

# Обзор операционной системы IoT: Contiki OS и ее коммуникационные модели

Паял Малик, Мальвика Гупта

**Аннотация:** Виртуально зависимые устройства, такие как интернет и компьютеры, зависят от человека в плане получения информации, которая фиксируется в виде набора текста, нажатия кнопки, цифрового изображения или сканирования штрих-кода с неравномерным объемом в 1024 терабайта и доступна в интернете. В этой статье автор описал операционную систему CONTIKI OS, основанную на IoT, которая используется для передачи данных от одного узла к другому с достоверностью. Интерфейс CONTIKI OS основан на JAVA, который является эффективным и безопасным благодаря множеству уровней безопасности. Protothread, системный цикл и micro ip - это функции, которые поддерживает CONTIKI OS, а также поддерживает сетевые протоколы, такие как TCP/IP и IPv6 от cisco. Автор также рассказал о коммуникационных моделях, которые помогают передавать данные по сети с безопасным соединением. Таким образом, можно сделать вывод, что CONTIKI OS обеспечивает безопасность при передаче данных от одного узла к другому, используя коммуникационные модели.

**Ключевые слова:** IoT, операционная система IoT, Contiki OS, модели IoT.

## 1. ВВЕДЕНИЕ

Взаимосвязь объектов, которая углубляется в настоящее время, и объекты не только собирают информацию с датчиков с взаимодействием физического мира при предоставлении услуг по передаче информации и приложений [1]. Инновационная технология с открытым исходным кодом Contiki OS, которая делает реализацию эффективной с

памятью элементарного протопотока. Модель многопоточности сравнивается с моделью стека, в которой для параллельных процессов требуется меньше вычислений и памяти. Протопоток также поддерживается Contiki OS, где для написания кодов или программ используется событийно-ориентированный и явный способ, который трудно понять и поддерживать. Модель выполнения определяется смешиванием бесстадийного потока с линейным событием. Симулятор Cooja используется под операционной системой Contiki OS, которая работает на интерфейсе JAVA. Преимуществом Contiki OS является профилирование энергопотребления и маломощная радиосеть для сетевых датчиков с эффективной оптимизацией. Она ориентирована на протоколы TCP, ICMP и IP [38] с минимальной абсолютной характеристикой при наличии полного стека TCP/IP и ограниченных ресурсов устройств TCP/IP и узлов датчиков. Модели IoT предполагают объединение в сеть интеллектуальных объектов (RFC), использующих методы передачи данных/информации по сети. Коммуникационные модели для IoT такие, как Device-to-device, в которых небольшой пакет данных используется для передачи информации между устройствами с низкой скоростью требования в этой модели в основном такие, как системы домашней автоматизации. Device-to-cloud позволяет расширить потребности конечного пользователя за счет расширения возможностей связи между устройствами. Device-to-gateway, в которой связь устанавливается между устройствами и прикладным уровнем, и это помогает осуществлять транзакции между устройствами и облачными сервисами. Бэкэнд-обмен данными, который состоит из гибридизации двух

вышеупомянутых моделей "устройство-шлюз" и "устройство-облако", что помогает быстро и эффективно передавать данные.

## 2. ОПЕРАЦИОННАЯ СИСТЕМА

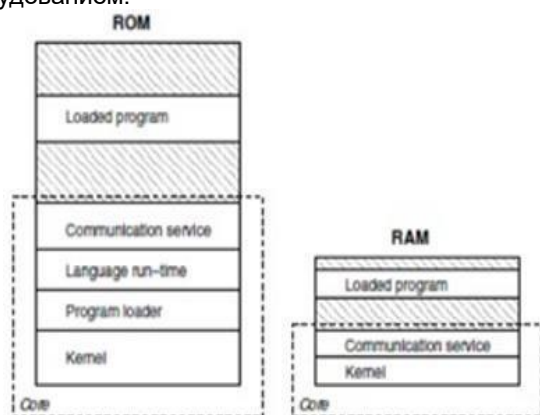
Теоретические исследования в области WSN, таких как сети датчиков, которые моделировали время жизни в реальных приложениях, которые были проведены

- Паял Малик 1, Мальвика Гупта 2
- 1 Инженерный колледж ABES, Газиабад, ЮП, Индия
- 2 Инженерный колледж ABES, Газиабад, УП, Индия

«Шведский институт компьютерных наук», и в его разработке участвует SICS (Адам Данкелс), в настоящее время генеральный директор Thinksquare (фокус на идеях IoT и инновациях через взаимосвязь устройств), в основном работающий над Contiki OS, которая является легкой и портативной ограниченной средой. TCP/IP и вытесняющая многопоточность - это пересекающиеся функции, которые поддерживаются Contiki OS с реализацией на стеке IPv6 от Cisco и Atmel [31]. Инновационная технология с открытым исходным кодом Contiki OS делает реализацию эффективной с памятью элементарного протопотока. Модель выполнения основана на смешении бесстадийного потока с линейным событием [32], а преимуществом Contiki OS является профилирование мощности [33] и маломощная радиосеть для сетевых датчиков с эффективной оптимизацией. В основном, она состоит из двух частей: -

- Ядро
- Загруженные программы

Разделение в основном работает на время компиляции при развертывании, ядро состоит из ядра Contiki, а язык времени выполнения с библиотеками состоит из загружаемых программ, которые помогают взаимодействовать драйверу устройства с оборудованием.



**Рис.1.** Разметка операционной системы Contiki

Язык интерфейса бэкенда Contiki OS - JAVA, а в инструменте симулятора Cooja используется язык C, что позволяет легко переключать различные платформы, а с 2003 года было выпущено около 20 платформ [34].

## 2.1. Различные функции Contiki OS в IoT:

### Ядро, основанное на событиях

Выполнение или реализация кода осуществляется обработчиком событий, что означает, что код полностью зависит от события и никогда не прерывается одним блоком кода. Модель многопоточности сравнивается с моделью стека, в которой для параллельных процессов требуется меньше вычислений и памяти [35].

### Protothread

Помимо вытесняющего потока, в ОС Contiki поддерживается протопоток, где для написания сложных для понимания и сопровождения кодов или программ используется событийно-ориентированный и явный способ. Это сохраняет высокоуровневую реализацию функций с абстракцией языка программирования и без накладных потоков выполняет условную блокировку. Для одного протопотока в Contiki OS требуется 2 байта оперативной памяти [34] [35].

### Системный контур

Для инициализации нескольких процессов при запуске Contiki OS многократно используется функция `process_run ()` [35]. Эта функция регистрируется обработчиком опроса, где процессы по событию из системы очереди событий после этого функция вызывается обратно обработчиком, который все еще находится в очереди. Если очередь событий освобождается, то для пробуждения функцией `process_run ()` с новым событием обработчика требуется внешнее прерывание.

```

Int
Main(void)
{
    Beep();
    While(1)
    {
        While(process_run())>0
        /*watchdog_reset();*/
        Lpm_sleep();
    }
    Возврат ();
}

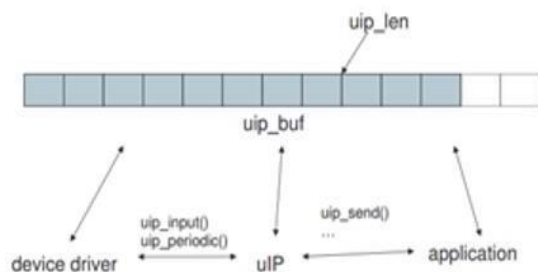
```

Приведенный выше исходный код взят из Contiki и рассказывает об общей демонстрации использования `process_run ()` [37] и аппаратной платформы. Всякий раз, когда узел снова просыпается, вызывается `Lpm_sleep()`, поэтому мы используем внешнее пробуждение `Lpm_awake()`.

### Микро IP (uIP)

Он ориентирован на протоколы TCP, ICMP и IP [34] с минимальной абсолютной характеристикой при наличии полного стека TCP/IP и ограниченных ресурсов устройств TCP/IP и узлов датчиков. Содержать один максимальный размер пакета в одном глобальном буфере, достаточно большом, чтобы вместить пакет. Уведомлять приложение uIP об анализе пакета данных и сетевые устройства, когда новый uIP вызывается для

обработки данных. Данные, которые должны быть перезаписаны в пакете другим входящим пакетом, чтобы избежать, мы используем приложение uIP, потому что оно имеет только один буфер.



**Рис.2.** Глобальное использование буфера в uIP

			кб
С	Дробный	Нет	Да
С++	Нет	Нет	Да
Многони- точные	Дробный	Дробный	Да
В режиме реального времени	Дробный	Нет	Да
Модульнос- ть	Дробный	Нет	Да

**Таблица. 1.** Требования к операционной системе

Текущий размер данных хранится в целочисленной переменной `uip_len`, а для глобального буфера используется `uip_buf` [33]. Функции, назначаемые для драйверов устройств, следующие: -

- `Uip_input()`: Пакет, полученный драйвером устройства и сохраненный в глобальном буфере, пока он проверяется драйвером устройства на отсутствие исходящего пакета.
- `Uip_periodic()`: Для любой ретрансляции необходимо периодически выполнять эту функцию через драйвер устройства. Симулятор `Cooja` используется под `Contiki OS`, которая работает на интерфейсе `JAVA`, а в инструменте `Cooja` мы используем язык `C` в качестве бэкенда для программирования. Вместо `Contiki OS` используются различные ОС:
- `mbed OS`: Для разработки IoT-приложений и их запуска на базе меньшей емкости, памяти, хранения и обработки данных используется фреймворк `C++` [32].
- `TinyOS`: Некоммерческая ОС, разработанная для беспроводных сенсорных сетей и работающих в условиях ограниченных ресурсов, с функцией управления сетью и повсеместными вычислениями [33].
- `MicroC OS`: Разработана для ОС реального времени и встраиваемых устройств для микроконтроллера с низким энергопотреблением [34].
- `RIOT OS`: использует фреймворк `C++` с эффективной высокой степенью программирования, что позволяет оптимально использовать микроконтроллер и основывается на микроядрах [34].
- `Brillo OS`: Новая ОС для устройств с низким энергопотреблением, представленная Google в 2015 году, например, платформы `android` (`android-смартфон`) [35].

OS	ContikiOS	TinyOS	RIOT OS
Минимальная оперативная память	Менее 2 кб	Менее 1 кб	Приблизительно 1,5 кб
Минимальное ПЗУ	Менее 30 кб	Менее 4 кб	Приблизительно 5

### 3. МОДЕЛЬ КОММУНИКАЦИИ В IoT

Сеть интеллектуальных объектов (RFC) [22], руководство было выпущено в марте 2015 года Архитектурным советом Интернета. К четырем основам коммуникационной модели относятся: -

#### Связь между устройствами

Прямое соединение устанавливается между двумя или более устройствами и взаимодействует друг с другом, а не требует немедленного использования сервера приложений. Для связи используется несколько сетей с несколькими протоколами, такими как Bluetooth [23], Z-wave [24] или ZigBee [25], которые устанавливаются для прямой связи между устройствами. Небольшой пакет данных используется для передачи информации между устройствами с низкой скоростью, требуемой в данной модели, в основном, для систем домашней автоматизации. В соответствии со статьей в журнале IETF, многие подходы, связанные с проблемой совместимости, подтверждены коммуникацией между устройствами: "Прямые отношения часто возникают между устройствами, встроенные механизмы конфиденциальности и доверия, которые они обычно имеют, но режимы данных обычно используют специфические для конкретного устройства, что требует избыточных усилий по разработке" [26]. Реализуется формат данных, специфичный для конкретного устройства, а не открытый подход, использующий стандартный формат.

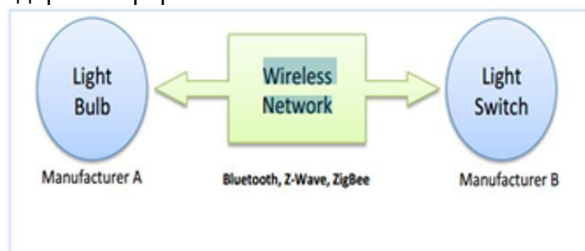


Рис.3. Связь между устройствами

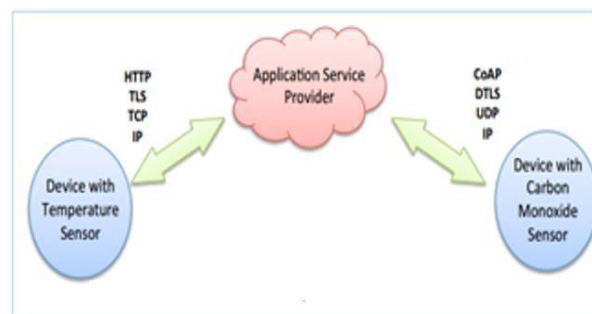


Рис.4. Связь устройства с облаком

#### Модель "устройство - шлюз"

Связь устанавливается между устройствами и прикладным уровнем, и это помогает проводить транзакции между устройствами и облачными сервисами. Облачные сервисы не могут получить доступ напрямую, поэтому для доступа к ним требуется носитель, например, Fitbit нуждается в носителе для отправки или получения данных.

#### Связь между устройствами и облаком

Между устройствами устанавливается прямое соединение с облаком Интернета, которое помогает в обмене информацией и управлении трафиком. Соединение устанавливается между IoT-устройствами с IP-сетью, которая ведет к облачным сервисам, что делается или используется с помощью традиционной проводной или беспроводной сети. "Умный телевизор" [27] - самый популярный пример IoT-устройства, использующего интернет-соединение для передачи информации для просмотра и интерактивного распознавания речи. Эта модель расширяется в соответствии с требованиями конечного пользователя по мере расширения возможностей продукта.

из облака, поэтому для подключения облачного сервиса используется смартфон [21].

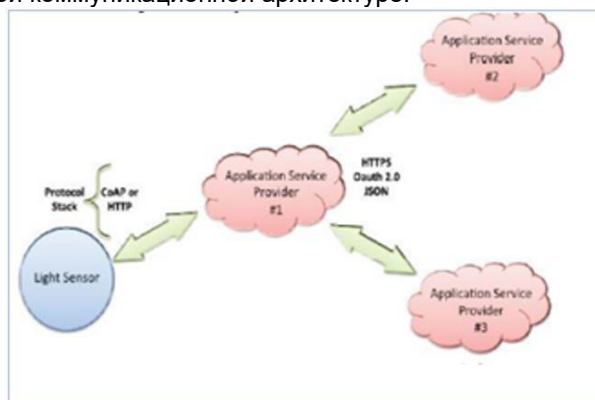
- [2] L. atzori, A. Iera, G. morabito, "Internet of Things: Обзор компьютерных сетей", № 54(15) (2010) 2787-2805.



**Рис.5.** Связь между устройствами и шлюзом

### Модель совместного использования данных на задней стороне

Комбинация данных и других объектов из облака экспортируется и анализируется пользователями в этой коммуникационной архитектуре.



**Рис.6.** Модель совместного использования данных на задней стороне

Данные датчиков могут быть загружены третьей стороной, и эта модель позволяет осуществлять связь между одним устройством и устройством.

## 4. ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШАЯ РАБОТА

Интернет вещей - это защита информации при обмене с малым бизнесом, который сталкивается с такими проблемами, как хранение огромного количества данных и т.д. CONTIKI OS работает под платформой IoT, а интерфейс основан на JAVA для обеспечения подлинности. Коммуникационные модели поддерживают CONTIKI OS для передачи данных по каналу связи от одного к другому. В будущем автор может сосредоточиться на архитектуре IoT с уровнем безопасности промежуточного программного обеспечения и архитектуре рисков в IoT. Эти архитектуры помогают предотвратить различные атаки при передаче информации от одного узла к другому.

## ССЫЛКА

- [1] Shen, guicheng and bing wu liu, "The vision, technologies, application and security issues of IoTII, E- Business and E-Government, International conference on IEEE 2011.



- [3] Серф, Винт и Морин Олхаузен, "Интернет вещей", семинар ФТК по IoT, Вашингтон, округ Колумбия, 19 ноября 2013 г.
- [4] IoT обзор с сайт  
ссылка:  
[https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151014\\_0.pdf](https://www.internetsociety.org/sites/default/files/ISOC-IoT-Overview-20151014_0.pdf).
- [5] Скотт Брейв, Хироши Исии и Эндрю Дали, "Осязаемый интерфейс для удаленного сотрудничества и общения", медиалаборатория MIT с группой Tangible media.
- [6] Ала-аль-фукаха, Мохсен Гизани, Мехди Мохаммади, мд. Aledhari, Moussa ayyash "Internet of Things: A survey on empowering technologies, protocols and application", IEEE communication survey and tutorial, vol-17, no. 4, четвертый квартал 2015 года.
- [7] Krushang sonar, Hardik Upadhyay, "A survey of security privacy issues of internet of things".
- [8] C. buckl, S. sommer, A. scholz, A. kvoll, A. kemper, J. heveret al, "Service to the field: Подход для сенсорной сети с ограниченными ресурсами", семинар по передовым информационным сетям и приложениям, представленный на Международной конференции IEEE 2009, стр. 968-975.
- [9] M U Farooq, Md. Waseem, Sadia mazhar, Anfum khairi, Kamal, "A review of IoT", International journal of computer application (0975-8887) vol. 11, no.-3, 1, march, 2015.
- [10] Tschofenig, H., et. al., "Internet Architecture Board: Architectural Considerations in Smart Object Networking", Tech. no. RFC 7452., Mar. 2015.
- [11] Тухин Боргохайн, Удай Кумар и Сугата Саньял, "Исследование вопросов безопасности и конфиденциальности Интернета вещей".
- [12] Презентация с сайта bluetooth.com для определения и подробного ознакомления с ним.
- [13] Презентация Z-wave.com для определения и подробного ознакомления с ним.
- [14] Презентация zigbee.com для определения и подробного ознакомления с ним.
- [15] Даффи Марсан, Кэролин, "IAB выпускает руководство для разработчиков Интернета вещей", журнал IETF, 11.1, июль 2015 г., веб. Ссылка: [https://www.internetsociety.org/sites/default/files/Journal\\_11.1.pdf](https://www.internetsociety.org/sites/default/files/Journal_11.1.pdf)
- [16] "Samsung Privacy Policy-Smart TV Supplement", Samsung corp.web, 29Sept2015, веб-ссылка: [www.samsung.com/sg/info/privacy/smart tv](http://www.samsung.com/sg/info/privacy/smart tv).
- [17] "Инициатива глобальных стандартов Интернета вещей", МСЭ, 26 июня 2015 г.
- [18] "Международный союз электросвязи: Обзор Интернета вещей", ITU-T Y.2060, июнь 2012 г.
- [19] Pew, Report: "An In-depth Look at Expert Responses", Исследовательский центр Интернета, науки и технологий в 14 мая 2014 года и получено 26 июня 2015 года.
- [20] Mathilde Durvy, Julien Abeillé, Patrick Wetterwald, Colin O'Flynn, Blake Leverett, Eric Gnoske, Michael Vidales, Geoff Mulligan, Nicolas Tsiftes, Niclas Finne, and Adam Dunkels, "Making Sensor Networks IPv6 Ready", ACM 6<sup>th</sup> Proceedings conference on Embedded network sensor systems, USA, 2008, pp. 421-422.
- [21] А. Дункельс, О. Шмидт, Т. Фойгт, А. Мунеб, "Protothreads: Simplifying event-driven programming of memory constrained embedded systems", ACM 4<sup>th</sup> Proceedings Conference on Embedded Networked Sensor Systems, Colorado, USA, 2006. Web link: <http://dunkels.com/adam/dunkels06 protothreads.ppt>. [Accessed: 02-Mar-2014].
- [22] A. Dunkels, "Contiki: Bringing IP to Sensor Networks", ERCIM News, № 76, pp. 59-60, Jan- 2009. Веб-ссылка: <http://ercim-news.ercim.eu/images/stories/EN76/EN76-web.pdf>.
- [23] А. Дункельс, "Операционная система "Contiki", ссылка на страницу: <http://www.sics.se/~adam/Contiki/> доступ получен 22 января 2006 года.
- [24] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki: Облегченная и расширяемая операционная система для миниатюрных сетевых датчиков", 1<sup>st</sup> Труды IEEE на семинаре по встроенным сетевым датчикам, Тампа, Флорида, США, ноябрь 2004 года.
- [25] A. Dunkels, O. Schmidt, and T. Voigt, "Real world wireless sensor networks: Использование Protothreads для программирования сенсорных узлов", Материалы семинара REALWSN, Стокгольм, Швеция, июнь 2005 года.
- [26] Онлайн-документация "Операционная система Contiki 2.x" была предоставлена А. Дункельсом со ссылкой на веб-страницу: <http://contiki.sourceforge.net/html>, доступ получен 22 января 2006 года.
- [27] A. Dunkels, "Full TCP/IP for 8 Bit Architectures", 1<sup>st</sup> Proceedings of ACM/Usenix, Applications and Services (MobiSys), International Conference on Mobile Systems, San Francisco, May 2003.
- [28] Тимоти Малче, Прити Махешвари, "Review: Harnessing the Internet of Things (IoT)", август 2015, Международный журнал передовых исследований в области компьютерных наук и программной инженерии, ISSN: 2277 128X, том 5, выпуск 8, стр. 320-323.
- [29] Веб-ссылка: <http://www.tinyos.net/> для детального ознакомления с Tiny OS, [Accessed: 13- May- 2016].
- [30] Ядра реального времени: µC/OS-II и µC/OS-III с подробными знаниями в веб-ссылке: <https://www.micrium.com/rtos/kernels/>, [Accessed: 13- Май - 2016].
- [31] Подробные знания о РИОТ в интернете по ссылке: <https://www.riot-os.org/>, [Accessed: 13- May- 2016].
- [32] Подробные сведения о Brillo в веб-ссылке: <https://developers.google.com/brillo/>, [Accessed: 13- May-2016].
- [33] А.М. Виламовска, Э. Хатциандреу, Р. Шиндлер, К. Ван Оранье, Х. Де Врис, Дж. Крапельсе, "Применение RFID в здравоохранении - оценка и определение областей для внедрения RFID в здравоохранении", RAND Europe, февраль 2009 года.
- [34] Е. Уэлборн, Л. Бэттл, Г. Коул, К. Гоулд, К. Ректор, S. Raymer, M. Balazinska, G. Borriello, "Building internet of things using RFID: the RFID ecosystem experience", IEEE Internet Computing 13, 2009, 48-55.
- [35] Сайт веб ссылка: <http://www.sensei-project.eu/>

Проект 7РП "SENSE" со сценарием портфеля,  
требованиями пользователя и контекста, результат.