

М. А. Шимченко
marina.shimchenko@intel.com
Московский физико-технический институт (государственный университет)
ЗАО «Интел А/О»

Методы моделирования ИОТ систем.

Современные симуляторы предоставляют широкий спектр возможностей для моделирования различных систем. Что такое классическая система в терминах моделирования? Это процессор, подключенные к нему внешние устройства общего пользования, т.е. usb-устройства, монитор, проектор и т.д. Таких устройств ограниченное количество, все они известны, и мы можем предсказать какие из них будут использоваться в моделируемой системе, а какие нет. Симулирование таких систем автоматизировано, процесс разработки симулятора не занимает много времени, а тестирование позволяет покрыть большинство сценариев использования. В таких системах преимущества виртуализации не вызывают сомнений:

1. Уменьшение затрат на конфигурирование аппаратной модели и поддержание ее в рабочем состоянии.
2. Возможность тестировать программное обеспечение на оборудовании, которое уже снято с производства и не представлено на рынке.
3. Возможность запускать тесты в одних и тех же условиях, на разном оборудовании.
4. Повышение скорости тестирования, возможность автоматизации тестирования, сохранение логов.
5. Стало легче делиться результатами с коллегами по всему миру.
6. Симуляция более предсказуема, чем аппаратура, а значит проще управлять процессом.
7. Естественные преимущества программного подхода: обратное исполнение, чекпоинты и т.д.[1]

Но на арену выходит новое перспективное[2] направление - ИОТ(Internet of things), интернет вещей, или как его называют в последнее время за его широкое распространение – интернет всего (рис. 1).

Интернет вещей — концепция вычислительной сети физических предметов (*«вещей»*), оснащённых встроенными технологиями для взаимодействия друг с другом или с внешней средой¹. Это принципиально новое направление, потому что при моделировании такой системы мы сталкиваемся с проблемой комбинационной сложности разрабатываемых систем. Это происходит потому, что основной составляющей ИОТ являются сенсоры и датчики. Сенсор (датчик) - это устройство для преобразования некоторой физической величины в электрический сигнал. Таких внешних устройств огромное количество – например, на сайте интернет-магазина «РОБОТОТЕХНИКА» представлено около 100 наименований², в дополнение существует несколько интерфейсов работы с такими внешними устройствами: I2C, SPI, GPIO, UART. Для работы с ними данные должны быть обработаны по-разному в каждом конкретном случае.

¹ Internet of Things (англ.). Gartner IT glossary. Gartner (5 May 2012). — «The Internet of Things is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment»

² <http://www.robototehnika.ru/e-store/catalog/202/> - сайт интернет-магазина «РОБОТОТЕХНИКА»

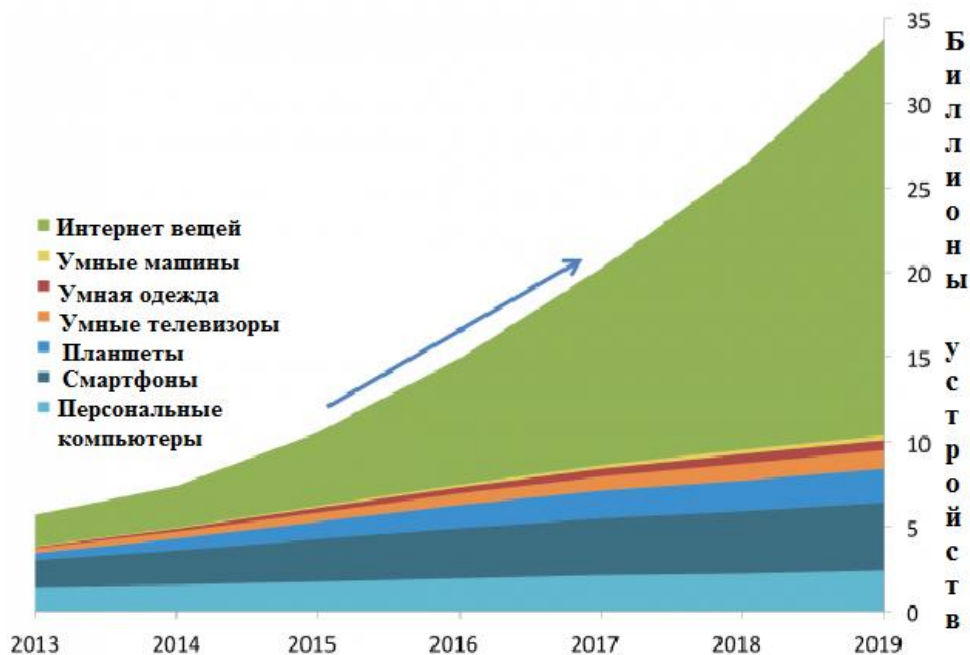


Рис. 1. Прогноз увеличения рынка рынка «Интернета всего» к 2019 г.[3]

«В типичном сценарии (прим. : для разработки IOT систем), только покупка 1000 устройств обойдется почти в 2 миллиона долларов. Ко всему прочему, трудности установки и поддержания работы всего этого оборудования делают задачу тестирования таких систем невыполнимой», - делится Эд Иллидж, вице-президент по управлению разработкой и тестированием в компании Wind River[1]. Филипп Мэйзонеv называется такое разнообразие конфигураций «матрицей боли», что еще раз подчеркивает актуальность данной темы в современных реалиях.

Конечно, такой мощный инструмент как Симикс³ предоставляет некоторые возможности для решения данной проблемы. Более того, этот симулятор является уникальным в своем роде, т.к. является единственным решением в области разработки IOT систем, требующих большого количества устройств.

«Симикс поддерживает тестирование любых IOT устройств, а также сценариев их конфигурирования... Такое тестирование не возможно любыми другими способами. »

Данная работа посвящена изучению способов моделирования сложных непредсказуемых IOT систем, в частности моделированию сенсоров, выявлению недостатков и достоинств каждого метода. На основе этих данных будет предложено решение, которое по возможности будет интегрировать преимущества каждого варианта и устранять негативные стороны.

³ <http://www.intel.com/content/www/us/en/intelligent-systems/software/simics-simulation-technology-to-accelerate-software-development.html>

На основании выявленных возможностей для улучшения различных показателей эффективности, предполагается реализовать в дальнейших работах предложенную здесь концепцию и провести ряд дополнительных исследований.

Существующие решения в полноплатформенном симуляторе Симикс⁴.

1. Программный способ.

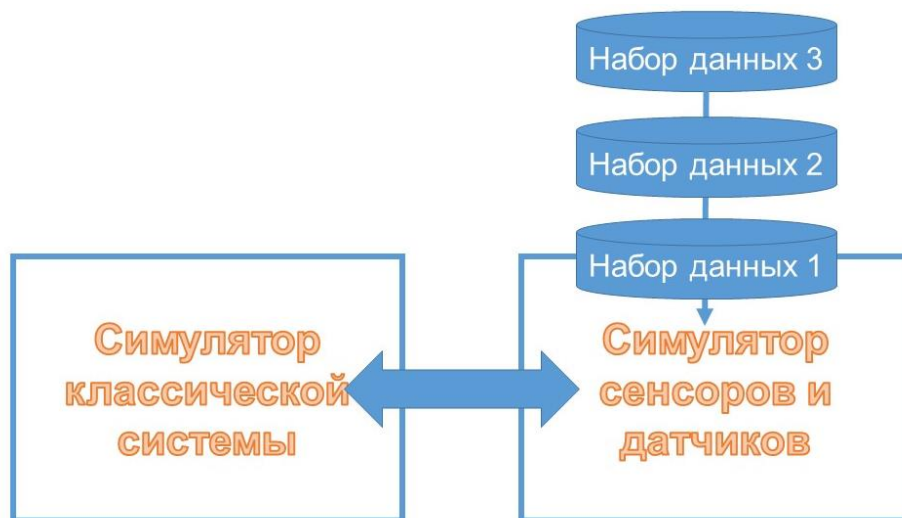


Рис. 2. Программный подход или симулятор-симулятор.

Пишется программа, которая может являться внешней по отношению к симулятору системы или быть его расширением. Эта программа является симулятором сенсоров и датчиков. На вход она может получать либо данные с симулятора системы, либо статичные наборы данных, которые были каким-то образом получены с реальных сенсоров и датчиков.

Основные преимущества:

- a. Среднее время реализации решения. Время в данном случае складывается из времени написания симулятора датчиков и интеграции его с симулятором системы, а также времени получения репрезентативного набора данных. Сложность симулятора датчиков, а также время интеграции может варьироваться в зависимости от конечной цели и требуемой скорости.

Основные недостатки:

- a. Слабое тестовое покрытие, т.к. применяется ограниченный набор данных.
- b. Не является унифицированным решением. Нужно заранее знать набор сенсоров, которые будут использоваться. В противном случае при добавлении нового датчика необходимо модифицировать симулятор.

Необходимые компоненты:

⁴ В данной работе будут рассматриваться в основном методы моделирования, которые используются в симуляторе Симикс. Но их можно рассматривать как обобщенные и абстрагированные на концептуальном уровне.

- a. Инженер(ы), который(е) напишет(ут) симулятор сенсоров и проинтегрирует(ют) его с симулятором системы.
- b. Сенсоры, с которых будут сниматься показатели.
- c. Дополнительное оборудование, чтобы снять показатели с сенсоров.

Конечно, такой подход может показать наивысшую эффективность в случае, если в разрабатываемой системе необходимо использовать всего несколько дополнительных внешних устройств и они заранее известны.

2. Низкоуровневое решение. «Классический» гибрид.

Концепция гибридной модели проста: часть функциональности симулируемого продукта находится в симуляторе, другая часть реализована вне симулятора (рис. 3). При этом модель симулятор-симулятор тоже может быть гибридной, если для интеграции двух симуляторов требуются дополнительные прослойки, обеспечивающие совместимость интерфейсов.



Рис. 3. Концептуальная структура гибридной модели.



Рис. 4. «Классическое» гибридное решение.

Обычно гибридные решения используются в качестве компромисса между шириной тестового покрытия и скоростью симуляции. Ту часть, которую хотят тщательно протестировать, помещают, например, на ПЛИС, все остальное оставляют в симуляторе. Чтобы обеспечить соединение между симулятором (программный продукт) и платой (аппаратное средство) требуется преобразователь интерфейсов между ними.

Основные достоинства:

- a. Широкое тестовое покрытие для определенного набора сенсоров.

Основные недостатки:

- a. Не является унифицированным решением. Т.к. при добавлении сенсора, который использует другой интерфейс взаимодействия с платой, нужно перепрошивать последнюю, а также изменять программные преобразователи.
- b. Низкая скорость интеграции. Нужно обеспечить правильное преобразование интерфейсов между симулятором и платой.

Необходимые компоненты:

- a. Инженер(ы), который(е) напишет(ут) преобразователь интерфейсов и проинтегрирует(ют) его с симулятором системы.
- b. Сенсоры.
- c. Плата.

3. Полугибридное решение на сокетaх.

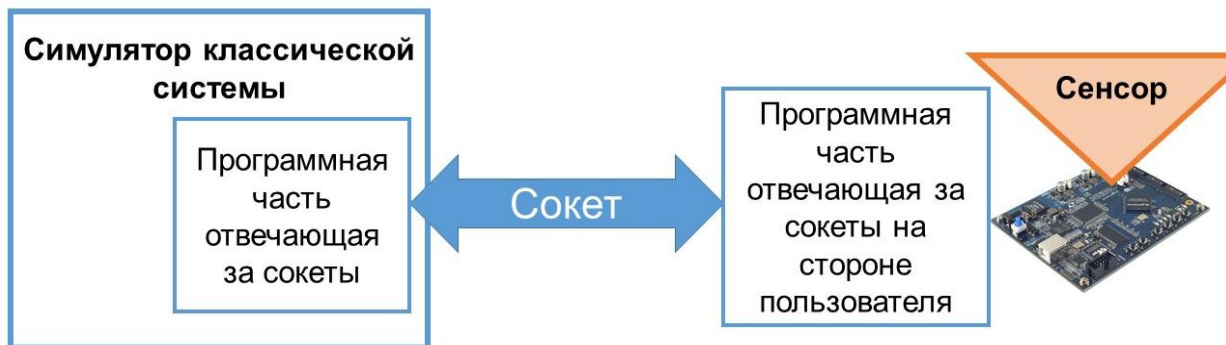


Рис. 5. Решение на сокетaх.

Такое решение называется полугибридным, потому что симулятор работает с аппаратурой не напрямую через преобразователи интерфейсов, а через специальную программную прослойку и программные средства. Одним из таких средств являются сокетa. На стороне симулятора появляется специальная часть, которая создает соединения, преобразует получаемые и отправляемые данные. Такая же часть нужна и на стороне пользователя. Схематично этот подход изображен на рисунке 6.

Это решение является пограничным между двумя предыдущими и объединяет достоинства обоих. Жаль, что и недостатки тоже в каком-то виде объединяются.

Основные достоинства:

- a. Широкое тестовое покрытие на определенном наборе тестов
- b. Быстрая интеграция. Имеется ввиду, что такое программное средство может быть имплементировано в симуляторе один раз и дальше просто переиспользоваться. При этом нет необходимости знать какое оборудование будет подключаться в дальнейшем.

Основные недостатки:

- a. Не является унифицированным решением. Набор сенсоров и интерфейсы взаимодействия должны быть заранее определены.
- b. Средняя скорость интеграции (со стороны пользователя). Требуется программная поддержка сокетов со стороны пользователя.

Необходимые компоненты:

- a. Инженер(ы), который(е) напишет(ут) программные части для поддержки соединения через сокет.
- b. Сенсоры.
- c. Плата.

Т.е. основное преимущество перед гибридным способом, в сокращении времени интеграции со стороны разработчиков симулятора. Поддержка сокетов может быть реализована один раз, а дальше переиспользоваться при необходимости. При этом остаются сложности с интеграцией для пользователя. И такой подход не уменьшает сложностей при использовании большого количества сенсоров с разными интерфейсами.

Как видно из обзора существующих подходов (рис. 7) для симуляции сенсоров при моделировании ИОТ систем, механизмов много, но все они содержат ряд критических недостатков. При этом преимущества каждого подхода содержатся в непересекающихся множествах, поэтому для разных сценариев использования удобно пользоваться только одним из них, что вынуждает разработчиков симулятора поддерживать сразу несколько вариантов. Хотелось бы иметь решение, которое объединяло бы достоинства всех вариантов, и не имело представленных недостатков (по возможности не имело бы новых).

Итак, максимальное тестовое покрытие не может быть обеспечено программными средствами. Значит, не обойтись без реальных сенсоров. При этом хотелось бы уменьшить время интеграции частей гибрида. Т.е. нужно уменьшить количество программных прослоек между симулятором и сенсором. В идеальном случае хотим получить модель, когда пользователь может без предварительной подготовки подключить сенсоры к симулятору. Как было сказано ранее, существует четыре интерфейса для работы с сенсорами, поэтому для унификации решения предполагается разработать подход, который поддерживал бы их все.

Подводя итог, хотим решение, поддерживающее I2C, SPI, GPIO, UART интерфейсы, которое бы легко интегрировалось с программным симулятором с одной стороны и аппаратной частью с другой.

Название способа моделирования / Показатели эффективности	Программный способ	Классический гибридный способ	Смешанный способ (сокеты)
Тестовое покрытие	Низкое	Высокое	Высокое
Унификация решения	Низкая	Средняя	Средняя
Среднее время разработки решения	Зависит от степени детализации требуемого решения	Небольшое	Небольшое
Среднее время интеграции компонент решения	Зависит от параметров эффективности конечного симулятора	Высокое	Среднее
Наиболее эффективная конфигурация	Заранее известный небольшой набор сенсоров, не изменяющийся на протяжении разработки.	Заранее известный набор сенсоров, использующих один и тот же интерфейс, не изменяющийся на протяжении разработки, но требующий тщательного тестирования.	Заранее известный набор сенсоров, использующих один и тот же интерфейс, не изменяющийся на протяжении разработки, но требующий тщательного тестирования.

Рис. 6. Сводная таблица преимуществ и недостатков для существующих решений моделирования сенсоров⁵ по предварительным показателям эффективности.

Решение.

AARDVARK I2C/SPI HOST ADAPTER – это переходник, который был специально создан для обеспечения взаимодействия между различными интерфейсами. Он поддерживает три из необходимых нам: I2C, SPI, GPIO. Обладает кроссплатформенностью – поддерживает Linux, Windows, Mac OS. Но главное идеально подходит для вышеописанной задачи, т.к. подсоединяется к компьютеру по USB-кабелю, а значит может быть легко подключен к симулятору. Он также легко может быть соединен с ПЛИС, т.к. имеет предусмотренные разъемы для этого. Новая конфигурация системы будет выглядеть как на рис. 7.

⁵Цвет выделяет преимущества и недостатки по степени градации их важности, где темно-красный – критический недостаток, зеленый – преимущество.



Рис. 7. Предложенное решение.

Использование такого решения, в первую очередь получает все преимущества, которые дает аппаратный подход. Пользователь может подключать любые сенсоры, а значит, увеличивается тестовое покрытие. При этом в каждом из предыдущих решений требовались сенсоры в их физическом воплощении, поэтому это никак не влияет на стоимость разработки системы. Интеграция в данном случае будет проходить эффективней, т.к. разработчикам симулятора требуется только раз поддержать совместимость симулятора с адаптером, а потом только переиспользовать это, вне зависимости от используемого сенсора. Со стороны пользователя не требуется каких-нибудь дополнительных трудозатрат по интеграции, они могут использовать адаптер как еще одно внешнее устройство (рис. 8).

Основные достоинства:

- a. Широкое тестовое покрытие, не ограниченное набором сенсоров.
- b. Быстрая интеграция, как для разработчиков симулятора, так и для пользователя.
- c. Является унифицированным решением для сенсоров, которые поддерживают I2C, SPI, GPIO.

Основные недостатки:

- a. Нет поддержки UART интерфейса.

Необходимые компоненты:

- a. Инженер(ы), который(е) напишет(ут) программные части для поддержки соединения через сокет.
- b. Сенсоры.
- c. Плата.
- d. AARDVARK I2C/SPI HOST ADAPTER.

Название способа моделирования	AARDVARK I2C/SPI HOST ADAPTER
Показатели эффективности	
Тестовое покрытие	Высокое
Унификация решения	Высокая
Среднее время разработки решения	Небольшое
Среднее время интеграции компонент решения	Небольшое
Наиболее эффективная конфигурация	Произвольные сенсоры.

Рис. 8. Информация по предварительным показателям эффективности для предложенного решения.

Заключение.

Проведя анализ существующих решений для моделирования сенсоров в симуляторе Симикс, было не выявлено унифицированного решения, которое бы покрывало все потребности при разработке ИОТ систем, комбинационная сложность которых подтверждается различными источниками, цитируемыми в работе. Данный симулятор является основным инструментом в данной области, но даже он, как было показано выше, не удовлетворителен с точки зрения удобства и эффективности по многим показателям. Анализ также выделил основные недостатки каждого решения, устранение которых стало основной задачей предложенного здесь способа. По теоретическим оценкам новый метод обеспечивает более широкое тестовое покрытие, проще для реализации и интеграции для разработчиков симулятора и пользователей Симикса, а также привносит в этот процесс все естественные преимущества программного моделирования. В последующих работах он будет реализован, и проведены дополнительные исследования и сравнения по другим показателям.

Литература:

1. *Wind River Systemc, Inc.* Wind River overcomes “Matrix of pain” in IOT challenges with Simics virtual testing environment» - 05/2016.
2. *IRONPAPER™*. Internet of Things Market Statistics - 5/03/2015.
3. *John Greenough*. The Internet of Everythingby. BI Intelligence Conference. – 2015.
4. *Asad Khan, Chris Wol*, Simics/SystemC Hybrid Virtual Platform. - 2015.
5. *Wind River Systemc, Inc.* Wind River Hindsight USER’S GUIDE. - 26/10/2012. - Revision 4078.