

Исследование производительности разных языков программирования на микроконтроллере ESP32

Андрей Дмитриевич Константинов, Лицей 64, Санкт-Петербург

Аннотация:

В данной работе рассматриваются вопросы производительности кода, написанного на C++, Python, JS. Тесты производятся на микроконтроллере ESP32. Оценка производительности выставляется по размеру занятой памяти и времени исполнения задачи. В итоге выбирается самый оптимальный язык

Введение:

В результате наблюдения за постоянными спорами с сети об производительности языков было решено провести сравнительное тестирование трех самых популярных языков разработки по под микроконтроллеры. В качестве объекта для исполнения кода будет использоваться ESP32-S (технические характеристики указаны в Приложении 1). Данный контроллер имеет 2 ядра, но в тестах мы будем использовать одно.

Методология исследования:

Описание алгоритма:

1) Для проверки производительности будет использоваться классический алгоритм со сложностью $O(n^2)$ – сортировка пузырьком. Данный алгоритм производит множество операций чтения/записи в оперативную память, а также постоянно производит операции сравнения.

2) Второй тест – рекурсивное вычисление факториала числа 41. Данный алгоритм будет очень затратным по памяти, на нем будет хорошо видно на сколько эффективно разные языки работают с памятью

Проведение тестов:

Для получения повторяемых результатов, были предприняты определённые меры:

Питание ESP32 происходит от мощного стабилизированного источника питания с величиной напряжения 5 вольт и током в 1.5 ампера.

Для исключения влияния внешних факторов, таких как температура и влажность, тестовая плата упаковывалась в герметичный контейнер. Далее контейнер помещался в среду с постоянной температурой в -10 градусов Цельсия. Таким образом перегрев и тротлинг не мог образоваться.

Тестирование происходило по 3 прогона, в каждом прогоне по 10 запусков теста. Между циклами устройство отключалось от питания на 10 минут.

Результаты тестирования были проанализированы, было высчитано среднее-арифметическое значение времени прогона теста.

Метод оценки результата:

Для оценки производительности используется параметр времени исполнения, так как другие значения, как, например, потребление тока и нагрев, не могут дать надежных результатов.

Настройки сред разработки и используемые фреймворки:

C++:

В качестве среды разработки была выбрана VS Code с плагином от PlatformIO, настройки стандартные, никаких дополнительных оптимизаций и изменений не вносилось. Код работает с использованием фреймворка Arduino. Компилятор и параметры компиляции стандартные.

JS:

Для языка программирования JS существует множество вариантов исполнения. Проанализировав количество доступных библиотек, и уже отлаженных функций, было решено использовать Espruino JS, как наиболее оптимальный по производительности и удобству использования вариант.

Версия Espruino JS – 2v08. Среда разработки – Espruino Web IDE

Нельзя не отметить, что при заливке чистой прошивки Espruino, микроконтроллер начинает разогреваться, доходя приблизительно до 45-50 градусов Цельсия.

Python:

Для языка программирования Python была выбрана среда MicroPython версии v1.12, поскольку это единственный развивающийся проект, все аналоги мертвы, или слабо развиты.

Результаты исследования:

В результате тестирования получились такие результаты:

Язык программирования	Bubble Sort N = 1000 (в секундах)	Factorial(49) Кол-во запусков 10000 В секундах
C++	$1 \cdot 10^{-6}$	$0,807615 \cdot 10^{-6}$
JavaScript	2868,435	871,0728 ¹
Python	7,66667	11

Выводы:

По полученным данным получается разгромное преимущество C++, как инструмента для разработки программ для микроконтроллеров. C++ быстрее JS в 197 миллионов раз. C++ быстрее MicroPython в 11 миллионов раз. Python быстрее JS в 194 раза.

Также не могу не отметить проблемы, которые наблюдались при работе с Espruino JS. Микроконтроллер сильно грелся. При попытке создать массив на 10000 элементов система уходила в перезагрузку. При глубине рекурсии больше 30 система выдавала Core Panic, поэтому пришлось считать время для вычисления факториала 30 и через математическую пропорцию прикидывать какой результат был бы при вычислении факториала 49. При работе с другими языками таких проблем не наблюдалось.

Исходя из всего вышесказанного, C++ быстрый, надежный и производительный язык с отличной поддержкой от производителя- лучший выбор для создания своих проектов. Если вы не знаете C++, то нормальным выбором будет MicroPython, он относительно быстрый, не имеет явных недостатков.

JS для использования на ESP32 не рекомендуется. У него удручающая производительность и высокое энергопотребление.

Ядра, Частота	Extensya Dual Core LX6, 160 mhz
Оперативная память	520 KB
Флеш-память	480 KB
Внешняя память	4 MB

¹ Результат примерный, получен через пропорцию, дополнительная информация в выводе