

УДК 004.457

Интернет вещей на основе raspberry pi

Линев Ф.А., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»*

Киселева С.Д., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»*

Научный руководитель: Еремин О.Ю.

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Компьютерные системы и сети»*

ereminou@bmstu.ru

Введение

Термином *Internet of things* немецкий футурист Кевин Эштон обозначил начало эры бытовых приборов, подключенных и управляемых через интернет. Тостеры, холодильники, кондиционеры будут передавать владельцу информацию о себе. В настоящее время эта тема активно развивается, все больше и больше устройств получают возможность быть подключенными к Интернету. В большинстве случаев эти устройства не связаны друг с другом, и к каждому необходимо обращаться по отдельности. При этом необходима реализация интерфейса для управления каждым из устройств. Следовательно, усложняются сами устройства и увеличивается их стоимость, необходимо применение более сложных технологий. Целью данной работы является разработка устройства, подключенного к сети Интернет, способное управлять портами ввода/вывода GPIO находящихся на микроконтроллере. Данная проблема является относительно простой, что позволяет использовать ее в целях обучения.

Постановка задачи

В рамках данной задачи необходимо разработать устройство с возможностью управления портами ввода/вывода GPIO и с возможностью управления ими через сеть Интернет. Управление портами ввода/вывода рассмотрим на примере подключения светодиода. Данная задача может быть решена множеством способов, но выбор конечного способа зависит от задач, которые должно выполнять это устройство, как-то:

безопасность, скорость обработки, количество подключаемых устройств, асинхронность взаимодействия, местоположение устройства и способ подключения к интернету.

Основные требования для устройства:

- Простота реализации;
- Переносимость;
- Возможность добавления других устройств;
- Поддержка выхода в интернет;
- Малые габариты устройства, без уменьшения функциональности;
- Простота модификации.

Выбор аппаратной реализации

Существует огромное количество микропроцессорных плат, с возможностью выхода в интернет. Рассмотрим плату raspberry pi model b+. Raspberry Pi — это миниатюрный недорогой компьютер, размещенный на одной плате размером с кредитную карту. Raspberry Pi основан на процессоре с архитектурой ARM 11, частотой в 700 МГц. В последних версиях платы официально разрешено разгонять процессор до 1000 МГц, что позволяет достичь приемлемой производительности при низком энергопотреблении.

В своем составе Raspberry pi b+ имеет не только разъем для подключения интернет кабеля, но и несколько портов USB, что позволяет подключить к ней модуль Wi-Fi и иметь практически независимый от расположения микроконтроллер, с возможностью выхода в Интернет.

Одно из основных преимуществ в raspberry pi – это возможность загрузить в памяти микроконтроллера Linux подобную операционную систему(Raspbian), что значительно упрощает проектирование и разработку системы, т.к. данная операционная система поддерживает языки, которые значительно ускоряют разработку программного обеспечения: Perl, Python, Java, C++. При этом не теряется возможность обращения к 26 выводам GPIO и отсутствует необходимость прошивать микросхему, как на других микроконтроллерах, все программы записываются во flash-память микроконтроллера.

Разработка архитектуры

Основной задачей является разработка архитектуры, которая будет обеспечивать работу устройства в любое время.

На рис.1 показана структура обычного интернет приложения.

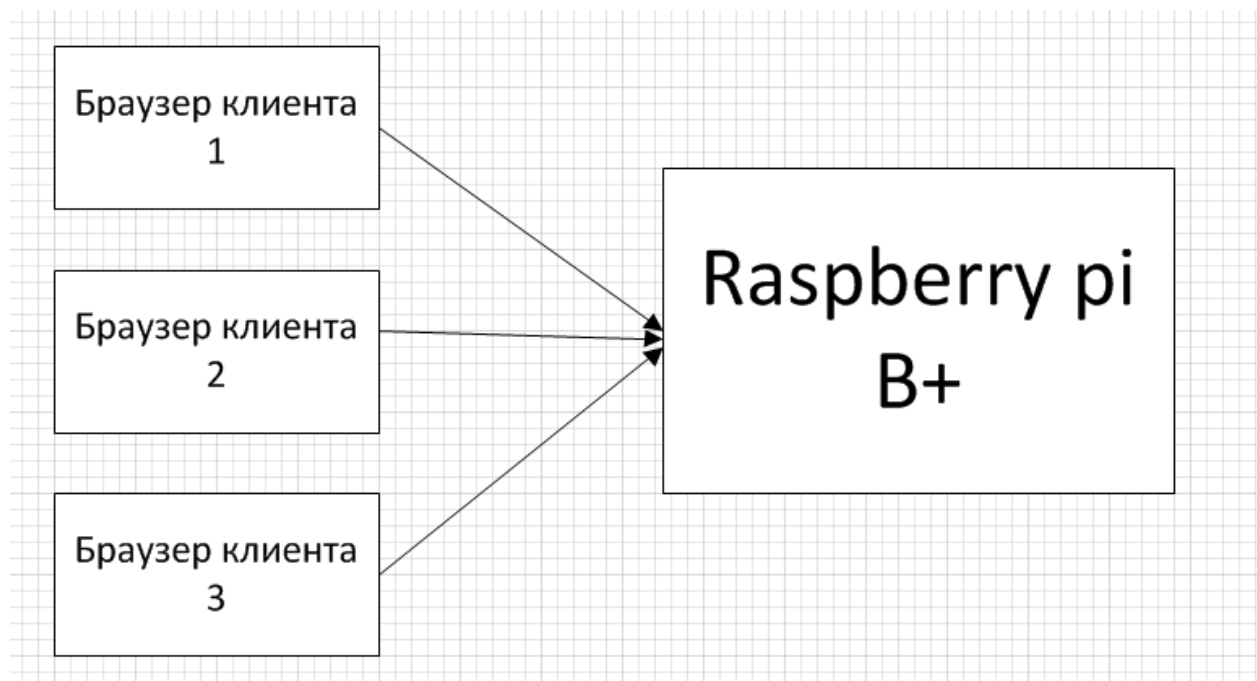


Рис. 1. Архитектура интернет приложения

Эта архитектура имеет простую реализацию, наличие только клиента (браузер) и сервера, обрабатывающего все запросы и управляющего портами ввода/вывода или светодиодом. Однако при проектировании подобной системы часто возникает следующая проблема: для управления портами ввода вывода программу необходимо запускать от пользователя с правами суперпользователя. Это влечет за собой определенные последствия, например, злоумышленник, который получил возможность подключиться к raspberry pi, может сильно навредить всей системе и микросхеме, удалить нужные файлы, или в худшем случае: получил управление другими устройствами(другими портами ввода/вывода, которые реализуют управление другими устройствами). Предлагается решение (рис. 2) - система, имеющая в своем составе два сервера: один внешний(frontend), любой пользователь общается именно с этим сервером и не имеет никакого доступа к микропроцессору.

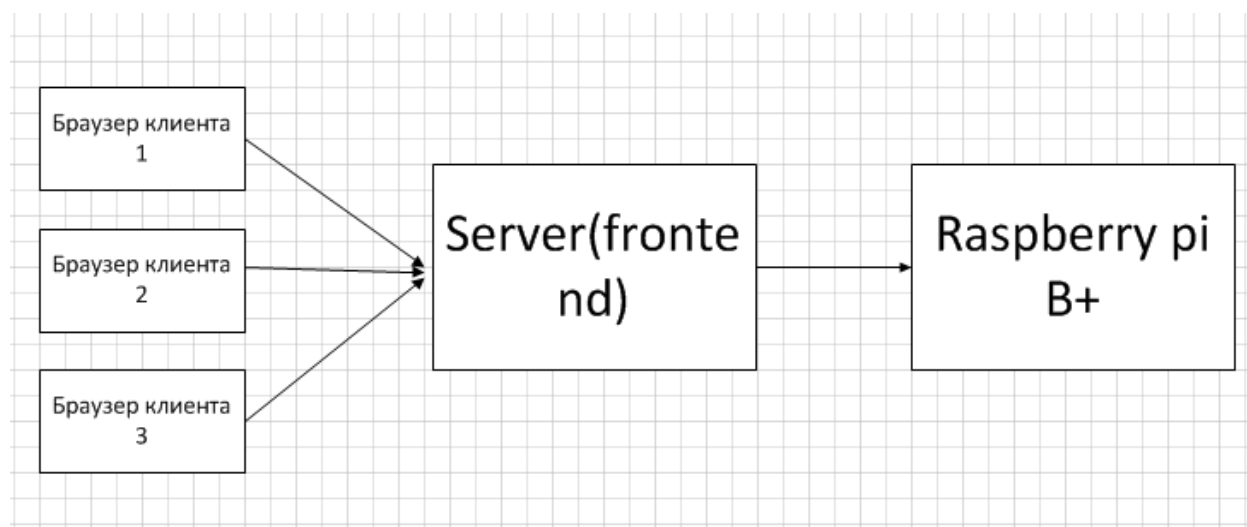


Рис. 2. Схема приложения, использующего raspberry pi.

Таким образом, общая суть схемы такова: любой пользователь не имеет прямого доступа к raspberry pi. Все пользователи подключаются к server(frontend). Пользователи не знают, ни физического, ни сетевого расположения платы raspberry pi, что существенно затрудняет ее взлом, следовательно, повышает защиту системы. Frontend сервер и raspberry pi взаимодействуют через сетевой протокол, следовательно, они могут быть разнесены на разные машины. Это увеличивает скорость raspberry pi, т.к. она не занимается ненужным вычислением и отображение страниц. Протокол общения между сервером и платой может быть намного проще, взаимодействия с браузером.

Разработка алгоритма взаимодействия между браузером, сервером и raspberry pi.

В связи с тем, что одновременно возможно подключение не одного пользователя, а 2 и более, появляется необходимость обновлять информацию о состоянии светодиода сразу на нескольких клиентах. Для этого используется постоянный опрос состояния датчика, и если приходят данные отличные от установленных, то необходимо изменить их в браузере. За счет такой простой системы достигается синхронизация между большим количеством браузеров.

Алгоритм на клиенте представлен на рис. 3.



Рис. 3. Алгоритм обновления данных у клиента

Сервер, который находится на raspberry pi, принимает данные от сервера frontend и записывает их в файл.

Рассмотрим алгоритм управления светодиодом (рис. 4). В начальный момент времени программа заходит в бесконечный цикл, далее осуществляется поиск файла, в котором содержится частота для мерцания. Затем необходимо получить системное время, в минутах и секундах, до тех пор, пока не пришло время менять значения светодиодов. Микропроцессор ждет сообщения от сервера frontend. В случае, если пришло время менять состояние светодиода, записывается время изменения значения, для последующего сравнения.

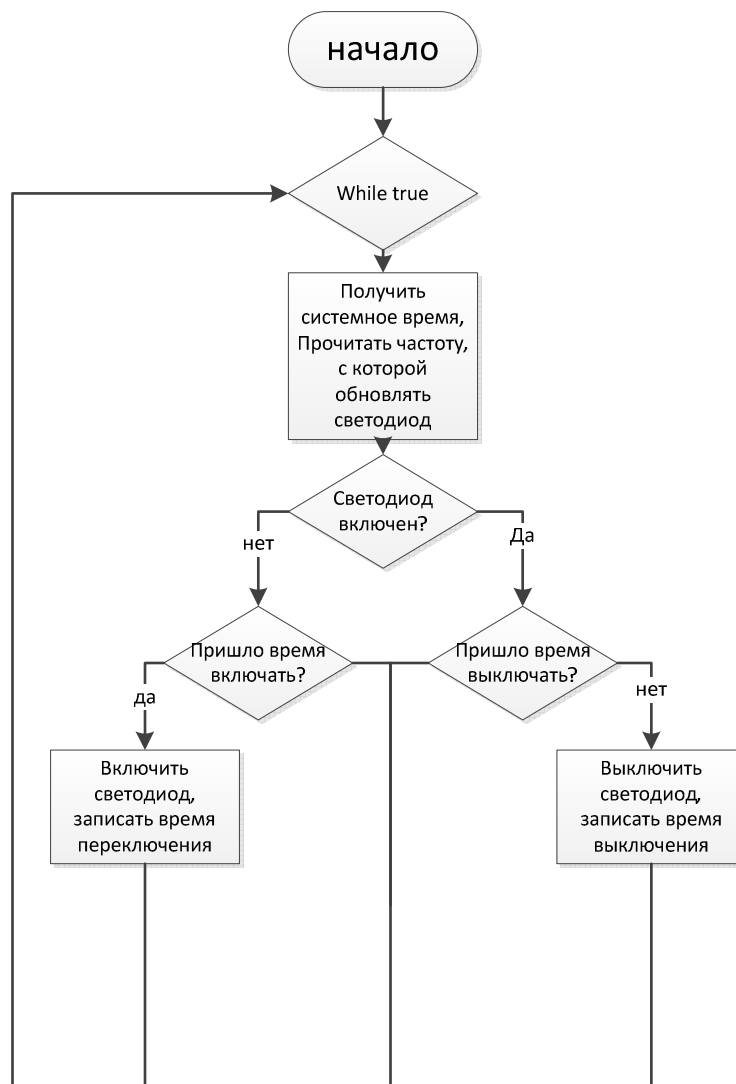


Рис. 4. Алгоритм управления светодиодом.

Разработка модели

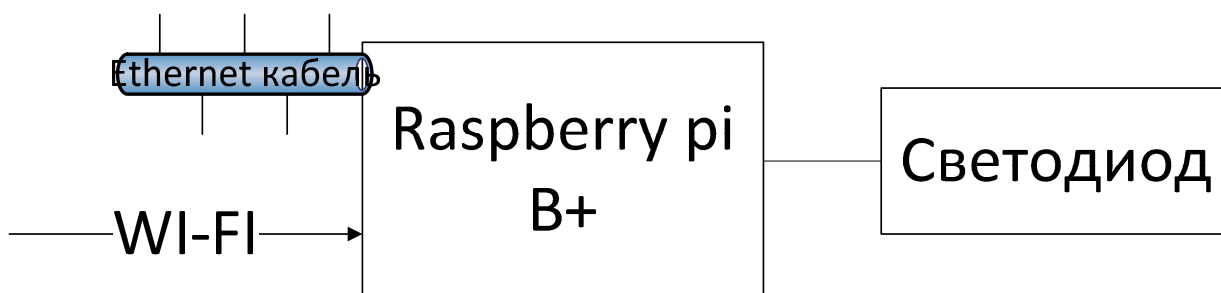


Рис. 5. Структурная схема системы на основе raspberry pi.

На рис. 5 приведена структурная схема системы на основе raspberry pi, учитывающая все указанные выше требования по контролю микросхемой.

В качестве светодиода подходит любой вариант с напряжением питания 5В.

Дальнейшие улучшения

В дальнейшем планируется спроектировать систему авторизации и аутентификации, для предотвращения несанкционированного управления светодиодом, так как вместо светодиода возможно использование и управление любым внешним устройством. Например, управление включение крана с водой, следовательно, необходимо ограничить доступ так, чтобы никто, кроме владельца, не мог воспользоваться системой.

Возможно введение многопоточной системы для того, чтобы избежать запись в файл данных, о частоте мерцания светодиода. Также можно выделить в отдельный поток алгоритм, отвечающий за переключение светодиода.

Дополнительно возможно использование другой алгоритм оповещения об изменении состояния лампочки в браузерах для снижения нагрузки.

Заключение

В результате тестирования, данная система показала возможность управления светодиодом, с помощью с браузера, при этом все клиенты, своевременно получали информацию об изменении состояния светодиода.

За счет выделения второго сервера raspberrу не испытывала проблем с нагрузкой, и смогла своевременно отвечать на все запросы.

Подобное решение, в дальнейшем может быть расширено, и использовано не только для мигания светодиодом, а в более серьезных и востребованных целях, например включения микроволновки подогреть еду заранее. Такая реализация системы позволяет с легкостью добавить сколько угодно устройств. При необходимости, с этими устройствами можно общаться по беспроводной связи.

Список литературы

1. Хартов В. Я. Микропроцессорные системы: учебное пособие. М.: Академия, 2014. 368 с.
2. Хоровиц П., Хилл. У. Искусство схемотехники. В 2-х т. Т. 2: пер. с англ. 4-е изд., стереотип. М.: Мир, 1993. 590 с., ил. [The Art of Electronics, 4th ed. Paul Horowitz, 1993, 590 p.].

3. Мартин Т., Микроконтроллеры ARM7. Семейство LPC2000 компании Philips. Вводный курс. М.:Издательский дом «Додэка-XXI», 2006. 240 с.