

УДК 004.41

## ПРОБЛЕМЫ РАЗРАБОТКИ НИЗКОУРОВНЕВОГО МОДУЛЯ ИНТЕРПРЕТАТОРА ESPRUIINO ДЛЯ ПРОГРАММИРОВАНИЯ МИКРОКОНТРОЛЛЕРОВ С ПОМОЩЬЮ JAVASCRIPT

Солнцев В. М. (Университет ИТМО)

Научный руководитель – к.п.н., доцент Государев И.Б.  
(Университет ИТМО)

Исследование нацелено на описание проблем и принципов разработки модулей интерпретатора JavaScript для программирования микроконтроллеров на базе проекта с открытым исходным кодом Espruino. Предлагается подход к имплементации функций стандарта ECMAScript, не реализованных на платформе Espruino. Также описывается доработка математической библиотеки для микроконтроллеров без динамического выделения памяти. Дополнительно раскрывается возможность разработки библиотек для взаимодействия с различными внешними устройствами, такими как радиопередатчик и радиоприемник частоты 433 МГц и др.

### Введение.

Программирование микроконтроллеров является достаточно сложной задачей, чтобы создавать комплексные устройства. Как показывает анализ документации и информационных ресурсов, для этого требуется минимальный набор знаний, который включает в себя:

- Языки C/C++;
- Язык Ассемблера;
- электротехника;
- схемотехника;
- понимание технического описания к микросхемам;
- протоколы передачи данных;
- принцип работы цифро-аналогового и аналого-цифрового преобразователей.

Тем самым создаётся высокий порог вхождения в эту область. В основном это связано с освоением синтаксиса C и C++.

Для снижения порога вхождения в разработку программного обеспечения микроконтроллеров, разрабатывается интерпретатор языка JavaScript. На данный момент существует популярный проект с открытым исходным кодом, в котором силами сообщества создается интерпретатор JavaScript для программирования различных микроконтроллеров.

Первой проблемой является отсутствие базового функционала, который необходим для более эффективного построения приложений:

- `Object.prototype.isEnumerable` – функция, возвращающая логическое значение, указывающее, является ли указанное свойство перечисляемым,
- `Multiline String Literals` – поддержка многострочных литералов,
- `lastIndexOf` – функция возврата индекса последнего вхождения указанного значения в строковый объект,
- `Date.prototype.toJSON` – функция представления объекта в виде JSON,
- `Symbol.toStringTag` – функция создания строкового значения свойства,
- `Number.isSafeInteger()` – функция, определяющая, является ли переданное значение безопасным целым числом.
- `Math.trunc` – функция выделения целой части числа, путем удаления дробной,
- `Math.sign` – функция возврата знака числа, указывающего на то, каким число является (отрицательным, положительным или нулем)

Второй проблемой является отсутствие большого количества функций в математической библиотеке интерпретатора, а также отсутствие динамического выделения памяти, которое не позволяет использовать математические функции языка С, поэтому количество функций и точность результатов ниже, чем в библиотеке Math JavaScript.

Третьей проблемой является модуль приёмника и передатчика радиосигнала rf 433. Приемник настроен таким образом, что не может получать данные, не сформированные в пакеты специального формата. Таким образом, на текущий момент, нет способа получения сигнала от иных устройств (пульта управления охранными комплексами, системы управления и др.).

### **Основная часть.**

Решение первой проблемы заключается в доработке ядра интерпретатора за счёт реализации модуля на языке С, содержащего вспомогательные функции и интерфейсы, необходимые при решении следующих проблем. При этом необходимо знание архитектуры проекта и умение оптимизировать программный код под ограниченное количество памяти микроконтроллера.

Вторая проблема реализации математического модуля может быть решена рефакторингом и оптимизацией текущих алгоритмов. Отсутствие возможности динамического выделения памяти снижает точность вычислений, но при определении нужного количества элементов ряда в численных методах, позволяет снизить погрешность. Аналогичным образом можно добавить отсутствующие функции.

Решение третьей проблемы с приемником и передатчиком радиосигнала 433 МГц может быть осуществлено путем миграции высокоуровневого интерфейса на JavaScript. В текущей реализации интерпретатора существует библиотека 433.js, которая занимается формированием пакетов данных в определенном формате, аналогично с приемником. Необходимо разработать универсальный способ взаимодействия этих устройств, чтобы подключать сторонние.

### **Выводы.**

Успешная апробация предлагаемых подходов обусловила предпочтительное внедрение в практику использования интерпретатора Espruino описанных в статье решений.

Солнцев В.М. (автор)

Подпись

Государев И.Б. (научный руководитель)

Подпись