

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра «Компьютерные технологии»

«Допущен к защите»
Заведующий кафедрой _____

(Ф.И.О., ученая степень, звание)

« _____ » _____ 20__ г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Использование протокола TCP/IP в аппаратном средстве

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Садыбаев Н.С. БВР-10
(Фамилия и инициалы) группа

Научный руководитель Ержан А.А.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Эрмашева З.Д. ст. преподаватель
(Фамилия и инициалы, ученая степень, звание)
Эрмашева « 23 » 05 20 14 г.
(подпись)

по безопасности жизнедеятельности:

Шайдарбекова Н.К., К.Х.Н., доцент
(Фамилия и инициалы, ученая степень, звание)
Шайдарбекова « 23 » 05 20 14 г.
(подпись)

по применению вычислительной техники:

Ержан А.А., ст. преподав.
(Фамилия и инициалы, ученая степень, звание)
Ержан « 24 » 05 20 14 г.
(подпись)

Нормоконтролер: Ержан А.А., ст. преподав.
(Фамилия и инициалы, ученая степень, звание)
Ержан « 24 » 05 20 14 г.
(подпись)

Рецензент: Архараев А.
(Фамилия и инициалы, ученая степень, звание)
Архараев « 24 » 05 20 14 г.
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет „Информационные технологии“
Специальность „Вычислительная техника и программное обеспечение“
Кафедра „Компьютерные технологии“

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Сазабенов Нурбол Сабитович
(фамилия, имя, отчество)

Тема проекта Использование протокола TCP/IP в аппаратном средстве

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «__» _____ 20__ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

1. Выбор для реализации аппаратных средств
2. Формы связи TCP/IP
3. Функциональная часть
4. Принцип работы системы
5. Создание ТЗ применительно к используемому оборудованию
6. Технические характеристики
7. Надежность и достоверность

Перечень графического материала (с точным указанием обязательных чертежей)

16 - рисунки

13 - таблицы

Рекомендуемая основная литература

1. Левин В.К. Защита информации в информационно-вычислительных системах и сетях // Программирование - 1994, - №5. -
2. Моктеев М. Электронный учебник Web-дизайн Руководство по созданию м.: компьютер, Юньюн, 1996, - 267с.
3. Кидерст В. Web-мастеринг для профессионалов настольный справочник м.: БИНАМ, 2003, - 313с.
4. Ковыцкий И.И. Основы менеджмента: организация и планирование производства, задачи и лабораторные работы. - м.: Финансы и статистика, 1998.
5. Полюбе по расчету и проектированию естественного, искусственного и смешанного освещения - м.: Стройиздат, 1985.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
безопасн. техникум	Майдарбекова Ж.В.	12-22.05.14	Майдарбекова
Жонолосова	Брессева З.Д.	15.04-23.05.14	Брессева
Основная часть	Ержан А.А.		Ержан

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Основы систем аппаратного управления	07.04.2014	
2	АС для системы аппаратного управления	25.04.2014	
3	Проектирование и разработка системы аппаратного управления	30.04.2014	
4	Технико-экономическое обоснование разработки модели аппаратного управления через протокол TCP/IP	15.04-23.05 2014 12-22.05.2014	
5	Безопасность жизнедеятельности		

Дата выдачи задания «17» апреля 2014 г.

Заведующий кафедрой _____
(подпись) (Фамилия и инициалы)

Руководитель _____
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент _____
(подпись) (Фамилия и инициалы)

АНДАТПА

Бұл дипломдық жобада Ethernet желісі арқылы аппараттық басқарма жүйесіне арнайы зерттеу қарастырылады. Жобаның барлық идеясының мәні, бақылау үрдісі мен басқарма, адамның қатысуысыз барлық деркетемелерді зерттеп, бақылап және басқару болып табылады. Дайындау процесі барысында бизнес-жоспар құрастырылады.

Сонымен қатар жобаның техникалық жақтармен қатар өміртіршілік қауіпсіздігі мәселелері қарастырылған.

АННОТАЦИЯ

В данном дипломном проекте рассматривается разработка специального программного обеспечения для систем аппаратного управления посредством Ethernet сетей. Суть всей идеи проекта заключена в том, чтобы текущие процессы контроля и управления, которыми на сегодняшний день требуют присутствия человека заменить системой сбора и анализа данных, которая сможет, анализируя данные о процессах управления и состояния агрегатов и приборов, вносить необходимые коррективы в работу оборудования без вмешательства человека. В процессе разработки также составляется бизнес-план.

Также дипломный проект включает вопросы безопасности жизнедеятельности и освещены технические стороны проекта.

THE SUMMARY

In the given degree project working out of the special software for systems of hardware control by means of Ethernet networks is considered. The essence of all idea of the project is concluded in that current processes of control and management with which for today presence of the person demand to replace with system of gathering and the analysis of the data which can, analyzing the data about managerial processes and conditions of units and devices, to introduce necessary corrective amendments in work of the equipment without intervention of the person. In the course of working out the business plan also is made.

Also the degree project includes safety issues of ability to live and project technical aspects are shined.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1 ОСНОВЫ СИСТЕМ АППАРАТНОГО УПРАВЛЕНИЯ	10
1.1 ВЫБОР ДЛЯ РЕАЛИЗАЦИИ АППАРАТНЫХ СРЕДСТВ.....	10
1.1.1 AVR32.....	11
1.2 ТСР/ІР ПРОТОКОЛ.....	22
1.2.1 УРОВНИ СТЕКА ТСР/ІР.....	23
1.2.2 ФИЗИЧЕСКИЙ УРОВЕНЬ	24
1.2.3 КАНАЛЬНЫЙ УРОВЕНЬ . ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.	24
1.2.4 СЕТЕВОЙ УРОВЕНЬ	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА. 24
1.2.5 ТРАНСПОРТНЫЙ УРОВЕНЬ	25
1.2.6 ПРИКЛАДНОЙ УРОВЕНЬ.....	25
1.3 ОТЛАДОЧНАЯ ПЛАТА EVK1105 ОШИБКА! ЗАКЛАДКА НЕ	
ОПРЕДЕЛЕНА.	26
1.3.1 ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ	26
1.3.2 ОСНОВНЫЕ ПАРАМЕТРЫ.....	27
2 ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ СИСТЕМЫ АППАРАТНОГО	
УПРАВЛЕНИЯ	28
2.1 ЯДРО ОПЕРАЦИОННОЙ СИСТЕМЫ	28
2.2 БИБЛИОТЕКИ ПОД АППАРАТНУЮ ЧАСТЬ.....	29
3 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ АППАРАТНОГО	
УПРАВЛЕНИЯ	32
3.1 ПРИНЦИП РАБОТЫ СИСТЕМЫ.....	32
3.2 БИБЛИОТЕКА УПРАВЛЯЮЩЕЙ ПРОГРАММЫ.	33
3.2.1 AVRSTUDIO	34
3.3 ВЕБ СЕРВЕР	36
3.3.1 ДНСР.	36
3.3.2 СТРУКТУРА СООБЩЕНИЙ ДНСР.	37
3.4 СОЗДАНИЕ ВЕБ ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ	
ПРОГРАММ УПРАВЛЕНИЯ.....	40
4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ	
МОДУЛЯ АППАРАТНОГО УПРАВЛЕНИЯ ЧЕРЕЗ ПРОТОКОЛ ТСР/ІР ...	42
4.1 ЦЕЛЬ ПРОЕКТА	42
4.2 ТЕМА ПРОЕКТА	42
4.3 ВВЕДЕНИЕ.....	42
4.4 РАСЧЁТ СТОИМОСТИ РАБОТЫ ПО ПРОЕКТИРОВАНИЮ И РАЗРАБОТКЕ	
ЧЕРЕЗ ПРОГРАММЫ ДЛЯ МОДУЛЯ АППАРАТНОГО УПРАВЛЕНИЯ ПРОТОКОЛА	
ТСР/ІР.....	46

4.5 Цена интеллектуального труда	55
5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	56
5.1 ХАРАКТЕРИСТИКА РАЗРАБАТЫВАЕМОГО ПРИБОРА И ПОМЕЩЕНИЙ ПО СТЕПЕНИ ОПАСНОСТИ ПОРАЖЕНИЯ ЧЕЛОВЕКА ЭЛЕКТРИЧЕСКИМ ТОКОМ.....	57
5.1.1 РАСЧЕТ ЗАЗЕМЛЕНИЯ АППАРАТУРЫ.....	61
5.2 ВЫВОД ПО РАЗДЕЛУ ОХРАНЫ БЕЗОПАСНОСТИ ЖИЗНЕДЕЯТЕЛЬНОСТИ	67
5.3 РАСЧЕТ СИСТЕМЫ КОНДИЦИОНИРОВАНИЯ.....	65
5.4 РАСЧЁТ ЕСТЕСТВЕННОГО ОСВЕЩЕНИЯ.....	66
5.4.1 РАСЧЁТ ИСКУССТВЕННОГО ОСВЕЩЕНИЯ МЕТОДОМ КОЭФФИЦИЕНТА ИСПОЛЬЗОВАНИЯ.....	68
5.4.2 РАСЧЕТ ИСКУССТВЕННОГО ОСВЕЩЕНИЯ ТОЧЕЧНЫМ МЕТОДОМ.....	70
ЗАКЛЮЧЕНИЕ.....	72
СПИСОК ЛИТЕРАТУРЫ	73

ВВЕДЕНИЕ

Целью данного проекта является разработка специального программного обеспечения для систем аппаратного управления посредством Ethernet сетей. С последующей возможностью её установки в специальное оборудование, которое имеет универсальные средства для контроля над различными агрегатами и устройствами.

Суть всей идеи проекта заключена в том, чтобы текущие процессы контроля и управления, которыми на сегодняшний день требуют присутствия человека заменить системой сбора и анализа данных, которая сможет, анализируя данные о процессах управления и состоянии агрегатов и приборов, вносить необходимые коррективы в работу оборудования без вмешательства человека.

На момент начала разработки проекта, процесс управления и контроля оборудованием представлял собой либо конечный автомат, либо компьютерные системы мониторинга. Недостаток их в том, что в одном случае система не делает отчета о происходящем процессе, а в другом постоянная необходимость в человеческом контроле. Человек в управлении оборудованием может допустить критические ошибки. А использование конечных автоматов лишает системы гибкости, а также любая модернизация становится дорогостоящей.

Система, разработанная в рамках данного проекта, позволяет:

- осуществлять удаленное управление, т.е. имея доступ к интернету вести, управление и мониторинг происходящих процессов в реальном времени без использования специализированного ПО, нужен только обычный интернет браузер;
- исключить в сбоях работы системы человеческий фактор, а за счет дублирования и перепроверки шанс ошибки свести к минимуму;
- возможность создать интеллектуальную систему управления с возможностью авто оптимизации на основании анализа предыдущих данных и данных принимаемых от схожего оборудования;
- возможность легкой модернизации и расширения системы за счет использования универсальных решений и возможности построения системы по принципу сотовых ячеек;
- автоматизировать кроме процесса управления и контроля и побочные процессы, таких как экономический учет, тех обслуживание.

1 ОСНОВЫ СИСТЕМ АППАРАТНОГО УПРАВЛЕНИЯ

1.1 Выбор для реализации аппаратных средств

Данный дипломный проект посвящен разработке системы удаленного контроля над различной аппаратурой и механизмов посредством использования TCP/IP протокола. Суть подобной системы заключается в том, что вместо того чтобы заменить непосредственный контроль, на удаленный контроль над различными приборами и агрегатами. Особенность данного проекта заключена в том, что удаленный контроль осуществляется посредством использования обычных локальных сетей и сети Internet. Т.е. на производстве для осуществления контроля над техпроцессом вместо того, чтобы покупать специализированное оборудование, использующее специфическое ПО, использовать универсальные устройства, соединенные в сеть и управляемые удаленно через Internet или внутри локальной сети используя обычный интернет браузер. Для этих целей был подобрана серия 32-х битных микроконтроллеров фирмы AtmelAT32UC3. Данные контроллеры отвечают необходимым требованиям для реализации цели проекта. Для упрощения разработки ПО, была использована на базе контроллера AT32UC30512 отладочная плата EVK1105 (рисунок 1.1), содержащая все необходимые аппаратные компоненты для разработки и тестирования ПО. Благодаря этому была исчерпана необходимость разработки первичного тестового модуля.

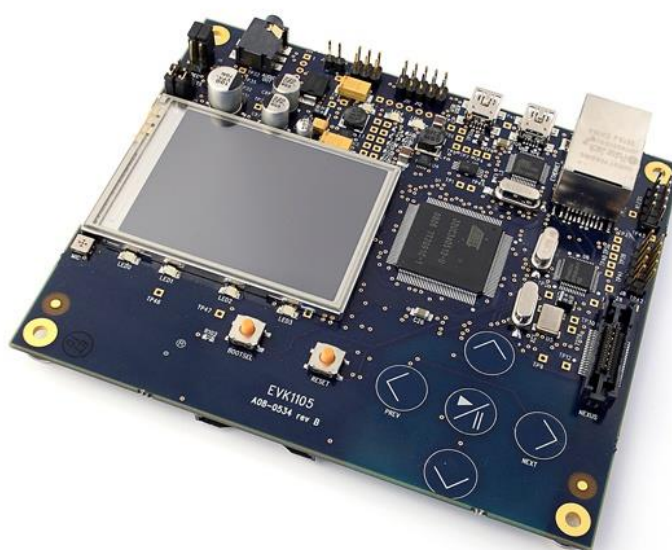


Рисунок 1.1 – Отладочная плата EVK1105

1.1.1 AVR32

AVR32 — 32 битные микроконтроллеры архитектуры RISC, анонсированные фирмой Atmel в 2006 году. Призваны конкурировать с архитектурами фирмы ARM. Может использоваться в КПК и других мобильных высокопроизводительных устройствах. Обладает примечательным соотношением производительность/тактовая частота. Имеет 2 семейства: AVR32 AP и AVR32 UC3.

В современном мире растет количество приборов для обработки нетекстовой информации, включаемой в понятие мультимедиа. Мультимедийные данные обрабатываются во всех биометрических приложениях, пользовательских системах позиционирования (GPS), развлекательных приборах для повседневной жизни и прочих устройствах, которые сегодня называют мультимедийными. В данной статье рассказывается о новом аппаратном решении фирмы Atmel, ориентированном на такие задачи - микроконтроллере AP7000 с архитектурой AVR32.

Всё новые и новые задачи, стоящие перед мультимедийными приборами, требуют выполнения всё более сложных алгоритмов обработки мультимедийной (и не только!) информации, что увеличивает вероятность достижения предела возможностей любого встраиваемого микроконтроллера. Алгоритмы быстрого преобразования Фурье (FFT), инверсного дискретного косинусного преобразования (iDCT) и другие математически-насыщенные алгоритмы, которые редко реализовывались на встраиваемых микроконтроллерах еще несколько лет назад, становятся более востребованными сегодня. В мультимедийных системах такие алгоритмы, как правило, используются для кодирования/декодирования данных, в т.ч. форматов MP3 и MPEG-4. Это требует высокой производительности элементной базы, на которой построены указанные приборы.

Исторически, задача повышения производительности решается за счет увеличения рабочей тактовой частоты процессора либо реализации многоядерных решений (ARM+DSP). По причине высокого энергопотребления такие решения могут использоваться не во всех встраиваемых системах, да и отладка двоядерных микроконтроллеров - задача не из легких. Массогабаритные характеристики портативных приборов определяются не только размерами печатной платы, но и батареей (а чем ниже энергопотребление, тем меньше и легче батарея).

Рассматривая задачу в таком контексте, складываются следующие требования к микроконтроллеру:

- Высокая производительность;

- Низкое энергопотребление;
- Высокая степень интеграции;
- Простота использования;
- Развитые средства разработки.

Задачей построения микроконтроллера с перечисленными выше требованиями еще в 2001 году занялась группа разработчиков компании Atmel, которая предложила решать её не повышением тактовой частоты, а улучшением архитектуры ядра процессора, который должен выполнить максимальное количество действий за один такт. Были начаты разработки в следующих направлениях:

1) максимально уменьшить количество инструкций пересылки данных (записи в память/чтения из памяти). Количество команд пересылки данных в RISC-контроллерах составляет порядка 30% от общего числа команд. Каждая из этих команд требует одного и более тактов. Избавление от этих операций должно дать существенный прирост производительности;

2) увеличить эффективность выполнения каждой инструкции в конвейере за счет операций с множественными данными (SIMD, SingleInstructionMultipleData). Многие алгоритмы, особенно в мультимедийных приложениях, требуют выполнения упорядоченных циклических операций над потоковыми данными. Так, например, при поиске смещения элементов движущихся фрагментов изображения при кодировании/декодировании MPEG-4, требуется вычисление суммы абсолютных разностей (SAD) в макроблоке размером 8x8 (или 16x16), которая находится для каждого пикселя в каждом кадре. Вычисление значения (1.1) потребует 24 операции на каждый пиксель (а их, допустим, 320x240):

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left| \text{pred}(x+i, y+j) - \text{ref}(x+dx+i, y+dy+j) \right| \quad (1.1)$$

где (x,y) - координаты точки, (dx,dy) - смещение относительно этой точки, Ниже приведен фрагмент кода на C, написанный для выражения (1.1).

```
/* Fromsad.c of the open source xvid codec */
uint32_t sad = 0;
uint32_t j;
```

```

uint8_tconst *ptr_cur = cur;
uint8_tconst *ptr_ref = ref;
for (j = 0; j < 8; j++) {
    //Compute SAD for 4 bytes
    sad += ABS(ptr_cur[0] - ptr_ref[0]);
    sad += ABS(ptr_cur[1] - ptr_ref[1]);
    sad += ABS(ptr_cur[2] - ptr_ref[2]);
    sad += ABS(ptr_cur[3] - ptr_ref[3]);
    //Compute SAD for next 4 bytes
    sad += ABS(ptr_cur[4] - ptr_ref[4]);
    sad += ABS(ptr_cur[5] - ptr_ref[5]);
    sad += ABS(ptr_cur[6] - ptr_ref[6]);
    sad += ABS(ptr_cur[7] - ptr_ref[7]);
    ptr_cur += stride;
    ptr_ref += stride;
};

```

3) Если взять за 100% всю операцию декодирования MPEG-4, то приведенный выше фрагмент кода занимает от 60 до 70% этого процессорного времени. Использование инструкций с множественными данными позволит оптимизировать данный алгоритм, так как за один такт можно выполнить несколько однотипных операций с разными входными данными;

4) повысить производительность машинного цикла за счет использования свободных ресурсов конвейера. Некоторые арифметические операции требуют одного такта на команду, а другие - существенно больше (если нет аппаратного математического акселератора, то на RISC-процессоре операция деления займет 32 такта). Если процессор будет ждать выполнения многотактовой операции до выполнения следующей команды, то эффективность работы всего конвейера снижается. Использование свободных ресурсов конвейера для несвязанных с выполняемой операцией вычислений позволит повысить его эффективность;

5) минимизировать задержки за счет прогнозирования переходов в циклах. В мультимедийных и криптографических алгоритмах всегда присутствуют блоки кода с внешним и внутренним циклом. Переход к внутреннему циклу может занимать от трех до пяти тактов. Так как таких циклических переходов может выполняться сотни (тысячи, десятки тысяч) в секунду, то задержки на переходы становятся заметными на фоне общего времени выполнения программы. Реализация логики, которая будет

прогнозировать переход, позволит снизить до нуля задержку на переход уже после первого прохождения цикла;

б) повышение плотности кода. В настоящее время относительная стоимость памяти программ невелика, поэтому мало кто обращает внимание на возможность оптимизации кода в своей программе. Но такой подход не является верным, так как в контроллерах, где используется кэш-память программ, не оптимально написанный код может привести к снижению производительности. Чем компактнее исполняемый код, тем большая полезная часть программы сможет уместиться в кэш-памяти, а это значит, что будет меньше промахов кэш-памяти и меньше обращений к внешней памяти.

В начале 2006 года корпорация Atmel анонсировала новое ядро AVR32 и микроконтроллер AT32AP7000 на его основе. Ядро AVR32 показало лучшие значения по производительности и энергопотреблению по сравнению с аналогичными микросхемами, ориентированными на рынок мультимедийных приложений. Так, например, алгоритм iDCT (инверсное дискретное косинусное преобразование) платформа AVR32 может выполнять на 35% быстрее, чем ядро ARM11. Такой запас можно проиллюстрировать по-другому: декодирование с частотой 30 кадров/с видеофрагмента размером "четверть" VGA (QVGA) формата MPEG-4 на платформе AVR32 может выполняться на частоте 100МГц, в то время как ARM11 сможет выполнять эту задачу только на частоте от 150МГц. При выполнении общепризнанных тестов на производительность консорциума EEMBC AVR32 демонстрирует превосходство над архитектурами ARM9 и ARM11. На рисунке 1.2 приведены нормализованные результаты оценки производительности AVR32, ARM11 и ARM9 (рассчитаны в условных единицах на одинаковой частоте, где большее значение на графике обозначает большую производительность).

Несомненно, разработчикам Atmel удалось создать оригинальное ядро нового поколения, способное конкурировать с лучшими микроконтроллерами для мультимедиа-приложений.

Среди оригинальных нововведений разработчики кристалла выделяют следующие параметры.

Инструкции записи в память/чтения из памяти могут оперировать с байтом (8-бит), полусловом (16-бит), словом (32-бит) и двойным словом (64-бит). Инструкции комбинированы с арифметическими операциями над указателями, что обеспечивает повышение производительности при доступе к таблицам, структурам и локальным переменным. При обмене байтами и полусловами поддерживаются знаковое и нулевое расширения данных.

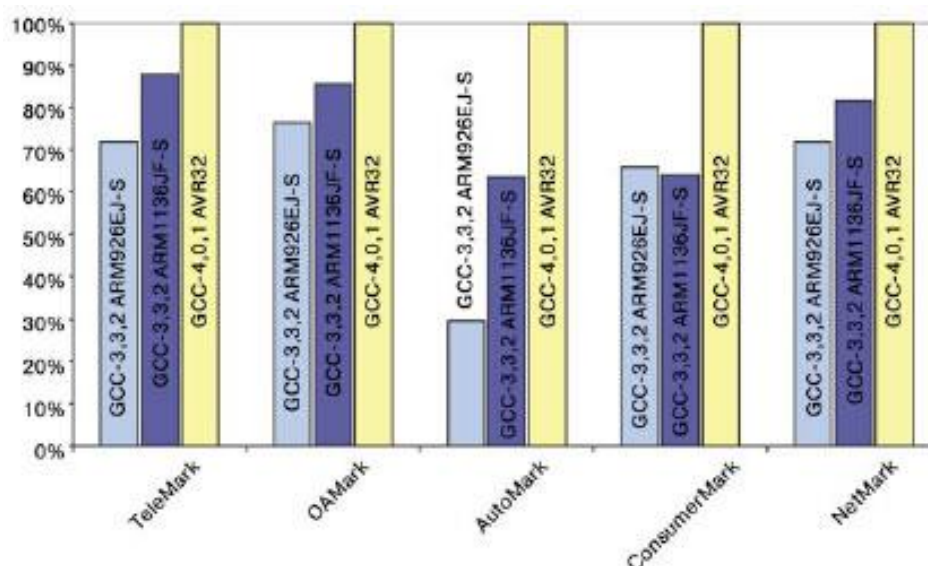


Рисунок 1.2 – Нормализованные результаты оценки производительности AVR32, ARM11 и ARM9

Среди популярных криптографических алгоритмов можно выделить алгоритмы блочного шифрования Blowfish и Triple-DES. Все эти алгоритмы используют особую адресацию в массиве данных, требующую длинной последовательности инструкций. Набор инструкций AVR32 оптимален для работы этих алгоритмов благодаря наличию инструкций чтения слова с извлечением указателя.

```

result = pointer0[offset0 >> 24] ^
pointer1[(offset1 >> 16) & 0xff] ^
pointer2[(offset2 >> 8) & 0xff] ^
pointer3[offset3 & 0xff];

```

(1.2)

Выражение (1.2) включает четыре операции доступа к памяти, которые извлекают по одному байту в 32-разрядном слове, дополняя их нулями, и добавляют к базовому указателю. На RISC-платформах такая операция будет выполнена за 14 инструкций, в то время как AVR32 потребует в два раза меньше операций за счет использования инструкций чтения с извлечением указателя, так как все четыре операции доступа к памяти будут выполнены за 4 такта со значениями смещения (offset0..3), хранящимися в одном регистре. Для повышения эффективности операций ввода-вывода могут быть использованы команды записи в память/чтения из памяти нескольких регистров (ldm, stm). Например, они могут использоваться для быстрого возврата из подпрограмм. В этом случае, при вызове подпрограммы, команда

stm будет сохранять в кэш-памяти инструкций адрес возврата, а команда ldm - извлекать его, одновременно записывая это значение в программный счетчик. Это избавит от необходимости выполнения команды возврата в конце подпрограммы.

Ядро AVR32 имеет 3 подконвейера (рисунок 1.3) и поддерживает одновременное исполнение несвязанных операций с изменением последовательности, предусмотренной программой. Например, 32-тактная команда деления поступает в блок умножения, а инструкции, следующие за ней, могут выполняться в ALU или блоке пересылки данных в течении этих 32-х тактов.

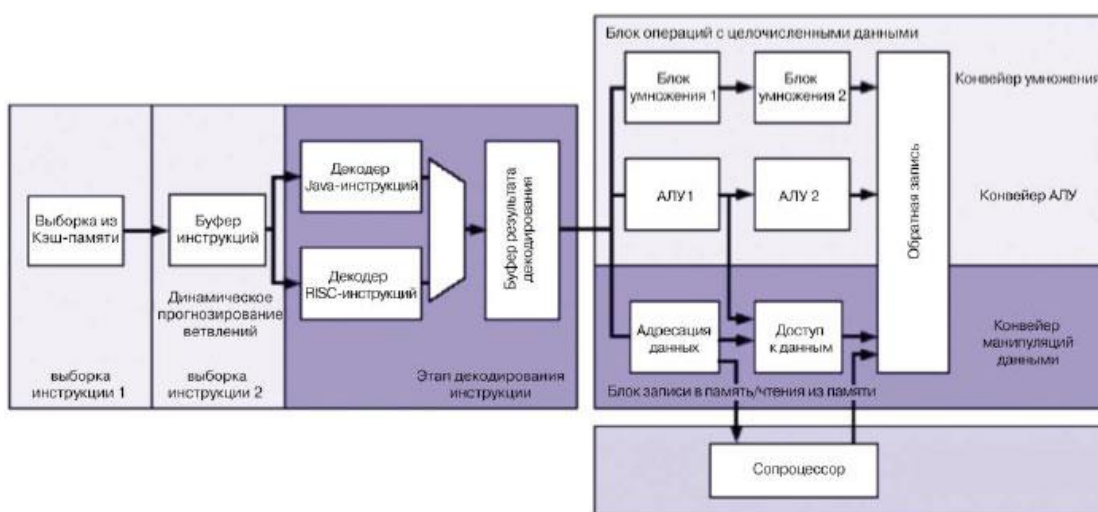


Рисунок 1.3 – Конвейер AVR32

Для экономии тактов также добавлена функция пересылки результата текущей операции на предыдущие ступени конвейера на случай, если следующая команда использует этот результат. Например, результат инструкции сложения (суммирования) в блоке ALU1 сразу пересылается на вход MUL1, ALU1 и блок адресации данных (DataAddress). Это позволяет сразу воспользоваться результатом сложения и не ждать до 3-х тактов, которые потребуются для программной пересылки данных на вход конвейера. Такая функция реализована для всех инструкций AVR32.

Операции с множественными данными (SIMD) позволяют увеличить скорость в 4 раза. Это относится к операциям с 8-разрядными значениями. Например, такая возможность нужна для описанного выше вычисления суммы абсолютной разности, где производится множество арифметических операций с 8-разрядными значениями пикселя.

Функция прогнозирования переходов (ветвлений) в AVR32 обеспечивает минимальные потери тактов при переходах с помощью команд `jmp`, `rcall` и `if` (в наборе Java-команд). Эта функция выполняется аппаратно и закрыта от программиста.

При разработке системы команд максимальное внимание было уделено плотности результирующего кода при выполнении типичных операций в мультимедиа-приложениях. Помимо этого, хорошие результаты дала совместная работа с фирмой IAR Systems, разработавшей C-компилятор с максимальной на сегодняшний день плотностью генерируемого кода для AVR32 (рис.3: чем меньше столбик, тем меньше размер кода).

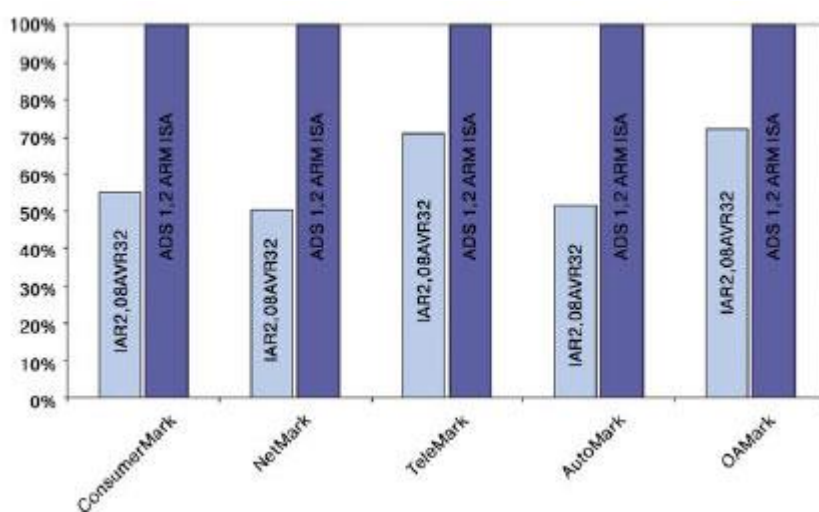


Рисунок 1.4 – Сравнительный размер кода IAR 2.06 и ADS 1.2

Введены дополнительные инструкции для операционных систем. Особое внимание было уделено обеспечению максимальной совместимости с ОС Linux: эта операционная система может полноценно использовать блок управления памятью (MMU) AVR32 и привилегированный/пользовательский режимы работы ядра AVR32. Для системных вызовов добавлена инструкция SCALL (System CALL), которая обращается непосредственно к подпрограммам ОС; для вызовов приложений добавлена функция ACALL (Application CALL), которая вызывает подпрограммы через 8-битные указатели в таблице переходов операционной системы.

Когда разработчики AVR32 говорят об ультранизком энергопотреблении этой платформы, они имеют ввиду, что AVR32 может выполнять те же задачи, что и другие платформы, но на меньшей тактовой частоте. По приближенным расчетам, можно сказать, что 80% работы по

энергосбережению решены за счет системы команд процессорного ядра AVR32, а 20% - на этапе реализации микроконтроллера в кремнии. Последнее достигается за счет размещения устройств хранения данных в непосредственной близости к ядру, так как это позволяет существенно сэкономить время на выполнение математически-насыщенных алгоритмов, требующих частого обращения к памяти, а, следовательно, позволяет снизить тактовую частоту.

Периферийные узлы микроконтроллера AT32AP7000

Первый микроконтроллер AT32AP7000 с ядром AVR32 выпущен с набором периферийных блоков, типичным для решения мультимедийных задач. Функциональная схема AT32AP7000 изображена на рисунке 1.5.

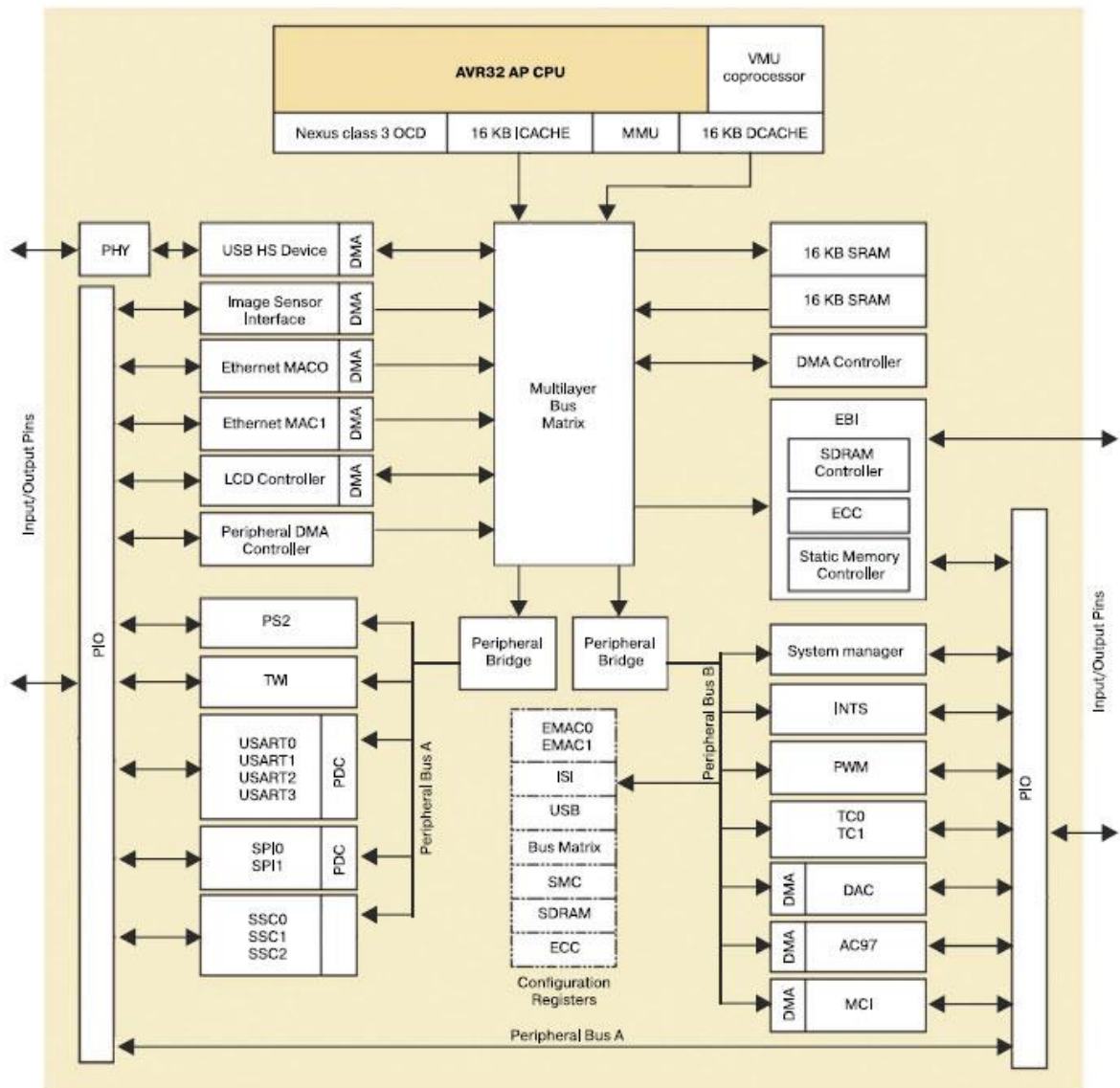


Рисунок 1.5 – Функциональная схема AP7000

Для сбалансированной работы ядра с быстродействующей и, наоборот, медленной периферией, все периферийные блоки были ранжированы по скорости. Самые быстрые узлы (ОЗУ, интерфейс внешней памяти, High-SpeedUSB и т.п.) были связаны с ядром многоуровневой матрицей шин (BusMatrix), которая обеспечивает до 16 параллельных и независимых связей между ведущими и ведомыми устройствами по шинам АНВ (AdvancedHostBus). Более медленные периферийные устройства (SPI, USART0..3, аудио-ЦАП) подключены к двум шинам APB (AdvancedPeripheralBus), работающим на разных частотах. Шина APB-A имеет рабочую тактовую частоту 37.5МГц, а шина APB-B - 75МГц. Согласование скорости передачи данных между шинами APB и BusMatrix производится мостами АНВ-APB (AdvancedHostBus - AdvancedPeripheralBus).

Для оптимизации энергопотребления микроконтроллера блок управления тактовой частотой (как системной, так и периферийной) имеет функцию динамического деления, которая позволяет регулировать тактовую частоту отдельно для каждой из 4-х вышеперечисленных подсистем. Такая функция полезна, когда интенсивные вычисления не производятся непрерывно во время работы микроконтроллера. Результат работы этой системы при декодировании фрагмента видеозаписи в формате MPEG-4 приведен на рисунке 1.6. Если принять тот факт, что ток, потребляемый микроконтроллером, пропорционален рабочей тактовой частоте, то можно сказать, что площадь на графике под кривой рабочей частоты пропорциональна энергии, потребленной за приведенный период.

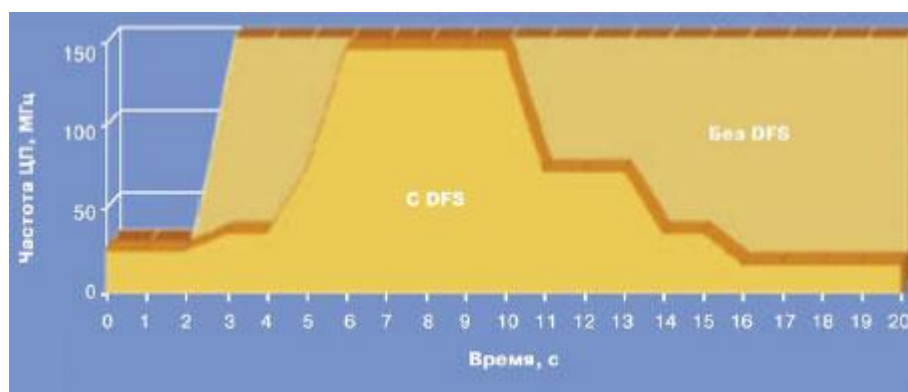


Рисунок 1.6 – Декодирование MPEG-4 с включением/выключением динамического деления частоты

Для освобождения вычислительных ресурсов ядра при операциях обмена потоковыми данными между периферией и памятью, имеется блок прямого доступа к памяти (DMA). Разработчики произвели расчет и получили, что если пересылка данных между памятью и быстродействующей периферией (USB High-Speed, Ethernet и др.) будет производиться под управлением ядра, то его свободных ресурсов для других задач просто не останется. Блок DMA в AP7000 поддерживает обмен данными не только со внутренней памятью, но и со внешней, подключенной по шине EBI (ExternalBusInterface). Что касается шины EBI, то по ней можно подключать различные виды памяти: SRAM, SDRAM, NAND Flash и CompactFlash.

При реализации типичных математических операций при работе с изображениями могут потребоваться операции векторной математики. Они нужны для таких распространенных задач как, например, цифровой фильтрации изображений (КИХ-фильтров), преобразования его цветовой модели (RGB \leftrightarrow YUV) или масштабирования. Так, например, после декодирования изображения в формате MPEG-4, представленного в YUV-формате, требуется его преобразование в цветовую модель RGB, требуемую для отображения на любом дисплее. Для таких задач в микроконтроллер AP7000 включен сопроцессор для операций векторного умножения (VMU, VectorMultiplicationUnit). При использовании этого блока, задача масштабирования изображения выполняется в 10 раз быстрее, чем с использованием ядра микроконтроллера.

Как было упомянуто выше, для ядра AVR32 фирма IAR Systems разработала компилятор языка C, стоимость которого на российском рынке составит порядка 5000\$. В качестве альтернативы этому коммерческому продукту предлагается бесплатный компилятор GCC, который по компактности кода проигрывает в 1.5-2 раза.

В состав набора включен дистрибутив ОС Linux. Единственным на сегодняшний день недостатком построения системы на базе этой ОС является необходимость использования компилятора GCC, так как продукт IAR Systems не может компилировать проекты на основе Linux.

Фактически, ЦПУ AVR32 выполняет в три раза больше действий за период синхронизации по сравнению с ближайшим конкурентом. Это означает, что требуемые в современных приложениях алгоритмы с интенсивными вычислениями могут выполняться при более низких тактовых частотах и, как следствие, при пониженном уровне потребления.

Современная пользовательская электроника, независимо от того карманная она или автомобильная, должна удовлетворять двум условиям

одновременно. Она должна выполнять сложные вычислительные алгоритмы цифровой обработки сигналов и при этом потреблять минимум энергии.

Большинство традиционных процессоров были разработаны в 1970-х и 1980-х годах до появления MP3-плееров, цифрового видео, GPS-навигации, распознавания голоса или, что более важно, миниатюрной портативной электронной продукции с батарейным питанием. Единственным способом достижения рабочих характеристик 21 века у процессоров 20 столетия - повышение частоты синхронизации. К сожалению, увеличение тактовой частоты процессора также приводит к повышению потребляемой мощности и выделения тепла.

Широкое распространение функций цифровой обработки привело к необходимости увеличения тактовой частоты процессоров, что в свою очередь увеличивает их потребляемую мощность до уровня, не позволяющего нашим карманным электронным устройствам работать без перезарядки сутки и более.

Архитектура 21-го столетия: высокая производительность с малой тактовой частотой (рисунок 1.7).



Рисунок 1.7 – Ядро архитектуры AVR32

Например, AVR32 может выполнять декодирование "quarter-VGAMPEG4" с кадровой частотой 30 Гц при тактировании частотой всего лишь 100 МГц. Для сравнения, сопоставимые архитектуры могут выполнить эту же операцию на тактовой частоте 266 МГц.

AVR32 - интеллектуальная собственность Atmel, которая будет использоваться для разработки микроконтроллеров общего назначения, а также специализированных и заказных интегральных схем.

Преимущества AVR32 AP:

- Пониженная потребляемая мощность и тепловыделение
 - Более низкая частота синхронизации ЦПУ
 - Улучшенное отношение производительность/МГц
- Простота проектирования
 - Стандартизированные средства для проектирования и встроенная отладочная система
 - Обширный набор инструкций
- Высококачественная разработка
 - Переходы без потерь тактов
 - SIMD-инструкции для цифровой обработки сигналов
 - 186 оригинальных RISC-инструкций
 - 7-ступенчатый конвейер с параллельным выполнением и измененным порядком
 - Исключительная плотность кода уменьшает потери кэш-памяти и увеличивает производительность

Таким образом, разработчики фирмы Atmel реализовали в ядре AVR32 функции, позволяющие повысить производительность при решении мультимедийных задач, и в выпускающийся на базе этого ядра микроконтроллер AT32UC30512 интегрировали специализированные периферийные узлы. На сегодняшний день ядро AVR32 поддерживается как коммерческими, так и бесплатными средствами разработки.

1.2 TCP/IP протокол

Стек протоколов TCP/IP (англ. TransmissionControlProtocol/InternetProtocol, Протокол управления передачей) — набор сетевых протоколов разных уровней модели сетевого взаимодействия DOD, используемых в сетях. Протоколы работают друг с другом в стеке (англ. stack, стопка) — это означает, что протокол, располагающийся на уровне выше, работает «поверх» нижнего, используя механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP.

Стек протоколов TCP/IP основан на модели сетевого взаимодействия DOD и включает в себя протоколы четырех уровней:

- прикладного (application);
- транспортного (transport);
- сетевого (internet);
- уровня доступа к среде (networkaccess).

Протоколы этих уровней полностью реализуют функциональные возможности модели OSI. На стеке протоколов TCP/IP построено всё взаимодействие пользователей в IP-сетях. Стек является независимым от физической среды передачи данных.

1.2.1 Уровни стека TCP/IP

Существуют разногласия в том, как вписать модель TCP/IP в модель OSI, поскольку уровни в этих моделях не совпадают.

К тому же, модель OSI не использует дополнительный уровень — «Internetworking» — между транспортным и сетевым уровнями. Примером спорного протокола может быть ARP или STP.

Вот как традиционно протоколы TCP/IP вписываются в модель OSI:

7	Прикладной	HTTP, SMTP, SNMP, FTP, Telnet, SSH, SCP, SMB, NFS, RTSP;
6	Представительский	XDR, AFP, TLS, SSL;
5	Сеансовый	ISO 8327 / CCITT X.225, RPC, NetBIOS, ASP;
4	Транспортный	TCP, UDP, SCTP, SPX, RTP, ATP, DCCP, GRE;
3	Сетевой	IP, PPP, ICMP, IGMP, CLNP, OSPF, RIP, IPX, DDP, ARP, RARP, BGP;
2	Канальный	Ethernet, Token ring, HDLC, X.25, Frame relay, ISDN, ATM, MPLS;
1	Физический	электрические провода, радиосвязь, волоконно-оптические провода, Wi-Fi;

Обычно в стеке TCP/IP верхние 3 уровня (прикладной, представительский и сеансовый) модели OSI объединяют в один — прикладной. Поскольку в таком стеке не предусматривается унифицированный протокол передачи данных, функции по определению типа данных передаются приложению. Упрощенно интерпретацию стека TCP/IP можно представить так:

5	Прикладной	HTTP, RTP, FTP, DNS;
---	------------	----------------------

4	Транспортный	TCP, UDP, SCTP, DCCP;
3	Сетевой	Для TCP/IP это IP (IP);
2	Канальный	Ethernet, IEEE 802.11 Wireless Ethernet, SLIP, Token Ring, ATM и MPLS;
1	Физический	физическая среда и принципы кодирования информации, T1, E1;

1.2.2 Физический уровень

Физический уровень описывает среду передачи данных (будь то коаксиальный кабель, витая пара, оптическое волокно или радиоканал), физические характеристики такой среды и принцип передачи данных (разделение каналов, модуляцию, амплитуду сигналов, частоту сигналов, способ синхронизации передачи, время ожидания ответа и максимальное расстояние).

1.2.3 Канальный уровень

Канальный уровень описывает, каким образом передаются пакеты данных через физический уровень, включая кодирование (то есть специальные последовательности бит, определяющих начало и конец пакета данных). Ethernet, например, в полях заголовка пакета содержит указание того, какой машине или машинам в сети предназначен этот пакет.

Примеры протоколов канального уровня — Ethernet, IEEE 802.11 WirelessEthernet, SLIP, TokenRing, ATM и MPLS.

PPP не совсем вписывается в такое определение, поэтому обычно описывается в виде пары протоколов HDLC/SDLC.

MPLS занимает промежуточное положение между канальным и сетевым уровнем и, строго говоря, его нельзя отнести ни к одному из них.

Канальный уровень иногда разделяют на 2 подуровня — LLC и MAC.

1.2.4 Сетевой уровень

Сетевой уровень изначально разработан для передачи данных из одной (под)сети в другую. Примерами такого протокола является X.25 и IPС в сети ARPANET.

С развитием концепции глобальной сети в уровень были внесены дополнительные возможности по передаче из любой сети в любую сеть, независимо от протоколов нижнего уровня, а также возможность запрашивать данные от удалённой стороны, например в протоколе ICMP

(используется для передачи диагностической информации IP-соединения) и IGMP (используется для управления multicast-потоками).

ICMP и IGMP расположены над IP и должны попасть на следующий — транспортный — уровень, но функционально являются протоколами сетевого уровня, и поэтому их невозможно вписать в модель OSI.

Пакеты сетевого протокола IP могут содержать код, указывающий, какой именно протокол следующего уровня нужно использовать, чтобы извлечь данные из пакета. Это число — уникальный IP-номер протокола. ICMP и IGMP имеют номера, соответственно, 1 и 2.

1.2.5 Транспортный уровень

Протоколы транспортного уровня могут решать проблему негарантированной доставки сообщений («дошло ли сообщение до адресата?»), а также гарантировать правильную последовательность прихода данных. В стеке TCP/IP транспортные протоколы определяют, для какого именно приложения предназначены эти данные.

Протоколы автоматической маршрутизации, логически представленные на этом уровне (поскольку работают поверх IP), на самом деле являются частью протоколов сетевого уровня; например OSPF (IP идентификатор 89).

TCP (IP идентификатор 6) — «гарантированный» транспортный механизм с предварительным установлением соединения, предоставляющий приложению надёжный поток данных, дающий уверенность в безошибочности получаемых данных, перезапрашивающий данные в случае потери и устраняющий дублирование данных. TCP позволяет регулировать нагрузку на сеть, а также уменьшать время ожидания данных при передаче на большие расстояния. Более того, TCP гарантирует, что полученные данные были отправлены точно в такой же последовательности. В этом его главное отличие от UDP.

UDP (IP идентификатор 17) протокол передачи датаграмм без установления соединения. Также его называют протоколом «ненадёжной» передачи, в смысле невозможности удостовериться в доставке сообщения адресату, а также возможного перемешивания пакетов. В приложениях, требующих гарантированной передачи данных, используется протокол TCP.

UDP обычно используется в таких приложениях, как потоковое видео и компьютерные игры, где допускается потеря пакетов, а повторный запрос затруднён или не оправдан, либо в приложениях вида запрос-ответ (например, запросы к DNS), где создание соединения занимает больше ресурсов, чем повторная отправка.

И TCP, и UDP используют для определения протокола верхнего уровня число, называемое портом.

1.2.6 Прикладной уровень

На прикладном уровне работает большинство сетевых приложений.

Эти программы имеют свои собственные протоколы обмена информацией, например, HTTP для WWW, FTP (передача файлов), SMTP (электронная почта), SSH (безопасное соединение с удалённой машиной), DNS (преобразование символьных имён в IP-адреса) и многие другие.

В массе своей эти протоколы работают поверх TCP или UDP и привязаны к определённым портам, например:

HTTP на TCP-порт 80 или 8080;

FTP на TCP-порт 20 (для передачи данных) и 21 (для управляющих команд);

SSH на TCP-порт 22;

запросы DNS на порт UDP (реже TCP) 53;

обновление маршрутов по протоколу RIP на UDP-порт 520.

Эти порты определены Агентством по выделению имен и уникальных параметров протоколов (IANA).

К этому уровню относятся: DHCP, Echo, Finger, Gopher, HTTP, HTTPS, IMAP, IMAPS, IRC, NNTP, NTP, POP3, POPS, QOTD, RTSP, SNMP, SSH, Telnet, XDMCP.

1.3 Отладочная плата EVK1105

EVK1105 – отладочная плата 32-разрядных контроллеров фирмы Atmel AT32UC3A0512. Полная совместимость отладочной платы с фирменным эмулятором JTAGICE mkII существенно облегчает процесс разработки и отладки программ пользователя. В комплекте с платой идут компилятор языка Си AVR32 GNU Toolchain (бесплатный) и среда разработки IAR EWAVR32 (с ограниченным Си компилятором).

Отладочный комплект предназначен для тестирования Wi-Fi модулей компании H&D Wireless SPB104 и SPB105. Модуль SPB104 подключается к SD-слоту платы, модуль SPB105 - к 10-контактному разъёму через переходник SPB400. Плата содержит графический дисплей с разрешением 320x240 пикселей, сенсорный интерфейс управления (5 сенсорных кнопок), память SDRAM 256Мбит, Flash-память 64Мбит, аудиовыход, слот для карт памяти SD/SDIO/MMC, интерфейсы Ethernet, USB HighSpeed, SPI и TWI, отладочные интерфейсы JTAG и NEXUS.

1.3.1 Отличительные особенности

- 1) установленный процессор AT32UC3A0512: 512 КБайт Flash, 64 КБайт SRAM, SPI, I2C, 4×UART, USB Full-Speed + OTG, Ethernet;
- 2) SDRAM 256 Мбит MT48LC16M16A2;
- 3) Ethernet интерфейс на DP83848I;
- 4) внешняя SPI Flash-память 64 Мбит AT45DB642D;
- 5) USB High Speed интерфейс и USB-UART для загрузки;
- 6) TFT LCD 2,4” ET0240B0DMU с сенсорным экраном;
- 7) аудио интерфейс: стерео кодек TLV32AIC23B, микрофон, стерео усилитель мощности на 138 мВт TPA6130A2;
- 8) сенсорный интерфейс управления на QT1081 5 сенсорных кнопок;
- 9) слота для SD/SDIO/MMC карт памяти;
- 10) отладочные интерфейсы: JTAG и NEXUS;
- 11) все свободные выводы микроконтроллера доступны на контактных площадках.

1.3.2 Основные параметры

Т а б л и ц а 1.1 – Основные параметры EVK1105

Параметр	Значение
Интерфейс подключения	USB / JTAG / NEXU
Ядро базового компонента	AVR32
Разрядность, бит	32
Базовый компонент	AT32UC3A0512
Вспомогательный компонент	DP83848I AT45DB642D TPA6130A2RT TLV320AIC23B
Целевое напряжение, В	3.3 / 5.0
Напряжение питания, В	5.0
Источник питания	внешний / USB

Области применения: портативные устройства, персональные мультимедиа устройства, пользовательские терминалы.

2 ОПЕРАЦИОННАЯ СИСТЕМА ДЛЯ СИСТЕМЫ АППАРАТНОГО УПРАВЛЕНИЯ

2.1 Ядро операционной системы

При реализации систем управления часто возникает задача одновременного управления несколькими объектами (двигателями, преобразователями и т.д.), обработки информации от датчиков (тока, напряжения, положения, температуры), опроса клавиатуры, управления выводом информации на дисплей ит.д. При этом опрос клавиатуры и вывод информации на дисплей не так критичны к скорости, как, на пример, программа формирования ШИМ(для двигателя) и обработка информации от датчиков. Следовательно, работа с клавиатурой и дисплеем имеет более низкий приоритет по отношению к работе с датчиками и программами формирования ШИМ. Все это усложняет приложение и накладывает высокие требования на производительность микроконтроллера. Операционные системы реального времени обеспечивают многозадачность системы управления при помощи сервисов, предоставляемых ядром, что заметно упрощает разработку систем управления и обеспечивает рациональное использование ресурсов процессора.

Ядро — это часть многозадачной системы, отвечающая за управление задачами (то есть за управление временем центрального процессора) и связью между задачами. Основная функция ядра—переключение контекста. Использование ядра реального времени позволяет упростить проект, путем разбиения приложения на множество задач, управляемых ядром. Ядро позволяет лучше использовать центральный процессор, предоставляя необходимые сервисы, такие как управление, семафорами, почтовые ящики, очереди, временные задержки, и т.д.

Однако МК как платформа для выполнения ОС имеет существенные отличия от современных компьютеров. Прежде всего, МК работает в режиме реального времени, то есть время реакции микроконтроллерного устройства па внешнее событие должно быть строго меньше заданной величины и должно быть сопоставимо со скоростью протекания внешних процессов. Типичный пример: время реакции на срабатывание датчика давления в промышленной установке должно быть не более 5 мс, иначе произойдет авария. Таким образом, ОС для МК — это операционная система реального времени (ОСРВ). К ОСРВ предъявляются жесткие временные требования в

отличие от распространенных ОС общего назначения (Windows, UNIX-подобные и др.).

Во-вторых, микроконтроллер, по сути, это однокристалльный компьютер с сильно ограниченными аппаратными ресурсами, хотя диапазон выпускаемых МК по производительности и объемам памяти очень широк. Тем не менее все МК имеют существенные аппаратные ограничения, что предъявляет специфические требования к ОСПВ для МК. Их основные особенности:

- Низкая производительность;
- Малый объем ОЗУ и ПЗУ;
- Отсутствие блока управления памятью (Memorymanagementunit, MMU), используемого большинством современных ОС, например Windows и UNIX-подобными;
- Отсутствие аппаратных средств поддержки многозадачности (например, средств быстрого переключения контекста).

В третьих, микроконтроллер сам, по себе предназначен для выполнения низкоуровневых задач, будь то опрос состояния кнопок, передача команды по I²C-интерфейсу или включение обмотки электромотора. Программа для МК, как правило, обращается к периферии напрямую, программист имеет полный контроль над аппаратной частью, нет необходимости в посредниках между аппаратурой и прикладной программой. Может показаться, что операционная система для МК вообще не нужна, что побитно программу можно написать и без ОС. Но есть один нюанс: микроконтроллер редко используют только для опроса состояния кнопок, только для передачи команды по I²C-интерфейсу или только для включения обмотки электромотора. Гораздо чаще из МК пытаются «выжать» все, на что он способен.

2.2 Библиотеки под аппаратную часть

FreeRTOS — это многозадачная, мульти платформенная, бесплатная операционная система жесткого реального времени с открытым исходным кодом. FreeRTOS была разработана компанией RealTimeEngineersLtd, специально для встраиваемых систем. На текущий момент (версия FreeRTOS 6.5.0) ОС официально поддерживает 23 архитектуры и 57 платформ (в подавляющем большинстве— микроконтроллеры). В течение 2010 и 2011 годов произошло более 125000 загрузок FreeRTOS с официального сайта, что делает ее одной из самых популярных ОСПВ на сегодня. Большая часть кода FreeRTOS написана на языке Си, ассемблерные вставки минимального

объема применяются лишь там, где невозможно применить Си из-за специфики конкретной аппаратной платформы.

Существуют так называемые официально поддерживаемые аппаратные платформы — официальные порты и неофициальные, которые поставляются «как есть» и не поддерживаются напрямую. Кроме того, для одного и того же порта могут поддерживаться несколько средств разработки. Основные характеристики FreeRTOS:

1) планировщик FreeRTOS поддерживает три типа многозадачности:

- вытесняющую;
- кооперативную;
- гибридную.

2) размер ядра FreeRTOS составляет всего 4-9 кбайт, в зависимости от типа платформы и настроек ядра;

3) FreeRTOS написана на языке Си;

4) поддерживает задачи (tasks) и сопрограммы (co-routines).

Сопрограммы специально созданы для МК с малым объемом ОЗУ;

5) богатые возможности трассировки;

6) возможность отслеживать факт переполнения стека;

7) нет программных ограничений на количество одновременно выполняемых задач;

8) нет программных ограничений на количество приоритетов задач;

9) нет ограничений в использовании приоритетов: нескольким задачам может быть назначен одинаковый приоритет;

10) развитые средства синхронизации «задача - задача» и «задача – прерывание» :

- очереди;
- двоичные семафоры;
- счетные семафоры;
- рекурсивные семафоры;
- мьютексы.

11) мьютексы с наследованием приоритета. 12 Поддержка модуля защиты памяти(Memoryprotectionunit, MPU) в процессорахCortex-M3;

12) поставляется с отлаженными примерами проектов для каждого порта и для каждой среды разработки.

Работа планировщика FreeRTOSв режиме вытесняющей многозадачности имеет много общего с алгоритмом переключения потоков в современных ОС общего назначения. Вытесняющая многозадачность пред-

полагает, что любая выполняющаяся задача с низким приоритетом прерывается готовой к выполнению задачей с более высоким приоритетом. Как только высокоприоритетная задача выполнила свои действия, она завершает свою работу или переходит в состояние ожидания, и управление снова получает задача с низким приоритетом. Переключение между задачами осуществляется через равные кванты времени работы планировщика, то есть высокоприоритетная задача, как только она стала готова к выполнению, ожидает окончания текущего кванта, после чего управление получает планировщик, который передаст управление высокоприоритетной задаче.

Таким образом, время реакции FreeRTOS на внешние события в режиме вытесняющей многозадач почти — не больше одного кванта времени планировщика, который можно задавать в настройках по умолчанию он равен 1 мс.

Если готовы к выполнению несколько задач с одинаковым приоритетом, то в таком случае планировщик выделяет каждой из них по одному кванту времени, по истечении которого управление получает следующая задача с таким же приоритетом, и так далее по кругу.

Кооперативная многозадачность отличается от вытесняющей тем, что планировщик самостоятельно не может прервать выполнение текущей задачи, даже если появилась готовая к выполнению задача с более высоким приоритетом. Каждая задача должна самостоятельно передать управление планировщику. Таким образом, высокоприоритетная задача будет ожидать, пока низкоприоритетная завершит свою работу и отдаст управление планировщику. Время реакции системы на внешнее событие становится неопределенным и зависит от того, как долго текущая задача будет выполняться до передачи управления.

Вытесняющая и кооперативная концепции многозадачности объединяются вместе в гибридной многозадачности, когда вызов планировщика происходит каждый квант времени, но, в отличие от вытесняющей многозадачности, программист имеет возможность сделать это принудительно в теле задачи. Особенно полезен этот режим, когда необходимо сократить время реакции системы на прерывание. Допустим, в текущий момент выполняется низкоприоритетная задача, а высокоприоритетная ожидает наступления некоторого прерывания. Далее происходит прерывание, но по окончании работы обработчика прерывания выполнение возвращается к текущей низкоприоритетной задаче, а высокоприоритетная задача ожидает, пока закончится текущий квант времени. Однако если после выполнения обработчика прерывания передать

управление планировщику, то он передаст управление высокоприоритетной задаче, что позволяет значительно сократить время реакции системы на прерывание, связанное с внешним событием.

3 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ АППАРАТНОГО УПРАВЛЕНИЯ

3.1 Принцип работы системы

Элементы системы связываются следующим образом:

- 1) контроль и управление аппаратурой происходит через модули управления;
- 2) модули управления связываются в сеть;
- 3) доступ клиента к модулям управления осуществляется через сеть.

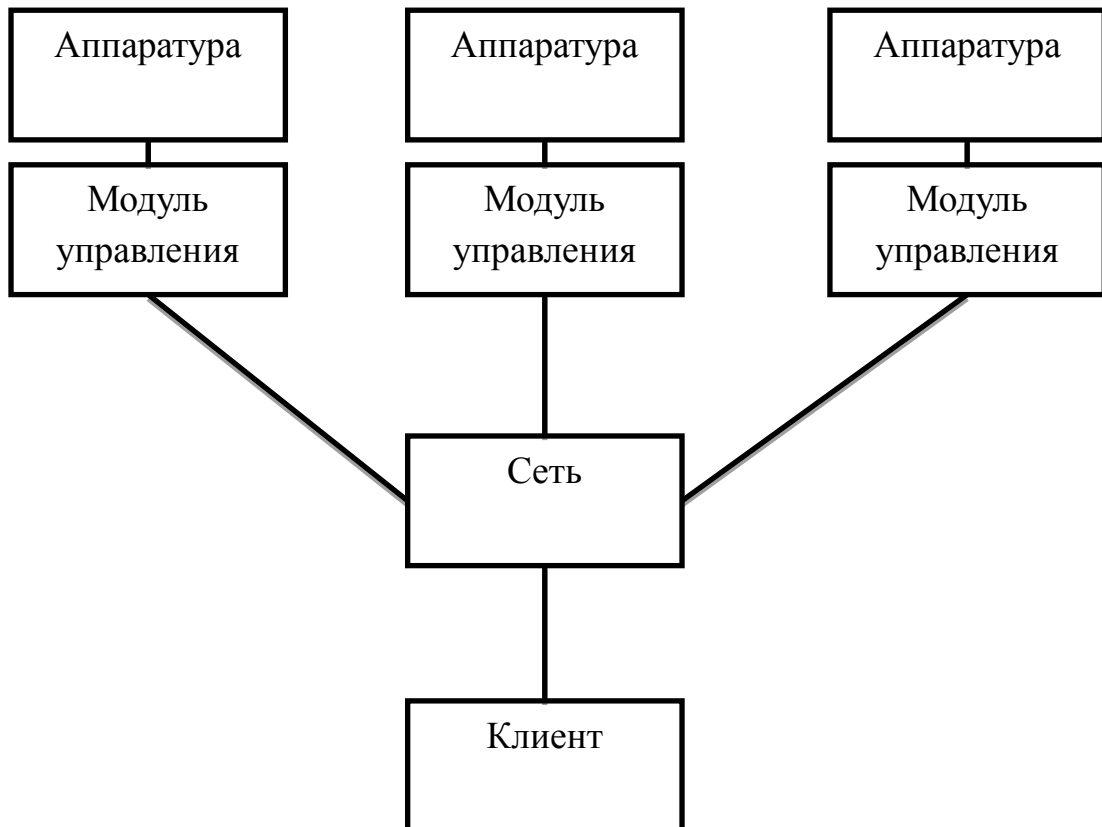


Рисунок 3.1 – Схема связей в системе

Каждый модуль управления берет на себя функции и сервера и системы управления. В итоге мы получаем, что каждый элемент работает

самостоятельно, причем в случае необходимости он связывается с другими модулями.

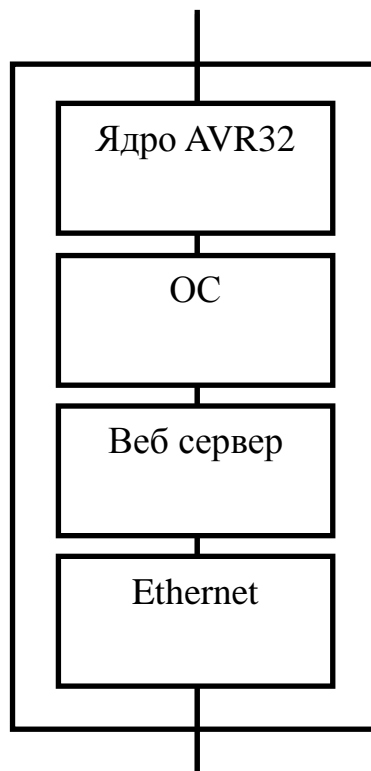


Рисунок 3.2 – Блок схема работы блока управления

В модуле управления контролем и управлением аппаратурой занимается непосредственно ядро микроконтроллера под управлением ОС, при этом ОС берет необходимую информацию об управлении и отправляет данные о состоянии Веб серверу, который по протоколу HTTP принимает и передает данные через Ethernet в сеть. Аналогично для связи с Internet только здесь потребуется для соединения устройств в сеть маршрутизаторы с возможностью подключения Internet.

3.2 Библиотека управляющей программы

3.2.1 AVRStudio

AVRStudio — интегрированная среда разработки (IDE) для разработки 8-битных AVR приложений от компании Atmel, работающая в операционных системах Windows NT/2000/XP/Vista/7. AVRStudio содержит ассемблер и симулятор, позволяющий отследить выполнение программы. Текущая версия поддерживает все выпускаемые на сегодняшний день контроллеры AVR и

средства разработки. AVRStudio содержит в себе менеджер проектов, редактор исходного кода, инструменты виртуальной симуляции и внутрисхемной отладки, позволяет писать программы на ассемблере или на C/C++. В силу перечисленных достоинств AVRStudio может считаться удобной средой разработки приложений для микроконтроллеров AVR.Содержание

Ранее существовал и фирменный ассемблер под Windows (wavrasm.exe) от Atmel, который совмещал ассемблер и редактор, подобно тому, как это делается в «больших» языках программирования.его можно извлечь из первых версий AVRStudio, но он довольно примитивный и неудобный. Затем, видимо, в корпорации решили его не развивать, ограничившись AVRStudio. Скачать AVRStudio можно совершенно бесплатно с сайта Atmel. Там же доступны и старые версии (к сожалению, последние версии пакета стали довольно объемными — более 70 Мбайт — так что на этом можно сэкономить), но только обратите внимание, что, чем старше версия, тем меньше ассортимент поддерживаемых контроллеров (так, версия 3 поддерживает только Classic).

AVRStudio 4 - профессиональная интегрированная среда разработки (IntegratedDevelopmentEnvironment - IDE), предназначенная для написания и отладки прикладных программ для AVR микропроцессоров в среде Windows 9x/NT/2000.AVRStudio 4 содержит ассемблер и симулятор. Также IDE поддерживает такие средства разработки для AVR как: ICE50, ICE40, JTAGICE, ICE200, STK500/501/502 и AVRISP. В ближайшие месяцы будет расширен список поддерживаемых AVRStudio 4 микроконтроллеров и средств разработки. Обо всех обновлениях можно будет узнать на интернет сайте.

AVRStudio поддерживает COFF как формат выходных данных для символьной отладки. Другие программные средства третьих фирм также могут быть сконфигурированы для работы с AVRStudio.

Ключевое окно в AVRStudio это окно исходного текста программы. Когда объектный файл открыт, автоматически создается окно исходного текста программ. В окне отображается код, который выполняется в отладочном окружении (эмуляторе или программном симуляторе) а текстовый маркер всегда находится на строке, которая будет выполнена в следующем цикле.

Пользователь может выполнять программу полностью в пошаговом режиме, трассируя блоки функций, или выполняя программу до места, где стоит курсор. В дополнение можно определять неограниченное число точек

останова, каждая из которых может быть включена или выключена. Точки останова сохраняются между сессиями работы.

В окне исходного текста программы выводится информация о процессе выполнения программы. В дополнение, AVRStudio имеет много других окон, которые позволяют управлять и отображать информацию о любом элементе микроконтроллера.

Watchwindow: Окно показывает значения определенных символов. В этом окне пользователь может просматривать значения и адреса переменных.

Tracewindow: Окно показывает хронологию программы, выполняемой в настоящее время.

Registerwindow: Окно показывает содержимое регистров. Регистры можно изменять во время остановки программы.

Memorywindows: Окна показывают содержимое памяти программ, данных, портов ввода/вывода и энергонезависимого ПЗУ. Память можно просматривать в HEX, двоичном или десятичном форматах. Содержимое памяти можно изменять во время остановки программы.

I/Owindow: Показывает содержимое различных регистров ввода/вывода:

- EEPROM;
- I/O порты;
- Таймеры;
- и т.д..

Messagewindow: Окно показывает сообщения от AVRStudio.

Processorwindow: В окне отображается важная информация о ресурсах микроконтроллера, включая программный счетчик, указатель стека, регистр статуса и счетчик цикла. Эти параметры могут модифицироваться во время остановки программы.

Настройки рабочего окружения сохраняются при выходе. При первом запуске требуется настроить окна для управления и вывода необходимой информации. Во время следующей загрузки настройки автоматически восстанавливаются.

В AVRStudio включена поддержка отладочных средств фирмы Atmel:

Внутрисхемный эмулятор AtmelICEPRO

Внутрисхемный эмулятор AtmelMegaICE

Внутрисхемный эмулятор AtmelAVRICE

Внутрисхемный эмулятор AtmelICE200

Внутрисхемный эмулятор AtmelAsicICE

Внутрисхемный эмулятор AtmelICE10

Внутрисхемный эмулятор AtmelICE30

С AVRStudio также совместимы любые программаторы и отладочные средства, которые поддерживают микроконтроллеры фирмы Atmel.

3.3 Веб сервер

3.3.1 DHCP

DHCP (англ. DynamicHostConfigurationProtocol — протокол динамической конфигурации узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP, и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве крупных (и не очень) сетей TCP/IP.

DHCP является расширением протокола BOOTP, использовавшегося ранее для обеспечения бездисковых рабочих станций IP-адресами при их загрузке. DHCP сохраняет обратную совместимость с BOOTP.

Стандарт протокола DHCP был принят в октябре 1993 года. Действующая версия протокола (март 1997 года) описана в RFC 2131. Новая версия DHCP, предназначенная для использования в среде IPv6, носит название DHCPv6 и определена в RFC 3315 (июль 2003 года).

Протокол DHCP предоставляет три способа распределения IP-адресов:

- ручное распределение. При этом сетевой администратор сопоставляет аппаратному адресу (для Ethernet сетей это MAC-адрес) каждого клиентского компьютера определённый IP-адрес. Фактически, данный способ распределения адресов отличается от ручной настройки каждого компьютера лишь тем, что сведения об адресах хранятся централизованно (на сервере DHCP), и потому их проще изменять при необходимости;

- автоматическое распределение. При данном способе каждому компьютеру на постоянное использование выделяется произвольный свободный IP-адрес из определённого администратором диапазона;

- динамическое распределение. Этот способ аналогичен автоматическому распределению, за исключением того, что адрес выдаётся

компьютеру не на постоянное пользование, а на определённый срок. Это называется арендой адреса. По истечении срока аренды IP-адрес вновь считается свободным, и клиент обязан запросить новый (он, впрочем, может оказаться тем же самым). Кроме того, клиент сам может отказаться от полученного адреса.

Некоторые реализации службы DHCP способны автоматически обновлять записи DNS, соответствующие клиентским компьютерам, при выделении им новых адресов. Это производится при помощи протокола обновления DNS, описанного в RFC 2136.

Помимо IP-адреса, DHCP также может сообщать клиенту дополнительные параметры, необходимые для нормальной работы в сети. Эти параметры называются опциями DHCP. Список стандартных опций можно найти в RFC 2132.

Некоторыми из наиболее часто используемых опций являются:

- IP-адрес маршрутизатора по умолчанию;
- маска подсети;
- адреса серверов DNS;
- имя домена DNS.

Протокол DHCP является клиент-серверным, то есть в его работе участвуют клиент DHCP и сервер DHCP. Передача данных производится при помощи протокола UDP, при этом сервер принимает сообщения от клиентов на порт 67 и отправляет сообщения клиентам на порт 68.

3.3.2 Структура сообщений DHCP

Все сообщения протокола DHCP разбиваются на поля, каждое из которых содержит определённую информацию. Все поля, кроме последнего (поля опций DHCP), имеют фиксированную длину.

Т а б л и ц а 3.1 – Структура сообщений DHCP

Поле	Описание	Длина (байт)
op	Тип сообщения. Может принимать два значения: BOOTREQUEST (1, запрос от клиента к серверу) и BOOTREPLY (2, ответ от сервера к клиенту).	1
htype	Тип аппаратного адреса. Допустимые значения этого поля определены в RFC «AssignedNumbers». Например, для MAC-адреса Ethernet10 Мбит/с это поле принимает значение 1.	1
hlen	Длина аппаратного адреса в байтах. Для MAC-адреса Ethernet — 6.	1
hops	Количество промежуточных маршрутизаторов (так называемых агентов ретрансляции DHCP), через которые прошло сообщение.	1

	Клиент устанавливает это поле в 0.	
Xid	Уникальный идентификатор транзакции, генерируемый клиентом в начале процесса получения адреса.	4
Secs	Время в секундах с момента начала процесса получения адреса. Может не использоваться (в этом случае оно устанавливается в 0).	2
Flags	Поле для флагов — специальных параметров протокола DHCP.	2
Ciaddr	IP-адрес клиента. Заполняется только в том случае, если клиент уже имеет собственный IP-адрес и способен отвечать на запросы ARP (это возможно, если клиент выполняет процедуру обновления адреса по истечении срока аренды).	4
Yiaddr	Новый IP-адрес клиента, предложенный сервером.	4
Siaddr	IP-адрес сервера. Возвращается в предложении DHCP (см. ниже).	4
Giaddr	IP-адрес агента ретрансляции, если таковой участвовал в процессе доставки сообщения DHCP до сервера.	4
Chaddr	Аппаратный адрес (обычно MAC-адрес) клиента.	16
Sname	Необязательное имя сервера в виде нуль-терминированной строки.	64
File	Необязательное имя файла на сервере, используемое бездисковыми рабочими станциями при удалённой загрузке. Как и sname , представлено в виде нуль-терминированной строки.	128
Options	Поле <i>опций DHCP</i> . Здесь указываются различные дополнительные параметры конфигурации. В начале этого поля указываются четыре особых байта со значениями 99, 130, 83, 99 («волшебные числа»), позволяющие серверу определить наличие этого поля.	переменная

Рассмотрим пример процесса получения IP-адреса клиентом от сервера DHCP. Предположим, клиент ещё не имеет собственного IP-адреса, но ему известен его предыдущий адрес — 192.168.1.100. Процесс состоит из четырёх этапов.

Вначале клиент выполняет широковещательный запрос по всей физической сети с целью обнаружить доступные DHCP-серверы. Он отправляет сообщение типа DHCPDISCOVER, при этом в качестве IP-адреса источника указывается 0.0.0.0 (так как компьютер ещё не имеет собственного IP-адреса), а в качестве адреса назначения — широковещательный адрес 255.255.255.255.

Клиент заполняет несколько полей сообщения начальными значениями:

В поле `xid` помещается уникальный идентификатор транзакции, который позволяет отличать данный процесс получения IP-адреса от других, протекающих в то же время.

В поле `chaddr` помещается аппаратный адрес (MAC-адрес) клиента.

В поле опций указывается последний известный клиенту IP-адрес. В данном примере это 192.168.1.100. Это необязательно и может быть проигнорировано сервером.

Сообщение DHCPDISCOVER может быть распространено за пределы локальной физической сети при помощи специально настроенных агентов ретрансляции DHCP, перенаправляющих поступающие от клиентов сообщения DHCP серверам в других подсетях.

Получив сообщение от клиента, сервер определяет требуемую конфигурацию клиента в соответствии с указанными сетевым администратором настройками. В данном случае DHCP-сервер согласен с запрошенным клиентом адресом 192.168.1.100. Сервер отправляет ему ответ (DHCPOFFER), в котором предлагает конфигурацию. Предлагаемый клиенту IP-адрес указывается в поле yiaddr. Прочие параметры (такие, как адреса маршрутизаторов и DNS-серверов) указываются в виде опций в соответствующем поле.

Это сообщение DHCP-сервер отправляет хосту, пославшему DHCPDISCOVER, на его MAC, при определенных обстоятельствах сообщение может распространяться как широковещательная рассылка. Клиент может получить несколько различных предложений DHCP от разных серверов; из них он должен выбрать то, которое его «устраивает».

Выбрав одну из конфигураций, предложенных DHCP-серверами, клиент отправляет запрос DHCP (DHCPREQUEST). Он рассылается широковещательно; при этом к опциям, указанным клиентом в сообщении DHCPDISCOVER, добавляется специальная опция — идентификатор сервера — указывающая адрес DHCP-сервера, выбранного клиентом (в данном случае — 192.168.1.1).

Наконец, сервер подтверждает запрос и направляет это подтверждение (DHCPACK) клиенту. После этого клиент должен настроить свой сетевой интерфейс, используя предоставленные опции.

Помимо сообщений, необходимых для первоначального получения IP-адреса клиентом, DHCP предусматривает несколько дополнительных сообщений для выполнения иных задач.

Если после, получения подтверждения (DHCPACK) от сервера клиент обнаруживает, что указанный сервером адрес уже используется в сети, он рассылает широковещательное сообщение отказа DHCP (DHCPDECLINE), после чего процедура получения IP-адреса повторяется. Использование IP-адреса другим клиентом можно обнаружить, выполнив запрос ARP.

Если по каким-то причинам сервер не может предоставить клиенту запрошенный IP-адрес, или если аренда адреса удаляется администратором,

сервер рассылает широковещательное сообщение отмены DHCP (DHCPNAK). При получении такого сообщения соответствующий клиент должен повторить процедуру получения адреса.

Клиент может явным образом прекратить аренду IP-адреса. Для этого он отправляет сообщение освобождения DHCP (DHCPRELEASE) тому серверу, который предоставил ему адрес в аренду. В отличие от других сообщений DHCP, DHCPRELEASE не рассылается широковещательно.

Сообщение информации DHCP (DHCPINFORM) предназначено для определения дополнительных параметров TCP/IP (например, адреса маршрутизатора по умолчанию, DNS-серверов и т. п.) теми клиентами, которым не нужен динамический IP-адрес (то есть адрес которых настроен вручную). Серверы отвечают на такой запрос сообщением подтверждения (DHCPACK) без выделения IP-адреса.

Компания Microsoft впервые включила сервер DHCP в поставку серверной версии Windows NT 3.5, выпущенной в 1994 году. Начиная с Windows 2000 Server реализация DHCP-сервера от Microsoft позволяет динамически обновлять записи DNS, что используется в Active Directory.

Internet Systems Consortium выпустил первую версию ISC DHCP Server (для Unix-подобных систем) 6 декабря 1997 года. 22 июня 1999 года вышла версия 2.0, более точно соответствующая стандарту.

Компания Cisco включила сервер DHCP в Cisco IOS 12.0 в феврале 1999 года. Sun добавила DHCP-сервер в Solaris 8 в июле 2001 года.

В настоящее время существуют реализации сервера DHCP для ОС Windows в виде отдельных программ, в том числе открытых [1], позволяющих выполнять роль сервера DHCP компьютерам под управлением несерверных версий данной ОС.

3.4 Создание веб приложения с использованием библиотеки программы управления

Сборки работы примера FreeRTOS с помощью GCC были сделаны следующие шаги:

- по инструкции собрантулчейн под Cortex-M3;
- из примеров от Atmel(AT91 SoftwarePackage) были скопированы:
 - скрипт компоновщика (линковщика, линкера) flash.lds ;
 - файл начальной инициализации board_cstartup_gnu.c ;
- написан Makefile;
- отредактирован файл начальной инициализации board_cstartup_gnu.c.

В Makefile были заданы опции линкера и компилятора:

- опции компилятора (CFLAGS) -mthumb -mcpu=cortex-m3 -O2 -Wall -mlong-calls;
- опции компоновщика (LDFLAGS) -mthumb -mcpu=cortex-m3 -O2 –nostartfiles.

При сборке проекта используются следующие файлы:

- файлы для работы с аппаратной частью: draw_hx8347.c font.c hx8347.c lcd.c lcd_hx8347.c nvic.cpio.cpmc.cusart.c;

- файлы тестов:

BlockQ.cblocktim.ccomtest.cflash.cGenQTest.cinteger.cPollQ.cQPeek.crecmutex.csemtest.c;

- файлыFreeRTOS: heap_2.c list.cqueue.cserial.ctasks.c;

- файлыдляинициализациисистемы:

board_lowlevel.cboard_memories.cexceptions.csyscalls.cboard_cstartup_gnu.c;

- основные файлы проекта: main.cParTest.c.

В файлеboard_cstartup_gnu.сизменена только таблица, которая задает обработчики прерываний/исключений, в нее добавлены следующие обработчики из FreeRTOS:

- vPortSVCHandler — обработчик программного прерывания, генерируемого командой svc, используется для запуска первого потока;

- xPortPendSVHandler — обработчик системного запроса, используется для переключения между потоками;

- xPortSysTickHandler — обработчик прерываний таймера, используется для вычисления системного времени;

- vSerialISR — обработчик прерываний последовательного порта.

При попытке собрать проект линкер выдал несколько ошибок:

```
undefinedreference to _sbrk
undefinedreference to _write
undefined reference to _close
undefined reference to _fstat
undefined reference to _isatty
undefined reference to _lseek
undefined reference to _read
```

Это связано с тем, что newlib был сконфигурирован с ключом --disable-newlib-supplied-syscalls

Для решения этой проблемы в проект был включен файл syscalls.c с необходимыми функциями-"заглушками".

4 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ МОДУЛЯ АППАРАТНОГО УПРАВЛЕНИЯ ЧЕРЕЗ ПРОТОКОЛ TCP/IP

4.1 Цель проекта

Целью данного проекта является разработка специального программного обеспечения для систем аппаратного управления. С последующей возможностью её установки в специальное оборудование.

4.2 Тема проекта

Проект посвящен разработке программного обеспечения для модуля аппаратного управления. Который при помощи разработанного программного обеспечения позволит осуществлять удаленный контроль над любыми производственными процессами через обычный интернет и без установки специального программного обеспечения на клиентской машине.

4.3 Введение

Модулем аппаратного управления является, какое либо электронное устройство, берущее на себя управление и контроль над любыми устройствами механического и электронного характера (станки, двигатели, электросети, вычислительные машины, измерительная аппаратура и т.д.). Суть подобных модулей заключена в дистанционном управлении над любыми устройствами, для которых необходим подобный контроль.

Стек протоколов TCP/IP (англ. TransmissionControlProtocol/InternetProtocol) — набор сетевых протоколов разных уровней модели сетевого взаимодействия DOD, используемых в сетях. Протоколы работают друг с другом в стеке (англ. stack, стопка) — это означает, что протокол, располагающийся на уровне выше, работает «поверх» нижнего, используя механизмы инкапсуляции. Например, протокол TCP работает поверх протокола IP. На сегодняшний день это основной протокол работы как локальных, так и глобальных сетей.

На сегодняшний день, в Казахстане в отличие от стран Европы отсутствуют системы промышленных сетей, в которых все оборудование связано друг с другом и с компьютерами операторов этого оборудования, посредством обычных Ethernet сетей. Подобная технология дает возможность

выпускать более качественную продукцию за счет тотального и высокоэффективного контроля над техпроцессом.

Подобные системы так же могут иметь и стратегическое значение, используя их в системе электронной охраны границ, или объектах на которых присутствие человека опасно для жизни.

Благодаря отсутствию на рынке подобного продукта, было принято решение о разработке программы под существующие без программные модули. В данном разделе мы проведем рассмотрение экономической составляющей реализации данного проекта, отражающей временные, трудовые и финансовые затраты на проект.

4.4 Расчёт стоимости работы по проектированию и разработке программы для модуля аппаратного управления через протокол ТСР/IP

Разработка проекта состоит из 8 этапов, общей продолжительностью 13 недель. Этапы и сроки проведения работ по разработке ПП представлены в таблице 4.1.

Т а б л и ц а 4.1 – Этапы реализации проекта по разработке ПП

Перечень работ		Сотрудник	Недели от начала работ														
			1	2	3	4	5	6	7	8	9	10	11	12	13		
1	1. Постановка технического задания на разработку ПП	Руководитель проекта	■														
	2. Содержание работы	Руководитель проекта, программист разработчик	■	■													
	3. Формирование ядра ОС	Программист разработчик		■													
2	4. Компиляция ОС	Программист разработчик			■												
3	5. Установка дополнительных библиотек	Программист разработчик			■												
	6. Настройка связей ОС сТСР/IPпротоколом	Программист разработчик				■											
4	7. Разработка процесса подключения к Ethernetсети	Программист разработчик					■										
	8. Разработка модуля авторизации	Программист разработчик						■	■								

5	9. Разработка программного модуля	Программист разработчик															
	10. Создание программного модуля	Программист разработчик															

Продолжение таблицы 4.1

5	11. Тестирование модуля с целью выявления ошибок	Программист разработчик															
	12. Разработка алгоритма для установки необходимого ПО на различных модулях	Программист разработчик															
6	13. Создание руководства по эксплуатации ПО	Программист разработчик															
	14. Установка и настройка веб сервера для связи с модулями	Программист разработчик															
	15. Подготовка раздела «экономика»	Консультант по части «экономика», программист разработчик															
	16. Подготовка раздела «БЖД»	Консультант по части «БЖД», программист разработчик															
7	17. Оформление НИР	Программист разработчик															
8	18. Проверка и сдача отчёта по проделанной работе	Руководитель проекта, программист разработчик															

Отбор персонала определяется по обязанностям штатного расписания, либо решением заведующего, соответствующего сектора.

В работе задействованы следующие трудовые ресурсы:

- руководитель – постановка задачи, разработка основных алгоритмов;
- инженер-разработчик – разработка алгоритмов и программирование, сопровождение и инструктаж персонала сопровождения;
- консультанты по экономической части и по части ОБЖД.

Количество сотрудников по каждому пункту и представлено в таблице 4.2

Т а б л и ц а 4.2 – Количество задействованных в проекте работников и их заработная плата

Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Руководитель	1	60000
Консультант по части “Экономика”	1	50000
Консультант по части “Безопасность жизни и труда человека”	1	45000
Инженер-разработчик	1	80000
Итого	6	235000

Затраты на разработку данного программного комплекса определяется по формуле:

$$C_{\text{пр}} = \text{ФОТ} + O_c + A + \text{Э} + \text{Н}, \quad (4.1)$$

- где ФОТ – фонд оплаты труда;
- O_c – социальный налог;
- A – амортизационные отчисления;
- Э – затраты на электроэнергию;
- $C_{\text{пр}}$ – прочие расходы;
- Н – накладные расходы;

Необходимый фонд оплаты труда (ФОТ) рассчитывается по формуле:

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (4.2)$$

где $Z_{\text{осн}}$ – основная заработная плата; $Z_{\text{доп}}$ – дополнительная заработная плата.

Так как участники проекта задействованы в различные промежутки времени для расчета затрат необходимо сначала произвести расчет почасовой и подневной заработной платы каждого участника проекта.

Труд сотрудников института, задействованных в работе, оплачивается согласно положению о заработной плате АИЭС, труд программиста-разработчика принят условно, на договорной основе в размере 80000 тенге.

Количество и заработная плата работников, задействованных в разработке, указано в таблице 4.1.

Заработная плата каждого работника за один рабочий день рассчитывается путем деления месячного оклада количество рабочих дней за прошедший месяц (это 24 дня – шестидневная рабочая неделя) по формуле:

$$D = \frac{ЗПм}{Др} \quad (4.4)$$

Для руководителя:

$$D = \frac{60000}{24} = 2500 \text{ тенге/день};$$

Для консультанта по части “Экономика”:

$$D = \frac{50000}{24} = 2083,33 \text{ тенге/день};$$

Для консультанта по части “Безопасность жизни и труда человека”:

$$D = \frac{45000}{24} = 1875 \text{ тенге/день};$$

Для инженера-разработчика:

$$D = \frac{80000}{24} = 3333,33 \text{ тенге/день};$$

Заработная плата работника за один час рассчитывается делением заработной платы работника за день на количество часов рабочего дня (7 часов) по формуле:

$$H = \frac{D}{Чр} \quad (4.5)$$

Для руководителя:

$$H = \frac{2500}{7} = 357,14 \text{ тенге/час};$$

Для консультанта по части “Экономика”:

$$H = \frac{2083,33}{7} = 297,62 \text{ тенге/ час};$$

Для консультанта по части “Безопасность жизни и труда человека”:

$$H = \frac{1875}{7} = 267,86 \text{ тенге/ час};$$

Для инженера-разработчика:

$$H = \frac{3333,33}{7} = 476,19 \text{ тенге/ час};$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n \cdot z \cdot K}, \quad (4.6)$$

где T – трудоёмкость этапа, норма-час;

q_n – количество исполнителей по этапу;

z – продолжительность рабочего дня, $z = 7$ часов;

K – коэффициент выполнения норм времени, $K = 1,1$;

Полученную величину t_n округляем в большую сторону до целых дней.

Руководитель проекта:

- постановка технического задания на разработку ПП

$$t_1 = \frac{36}{1 \cdot 7 \cdot 1,1} \approx 5; \text{дн}$$

- Содержание работы

$$t_2 = \frac{21,6}{1 \cdot 7 \cdot 1,1} \approx 3; \text{дн}$$

- Проверка и сдача отчёта по проделанной работе

$$t_{18} = \frac{24}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

Консультант по части «экономика»:

- Подготовка раздела «экономика»

$$t_{15} = \frac{16,8}{1 \cdot 7 \cdot 1,1} \approx 3; \text{дн}$$

Консультант по части «безопасность жизнедеятельности»:

- Подготовка раздела «БЖД»

$$t_{16} = \frac{32}{1 \cdot 8 \cdot 1,1} \approx 3; \text{дн}$$

Программист разработчик:

- 1) Содержание работы

$$t_2 = \frac{24}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

- 2) Формирование ядра ОС

$$t_3 = \frac{32}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

- 3) Компиляция ОС

$$t_4 = \frac{50,4}{1 \cdot 7 \cdot 1,1} \approx 7; \text{дн}$$

- 4) Установка дополнительных библиотек

$$t_5 = \frac{21,6}{1 \cdot 7 \cdot 1,1} \approx 3; \text{дн}$$

- 5) Настройка связей ОС с TCP/IP протоколом

$$t_6 = \frac{72}{1 \cdot 7 \cdot 1,1} \approx 10; \text{дн}$$

- 6) Разработка процесса подключения к Ethernet сети

$$t_7 = \frac{50.4}{1 \cdot 7 \cdot 1,1} \approx 7; \text{дн}$$

7) Разработка модуля авторизации

$$t_8 = \frac{168}{1 \cdot 7 \cdot 1,1} \approx 22; \text{дн}$$

8) Разработка программного модуля

$$t_9 = \frac{144}{1 \cdot 7 \cdot 1,1} \approx 19; \text{дн}$$

9) Создание программного модуля

$$t_{10} = \frac{180}{1 \cdot 7 \cdot 1,1} \approx 24; \text{дн}$$

10) Тестирование модуля с целью выявления ошибок

$$t_{11} = \frac{24}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

11) Разработка алгоритма для установки необходимого ПО на различных модулях

$$t_{12} = \frac{3.6}{1 \cdot 7 \cdot 1,1} \approx 1; \text{дн}$$

12) Создание руководства по эксплуатации ПО

$$t_{13} = \frac{24}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

13) Установка и настройка веб сервера для связи с модулями

$$t_{14} = \frac{30}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

14) Подготовка Раздела «экономика»

$$t_{15} = \frac{26.4}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

15) Подготовка раздела «Безопасность Жизнедеятельности»

$$t_{16} = \frac{26.4}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

16) Оформление НИР

$$t_{17} = \frac{156}{1 \cdot 7 \cdot 1,1} \approx 21; \text{дн}$$

17) Проверка и сдача отчёта по работе

$$t_{18} = \frac{24}{1 \cdot 7 \cdot 1,1} \approx 4; \text{дн}$$

Сводные результаты расчета затрат по основной заработной плате представлены в таблице 4.3.

Т а б л и ц а 4.3 – Затраты по основной заработной плате персонала, задействованного в проекте

	Наименование этапов и содержание работ	Исполнитель	Трудоёмкость		цикл, дни	Заработная плата за час работы, тенге	Сумма заработной платы, тенге
			Нормы часы	% от общей трудоёмкости			
1	2	3	4	5	6	7	8
1	Постановка задачи	Руководитель	36	3	5	357,14	12857,04
2	Разработка содержания работы	Руководитель	21,6	1,8	3	357,14	7714,224
		Инженер-разработчик	24	2	4	476,19	11428,56
3	Формирование ядра ОС	Инженер-разработчик	30	2,5	4	476,19	14285,7
4	Компиляция ОС	Инженер-разработчик	50,4	4,2	7	476,19	23999,976
5	Установка дополнительных библиотек	Инженер-разработчик	21,6	1,8	3	476,19	10285,704
6	Настройка связей ОС сТСР/IPпротоколом	Инженер-разработчик	72	6	10	476,19	34285,68
7	Разработка процесса подключения к Ethernetсети	Инженер-разработчик	50,4	4,2	7	476,19	23999,976
8	Разработка модуля авторизации	Инженер-разработчик	168	14	22	476,19	79999,92
9	Разработка программного модуля	Инженер-разработчик	144	12	19	476,19	68571,36
10	Создание программного модуля	Инженер-разработчик	180	15	24	476,19	85714,2
11	Тестирование модуля с целью выявления ошибок	Инженер-разработчик	24	2	4	476,19	11428,56
12	Разработка алгоритма для установки необходимого ПО на различных модулях	Инженер-разработчик	3,6	0,3	1	476,19	1714,284
13	Создание руководства по эксплуатации ПО	Инженер-разработчик	24	2	4	476,19	11428,56

<i>продолжение - таблицы 4.3</i>							
14	Установка и настройка веб сервера для связи с модулями	Инженер-разработчик	30	2,5	4	476.19	14285,7
15	Подготовка раздела “Экономика”	Консультант по экономической части	16,8	1,4	3	297.62	5000,016
		инженер-разработчик	26,4	2,2	4	476.19	12571,416
16	Подготовка раздела “Безопасность жизнедеятельности”	Консультант по части БЖД	16,8	1,4	3	267.86	4500,048
		Инженер-разработчик	26,4	2,2	4	476.19	12571,416
17	Оформление НИР	Инженер-разработчик	156	13	21	476.19	74285,64
18	Проверка и сдача отчёта по проделанной работе	Инженер-разработчик	42	1,5	6	476.19	8571,42
19		Руководитель	24	2	4	357,14	8571,36
		ИТОГО:	1200	100	169		448785,08

Таким образом, затраты по основной заработной плате всех участников проекта составляет 448785,08 тенге.

Дополнительная заработная плата рассчитывается по формуле:

$$Z_{доп} = Z_{осн} \cdot 1,1 \quad (4.4)$$

И составляет

$$Z_{доп} = 493663,5 \text{ тенге};$$

Затраты на оплату труда согласно формуле 4.3 составляют:

$$Z_{осн} = 448785,08 \text{ тенге};$$

Отчисления по социальному налогу рассчитываются по формуле:

$$C_{соц} = Z_{осн} \cdot 0,1 \quad (4.5)$$

где ПО — пенсионные отчисления и рассчитываются по формуле:

$$П_{отч} = Z_{осн} \cdot 0,08 \quad (4.7)$$

$$П_{отч} = 35902,81 \text{ тенге};$$

Таким образом, отчисления по социальному налогу составляют:

$$C_{соц} = 44878,51 \text{ тенге};$$

Затраты на амортизацию по используемому оборудованию рассчитывается по формуле. Результаты расчетов представлены в таблице 4,4:

$$A = \frac{H \cdot C_{тип}}{i \cdot 100}, \quad (4.8)$$

где H_A - норма амортизации;
 $C_{\text{ПЕР}}$ – первоначальная стоимость оборудования;
 N - количество дней на выполнение работ;
 n - количество дней в рабочем месяце.

По существующему положению, в настоящее время, норма амортизации H_A на оборудование связи составляет 40% от стоимости всего оборудования.

$$A = \frac{C_{\text{ПЕР}} \cdot H_A \cdot N}{n}$$

$$A = \frac{25 \cdot 2500}{1000}$$

Т а б л и ц а 4.4 – Затраты на амортизацию оборудования

1	2	3	4	5
Наименование оборудования	Кол-во	Цена за единиц, тенге	Норма амортизации, %	Сумма амортизации, тенге
Персональный компьютер IntelCore 2 Duo 2,7GHz/ MBintel975PE, GF FX 256MB, 2048MB DDR/ SeagateBarracuda IV 160 Gb/CD-RWAsus 52x24x16»	1	65500	40	72996
Лазерный принтер HP1120	1	12500	25	52,1
Итого	-	-	-	73048,1

Поскольку в производстве используется электрооборудование, то необходимо посчитать затраты на электроэнергию.

Затраты электроэнергии на основное и дополнительно оборудование рассчитываются по формуле:

$$Z_{\text{ЭЭ}} = \frac{W \cdot T \cdot S \cdot K_{\text{ИСП}}}{1000} \quad (4.9)$$

Затраты на электроэнергию в соответствии с формулой

$$Z_{\text{ЭЭ}} = \frac{W \cdot T \cdot S \cdot K_{\text{ИСП}}}{1000} \quad (4.10)$$

где W – потребляемая мощность, Вт;
 T – время работы ($T = 8760$ ч/год);
 S – тариф (1 кВтч = 12 тг, для юридических лиц);
 $K_{\text{ИСП}}$ – коэффициент использования ($K_{\text{ИСП}} = 0,9$).

~~320791974~~ тенге;

~~320791974~~ тенге;

Сумма затрат на электроэнергию основного оборудования составляют:

~~320791974~~ тенге;

Затраты на дополнительные нужды рассчитывают по формуле:

$$Z_{\text{доп}} = 53000 \quad (4.11)$$

Составляет

~~320791974~~ тенге;

Итого затраты на электроэнергию составляют:

~~320791974~~ тенге;

Результаты расчета затрат на электроэнергию представлены в таблице 4.5

Т а б л и ц а 4.5 – Расходы на электроэнергию

Наименование приборов	W, кВт	Число рабочих дней	$K_{\text{им}}$	Время работы прибора, час	W, кВт/ч	Стоимость, тенге
3 компьютер с монитором	0,9	168	0,9	1176	952,6	7791
Принтер	0,5	6	0,9	42	18,9	389,6
Итого W:	-	-	-	-	971,5	8180,6

Прочие расходы:

– расходы на Интернет – 5000 тенге;

Накладные расходы на разработку программного продукта принимаем в размере 75% от суммы затрат рассчитываются по формуле:

$$H = (\text{ФОТ} + O_c + A + \text{Э} + C_{\text{пр}}) \cdot 75\% \quad (4.12)$$

И составляет:

$$H = (493663,588 + 48872,7 + 73048,1 + 8180,7 + 5000) \cdot 75\% = 471573,816 \text{ тг.};$$

Суммарные затраты по разработке базы данных согласно формуле (4.2) по всем статьям составляют:

$$C = 493663,588 + 48872,7 + 73048,1 + 8180,7 + 5000 + 471573,82 = 1100338,91 \text{ тг.};$$

Сводные результаты расчетов сметы затрат на разработку управляющей программы и её структура представлены в таблице 4.6 и на рис 4.1.

Т а б л и ц а 4.6 – Стоимость разработки программы по всем статьям затрат

Наименование статей затрат	Сумма, тенге
ФОТ	493663,59
Отчисления на социальные нужды	48872,7
Амортизация	73048,1
Затраты на электроэнергию	8180,7
Прочие расходы	5000
Накладные расходы	471573,82
Итого:	1100338,91

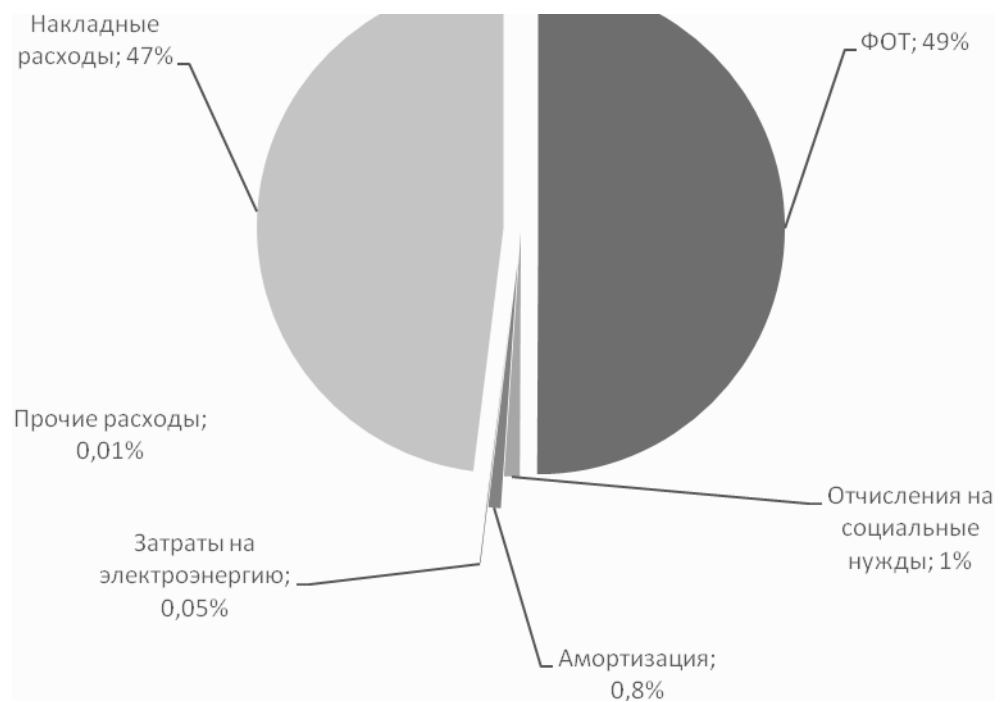


Рисунок 4.1 - Структура затрат разработке программы и её структуры

4.5 Цена интеллектуального труда

Цена реализации программного продукта складывается из стоимости и чистого дохода:

$$C = C + \Pi, \quad (4.13)$$

где C – стоимость продукта;

Π – чистый доход;

При определении первоначальной цены задаёмся желаемым уровнем рентабельности (40%) для реализации программных продуктов:

$$C_{\Pi} = C * \left(1 + \frac{P}{100}\right), \quad (4.14)$$

где P – рентабельность (40%);

$$C_{\Pi} = 1100338,91 * \left(1 + \frac{40}{100}\right) = 1540474,47$$

Цена реализации готовой продукции рассчитывается по формуле:

$$C_{\text{гп}} = C_{\Pi} + \text{НДС}, \quad (4.15)$$

Поскольку на сегодняшний день размер НДС в РК составляет 12%, следовательно:

$$\text{НДС} = C_{\Pi} \cdot 12\%, \quad (4.16)$$

$$\text{НДС} = 1540447,47 * 12\% = 184856,936; \text{ тенге};$$

$$C = 1540447,47 + 184856,936 = 1725304,41 \text{ тенг}$$

Окончательная цена в условиях рыночных отношений будет зависеть и от спроса на интеллектуальный труд.

Вывод

Себестоимость разработки программы составила 1100,338,91 тг. Наибольшую долю в общей сумме расходов занимают затраты на оплату труда - 493 664 или 49 %. Цена реализации программного продукта составила 1 725304,41 тг. Программа может свободно меняться, дополняться и улучшаться под изменяющимися требованиями. Данное программное обеспечение не имеет аналогов на рынке Республики Казахстан. Программа позволит выбирать на рынке любое необходимое оборудование и конфигурировать его под собственные нужды.

5 Безопасность жизнедеятельности

Данный дипломный проект посвящен разработке системы удаленного контроля над различной аппаратурой и механизмов посредством использования ТСР/ІР протокола. Суть подобной системы заключается в том, что вместо того чтобы заменить непосредственный контроль, на удаленный контроль над различными приборами и агрегатами. Особенность данного проекта заключена в том, что удаленный контроль осуществляется посредством использования обычных локальных сетей и сети Internet. Т.е. на производстве для осуществления контроля над техпроцессом вместо специализированного оборудования, используется специфическое ПО, и универсальные устройства, соединенные в сеть и управляемые удаленно через Internetили внутри локальной сети используя обычный интернет браузер.Для этих целей была подобрана серия 32-х битных микроконтроллеров фирмы AtmelAT32UC3. Данные контроллеры отвечают необходимым требованиям для реализации цели проекта. Для упрощения разработки ПО, была использована на базе контроллера AT32UC30512 отладочная платаEVK1105,содержащая все необходимые аппаратные компоненты для разработки и тестирования ПО. Благодаря этому была исчерпана необходимость разработки первичного тестового модуля.



Рисунок 5.1 – Отладочная плата EVK1105

5.1 Характеристика разрабатываемого прибора и помещений по степени опасности поражения человека электрическим током.

В данном дипломном проекте разрабатывается управляющая программа для модуля аппаратного управления с поддержкой протокола TCP/IP. Для этого используется отладочное средство EVK1105. Работа с EVK1105 производится через компьютер, на плате расположены необходимые разъемы для подключения к компьютеру USB, Ethernet. Плата питается непосредственно от компьютера через разъем USB.

Следовательно, главным орудием труда является компьютер, на котором установлен необходимая программа для разработки управляющей программы.

Разработка ведется в помещении (рис. 5.2), которое находится на втором этаже двухэтажного здания, имеет две двери, окно длиной 2 и высотой 1.5 метра; параметры помещения: Длина 5, ширина 4, высота 3 метра; имеется искусственное освещение в виде 4-х потолочных ламп, в которые вставлены по 4 люминесцентных лампы по 23 Вт; внутренняя отделка стен светлая; в отделе ведется постоянная работа с компьютерами, по зрительным условиям работы относится к V разряду, т. к. наименьший объект различения от 1 до 5 мм. Ближайшее здание находится в 20-ти метрах, одноэтажное и не препятствует проникновению света через окна.

В отделе два рабочих места, имеется орг. техника: два компьютера типа Notebook, принтер и плата EVK1105.

Технические характеристики устройств:

- 1) Asus U50Vg Intel Core2 Duo Processor P8700, 4GB SDRAM , LCD 15.6" HD (1366x768) LED, NVIDIA® GeForce® G 105M , with 512MB VRAM, 320GB, 5400rpm, Integrated 802.11a/b/g/n;
- 2) Acer Aspire 5573ZWXMii Intel Core2 Duo T2080, 1GB SDRAM, LCD 14.1", GMA 945 128mb, 320 GB 5400rpm, Integrated 802.11a/b/g/n;
- 3) принтер HP LaserJet 1100;
- 4) плата EVK1105.

При работе с компьютером человек подвержен воздействию ряду опасных и вредных производственных факторов. Для данного помещения это:

- 1) опасный уровень напряжения в электрической цепи, замыкание которой может пройти через тело человека;

2) длительное пребывание в одном и том же положении и повторение одних и тех же движений приводит к синдрому длительных статических нагрузок (СДСН);

3) умственное перенапряжение, которое обусловлено характером решаемых задач, приводит к синдрому длительных психологических нагрузок (СДПН);

4) большой объем перерабатываемой информации приводит к значительным нагрузкам на органы зрения;

5) монотонность труда.

Наиболее опасным является поражение электрическим током, поскольку может привести к гибели человека. При работе с орг. техникой человек может пострадать от удара током за счет того, что может произойти короткое замыкание, техника выйдет из строя и может навредить пользователю. А при работе с отладочной платой можно произвести короткое замыкание, которое может вывести из строя Блок питания компьютера. И как следствие может привести к короткому замыканию.

Кроме того, при работе с отладочной платой необходимо соблюдать определенные правила техники безопасности, как при работе с любым иным стендовым или макетным оборудованием, во избежание поражения током.

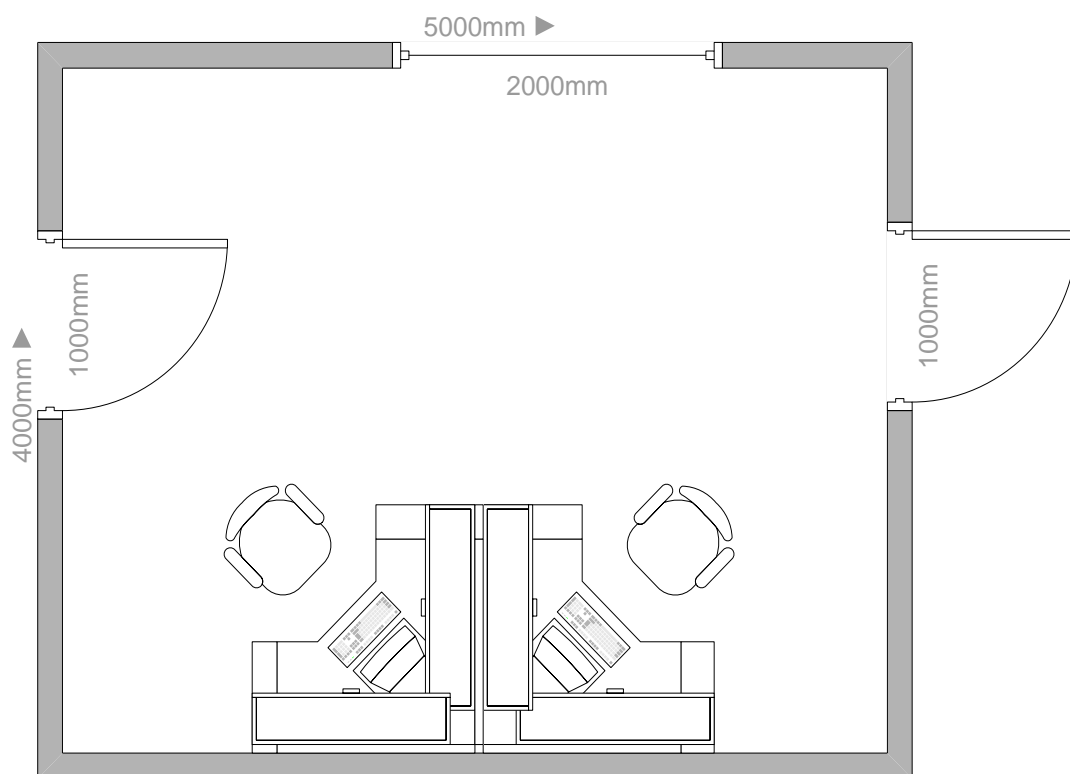


Рисунок 5.2 - План помещения, в котором ведутся работы

Поэтому нам необходимо решить следующие задачи:

- 1) описать правила ТБ при работе с отладочной платой;
- 2) произвести расчет электробезопасности.

Отладочная плата EVK1105 представляет собой электрический прибор без внешнего защитного корпуса экрана. Это связано с тем, что к элементам прибора должен быть прямой доступ. Прибор имеет четыре силиконовых ножки обеспечивающих изоляцию с поверхностью. Плату покрывает слой уретанового лака обеспечивающего изоляцию окружающей средой.

Порт USB подает питание 5В 300мА и имеет схемотехническую развязку с внешним питанием. Т.е. в случае неверного подключения или наличия повреждений во внешнем питании устройство не включится до устранения проблем с внешним питанием. В случае если питание не стабилизировано или имеет превышение, микросхема U14 стабилизирует питание и на выходе будут необходимые номинальные значения для работы элементов схемы, скачки напряжения сглаживают конденсаторы C129, C130. Диодный мост со стабилитроном D11 обеспечивает защиту сигнальных линий. В случае короткого замыкания на плате, может произойти выход из строя питающих схем, но при этом контакт с внешним питанием разрывается. Тем самым защищая внешний источник питания.

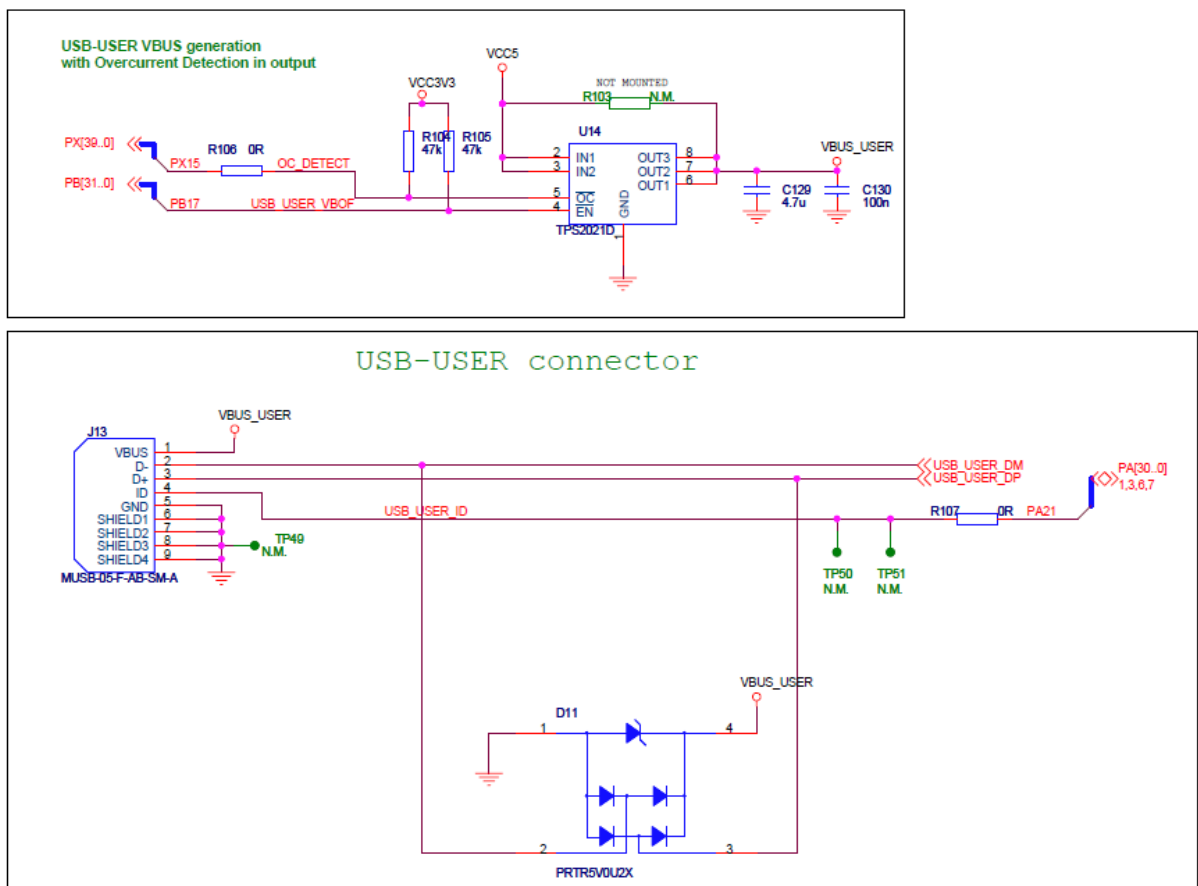


Рисунок 5.3 – Развязка питания в порта USB

Порт Ethernet имеет трансформаторную развязку питания и сигнальной шины, тем самым обеспечивая защиту самой схемы от внешних факторов, также обеспечивается сохранность подключенного к плате устройства при коротком замыкании или иной поломке платы.

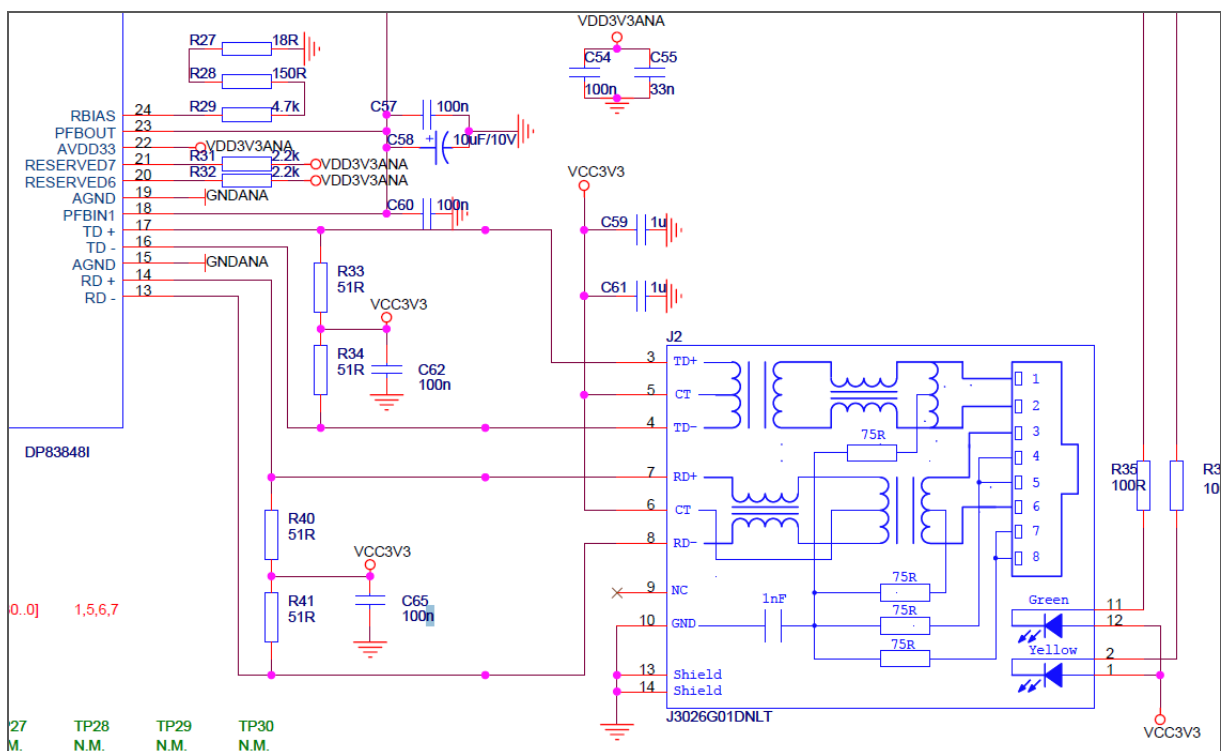


Рисунок 5.4 – Трансформаторная развязка питания и сигнальной шины порта Ethernet

При проведении работ с платой было сделано:

- 1) установлена плата на ровной изолированной поверхности;
- 2) проведены все подключения и установки при отключенном питании;
- 3) проведенные измерения сняты с использованием выделенных внешних разъемов;
- 4) работа, связанная с включённым устройством, (контакт с органами управления) произведена с использованием одной руки (во избежание прохода тока через путь рука-рука);
- 5) перед выполнением работ был проверен необходимый уровень изоляции пользователей;
- 6) при работе был использован антистатический браслет;
- 7) изоляционное покрытие платы не повреждено.

5.1.1 Расчет заземления аппаратуры

Согласно требованиям ПУЭ в электроустановках с напряжением до 1000 В сопротивление заземляющего устройства не должно превышать 4 Ом.

Определим требуемое сопротивление заземляющего устройства R_H .

а) для электроустановок напряжением до 1000 В $R_H \leq 40$ Ом.

б) для электроустановок напряжением 1000 В и выше с большими заземляющими токами ($I_3 < 500$ А) $R_H \leq 0,5$ Ом;

в) если заземляющее устройство используется одновременно для установок напряжением до 1000 В и выше, то $R_H = 125/I_3$, не более R_H , указанно для установок до 1000 В (4 или 10 Ом);

г) для электроустановок напряжением выше 1000 В с малыми токами на землю ($I_3 > 500$ А) $R_H \leq 250 \text{ Ом} / I_3$, но не более 10 Ом.

В пунктах в) и г) I_3 - ток на землю.

Из выше перечисленного следует принимать $R_H \leq 4$ Ом, так как наша аппаратура ниже 1000 В.

В соответствии с климатическим признакам по таблице 5.4 определим климатическую зону. Для наших условий приемлема IV - климатическая зона.

Затем определим расчетное значение отдельное сопротивление грунта.

$$R_v = R_{уд} * \psi_v; \quad (5.1)$$

$$R_g = R_{уд} * \psi_g; \quad (5.2)$$

где R_v , R_g - расчетное и удельное сопротивление грунта, соответственно для вертикального и горизонтального электродов;

ψ_v , ψ_g - коэффициенты сезонности по таблице 5.5, учитывающие возможное повышение сопротивления грунта в течении года и состоянии земли во время измерений соответственно для вертикального и горизонтального электродов.

Т а б л и ц а 5.4 - Признаки климатических зон для определения коэффициента сезонности ψ .

Характеристика климатической зоны.	Климатические зоны				
	1	2	3	4	5
редняя многолетняя низшая температура (Январь) °С	от 20 до 15	от 14 до 10	от 10 до 0	от 0 до 5	
Средняя многолетняя высшая температура (Июль) °С	от 16 до 18	от 18 до 22	от 22 до 24	от 24 до 26	

Продолжение таблицы 5.4

Среднегодовое количество осадков, см	=40	= 50	= 50	30-50
Продолжительность заморзания вод, дни	170-190	= 150	= 100	0

Т а б л и ц а 5.5 - Коэффициент сезонности ψ для однородной земли (вертикальный электрод длиной 3м, горизонтальный электрод длиной 10м)

Климатическая зона	Состояние земли во время измерений ее сопротивления и при влажности		
	малой	нормальной	повышенной
Вертикальный			
	1,5	1,7	1,9
	1,3	1,5	1,7
	1,2	1,3	1,5
	1	1,1	1,3
Горизонтальный			
	4,1	5,5	9,3
	2,6	3,5	5,9
	2	2,5	4,2
	1,1	1,5	2,5

Из таблицы 5.6 выбираем удельное электрическое сопротивление грунта:

$$\rho_{уд} = 150 \text{ Ом*м};$$

$$\psi_{в} = 1,1; \psi_{г} = 1,5 \text{ (из таблицы 5.5)}$$

$$\rho_{в} = 150 * 1,1 = 165 \text{ Ом*м}$$

$$\rho_{г} = 150 * 1,5 = 225 \text{ Ом*м}$$

Т а б л и ц а 5.6 - Приближенные значения удельных электрических сопротивлений различных грунтов, Ом*м

Грунт	Возможные пределы колебаний	При влажности 10-20% к массе грунта
Суглинок	40-150	100

За естественный заземлитель примем фундамент здания.

Определим сопротивление растекания тока, имеющих естественных заземлителей.

Сопротивление растекания тока железобетонного здания

$$R_{\phi} = 0,5 \cdot \zeta \cdot \rho \cdot \sqrt{S} \quad (5.3)$$

где ζ - коэффициент, учитывающий наличие бетонного слоя и его промерзания $\zeta = 1,5 - 1,8$;

S_{ϕ} - площадь, ограниченная фундаментом здания;

$$R_{\phi} = 0,5 \cdot 1,6 \cdot 225 / \sqrt{100 \cdot 12} = 5,2 \text{ Ом.}$$

Сопротивление рассеяния искусственных заземлителей $R_{Н\text{доп}}$ можно определить из выражения $1/R_{Н\text{доп}} \leq 1/R_{Н} - 1/R_{с}$;

$$R_{Н\text{доп}} = (4 \cdot 5,2) / (5,2 - 4) = 17,32 \text{ кОм.}$$

Выбирают тип заземляющего устройства и размеры заземлителя.

В нашем случае выносной. Вертикальный электрод, заглублен в грунт на расстояние от поверхности земли до середины электрода T . Длина электрода $L=3\text{м}$, $d = 0,05\text{м}$, $t_0 = 0,5\text{м}$.

Определим сопротивление рассеяния тока заземлителя, размеры и форму определили.

Определим минимальное необходимое количество электродов

$$N = R_{в0} / R_{Н}$$

При этом

$$R_{в0} = \frac{\rho_a}{2 \cdot \pi \cdot L} \cdot \left(\ln \frac{2 \cdot L}{d} + \frac{1}{2} \cdot \ln \left(\frac{4 \cdot T + L}{4 \cdot T - L} \right) \right) = \frac{165}{2 \cdot \pi \cdot 3} \cdot \left(\ln \frac{2 \cdot 3}{0,04} + \frac{1}{2} \cdot \ln \left(\frac{4 \cdot 2 + 3}{4 \cdot 2 - 3} \right) \right) = 47,34 \text{ Ом}$$

Тогда $N = 47,34 / 17,32 = 2,73$;

Зададимся расстоянием между электродами, а так как электроды в ряд примем $a/L=2$. Необходимое количество электродов $n_{в} = 3$.

Вертикальные электроды размещают на плане участка, введенного под заземлитель и находят длину $L_{г}$. При расположении электродов в ряд.

$$L_{г} = 1,05 \cdot (n_{в} - 1) \cdot a = 9\text{м}$$

Для горизонтального электрода определим минимальный диаметр сечения $d=9,5\text{мм}$. По формуле:

$$R_{г0} = \frac{\rho}{2 \cdot \pi \cdot L} \cdot \frac{L}{d} \quad (5.4)$$

Определим сопротивление растекания горизонтального электрода.

$$R_{г0} = 50,84 \text{ Ом.}$$

Тогда сопротивление группового заземления

$$R_{г0} = \frac{R_{в0} \cdot R_{г0}}{R_{в0} + R_{г0}} \quad (5.5)$$

где $\eta_{г}$, $\eta_{в}$ - соответственно коэффициенты использования горизонтального и вертикального электродов группового заземления.

$$\eta_{г} = 0,94; \eta_{в} = 0,96; R_{г0} = 17,07 \text{ Ом;}$$

$R_{гр} < R_{Н доп}$ - рассчитанное по формуле $R_{гр}$ не превышает предельно допустимого значения $R_{Н доп}$.

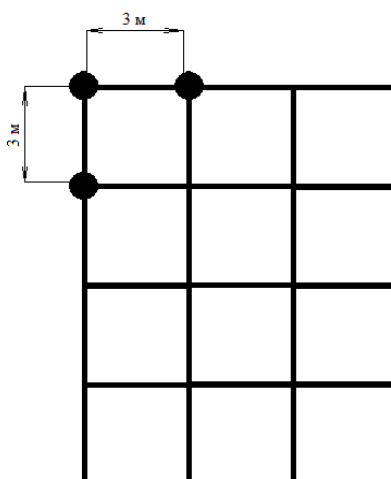


Рисунок 5.3 — Схема расположения электродов

5.2 Вывод по разделу охраны безопасности жизнедеятельности

В данном разделе дипломного проекта были изложены основные положения, связанные с организацией безопасной работы программиста работающего с компьютером и отладочной платой EVK1105. Проанализировали примененные методы для обеспечения электробезопасности в отладочной плате EVK1105. Произвели расчеты электробезопасности и необходимые требования для проведения заземления.

5.3 Расчет системы кондиционирования

Определим необходимое количество кондиционеров для создания комфортных условий труда в помещении. В помещении за счёт тепловыделений производственного оборудования могут иметь место значительные избытки тепла (разность между тепловыделениями в помещении и теплоотдачей через стены, окна, двери и т.д.), удаление которых, прежде всего, должна обеспечить система вентиляции.

Избыточное тепло

$$Q_{\text{изб}} = Q_{\text{об}} + Q_{\text{осв}} + Q_{\text{л}} + Q_{\text{р}} - Q_{\text{отд}} \quad (5.12)$$

где $Q_{\text{об}}$, $Q_{\text{осв}}$, $Q_{\text{л}}$ – тепло, выделяемое производственным оборудованием, системой искусственного освещения помещения и работающим персоналом (людьми) соответственно, ккал/ч;

$Q_{\text{р}}$ – тепло, вносимое в помещение солнцем (солнечная радиация), ккал/ч;

$Q_{\text{отд}}$ – теплоотдача естественным путём, ккал/ч.

Тепло, выделяемое производственным оборудованием

$$Q_{об} = 860 \cdot P_{об} \cdot \eta \quad (5.13)$$

где 860 тепловой эквивалент 1 кВт/ч;

$P_{об}$ – мощность, потребляемая оборудованием, кВт/ч;

η – коэффициент перехода тепла в помещение.

Для 1 компьютера имеем:

$$Q_{иА} = 860 \cdot (1 \cdot 0,250) \cdot 0,95 = 204,25 \text{ ккал/ч},$$

где $\eta=0,95$ – норма потерь потребляемой мощности на тепловыделения компьютерного оборудования.

Тепло, выделяемое осветительными установками

$$Q_{осв} = 860 \cdot \eta \cdot N \quad (5.14)$$

где N – расходуемая мощность светильников, кВт;

$\eta=0,55$ – норма потерь потребляемой мощности на тепловыделения люминесцентных ламп.

$$Q_{осв} = 860 \cdot 0,55 \cdot 0,52 = 246 \text{ ккал/ч};$$

Тепло, выделяемое людьми

$$Q_{л} = K_{л} \cdot (q - q_{исп}) \quad (5.15)$$

где $K_{л}$ – количество работающих;

$(q - q_{исп})$ – явное тепло, ккал/ч;

q – тепловыделения одного человека при данной категории работ I-III, ккал/ч.

Работа, производимая в помещении, относится к I категории работ:

$q=100$ Вт или 0,1 кВт для офисных помещений;

$$Q_{л} = 4 \cdot 860 \cdot 0,1 = 344 \text{ ккал/ч}$$

Тепло, вносимое солнечной радиацией

$$Q_{р} = m \cdot F \cdot q_{ост} \quad (5.16)$$

где m – количество окон в помещении;

F – площадь одного окна, м²;

$Q_{ост}$ – солнечная радиация через остеклённую поверхность, т.е. количество тепла, вносимое за один час через остеклённую поверхность площадью 1 м².

Для окна с двойным остеклением с деревянными переплетами $q_{ост} = 105$ (окна выходят на север, Алматы находится на широте 43° сев. широты). Количество окон равно 1. Площадь окна равна $7,02 \cdot 1,5 = 10,53 \text{ м}^2$.

$$Q_p = 1 \cdot 10,53 \cdot 105 = 1105,65 \text{ ккал/ч};$$

Для тёплого периода года при расчётах можно принять $Q_{отд} = 0$;

$$Q_{изб} = 204,25 + 246 + 344 + 1105,65 = 1899,9 \text{ ккал/ч};$$

При наличии теплоизбытков количество воздуха, которое необходимо удалить из помещения

$$L_b = \frac{Q_{изб}}{C_b \cdot \Delta t \cdot \gamma_b} \quad (5.17)$$

где $Q_{изб}$ – избыточное тепло, ккал/ч;

C_b – теплоёмкость воздуха ($0,24 \text{ ккал/кг}^\circ\text{C}$);

$$\Delta t = t_{вых} - t_{вх};$$

$t_{вых}$ – температура воздуха выходящего из помещения, $^\circ\text{C}$;

$t_{вх}$ – температура воздуха поступающего в помещение, $^\circ\text{C}$;

$\gamma_b = 1,206 \text{ кг/м}^3$ – удельная масса приточного воздуха.

Величина Δt при расчётах выбирается в зависимости от теплонапряжённости воздуха

$$Q_H = \frac{Q_{изб}}{V_H} \quad (5.18)$$

$$Q_H = \frac{1899,9}{96} = 19,8 \text{ ккал/м}^3;$$

Если теплонапряжённость воздуха $Q_H < 20 \text{ ккал/м}^3$, то принимают $\Delta t = 6^\circ\text{C}$, а при $Q_H > 20 \text{ ккал/м}^3$, $\Delta t = 8^\circ\text{C}$.

$$L_b = \frac{1899,9}{0,24 \cdot 6 \cdot 1,206} = 1094 \text{ м}^3/\text{ч};$$

Существующий оконный кондиционер имеет расход воздуха $450 \text{ м}^3/\text{ч}$. Определим требуемое количество таких кондиционеров:

$$N = \frac{1094}{450} \approx 2 \text{ кондиционера.}$$

5.4 Расчёт естественного освещения

Расчитаем площадь боковых световых проёмов в помещении, необходимую для создания нормируемой освещённости на рабочем месте.

Помещение имеет размеры: длина $a = 8 \text{ м}$, ширина $b = 4 \text{ м}$, высота $h = 2,8 \text{ м}$. Высота рабочей поверхности над уровнем пола – $0,7 \text{ м}$, окно начинается с высоты $0,8 \text{ м}$, высота окна $1,5 \text{ м}$. Рабочее помещение находится в IV часовом поясе – город Алматы. Со всех сторон затеняющих зданий нет.

Рабочее место расположено в 0,5 м от наружной стены помещения, где проектируем оконные проёмы. Общую требуемую площадь окон S_0 , m^2 определим по формуле

$$100 \cdot \frac{S_0}{S_n} = \frac{e_n \cdot \eta_0}{\tau_0 \cdot r_1} \cdot k_{зд} \cdot k_3 \quad (5.1)$$

$$S_0 = \frac{S_n \cdot e_n \cdot \eta_0 \cdot k_{зд} \cdot k_3}{100 \cdot \tau_0 \cdot r_1} \quad (5.2)$$

где S_n – площадь помещения, m^2 ;

$$S_n = a \cdot b = 4 \cdot 8 = 32 m^2;$$

e_n – нормированное значение КЕО.

$$e_n^{IV} = e_n \cdot m \cdot c \quad (5.3)$$

где $m=0,9$;

$c=0,75$ – для IV часового пояса (таблица 5.3)

Т а б л и ц а 5.3 – Значения коэффициентов m, c (СНиП II-4-79)

Пояс светового климата	с, при световых проемах				
		в наружных стенах зданий	в прямоугольных и трапециевидных фонарях	в фонарях типа шед.	при зенитных фонарях
IV 50° северной широты и Южнее (Алматы и Караганда)	.9	0.8	0.9	1.0	0.9
	.9	0.75	0.85	0.95	0.85

$e_n=1,2$ для работ средней точности V подряда;

$$e_n^{IV} = 1,2 \cdot 0,9 \cdot 0,75 = 0,81;$$

k_3 – коэффициент запаса;

$k_3=1,2$ (учебные помещения, лаборатории, конструкторские бюро);

τ_0 – общий коэффициент светопропускания равный.

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 \quad (5.4)$$

В качестве светопропускающего материала используем:

– стекло оконное листовое, двойное:

- $\tau_1 = 0,8$;
- вид переплѐта – двойной раздельный:
 $\tau_2 = 0,6$;
- вид несущей конструкции – железобетонные фермы:
 $\tau_3 = 0,8$;
- солнцезащитные устройства – убираться регулируемые жалюзи.
 $\tau_4 = 1$;
- общий коэффициент светопропускания:
 $\tau_0 = 0,8 \cdot 0,6 \cdot 0,8 \cdot 1 = 0,384$;
- η_0 – световая характеристика окон.

Отношение длины комнаты к ее ширине $\frac{8}{4} = 2$. Отношение ширины помещения к высоте от уровня рабочей поверхности до верха окна равно $\frac{4}{2,2} = 1,8$. Отсюда $\eta_0 = 13$

$k_{зд}$ – коэффициент, учитывающий затенение окон противостоящими зданиями;

Поскольку затеняющих зданий поблизости нет, то $k_{зд} = 1$.

Вычислим общую площадь окон:

$$S_0 = \frac{32 \cdot 0,81 \cdot 13 \cdot 1 \cdot 1,2}{100 \cdot 0,384 \cdot 1} = 10,53 \text{ м}^2 ;$$

Так как в кабинете общая площадь окон составляет $10,53 \text{ м}^2$, следовательно они соответствуют нормативам естественного освещения рабочего помещения.

5.4.1 Расчѐт искусственного освещения методом коэффициента использования

Разряд зрительной работы – V. Нормируемая освещѐнность по таблице **Ошибка! Источник ссылки не найден.** 5.4 – 400 лк.

Т а б л и ц а 5.4 – Технические характеристики газоразрядной лампы ЛБ

Номинальная мощность, Вт	Номинальный световой поток ламп типа ЛБ, лм	Размеры ламп, мм	
		Диаметр	Длина по штырькам
65	3570	40	1514,2

В качестве светильника возьмем ЛСП64-2. Длина светильника 1540 мм, ширина 276 мм.

Расчёт искусственного освещения производим методом коэффициента использования.

Коэффициенты отражения от потолка стен и пола соответственно равны:

$$\rho_{\text{пот}} = 70\% ;$$

$$\rho_{\text{ст}} = 50\% ;$$

$$\rho_{\text{пол}} = 30\% ;$$

Вычислим высоту подвеса светильника над рабочей поверхностью

$$H = h - h_p - h_c \quad (5.5)$$

где h_c – расстояние от светильника до перекрытия, $h_c = 0,05$ м;

h_p – высота рабочей поверхности над полом, $h_p = 0,7$ м;

h – высота помещения, $h = 2,8$ м.

$$H = 2,8 - 0,7 - 0,05 = 2,05 \text{ м};$$

Наивыгоднейшее расстояние от окна до светильника определяется как

$$L = \lambda \cdot H \quad (5.6)$$

где $\lambda = 1,2 \div 1,4$;

$$L = 1,25 \cdot 2,05 = 2,56 \text{ м};$$

Расстояние от стены до ближайшего светильника, когда работа у стены не проводится, определяем по формуле

$$l_1 = (0,4 \div 0,5) \cdot L \quad (5.7)$$

$$l_1 = 0,4 \cdot 2,56 = 1,024 \text{ м};$$

Определяем индекс помещения

$$i = \frac{S}{(h_1 - h_2) \cdot (a + b)} \quad (5.8)$$

$$i = \frac{32}{(2,8 - 0,7) \cdot (4 + 8)} = 1,269 ;$$

Коэффициент использования в данном случае равен $\eta = 60\%$, коэффициент запаса равен $k_z = 1,2$

Определим количество люминесцентных ламп по формуле

$$N = \frac{E \cdot k_z \cdot S_{OC} \cdot Z}{n \cdot \Phi_{л} \cdot \eta} \quad (5.9)$$

где S_{OC} – площадь помещения;
 k_z – коэффициент запаса;
 E – заданная минимальная освещённость, $E=400$ лк;
 Z – коэффициент неравномерности освещения, $Z=1,1$;
 n – количество ламп в светильнике;
 $\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л}=3570$ лм;
 η – коэффициент использования, $\eta=60\%$;

$$N = \frac{400 \cdot 1,2 \cdot 32 \cdot 1,1}{2 \cdot 3570 \cdot 0,60} \approx 4$$

Всего для создания нормируемой освещенности 400 лк необходимо четыре люминесцентные лампы серии ЛБ, мощность каждой лампы должна быть не меньше 51 Вт, что соответствует действительности, а значит имеющегося в наличии освещения достаточно для соответствия санитарным нормам.

5.4.2 Расчет искусственного освещения точечным методом

Разряд зрительной работы – V.

Нормируемая освещенность по таблице 5.2 – 400 лк.

Для искусственного освещения применяем люминесцентные лампы ЛБ 40-4. Параметры люминесцентных ламп приведены в таблице 5.10.

Высота подвеса светильников над освещаемой поверхностью $H=2,05$ м, коэффициент запаса равен $k_z = 1,2$.

Схема освещенности представлена на рисунке 5.2.

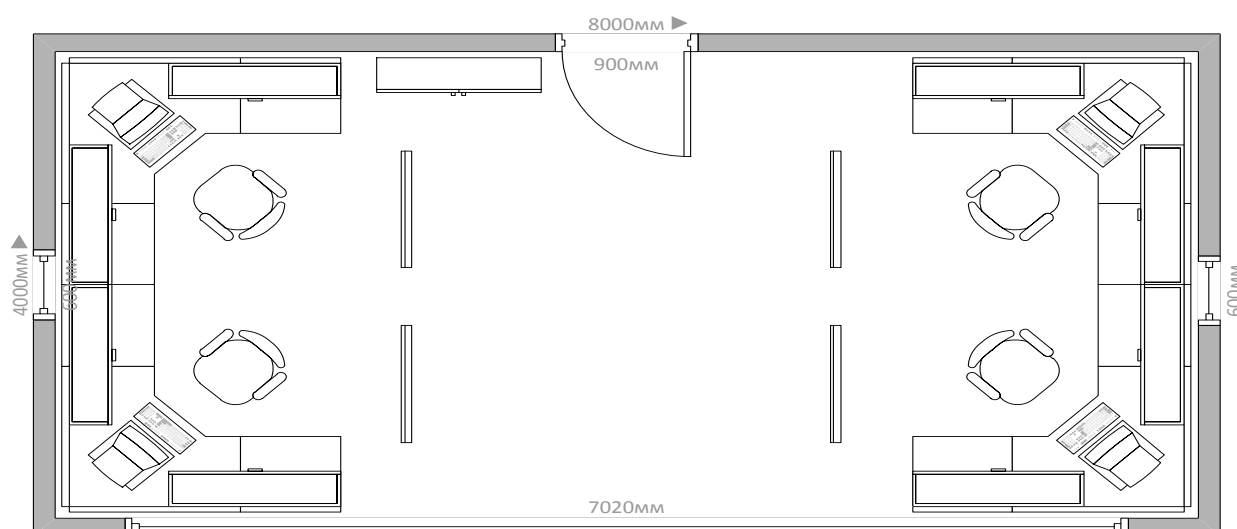


Рисунок 5.2 – Схема освещенности.

Намечаем контрольную точку O. Находим расстояние от этой точки до каждого светильника и затем по графику пространственных изолюкс находим освещенность каждого светильника.

Первый светильник:

$$d_1=2,3 \text{ м};$$

$$E_1=9 \text{ лк};$$

Второй светильник:

$$d_2=2,5 \text{ м};$$

$$E_2=8 \text{ лк};$$

Третий светильник:

$$d_3=0,8 \text{ м};$$

$$E_3=30 \text{ лк};$$

Четвертый светильник:

$$d_4=1,2 \text{ м};$$

$$E_4=20 \text{ лк}.$$

Суммарная освещенность от всех светильников в контрольной точке O

$$\sum E_o = E_1 + E_2 + E_3 + E_4 \quad (5.10)$$

$$\sum E_o = 9 + 8 + 30 + 20 = 67 \text{ лк};$$

Световой поток вычисляется по формуле

$$\phi = \frac{1000 \cdot E \cdot k_z}{\mu \cdot \sum E_o} \quad (5.11)$$

где μ – коэффициент, учитывающий действие «удаленных» светильников (принимается 1,1-1,2);

E – нормируемая освещенность помещения.

$$\phi = \frac{1000 \cdot 400 \cdot 1,2}{1,2 \cdot 67} = 5970,15 \text{ лм};$$

ЗАКЛЮЧЕНИЕ

По условиям поставленной задачи была система аппаратного управления, поддерживающая для связи TCP/IP протокол через Ethernet порт. Аппаратным решением стала плата EVK1105 на базе микроконтроллера AT32UC3A0512. Веб сервер был реализован через DHCP. Была использована своя сборка операционной системы FreeRTOS под архитектуру микроконтроллера.

Во время испытаний система зарекомендовала себя с хорошей стороны и не имеет никаких нареканий, она успешно работает. На экран выводится сводка текущих параметров работы платы.

В общем, система укладывается в концепцию широкополосных систем вычисления состоящих из простых типовых элементов соединённых в сеть, благодаря чему необходимые требования к системе в целом при её росте всегда удовлетворяются т.к. с ростом общего количества элементов растёт и мощность системы в целом. Подобная концепция исключает перерасход ресурсов.

Список использованных литератур

- 1 Гилстер П. Новый навигатор Интернет. / Пер. с англ. – Киев: Диалектика, 1996. – 495 с.
- 2 Игер Б. Работа в Интернет. / Под ред. А. Тихонова. Пер. с англ. – М.: БИНОМ, 1996. – 313 с.
- 3 Кент П. Интернет. / Пер. с англ. В.Л. Григорьева. – М.: Компьютер, ЮНИТИ, 1996. – 267 с.
- 4 Колесников О.Э. Интернет для делового человека. – М.: МЦФ. Изд-во фирма Яуза, 1996. – 281 с.
- 5 Крол Эд. Все об Интернет: Руководство и каталог. / Пер. с англ. С.М. Тимачева. – Киев: BNV, 1995. 591 с.
- 6 Левин В.К. Защита информации в информационно-вычислительных системах и сетях // Программирование. – 1994. – №5. – С. 5-16.
- 7 Нольден М. Ваш первый выход в Интернет. Для начинающих пользователей Интернет и широкого круга пользователей РС. / Гл. ред. Е.В. Кондукова; Пер с нем. К.А. Шиндер. – Спб.: ИКС, 1996. – 238 с.
- 8 Крамер Д. Электронный учебник HTML: наглядный курс Web-дизайна – Спб.: ИКС, 1999. – 238 с.
- 9 Леонтьев М. Электронный учебник Web-дизайн. Руководство пользователя – М.: Компьютер, ЮНИТИ, 1996. – 267 с.
- 10 Нидерст Б. Web-мастеринг для профессионалов. Настольный справочник – М.: БИНОМ, 2003. – 313 с.
- 11 Хавронская А.М. Организация и планирование деятельности производственных подразделений предприятия. Методические указания по выполнению курсовой работы по дисциплине. – Алматы: АУЭС, 2012.
- 12 Новицкий Н.И. Основы менеджмента: организация и планирование производства. Задачи и лабораторные работы. – М.: Финансы и статистика, 1998.
- 13 СНиП РК 2.04.-05.2002 Естественное и искусственное освещение.
- 14 Лесман Е. А. Освещение административных зданий и помещений. – Л.: Энергоатомиздат, 1985.
- 15 Пособие по расчету и проектированию естественного, искусственного и совместного освещения. – М.: Стройиздат, 1985.
- 16 Ананьев В. А., Балужева Л. Н., Гальперин А. Д. и др. Системы кондиционирования. Теория и практика: Учеб. пособие – М.: Евроклимат Изд-во Арина, 2000 – 416 с.
- 17 СНиП РК 2.02-05-2002 огнестойкость помещения.