



УДИВИТЕЛЬНЫЙ МИР

Открытая научно-практическая конференция в области естествознания и инновационных технологий

УДК 004.451

JSON-RPC ДЛЯ УПРАВЛЕНИЯ МИКРОКОНТРОЛЛЕРАМИ НА БАЗЕ ОПЕРАЦИОННОЙ СИСТЕМЫ MICROPYTHON

Абдулов Александр Алексеевич, студент 3-го курса ФСиР,
Руководитель: Киселев Александр Викторович, старший преподаватель ФСиР;
ФГБОУ ВО «Иркутский государственный университет», Иркутск, РФ

*Работа посвящена разработке модуля, позволяющего посредством протокола JSON-RPC обеспечить высокоуровневую коммуникацию с микроконтроллерами на операционной системе MicroPython. В статье рассмотрены протоколы передачи данных, микроконтроллеры, программируемые на языке Python 3, а также архитектура созданной системы.
Ключевые слова: RPC; JSON-RPC; MicroPython; ESP8266.*

JSON-RPC FOR MICROCONTROLLER MANAGEMENT ON BASED OPERATING SYSTEM MICROPYTHON

Abdulov Alexander Alekseevich, 3rd year student of the faculty of service and advertising,
Adviser: Kiselev Alexander Viktorovich, senior lecturer of the faculty of service and advertising;
Irkutsk State University, Irkutsk, Russia

*The work is devoted to the development of the module that allows by the protocol JSON-RPC to provide high-level communication with microcontrollers on operating system MicroPython. In the article data transfer protocols, microcontrollers programmable in language Python 3, and created system architecture.
Keywords: RPC; JSON-RPC; MicroPython; ESP8266.*

Для цитирования: Абдулов А.А. JSON-RPC для управления микроконтроллерами на базе операционной системы MicroPython// Наука без границ. 2019. № 7(35). С. 116-118.

Во многих проектах, в том числе, основанных на использовании микроконтроллеров, появляется задача высокоуровневого управления системой.

Для решения данной задачи должен присутствовать транспортный протокол (в случае беспроводного подключения), серверное приложение, клиентское прило-

жение и протокол, по которому они будут взаимодействовать.

В качестве протокола взаимодействия было решено выбрать RPC [1], т.к. в нем присутствует четкая стандартизация и отсутствует лишняя информация, значительно увеличивающая размер передаваемых данных.

Из всех разновидностей этого протокола был выбран JSON-RPC [2]. Это популярный, гибкий и простой протокол, который еще не реализован для MicroPython [3].

Данную систему можно запускать на всех устройствах, поддерживающих Python 3, благодаря обратной совместимости с MicroPython.

Реализация состоит из двух частей: серверной и клиентской. Серверная часть ле-

жит на микроконтроллере. Она принимает на вход пользовательский класс с какой-то реализованной логикой по управлению микроконтроллером и, в зависимости от запросов клиентской части, исполняет его методы, формирует ответы. Клиентская часть находится на компьютере пользователя. Она отвечает за конвертацию вызываемых методов в JSON соответственно протоколу. Взаимодействие клиентской и серверной частей происходит с помощью подключения через сокет TCP.

На рис. 1 представлен пример скрипта для подключения и управления пользовательским классом, находящимся на микроконтроллере. Пользователь импортирует библиотеку, инициализирует клиент, передавая ip-адрес и порт, авторизуется и далее работает как с обычными классами.

```
1 import client #импорт библиотеки
2
3 cl=client.Client('192.168.4.1',80) #подключение
4 answer=cl.auth() #аутентификация
5 print(answer)
6 answer=cl.echo("add") #вызов метода реализованного на сервере
7 print(answer)
8 cl.close() #закрытие сокета
```

Рис. 1. Пример использования библиотеки

Пример вызова метода класса:

```
obj.subtract(42, 23)
```

Пример сформированного запроса:

```
{"jsonrpc": "2.0", "method": "subtract",
"params": [42, 23], "id": 1}
```

Сформированные запросы отправляются на микроконтроллер пакетами размером 1 кБ. Из-за малого объема оперативной памяти, доступной для исполнения скрипта, пакеты записываются на flash-память микроконтроллера.

Было реализовано одновременное подключение нескольких клиентов к серверу

(тестировалось до 9 подключений). Они могут параллельно отправлять запросы на сервер. Но предварительно каждый клиент должен авторизоваться. Пользователь через клиента задает пароль доступа, он шифруется согласно ключу. При успешной проверке клиент допускается к использованию сервера.

В результате работы получен мультиклиентный сервер, реализующий стандарт JSON-RPC с аутентификацией и с AES шифрованием ключей.

СПИСОК ЛИТЕРАТУРЫ

1. Remote Procedure Call. Available at: https://www.ibm.com/support/knowledgecenter/en/ssw_aix_71/com.ibm.aix.progcom/ch8_rpc.htm
2. JSON-RPC. Available at: <https://www.jsonrpc.org/specification>
3. MicroPython. Available at: <https://micropython.org>

© Абдулов А.А., 2019