

Программная платформа Node.js.

Долгов А. Н.¹, Нуруллин Р. Ю.²

¹Долгов Антон Николаевич / Dolgov Anton Nikolaevich – студент;

²Нуруллин Роман Юрьевич / Nurullin Roman Yurievich – студент,

кафедра систем автоматического управления и контроля, факультет интеллектуальных технических систем,

Национальный исследовательский университет

Московский институт электронной техники, г. Зеленоград

Аннотация: в данной статье рассматриваются основные теоретические сведения об инструменте программирования на языке JavaScript, платформе Node.js. Здесь приведены определения, принципы и механизмы работы, исторические сведения, а также описаны сравнения Node.js с другими платформами, некоторые преимущества инструмента и недостатки. На примере микроконтроллера Iskra JS продемонстрирована работа простейшей программы на Node.js. Далее приведено небольшое описание среды разработки на Node.js. Кроме того, описывается важность внедрения данной технологии в современную IT-индустрию.

Ключевые слова: Node.js, JavaScript, микроконтроллер.

Node.js – платформа, основанная на языке программирования JavaScript движке V8, и предлагающая асинхронное API для работы с сетью и диском. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода. Node.js применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и Mac OS) [1]. Таким образом, Node.js превращает JavaScript из узкоспециализированного языка в язык программирования общего назначения. Одной из интересных особенностей, представляемых нам данной платформой, является возможность программировать некоторые модели микроконтроллеров.

Node.js был создан Райаном Далом (Ryan Dahl) в 2009 году. Что же такого сделал Райан? Почему именно Node.js получил такое большое признание, ведь на самом деле до него существовали другие попытки сделать тоже самое? Ответ на данный вопрос весьма очевиден. Node.js был создан на основе виртуальной машины V8, изначально разработанной компанией Google для браузера Chrome. Node.js разработан с целью интерпретации JavaScript на любой платформе. Простыми словами, данный инструмент умеет выполнять написанный программистом код. И не просто умеет – Node.js делает это хорошо. Он работает очень быстро, поддерживает практически все возможности последних версий JavaScript, а также довольно экономно расходует память. Есть несколько причин, почему многие веб-разработчики отдают предпочтение Node.js. Веб-разработчики, могут применять на сервере язык JavaScript, который изначально был задуман для реализации на клиентской стороне разработки. Таким образом, получается, что программист может применять один и тот же код и на клиентской стороне, и на серверной. Для программирования оконных приложений и микроконтроллеров наиболее предпочтительными будут, конечно, другие инструменты. Но с правильно поставленным кодом Node.js также может прекрасно выполнять свою работу и здесь.

В качестве примера вполне подойдет плата Iskra JS, разработанная российской компанией «Амперка», со встроенным интерпретатором JavaScript. Она является развитием платформы Espruino, но совместима с платами шилдами для Arduino. Если для вашего проекта важны скорость и комфорт разработки, максимальная совместимость с платами расширения, сенсорами и другими электронными модулями, Iskra JS — это оптимальный выбор [2]. Ниже представлена простейшая программа, которая попеременно включает и выключает светодиод.

```
var on = false;
setInterval(function() {
    on = !on;
    LED1.write(on);
}, 500);
```

Средой программирования является приложение для браузера Chrome - Espruino Web IDE. Для пользователя она заменяет Arduino IDE. Стандартная и встроенные библиотеки реализованы на чистом Си, пользователю доступен JavaScript-интерфейс в виде набора объектов, функций и модулей. Вся прелесть данного подхода становится очевидна в сложных с архитектурной точки зрения проектах, в которых сложность работы на низкоуровневом языке будет неоправданно велика. И тут JavaScript нам в помощь. Благодаря замыканию и нестрогой типизации данных, работа с микроконтроллером для многих будет по душе.

В современном мире наблюдается бурное развитие технологий. Если раньше 10 мегабайт памяти было чем-то колоссальным, сегодня подобные размеры могут разве что улыбнуть. Еще в начале XXI века невозможно было представить, что компьютерную игру можно написать на чем-то, кроме C/C++. Затем

техника могла себе позволить нечто «более тяжелое», и, к началу 2010-х создание полноценного продукта стала возможна без единой строки на Си. Как следствие, за счёт в разы более быстрой производительности и большего количества доступных разработчиков, стало возможным больше ресурсов тратить на интересность игры, а не на поиск утечек памяти. Сейчас уже актуальность базовых и низкоуровневых языков постепенно уменьшается. Будущее за высокоуровневыми языками, позволяющими осуществлять интересные задумки быстро и с минимальным количеством ошибок. «Железо» сейчас недорогое и постоянно развивается, время людей – дорогое и ниоткуда не берется. Такие технологии, как Node JS позволяют реализовать свои идеи на более привычном для многих языке программирования JavaScript. Программирование микроконтроллеров здесь напоминает обычную веб-разработку. Что очень даже радует.

Литература

1. Романюк А. А. Синтезаторы частот на основе автогенераторов с ФАПЧ: уч. пособие. М.: МИЭТ, 2005. 100 с.
2. Цифровая техника в радиосвязи. [Электронный ресурс]. Режим доступа: <http://digteh.ru/WLL/synt.php> (дата обращения: 13.12.2016).