

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»
(БГТУ им. В.Г.Шухова)

(ТИТУЛЬНИК ВРЕМЕННЫЙ) Выпускная квалификационная
работа
дисциплина «.»
«.»

Выполнил: студент группы ВТ-41

Макаров Д.С.

Проверил:

Шамраев А.А.

Белгород 2021

Содержание

Содержание	1
1 Введение	2
2 Основная часть	2
2.1 Описание предметной области, анализ и выбор методов решения задач	2
2.1.1 Описание и анализ предметной области	2
2.1.2 Выбор методов решения задачи	4
2.2 Проектирование аппаратного обеспечения	4
2.2.1 Разработка принципиальной схемы	4
2.2.2 Разработка печатной платы	4
2.3 Проектирование программного обеспечения	5
2.3.1 Разработка методов решения задач	5
2.3.2 Разработка и описание алгоритмов	5
2.3.3 Разработка структур данных	5
2.4 Программная реализация	5
2.4.1 Описание модульной структуры программы	5
2.4.2 Спецификации подпрограмм (методов)	5
2.4.3 Описание использованных внешних компонент и библиотек	5
2.4.4 Руководство пользователя	6
2.4.5 Тестирование и экспериментальная проверка программного- аппаратного комплекса	6
2.4.6 Оценка качества разработанного комплекса	6
3 Заключение	6
4 Список литературы	6

Введение

Основная часть

2.1 Описание предметной области, анализ и выбор методов решения задач

2.1.1 Описание и анализ предметной области

В данной выпускной квалификационной работе описано создание межпротокольного шлюза MQTT - CAN, в рамках программно-аппаратного комплекса умный дом.

Программно-аппаратный комплекс - набор аппаратных и программных средств, работающих для выполнения одной или нескольких связанных задач.

Аппаратная часть комплекса представляет собой: компьютеры и микроконтроллеры объединенные в одну вычислительную сеть, посредством гетерогенных сетей, а так же различные датчики и исполнительные устройства.

Программной частью комплекса являются встроенное программное обеспечение, реализующее функционал исполнительных устройств и датчиков, сетевое взаимодействие, различные программные модули сетевых протоколов.

Система домашней автоматизации решает ряд задач таких как:

- сбор и обработка данных полученных с датчиков системы, а так же с внешних источников.
- регулирование внутренних параметров системы, в зависимости от обработанных данных и предпочтений пользователей.
- обеспечение безопасности дома (пожарная безопасность, предупреждение аварийных ситуаций).
- упрощение однотипных процессов пользователей при помощи программируемых сценариев.

- экономия ресурсов домовладения, при помощи эффективного их использования.
- предоставление различных отчетов, по использованным ресурсам домовладения.

Саму систему можно разделить на подсистемы по функциональным возможностям.

- подсистема управления освещением
 - элементы управления освещением
 - модули управления естественным освещением (шторы, рольставни, электрохромные стекла)
 - датчики присутствия, освещения
 - контроллеры сцен освещения (RGB контроллеры)
- подсистема управления климатом
 - внутренние и внешние метеорологические датчики (термометр, барометр, гигрометр, датчик уровня CO_2 и чистоты воздуха)
 - модули управления климатом
 - термостаты, рекуператоры, гигростаты, приточная и вытяжная вентиляция (с электрическим управлением)
- подсистема безопасности
 - подсистема видеонаблюдения
 - модули детектирования утечек природного газа, утечек воды
 - умные замки
 - модули сигнализации и сирены
- управляющая подсистема

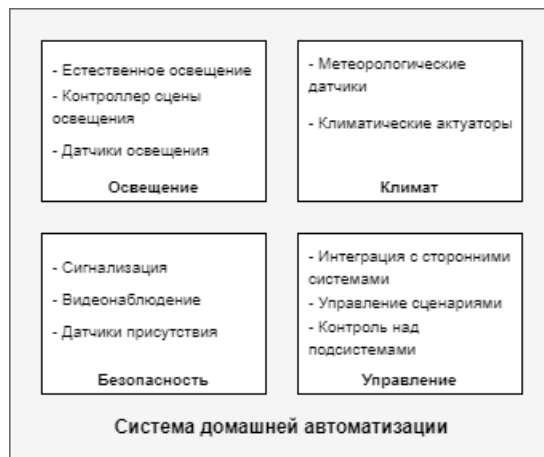


Рисунок 1 - Схема подсистем системы домашней автоматизации

@todo (почему то что на рынке плохо)

@todo (сравнительный анализ того что есть)

2.1.2 Выбор методов решения задачи

@todo (предлагаем гетерогенные приколы)

@todo (рассказываем про CAN)

@todo (рассказываем про MQTT)

@todo (рассказываем про Bluetooth LE Mesh)

@todo (рассказываем про все вместе)

@todo (схема концепции умного дома)

2.2 Проектирование аппаратного обеспечения

2.2.1 Разработка принципиальной схемы

@todo (нарисовать схему шлюза и рассказать про ее модули)

2.2.2 Разработка печатной платы

@todo (нарисовать плату и рассказать про нее)

2.3 Проектирование программного обеспечения

2.3.1 Разработка методов решения задач

@todo (схема модели OSI но с CAN)

2.3.2 Разработка и описание алгоритмов

@todo Адресная таблица: блок схемы

2.3.3 Разработка структур данных

@todo CAN_ID, DHCP_CAN_IP: структуры данных

@todo Адресная таблица: структуры данных

2.4 Программная реализация

2.4.1 Описание модульной структуры программы

2.4.2 Спецификации подпрограмм (методов)

@todo Адресная таблица: спецификации

2.4.3 Описание использованных внешних компонент и библиотек

Библиотека для работы с периферией микроконтроллера STM32CUB

Реализация семейства функции printf без зависимостей от stdlib.h

Для генерации символьных сообщений протокола MQTT удобно использовать функцию *sprintf*, из семейства функций *print formatted*, позволяющая выводить строки содержащие значения различных типов. В случае функции *sprintf*, она позволяет выводить в отформатированную функцию в переменную строки а не потоки ввода-вывода, в отличии от других функций семейства. Но функции входящие в стандартную библиотеку языка Си, не подходят для использования в встраиваемых система из-за ряда особенностей.

Во первых стандартная реализация функций *printf* имеет ряд зависимостей из стандартной библиотеки языка Си, которые после компиляции занимают

значительный объем в флеш памяти микроконтроллера (около 32 Кб).

Во вторых стандартная реализация *printf* использует динамическое выделение памяти, что не рекомендовано стандартами и руководствами разработки программно-аппаратных комплексов высокой надежности и ответственности (IEC 61508, MISRA C).

Для решения этой проблемы была использована сторонняя реализация функций семейства *printf* (ссылка на репозиторий) для языка Си, разработанная специально для использования в встраиваемых системах.

Данная реализация имеет совместимость с реализацией из стандартной библиотеки, но занимает значительно меньший объем в флеш памяти микроконтроллера (14 Кб против 32 Кб).

Так же реализация не использует динамическое выделение памяти для работы функций, что соответствует мировым стандартам разработки встраиваемых систем, а так же исправляет некоторые особенности стандартной реализации функций семейств *printf*, связанных с потоко-безопасностью.

Библиотека с сторонней реализацией распространяется с открытой лицензией MIT, что позволяет использовать данную реализацию в любых коммерческих и не коммерческих разработках.

2.4.4 Руководство пользователя

2.4.5 Тестирование и экспериментальная проверка программного-аппаратного комплекса

2.4.6 Оценка качества разработанного комплекса

Заключение

Список литературы