

1.

2. Интернет вещей

Концепция Интернета вещей в сегодняшние дни активно развивается и находит своё применения в огромном множестве сфер личного и промышленного использования. При этом, программирование устройств Интернета вещей всё ещё остаётся достаточно трудоёмким процессом и требует значительной квалификации разработчика, что затрудняет использование IoT для решения прикладных задач специалистами в их предметных областях.

3. ODEC

Одним из подходов, решающих данную проблему являются онтологически управляемые периферийные вычисления. Данный подход предлагает заменить использование классических прошивок (встраиваемых программы) периферийных устройств на комбинацию из:

- Доменной онтологии, описывающей общие знания.
- Онтологии задачи, описывающей действия, необходимые для решения конкретной задачи.
- Механизма логического вывода, запускаемого на периферийных устройствах и выполняющего эти действия.

Такой подход обладает массой преимуществ, и, в конечном счёте, позволяет добиться автоматизации процесса программирования устройств и снижения минимального необходимого уровня квалификации разработчика.

4. SciVi

Данный подход используется платформой научной визуализации и визуальной аналитики SciVi. Платформа SciVi позволяет описывать алгоритмы сбора, обработки и отображения данных, и выполнять эти алгоритмы, при этом храня и передавая их в виде онтологий.

5. Редактор SciVi

Причём описание алгоритмов происходит декларативно, с помощью графического редактора, в котором пользователь может легко описать необходимые действия в виде диаграмм потоков данных. Часть из этих действий как раз и может быть выполнена на устройствах Интернета вещей. Использование онтологически управляемых периферийных вычислений помогает платформе автоматизировать процесс их программирования. **Однако, уровень автоматизации программирования этих устройств остаётся недостаточным, в следствии того, что часть связанных с этим задач остаётся нерешённой в рамках платформы SciVi.** Решению этих и посвящена данная работа.

6. Объект и предмет

Таким образом: **Объект исследования:** автоматизация периферийных вычислений. **Предмет исследований:** средства платформы SciVi для организации онтологически управляемых периферийных вычислений.

7. Цель ВКР:

Создание комплексного решения по автоматизации программирования устройств Интернета вещей на базе платформы SciVi путём реализации недостающей функциональности в рамках концепции онтологически управляемых периферийных вычислений

8. Недостающая функциональность

К упомянутым в постановке цели возможностям, которых недостаёт платформе для достижения желаемого уровня автоматизации:

- **Автоматическое обнаружения периферийных устройств в локальной сети**
- **Возможности эффективного взаимодействия с энергонезависимой памятью периферийных устройств**

Два этих пункта и являются необходимо функциональность, в реализации которой заключалась данная работы.

9. Задачи

10. ESP8266

Основным устройством IoT, используемым платформой SciVi являются микроконтроллеры ESP8266. В следствии этого, именно они являются целевой аппаратной платформой для всей последующей работы. Для программирования ESP8266 используется язык программирования C++ и средства среды разработки Arduino IDE.

11. Менеджер EEPROM

Первой практической задачей данной работы стала разработка библиотеки управления энергонезависимой памятью устройств Интернета вещей. В качестве такой памяти в целевых устройствах используется EEPROM, т.е. электрически стираемое перепрограммируемое постоянное запоминающее устройство.

12. Требования

Перед такой библиотекой были поставлены требования:

- (a) Наличие возможности сохранять и считывать данные произвольной структуры из EEPROM
- (b) Высокоуровневый интерфейс
- (c) Обращение к хранимым данным по именам, а не физическим адресам

В качестве второстепенного требования библиотека должна минимизировать количества операций записи, т.к. оно ограничено для EEPROM каждого устройства.

13. Существующие решения

Были изучены существующие решения для управления EEPROM, которые обладают общими недостатками в виде:

- (а) Отсутствие возможности именования данных
- (b) Отсутствие механизмов оптимизации под целевые устройства. Они необходимы, так как в EEPROM ESP8266 обладает рядом особенностей
- (с) Ряд из них также позволяют хранить данные только базовых тип, но не создаваемых пользователем.

В конечном счёте эти недостатки делают невозможным выполнение поставленных платформой требований с помощью существующих средств.

14. Диаграмма классов

В следствии этого была спроектирована собственная библиотека, диаграмма классов которой представлена на слайде. На этой диаграмме мне хотелось бы обратить ваше внимание на два участка.

15. Класс чтения/записи

Первый из них - класс, предоставляющий низкоуровневые возможности для чтения и записи произвольных данных по конкретным адресам. Стандартные библиотеки для работы с EEPROM различных устройств предоставляют различные интерфейсы, в следствии чего часть методов данного класса является абстрактной и может быть реализована для различных устройств.

16. Реализации класса чтения/записи

В рамках работы были разработаны две их реализации: для микроконтроллеров ESP8266 и ATmega как для наиболее популярной платформы в целом.

17. Конструктор. Автоматическая адресация

Также важно обратить внимание на интерфейс, который предоставляет библиотека. Показанный на слайде класс позволяет описать один блок пользовательских данных в EEPROM. Как можно увидеть, его конструктор, в состоянии с требованиями, позволяет обращаться к данным по именам, а не физическим адресам. Уточню, что второй параметр конструктора - значение по умолчанию, а не адрес, как могло бы показаться.

18. Переопределения операторов для приведения типов

Также для данного класса переопределён ряд операторов, реализующих приведение типов от данного класса к типу хранимых им данных и обратно. Что позволяет пользователю оперировать объектом этого класса, как самими данными, хранимыми в нём.

19. Преимущества разработанной библиотеки

Помимо озвученного, разработанная библиотека реализует механизм уменьшения износа EEPROM специально с учётом особенностей целевых устройств.

Таким образом, разработанная библиотека полностью удовлетворяет поставленным требованиям для использования в рамках платформы SciVi.

20. Средство автоматизации обнаружения периферийных устройств в локальной сети

21. Анализ протоколов самоидентификации

22. SSDP

23. Диаграмма состояний синтаксического анализатора

24. Диаграмма рабочего цикла. Анализ полученного запроса

25. Диаграмма рабочего цикла. Добавление необязательных заголовков

26. Расширение протокола. Фрагмент диаграммы классов

27. Достигнутые результаты

28. Список слайдов