, подключенными к нему, например, пьезодинамиками, ЖК-индикаторами, сервоприводами и т.п. Во-вторых, это блоки входа, позволяющие читать цифровые и аналоговые данные непосредственно с выводов Arduino, а также получать данные из последовательного интерфейса, с ультразвукового датчика расстояния, с температурного датчика, а также получать данные об угле поворота сервопривода. Четыре оставшиеся категории: замечания, управление, мат. данные и переменные – схожи с категориями классического Scratch.

Ещё одной особенностью реализации Scratch в Tinkercad является то, что он позволяет на основе программы на языке Scratch получить программу на языке C++, благодаря чему учащиеся могут постепенно привыкать к виду программ на этом языке и изучать отдельные особенности его синтаксиса и его операторы. К сожалению, перевода программы с языка C++ на язык Scratch Tinkercad делать не умеет.

Так как Tinkercad является полноценным эмулятором электронных схем, то работу программы можно проверить непосредственно в этой среде. При этом в Tinkercad имеется возможность пошагового выполнения программ, так что ученики могут посмотреть действия, выполняемые отдельными операторами и оценить их влияние на работу схемы в целом.

Обучение программированию микроконтроллеров на основе среды Tinkercad строится следующим образом. Вначале занятия формулируется описание проекта. Если проект сложный, например, обработка нажатия кнопки требует понимания не только программной части, но и схемотехнической реализации, то приводится пример упрощенного проекта и разбираются принципы работы программной и аппаратной части. Далее обсуждаются особенности реализации предлагаемого проекта, его отличия от прототипа. После этого дается некоторая заготовка проекта, содержащая часть элементов, комментарии, по поводу подключения остальных элементов, а также часть кода и комментарии, что ещё в код надо добавить. То, насколько заготовка проекта будет неполной, зависит от сложности проекта, от возраста и подготовки учащихся. На этом этапе задачей учащихся является доделать в эмуляторе аппаратную и программную реализацию проекта и проверить их работоспособность. При этом рекомендуется проверять работу схемы в пошаговом режиме исполнения программы микроконтроллера, даже если результат сразу получился рабочим. Если что-то не работает, то учитель может задать наводящие вопросы, чтобы учащийся обратил внимание на конкретные части схемы или программы. После того, как схема сделана в эмуляторе, учащиеся собирают эту схему из реальных компонентов и добиваются её работоспособности. В конце занятия обсуждаются как отдельные используемые приемы, так и некоторые операторы на языке C++ и их соответствие блокам Scratch – это нужно для того, чтобы учащиеся параллельно осваивали его, чтоб в дальнейшем начать программировать без Tinkercad.

Хотя Tinkercad и хорош для начала обучения программированию микроконтроллеров, есть у него и недостатки. Один из них заключается в отсутствии эмуляции плат расширения и значительного количества оборудования, например, ЖК-экранов, управляющихся по параллельной шине данных. Второй недостаток тесно связан с первым: у Arduino имеется возможность использования не только стандартных библиотек, но и библиотек, создаваемых производителями оборудования – набор же блоков Tinkercad фиксированный и никак не расширяется. Так что на определенном этапе учащиеся будут вынуждены переходить от программирования микроконтроллеров в Tinkercad к программированию их непосредственно в Arduino IDE. Но к тому времени многие аспекты создание проектов с использованием микроконтроллеров Arduino и особенности языка C++ должны стать для них понятными.

Третьим недостатком Tinkercad стал тот, что данный инструмент в любой момент времени может стать недоступным в России благодаря очередным санкциям. Единственным способом его преодоления является создание российского инструмента со схожей функциональностью. К сожалению, к настоящему времени автору ничего не известно о работе над подобным инструментом в РФ.

Однако, до тех пор, пока Tinkercad доступен в России, он является прекрасным средством для приобщения учеников начальной и средней школы к программированию микроконтроллеров и к схемотехнике.

### Список литературы

1. Магда, Ю. С. Современные микроконтроллеры. Архитектура, программирование, разработка устройств [Текст]. / Ю. С. Магда. – М. : ДМК Пресс, 2012. – 224 с. : ил.
2. Болл, С. Р. Аналоговые интерфейсы микроконтроллеров [Текст]. / С. Р. Болл, пер. с англ. – М. : Издательский дом «Додэка XXI», 2010. – 360 с.: ил. (Серия «Программируемые системы»).
3. Arduino Hardware [Электронный ресурс] // Arduino Store. – URL : <https://www.arduino.cc/en/hardware> (дата обращения : 17.11.2022).
4. Arduino Uno Rev3 [Электронный ресурс] // Arduino Store. – URL : <https://store.arduino.cc/products/arduino-uno-rev3> (дата обращения : 20.11.2022).
5. Arduino Leonardo with Headers [Электронный ресурс] // Arduino Store. – URL : <https://store.arduino.cc/products/arduino-leonardo-with-headers> (дата обращения : 20.11.2022).
6. Arduino Mega 2560 Rev3 [Электронный ресурс] // Arduino Store. – URL : <https://store.arduino.cc/products/arduino-mega-2560-rev3> (дата обращения : 20.11.2022).
7. Arduino MKR WiFi 1010 [Электронный ресурс] // Arduino Store. – URL : <https://store.arduino.cc/products/arduino-mkr-wifi-1010> (дата обращения : 20.11.2022).
8. Arduino Due [Электронный ресурс] // Arduino Store. – URL : <https://store.arduino.cc/products/arduino-due> (дата обращения : 20.11.2022).
9. ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash datasheet [Электронный ресурс] // Microchip Technology Incorporated. – URL : [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) (дата обращения : 27.11.2022).
10. Петин, В. А. 77 проектов для Arduino [Текст]. / В. А. Петин. – М. : ДМК Пресс, 2019. – 256 с. : ил.
11. Language Reference [Электронный ресурс]. – URL : <https://www.arduino.cc/reference/en/> (дата обращения : 12.11.2022).
12. Елисеева, О. Е. Обучение детей основам создания компьютерных игр на языке программирования Scratch : пособие для учителей учреждений общ. сред. образования с белорус. и рус. яз. обучения : 5-6 классы [Текст]. / О. Е. Елисеева. – Минск : Народная асвета, 2017. – 166 с. : ил. – (Асветик-айтишник).
13. Голиков, Д. В. 42 проекта на Scratch 3 для юных программистов [Текст]. / Д. В. Голиков. – СПб. : БХВ-Петербург, 2019. – 184 с.: ил.
14. About S4A datasheet [Электронный ресурс] // S4A. – URL : <http://s4a.cat/> (дата обращения : 22.11.2022).
15. Make with Code. One-stop coding platform tailored to coding education, trusted by 20 million educators, and learners [Электронный ресурс] // makeblock | mBlock. – URL : <https://mblock.makeblock.com/en-us/> (дата обращения : 19.11.2022).
16. Autodesk Tinkercad [Электронный ресурс]. – URL : <https://www.tinkercad.com/> (дата обращения : 20.11.2022).