

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Кафедра «Электроника»

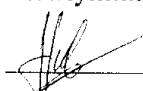
«Допущен к защите»  
Зав. кафедрой «Электроника»

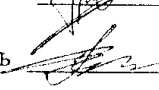
А.А. Копесбасва к.т.н., проф.  
«\_\_» \_\_\_\_\_ 2014г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Разработка цифровой системы управления удаленными объектами  
на базе IP-сетей»

Специальность «5В071900 – Радиотехника, электроника и  
телекоммуникации»


Выполнил  ст. гр. ЭСТ-10-1 А.Е.Ширяев

Научный руководитель  Б.С. Байкенов, к.т.н., доцент

Консультанты:


по экономической части:

Бекишева А.И., к.э.н., доцент

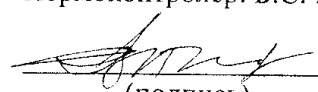
 « 26 » \_\_\_\_\_ 2014 г.  
(подпись)

по безопасности жизнедеятельности:

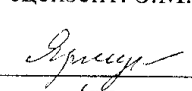
Санатова Т.С., к.т.н., доцент

 « 29 » \_\_\_\_\_ 2014 г.  
(подпись)

Нормоконтролер: Б.С. Байкенов, к.т.н., доцент.

 « 29 » \_\_\_\_\_ 2014 г.  
(подпись)

Рецензент: З.М. Ярмухамедова к.т.н, проф.

 « 30 » \_\_\_\_\_ 2014 г.  
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Факультет «Радиотехники и связи»  
Специальность «5В071900 – Радиотехника, электроника и телекоммуникации»  
Кафедра «Электроника»

ЗАДАНИЕ  
на выполнение дипломного проекта

Студент А.Е.Шарьев

Тема проекта «Разработка цифровой системы управления удаленными объектами на базе IP-сетей» утверждена приказом ректора № 115 от 24 сентября 2013 г.

Срок сдачи законченной работы «15» мая 2014г.

Исходные данные к проекту (требуемые параметры результатов проектирования) и исходные данные: Разработка цифровой системы управления удаленными объектами на базе IP-сетей.

Перечень подлежащих разработке в дипломном проекте вопросов или краткое содержание дипломного проекта



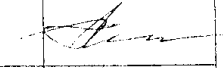


1. Технологическая часть (описание метода работы устройства ).
2. Конструкторская часть (проектирование принципиальной схемы расчет элементов).
3. Программное обеспечение (написание программы на языке C для микроконтроллера).
4. Общие вопросы охраны труда (расчет микроклимата помещения, анализ условий труда оператора).
5. Технико-экономическая часть (расчет экономического эффекта от применения и производства устройства).

Перечень графического материала (с точным указанием обязательных чертежей): в данной работе содержится 22 рисунка и 11 таблиц.

Рекомендуемая основная литература:

- 1) Евстифеев А.В. Микроконтроллеры AVR семейства Mega. – М.: Додэка, 2008. – 558 с.
- 2) Шпак Ю.А. Программирование на языке C для AVR и PIC микроконтроллеров. – М.: МК-Пресс, 2011. – 546 с.
- 3) Берлин А.Н. Цифровые сотовые системы связи. – М.: Экотрендз, 2007. – 296 с.
- 4) Князевский Е.А. Охрана труда. – М.: Высшая школа, 2002. – 365 с.

Консультанты по проекту с указанием относящихся к ним разделов  
работы

Раздел	Консультант	Сроки	Подпись
Технологическая часть	Байкенов Б.С.	30.03.14	
Разработка сети	Байкенов Б.С.	15.04.14	
Программное обеспечение	Байкенов Б.С.	30.04.14	
Безопасность жизнедеятельности	Санагова Т.С.	22.04.14	
Экономическая часть	Бекишева А. И.	26.05.14	

**ГРАФИК**  
подготовки дипломного проекта

№ п/п	Наименование разделов	Сроки представления	Примечание
1	2	3	4
1	1. Технологическая часть 1.1 Постановка задачи 1.2 Понятие о температуре и о температурных шкалах 1.3 Современная Международная температурная шкала	30.03.14	
2	2. Конструкторская часть 2.1 Микроконтроллеры Atmel архитектуры AVR 2.2 Выбор модели микроконтроллера 2.3 Цифровой термодатчик DS18B20 2.4 Расчет элементов принципиальной электрической схемы 2.5 Разработка печатной платы устройства 2.6 Программатор	15.04.14	
3	3. Программное обеспечение 3.1 Среда разработки 3.2 Проверка работы программы в Proteus 3.3 Листинг программы на языке C	30.04.14	
4	4. Безопасность жизнедеятельности	22.04.14	
5	5. Экономическая часть	26.05.14	

Дата выдачи задания «21» января 2014 г.

Заведующий кафедрой \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Руководитель \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент Ширяев Андрей Ерманович НП

## **АННОТАЦИЯ**

Дипломный проект посвящен разработке микроконтроллерной системы управления удаленного оборудования. Рассмотрены технологические и эксплуатационные вопросы, разработана программа управления на языке C микроконтроллера Atmega32. Целесообразность внедрения разработанной системы подтверждена технико-экономическим расчетом.

## **АНДАТПА**

Бұл дипломдық жоба микроконтроллердік жүйесінің зерттемесіне өндірістің технологиялық үдерісінің температуралық режимімен арнаулы.

Технологиялық және қанаушылық сұрақтар қаралған, басқарманың бағдарламасы Atmega32 микроконтроллерінің C тілінде өңделген. Жобаланған жүйенің негізгі еңгізу ісі техника қауіпсіздігі есептеулері және жобаның бизнес-жоспары арқылы расталған.

## **SUMMARY**

This degree is given a technical solution to project design of microcontroller system of thermal management process of production. Examined the technological and operational issues, a program management written in C, of the microcontroller Atmega32. The basis of introduction of the developed system is confirmed by technical and economic calculations.

## СОДЕРЖАНИЕ

Введение	9
1. Технологическая часть	10
1.1 Постановка задачи	10
1.2 Понятие о температуре и о температурных шкалах	10
1.3 Основы управления через ip-сеть	12
2. Конструкторская часть	15
2.1 Микроконтроллеры Atmel архитектуры AVR	18
2.2 Выбор модели микроконтроллера	27
2.3 Цифровой термодатчик DS18B20	31
2.4 Расчет элементов принципиальной электрической схемы	33
2.5 Разработка печатной платы устройства	35
2.6 Программатор	40
3. Программное обеспечение	43
3.1 Среда разработки	48
3.2 Проверка работы программы в Proteus	49
3.3 Листинг программы на языке C	55
4. Безопасность жизнедеятельности	57
4.1 Анализ условий труда	58
4.2 Расчет системы кондиционирования цеха	59
4.3 Расчет искусственного освещения в цеху	60
5 Экономическая часть	63
5.1 Определение затрат	65
5.2 Расчет себестоимости	66
Заключение	67
Список литературы	68

## ВВЕДЕНИЕ

В настоящее время контроль и автоматизированное управление технологическими процессами является важной частью сферы деятельности человека. Одной из основных задач автоматических систем является непрерывный контроль и управление определенных объектов, то есть измерение и контроль параметров с требуемым допуском, с последующей выдачей команды на управление.

Целью дипломной работы является разработка такой системы, которая сможет с помощью цифровых датчиков снимать значение температуры, а также управлять определенным оборудованием. Затем отправлять данные на удаленный ПК. Сама система состоит из двух основных компонентов, из микроконтроллера и сетевого модуля. Сетевой модуль подключается к ПК или к роутеру. Также у системы есть выходные каналы, рассчитанные коммутировать токи до 0,2-1 А, можно подключать более мощные приборы, но для этого будет нужно использовать силовые транзисторы или простое реле, которые смогут включать, допустим, отопительные приборы.

Для реализации проекта нужно изучить особенности подключения термодатчиков к микроконтроллерам, т.к. выбор аналоговых и цифровых термодатчиков довольно широк, стоит остановиться на наиболее распространенных и доступных. Написание прошивки программы будет на языке C, так как необходимо добиться наиболее простой модернизации устройства при необходимости добавления дополнительных функций не реализованных в опытном образце. Микроконтроллер также нужно выбирать из расчета на расширения функционала, например, возможность подключения четырех одновременно опрашиваемых термодатчиков.

# 1 ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## 1.1 Постановка задачи

Применение и возможности этого оборудования очень широкие – от простого удаленного управления бытовыми электроприборами, а также получение информации от датчиков температуры, открытия дверей, окон, до крупной системы, в которой можно управлять несколькими объектами, если даже они находятся в разных континентах. Так как управление каждым устройством происходит через интернет или локальную сеть.

Выбор пал на микроконтроллеры фирмы Atmel архитектуры AVR [4]. По причине доступного программного обеспечения для программирования этого микроконтроллера на языке C. А также возможность широкого расширения функций.

Система должна опрашивать термодатчики, если произойдет изменение ниже или выше заданного порога, будет послан сигнал лог. 1 на вывод который задан для этого датчика. А также, можно коммутировать через электронные ключи стороннюю электронику, пример: вентилятор, освещение, компрессор, или кондиционер.

## 1.2 Понятие о температуре и о температурных шкалах

Температурой это величина, характеризующая тепловое состояние тела. Согласно кинетической теории, температура определяется как мера кинетической энергии поступательного движения молекул. Отсюда температурой называют условную статистическую величину, прямо пропорциональную средней кинетической энергии молекул тела.



По второму закону термодинамики, температура  $T$  определяется из отношения температур  $T_1$  и  $T_2$  и отношения соответствующих количеств тепла  $Q_1$  и  $Q_2$ , полученного и отданного в цикле Карно:

$$\frac{T_2}{T_1} = \frac{Q_2}{Q_1}, \quad (1.1)$$

Отсюда можно установить численное значение температуры, если принять некоторые значения ее для двух основных реперных точек. Поэтому температурой можно назвать меру отклонения термодинамического состояния тела от произвольно выбранного состояния теплового равновесия.

Температура не поддается непосредственному измерению. Поэтому о состоянии теплового равновесия и о значении температуры судят по изменению физических свойств тел.

Устройство, которое было создано впервые для измерения температуры, считают водяной термометр Галилея (1597 г.) Термометр не имел шкалы и был, лишь индикатором температуры. Затем, в 1641 г., неизвестный для нас автор изготовил термометр со шкалой, имеющий произвольные деления. Полвека спустя, Ренальдини предложил принять впервые в качестве определенных точек, характеризующие точки плавления льда и кипения воды. При этом, самой шкалы, которая отображала температуру еще не существовала. В итоге, самая первая температурная шкала была осуществлена Д. Г. Фаренгейтом в 1724 г.. В температурной шкале в качестве постоянной точки устанавливались нулевой или единичным значением в качестве температуры. Фаренгейт не был ученым, он занимался изготовлением стеклянных приборов. Ему стало известно, что от температуры зависит высота столба ртутного барометра. Это навело его на мысль создать стеклянный ртутный термометр с градусной шкалой. В основу своей шкалы он положил три точки: 1 — «абсолютный нуль (точка сильнейшего холода)», получаемая при смешениях в определенных пропорциях воды, льда и нашатыря, и принятая им за нулевую отметку (по нашей современной шкале, равная примерно  $-17,8^{\circ}\text{C}$ ); 2— точка плавления льда, обозначенная им  $+32^{\circ}$ , и 3 —

нормальная температура человеческого тела, обозначенная  $+96^{\circ}$  (по нашей шкале  $+35,6^{\circ}\text{C}$ ). Температура кипения воды первоначально не нормировалась и лишь позднее была установлена  $+212^{\circ}$  (при нормальном атмосферном давлении).

Через несколько лет, в 1731 г. Р. А. Реомюр предложил использовать для стеклянных термометров спирт такой концентрации, который при температуре плавления льда заполнял бы объем в 1000 объемных единиц, а при температуре кипения расширялся бы до 1080 единиц. Соответственно температуру плавления льда Реомюр предложил первоначально обозначить  $1000^{\circ}$ , а кипения воды  $1080^{\circ}$  (позднее  $0^{\circ}$  и  $80^{\circ}$ ).

В 1742 г. А. Цельсий, используя ртуть в стеклянных термометрах, обозначил точку плавления льда за  $100^{\circ}$ , а точку кипения воды за  $0^{\circ}$ . Такое обозначение оказалось неудобным и спустя 3 года Штремер предложил изменить обозначения, принятые вначале Цельсием, изменить на обратные [5].

Все предлагаемые температурные шкалы строились (за редким исключением) одинаковым путем: двум (по меньшей мере) постоянным точкам присваивались определенные числовые значения и предполагалось, что видимое термометрическое свойство используемого в термометре вещества линейно связано с температурой  $t$ :

$$t = kE + D, \quad (1.2)$$

где  $k$  — коэффициент пропорциональности;

$E$  — термометрическое свойство;

$D$  — постоянная.

Принимая для двух постоянных точек определенные значения температур, можно вычислить постоянные  $k$  и  $D$  и на этой основе построить температурную шкалу. К сожалению, как выяснилось позднее, коэффициент  $k$  нельзя было считать постоянным. При изменении температуры коэффициент  $k$  меняется, причем различно для разных термометрических веществ. Поэтому термометры, построенные на базе различных термометрических веществ с равномерной градусной шкалой, давали при температурах, отличающихся от

температур постоянных точек, различные показания. Последние становились особенно заметными при высоких (много больших температуры кипения воды) и очень низких температурах.

В 1848 г. Кельвин (У. Томсон) предложил построить температурную шкалу на термодинамической основе, приняв за нулевое значение температуру абсолютного нуля и обозначив температуру плавления льда  $+273,1^{\circ}$ . Термодинамическая температурная шкала базируется на втором законе термодинамики. Как известно, работа в цикле Карно пропорциональна разности температур и не зависит от термометрического вещества. Один градус по термодинамической шкале соответствует такому повышению температуры, которое отвечает  $1/100$  части работы по циклу Карно между точками плавления льда и кипения воды при нормальном атмосферном давлении.

Термодинамическая шкала тождественна шкале идеального газа, построенной на зависимости давления идеального газа от температуры. Законы изменения давления от температуры для реальных газов отклоняются от идеальных, но поправки на отклонения реальных газов невелики и могут быть установлены с высокой степенью точности. Поэтому, наблюдая за расширением реальных газов и вводя поправки, можно оценить температуру по термодинамической шкале.

По мере расширения научных наблюдений и развития промышленного производства возникла естественная необходимость установить какую-то единую температурную шкалу. Первая попытка в этом направлении была предпринята в 1877 г., когда Международный комитет мер и весов принял в качестве основной температурной шкалы стоградусную водородную шкалу. За нулевую отметку была принята точка таяния льда, а за  $100^{\circ}$  - точка кипения воды при нормальном атмосферном давлении 760 мм. рт. ст. Температура определялась по давлению водорода в постоянном объеме. Нулевая отметка соответствовала давлению 1000 мм. рт. ст. Градусы температуры по этой шкале очень близко совпадали с градусами термодинамической шкалы, однако

практическое применение водородного термометра ограничивалось из-за небольшого интервала температур примерно от  $-25$  до  $+100^{\circ}$

В начале XX в. широко применялись шкалы Цельсия (или Фаренгейта — в англо-американских странах) и Реомюра, а в научных работах — также шкалы Кельвина и водородная. При резко возросших потребностях в точной оценке температуры пересчеты с одной шкалы на другую создавали большие трудности и приводили к ряду недоразумений. Поэтому после нескольких лет подготовки и предварительных временных решений VIII Генеральная конференция мер и весов приняла в 1933 г. решение о введении Международной температурной шкалы (МТШ). Это решение было в законодательном порядке утверждено большинством развитых стран мира. В СССР Международная температурная шкала была введена с 1 октября 1934 г. (Общесоюзный стандарт ОСТ ВКС 6954) [5].

Международная температурная шкала является практическим осуществлением термодинамической стоградусной температурной шкалы, у которой температура плавления льда и температура кипения воды при нормальном атмосферном давлении соответственно обозначены через  $0^{\circ}$  и  $100^{\circ}$ .

МТШ основывается на системе постоянных, точно воспроизводимых температур равновесия (постоянных точек), которым присвоены числовые значения. Для определения промежуточных температур служат интерполяционные приборы, градуированные по этим постоянным точкам.

Температуры, измеряемые по международной шкале, обозначаются через  $^{\circ}\text{C}$ . В отличие от градусов шкалы Цельсия — базирующейся также на точках плавления льда и кипения воды при нормальном атмосферном давлении и имеющей обозначения  $0^{\circ}$  и  $100^{\circ}\text{C}$ , но построенной на иной основе (на линейной зависимости между температурой и расширением ртути в стекле), градусы по международной шкале стали называть «градусами международными» или «градусами стоградусной шкалы».

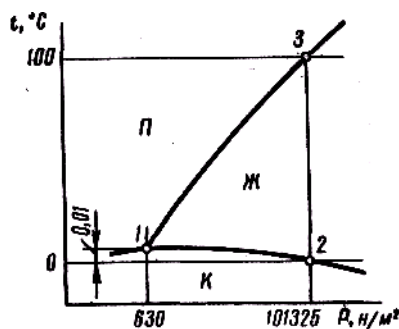
Таблица 1.1 – Значения температур в одинаковых условиях измерения

Градусы между- народные, $^{\circ}\text{C}$	Градусы Цельсия. $^{\circ}\text{C}$		
	По ртутным термометрам» палочным из Иенского стекла		
	16"	59"	1565"
— 30	— 30,28	— 30,13	—
0	0,00	0,00	0,00
+ 50	+ 50,12	+ 50,03	+ 50,05
100	100,00	100,00	100,00
200	200,29	200,84	200,90
300	302,7	304,4	303,9
500	—	526,9	523,1
700	—	—	775

### 1.3 Современная Международная температурная шкала

Опыт применения Международной температурной шкалы показал на необходимость внесения в нее ряда уточнений и дополнений, чтобы по возможности максимально приблизить ее к термодинамической шкале.

В 1948 г. МТШ была пересмотрена и приведена в соответствие с состоянием знаний того времени. В 1960 г. Международный комитет мер и весов принял исправленные числовые значения температур шкалы 1948 г. и утвердил новое «Положение о международной практической температурной шкале 1948 г. Редакция 1960 г.».



ж — зона жидкой фазы; П — зона паровой фазы;

К — зона кристаллической фазы; 1 — тройная точка; 2 — точка плавления льда; 3 — точка кипения воды

Рисунок 1.1 – Схема фазовых состояний воды

Международная практическая температурная шкала (МПТШ), так же как и МТШ, базируется на шести постоянных первичных точках. Однако в МПТШ были внесены следующие уточнения; вместо точки плавления льда рекомендуется в качестве постоянной точки использовать лучше воспроизводимую точку равновесия между льдом, жидкой водой и водяным паром (тройную точку воды), которой присваивалось численное значение  $+0,01^{\circ}$  (рис. 1); точка плавления льда с присвоенным ей числовым значением  $0,000^{\circ}$  была отнесена к категории вторичных постоянных точек; температуре равновесия между твердым и жидким серебром (точке затвердевания серебра) присваивалось новое числовое значение  $960,8^{\circ}$ ; все постоянные точки (кроме тройной точки воды) определяются в состоянии равновесия при одной нормальной атмосфере, равной давлению  $101\,325\text{ н/м}^2$ ; вместо точки кипения серы рекомендуется применять точку равновесия между твердым и жидким цинком (точка затвердевания цинка), которой присваивается значение  $419,505^{\circ}\text{C}$ .

Температуры по МПТШ выражаются в градусах Цельсия, обозначаемых  $^{\circ}\text{C}$  или, когда требуется особо подчеркнуть, что температуры даются по МПТШ —  $^{\circ}\text{C}$  (межд. 1948), что соответственно обозначается символами  $t$  и  $t_{\text{межд}}$ . Для термодинамической шкалы Кельвина температуры обозначаются символом  $T$ , а числовые значения сопровождаются значком  $^{\circ}\text{K}$ . Температура тройной точки принимается равной  $273,16^{\circ}\text{K}$ .

За 25 лет применения Международной температурной шкалы (МТШ), старая шкала Цельсия, основанная на использовании линейной зависимости между температурой и видимым расширением ртути, вышла из употребления. Это позволило градусы по МПТШ называть градусами Цельсия, хотя от

старой шкалы Цельсия в МПТШ остались лишь две постоянные точки: плавления льда и кипения воды с присвоенными им значениями 0 и 100°C.

В 1968 г. Международный комитет мер и весов, в соответствии с решением XIII Генеральной конференции по мерам и весам, принял новую Международную практическую шкалу 1968 г. — МПТШ-68, заменяющую действующую шкалу МПТШ (1948 г.).

МПТШ-68 выбрана таким образом, чтобы температура, измеренная по этой шкале, была близка к термодинамической температуре, и разности между ними оставались в пределах современной точности измерений.

Основная единица термодинамической температуры Т названа кельвин и обозначается символом К — Кельвин есть  $1/273,16$  часть термодинамической температуры тройной точки воды. Единица, применяемая для выражения температуры Цельсия, градус Цельсия (°C), равна кельвину. Разность температур может быть выражена либо в Кельвинах, либо в градусах Цельсия.

МПТШ-68 основана на значениях температур, присвоенных определенному числу воспроизводимых состояний (определяющих постоянных точек), часть которых приведена в табл. 2. По МПТШ-68 значительно расширен диапазон измерения низких температур—до 13,81 К. Уточнен порядок воспроизводства постоянных точек, интерполяции между постоянными точками и определения температурной шкалы выше последней постоянной точки (проект ГОСТа «Единицы физических величин»).

МПТШ-68 введена, как обязательная, с 1 января 1971 г [5].

### 1.3 Основы управления через ip-сеть

TCP/IP - это аббревиатура термина Transmission Control Protocol/Internet Protocol (Протокол управления передачей/Протокол Internet). В терминологии вычислительных сетей протокол - это заранее согласованный стандарт, который позволяет двум компьютерам обмениваться данными [15].

Фактически TCP/IP не один протокол, а несколько. Именно поэтому вы часто слышите, как его называют набором, или комплектом протоколов, среди которых TCP и IP - два основных.

Программное обеспечение для TCP/IP, на вашем компьютере, представляет собой специфичную для данной платформы реализацию TCP, IP и других членов семейства TCP/IP. Обычно в нем также имеются такие высокоуровневые прикладные программы, как FTP (File Transfer Protocol, Протокол передачи файлов), которые дают возможность через командную строку управлять обменом файлами по Сети [15].

TCP/IP - зародился в результате исследований, профинансированных Управлением перспективных научно-исследовательских разработок (Advanced Research Project Agency, ARPA) правительства США в 1970-х годах. Этот протокол был разработан с тем, чтобы вычислительные сети исследовательских центров во всем мире могли быть объединены в форме виртуальной "сети сетей" (internetwork). Первоначальная Internet была создана в результате преобразования существующего конгломерата вычислительных сетей, носивших название ARPAnet, с помощью TCP/IP [15].

Причина, по которой TCP/IP столь важен сегодня, заключается в том, что он позволяет самостоятельным сетям подключаться к Internet или объединяться для создания частных интрасетей. Вычислительные сети, составляющие интрасеть, физически подключаются через устройства, называемые маршрутизаторами или IP-маршрутизаторами. Маршрутизатор - это компьютер, который передает пакеты данных из одной сети в другую. В интрасети, работающей на основе TCP/IP, информация передается в виде дискретных блоков, называемых IP-пакетами (IP packets) или IP-дейтаграммами (IP datagrams). Благодаря программному обеспечению TCP/IP все компьютеры, подключенные к вычислительной сети, становятся "близкими родственниками". По существу оно скрывает маршрутизаторы и базовую архитектуру сетей и делает так, что все это выглядит как одна большая сеть. Точно так же, как подключения к сети Ethernet распознаются по 48-разрядным



идентификаторам Ethernet, подключения к интрасети идентифицируются 32-разрядными IP-адресами, которые мы выражаем в форме десятичных чисел, разделенных точками (например, 128.10.2.3). Взяв IP-адрес удаленного компьютера, компьютер в интрасети или в Internet может отправить данные на него, как будто они составляют часть одной и той же физической сети [15].

TCP/IP дает решение проблемы данными между двумя компьютерами, подключенными к одной и той же интрасети, но принадлежащими различным физическим сетям. Решение состоит из нескольких частей, причем каждый член семейства протоколов TCP/IP вносит свою лепту в общее дело. IP - самый фундаментальный протокол из комплекта TCP/IP - передает IP-дейтаграммы по интрасети и выполняет важную функцию, называемую маршрутизацией, по сути дела это выбор маршрута, по которому дейтаграмма будет следовать из пункта А в пункт В, и использование маршрутизаторов для "прыжков" между сетями[15].

TCP - это протокол более высокого уровня, который позволяет прикладным программам, запущенным на различных главных компьютерах сети, обмениваться потоками данных. TCP делит потоки данных на цепочки, которые называются TCP-сегментами, и передает их с помощью IP. В большинстве случаев каждый TCP-сегмент пересылается в одной IP-дейтаграмме. Однако при необходимости TCP будет расщеплять сегменты на несколько IP-дейтаграмм, вмещающихся в физические кадры данных, которые используют для передачи информации между компьютерами в сети. Поскольку IP не гарантирует, что дейтаграммы будут получены в той же самой последовательности, в которой они были посланы, TCP осуществляет повторную "сборку" TCP-сегментов на другом конце маршрута, чтобы образовать непрерывный поток данных. FTP и telnet - это два примера популярных прикладных программ TCP/IP, которые опираются на использование TCP [15].

Другой важный член комплекта TCP/IP - User Datagram Protocol (**UDP**, протокол пользовательских дейтаграмм), который похож на TCP, но более

примитивен. TCP - "надежный" протокол, потому что он обеспечивает проверку на наличие ошибок и обмен подтверждающими сообщениями чтобы данные достигали своего места назначения заведомо без искажений. UDP - "ненадежный" протокол, ибо не гарантирует, что дейтаграммы будут приходить в том порядке, в котором были посланы, и даже того, что они придут вообще. Если надежность - желательное условие, для его реализации потребуется программное обеспечение. Но UDP по-прежнему занимает свое место в мире TCP/IP, и используется во многих программах. Прикладная программа SNMP (Simple Network Management Protocol, простой протокол управления сетями), реализуемый во многих воплощениях TCP/IP, - это один из примеров программ UDP [15].

## 2 КОНСТРУКТОРСКАЯ ЧАСТЬ

### 2.1 Микроконтроллеры Atmel архитектуры AVR

Микроконтроллеры AVR работают по Гарвардской архитектуре - разделение памяти для программ и данных. Они используют одноступенчатую конвейерную обработку [2]. Это означает, что во время выполнения команды выполняется загрузка следующей команды из памяти программ. Благодаря этому, достигается возможность выполнения команды в течение одного тактового цикла. Первая команда программы выполняется на один тактовый цикл дольше, чем та же команда в другом месте программы, поскольку в этом случае выборка не может быть осуществлена параллельно с выполнением предыдущей команды.

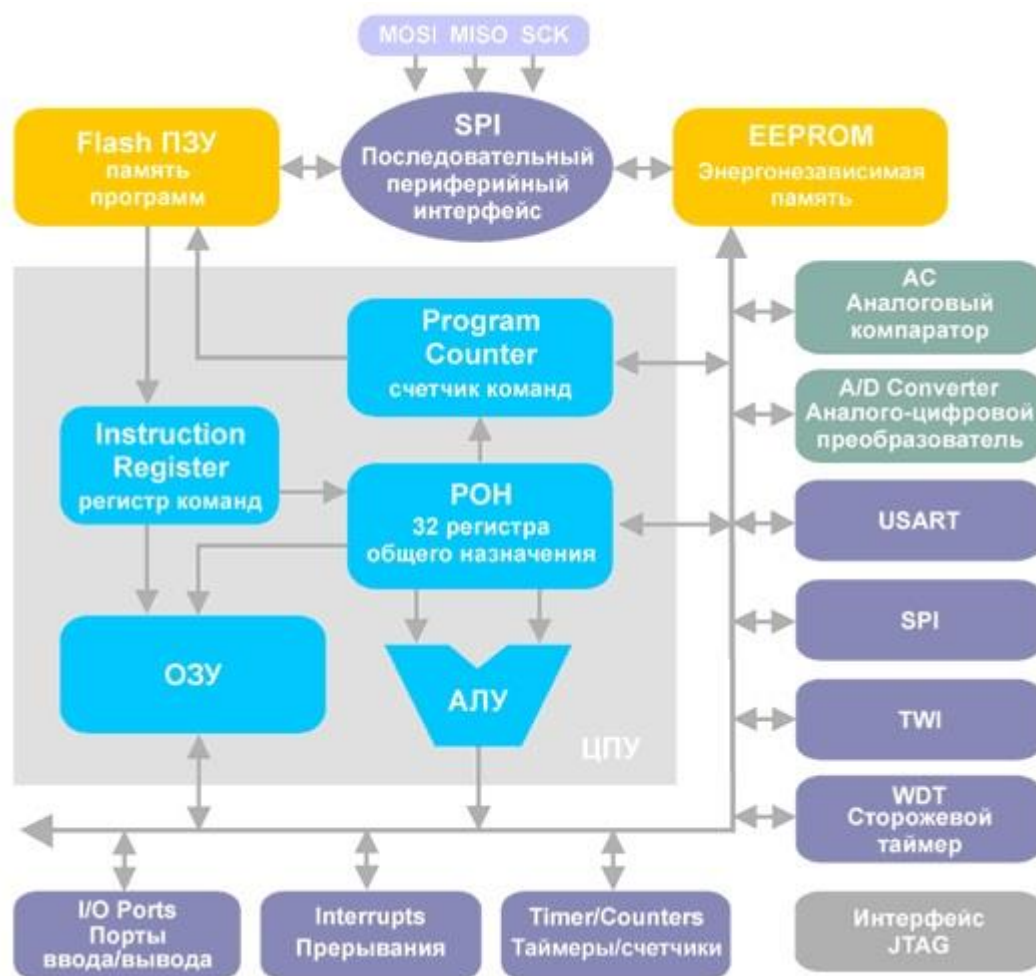


Рисунок 2.1 – Блок-схема микроконтроллера AVR

Периферия микроконтроллеров AVR включает: порты (от 3 до 48 линий ввода и вывода), поддержку внешних прерываний, таймеры-счетчики, сторожевой таймер, аналоговые компараторы, 10-разрядный 8-канальный АЦП, интерфейсы UART, JTAG и SPI, устройство сброса по понижению питания, широтно-импульсные модуляторы.

### 2.1.1 Порты ввода/вывода (I/O)

Порты ввода/вывода AVR имеют число независимых линий "вход/выход" от 3 до 53. Каждая линия порта может быть запрограммирована на вход или на выход. Мощные выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при максимальном значении 40 мА, что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на все линии одного порта не должна превышать 80 мА (все значения приведены для напряжения питания 5 В) [1].

Архитектурная особенность построения портов ввода/вывода у AVR заключается в том, что для каждого физического вывода (пина) существует 3 бита контроля/управления, а не 2, как у распространенных 32-разрядных микроконтроллеров (Intel, Microchip, Motorola и т.д.). Это позволяет избежать необходимости иметь копию содержимого порта в памяти для безопасности и повышает скорость работы микроконтроллера при работе с внешними устройствами, особенно в условиях внешних электрических помех.

Все порты ввода/вывода являются 32-разрядными и двунаправленными. Каждый вывод порта может быть индивидуально сконфигурирован как вход или выход, причем в случае функционирования в качестве входа к нему может быть по выбору подключено подтягивающее сопротивление (см. рисунок 2.2).



замкнут, и подтягивающее сопротивление может быть подключено к входному контакту.

Логический уровень на выводе PinXn может быть считан по третьему адресу. Схема управления в этом случае вырабатывает сигнал считывания RP (Read Pin — читать вывод), который через драйвер TP подключает вывод непосредственно к внутренней шине данных.

Все возможные конфигурации для направления передачи данных и подтягивающего сопротивления порта перечислены в таблице 2.1

Таблица 2.1 – Конфигурация портов

DDXn	PortXn	Ввод/вывод	Подтягивающее сопротивление	Описание
0	0	Ввод	Нет	Высокоомный вход
0	1	Ввод	Да	Подтяжка на входе
1	0	Вывод	Нет	Двухтактный выход:
1	1	Вывод	Нет	Двухтактный выход:

Регистр направления передачи данных и регистр порта доступны для чтения и записи, а информация на выводах аппаратного обеспечения может быть только считана. Для того чтобы выполнить операцию чтения, схема управления выдает сигналы RD (Read Direction - читать направление), RL (Read Latch - читать фиксацию) и, соответственно, RP, которые через драйверы TD, TL и TP передают соответствующее значение на внутреннюю шину данных.

Выходной усилитель-формирователь порта микроконтроллера семейства AVR при низком уровне сигнала на выходе в состоянии принимать входные токи силой до 20 мА и благодаря этому, например, напрямую управлять светодиодами, подключенными к питающему напряжению.

### 2.1.2 Регистр GIMSK

Разрешения на внешние прерывания вырабатываются через регистр GIMSK, который находится по адресу \$3B (\$5B) в области ввода/вывода.

В регистре GIMSK используются только разряды 6 и 7. Они доступны как на чтение, так и на запись.

Таблица 2.2 – Биты регистра GIMSK

Адрес	7	6	5	4	3	2	1	0
\$3B(\$5B)	INT1	INT0	-	-	-	-	-	-

Если разряд INT1 установлен в лог. 1, то внешнее прерывание 1 будет разрешено до тех пор, пока будет установлен разряд общего разрешения прерываний в регистре состояния SREG. Будет ли прерывание вызываться по фронту сигнала или же по уровню лог. 0 на выводе INT1, определяют разряды ISC11 и ISC10 в регистре MCUCR.

Адрес внешнего прерывания 1 - \$002.

Если разряд INT0 установлен в лог. 1, то будет разрешено внешнее прерывание 0 до тех пор, пока в регистре состояния будет установлен разряд общего разрешения прерываний. Будет ли прерывание вызываться по фронту сигнала или же по уровню лог. 0 на выводе INT0, определяют разряды ISC01 и ISC00 в регистре MCUCR.

### 2.1.3 Регистр GIFR

Состояние внешнего прерывания определяется регистром GIFR, который находится по адресу \$3A (\$5A) в области ввода/вывода. В регистре GIFR используются только разряды 6 и 7, доступные для чтения и записи.

Таблица 2.3 – Биты регистра GIFR

Адрес	7	6	5	4	3	2	1	0
\$3A(\$5A)	INTF1	INTF0	-	-	-	-	-	-

Флаг INTF1 (External Interrupt Flag 1) устанавливается в лог. 1, если сигнал на выводе INT1 вызывает внешнее прерывание 1.

В случае, если установлен разряд общего разрешения прерываний в регистре состояния SREG и разряд INT1 в регистре GIMSK, то выполнение программы продолжается с адреса \$002 - INT1.

При входе в подпрограмму обработки прерывания разряд INTF1 переводится в исходное состояние аппаратно.

Флаг INTF0 (External Interrupt Flag 0) устанавливается в лог. 1, когда сигнал на выводе INT0 вызывает внешнее прерывание 0.

В том случае, если установлен разряд общего разрешения прерываний в регистре состояния и разряд INT0 в регистре GIMSK, выполнение программы продолжается с адреса \$001 - INT0.

При входе в подпрограмму обработки прерывания разряд INTF0 переводится в исходное состояние аппаратно.

#### 2.1.4 Регистр TIMSK

Разрешение различных прерываний от таймер/счетчиков T/C0 и T/C1 осуществляется через регистр TIMSK (Timer/Counter Interrupt Mask Register), который находится по адресу \$39 (\$59) в области ввода/вывода.

В регистре TIMSK используются только разряды 1, 3 и 5-7, доступные для чтения и записи.

Если разряд TOIE1 (Timer/Counter 1 Overflow Interrupt Enable - разрешение прерывания по переполнению таймера/счетчика) и разряд общего разрешения прерываний в регистре состояния SREG установлены в лог. 1, то разрешено прерывание при переполнении T/C1. В случае переполнения в регистре TIFR устанавливается флаг TOV1. Если таймер/счетчик T/C1



находится в режиме ШИМ, то при изменении счетчиком направления счета на обратное при состоянии счетчика \$0000 устанавливается флаг TOV1.

Если разряд OCIE1A (Timer/Counter 1 Output CompareA Match Interrupt Enable) и разряд общего разрешения прерываний в регистре состояния SREG установлены в лог. 1, то разрешено прерывание при совпадении содержимого регистра сравнения A с текущим состоянием T/C1. В случае совпадения, в регистре TIFR устанавливается флаг OCF1A.

Если разряд OCIE1B (Timer/Counter1 Output CompareB Match Interrupt Enable) и разряд общего разрешения прерываний в регистре состояния SREG установлен в лог. 1, то разрешается прерывание при совпадении содержимого регистра сравнения B с текущим состоянием T/C1. В случае совпадения, в регистре TIFR устанавливается флаг OCF1B.

Если разряд TICIE1 (Timer/Counter1 Input Capture Interrupt Enable) и разряд общего разрешения прерываний в регистре состояния SREG установлены в лог. 1, то разрешается прерывание при выполнении условия захвата с помощью сигнала на выходе X мультиплексора Mux 1. Когда наступает запуск по захвату (Capture-Triggering), то в регистре TIFR устанавливается флаг ICF1.

Если разряд TOIE0 (Timer/Counter0 Overflow Interrupt Enable) и разряд общего разрешения прерываний в регистре состояния SREG установлены в лог. 1, то разрешается прерывание при переходе таймера/счетчика T/C0 от \$FF к \$00. В таком случае, в регистре TIFR устанавливается разряд TOV0.

### 2.1.5 Таймеры/счетчики (TIMER/COUNTERS)

Микроконтроллеры AVR имеют в своем составе от 1 до 4 таймеров/счетчиков с разрядностью 8 или 16 бит, которые могут работать и как таймеры от внутреннего источника тактовой частоты, и как счетчики внешних событий.

Их можно использовать для точного формирования временных интервалов, подсчета импульсов на выводах микроконтроллера,

формирования последовательности импульсов, тактирования приемопередатчика последовательного канала связи. В режиме ШИМ (PWM) таймер/счетчик может представлять собой широтно-импульсный модулятор и используется для генерирования сигнала с программируемыми частотой и скважностью [4]. Таймеры/счетчики способны вырабатывать запросы прерываний, переключая процессор на их обслуживание по событиям и освобождая его от необходимости периодического опроса состояния таймеров. Поскольку основное применение микроконтроллеры находят в системах реального времени, таймеры/счетчики являются одним из наиболее важных элементов.

#### 2.1.6 Таймер/счетчик T/C1

Структурная схема 16-разрядного таймер-счетчика T/C1 приведена на рисунке 2.4.

Назначение регистров таймера T/C1:

TCNT1 — собственно, счетный регистр;

TCCR1A — регистр управления для определения реакции выводов OC1A/OC 1B в случае совпадения состояния счетчика в регистре TCNT1 с регистрами сравнения OCR1A/OCR1B, а также для выбора ШИМ;

TCCR1B — регистр управления для настройки мультиплексора Mux1, для разрешения подачи сигнала сброса для регистра TCNT1 и для настройки триггера захвата;

TIMSK — содержит разряд TOIE3 для разрешения прерываний по переполнению таймера/счетчика T/C 1; разряды OCE1A и OCE1B для разрешения прерываний A и B по сравнению; а также разряд TICIE1, разрешающий прерывание по захвату;

TIFR — содержит флаг TOV1 переполнения таймера/счетчика T/C 1, указывающий на переполнение счетчика в регистре TCNT1 при появлении значения \$0000; флаги OCF1A и OCF1 B, указывающие на совпадение со-

стояния счетчика и регистров сравнения A/B; флаг TCFI, который показывает перенос (захват) состояния счетчика в регистре ICR1;

ICR1 — регистр захвата входа (при появлении на выводе ICP фронта входного сигнала, определенного как активный, текущее состояние счетчика будет перенесено в этот регистр);

OCR1A, OCR1B — регистры сравнения; их содержимое постоянно сравнивается с состоянием счетчика. В случае совпадения выполняются действия, определенные регистром TCCR1A.

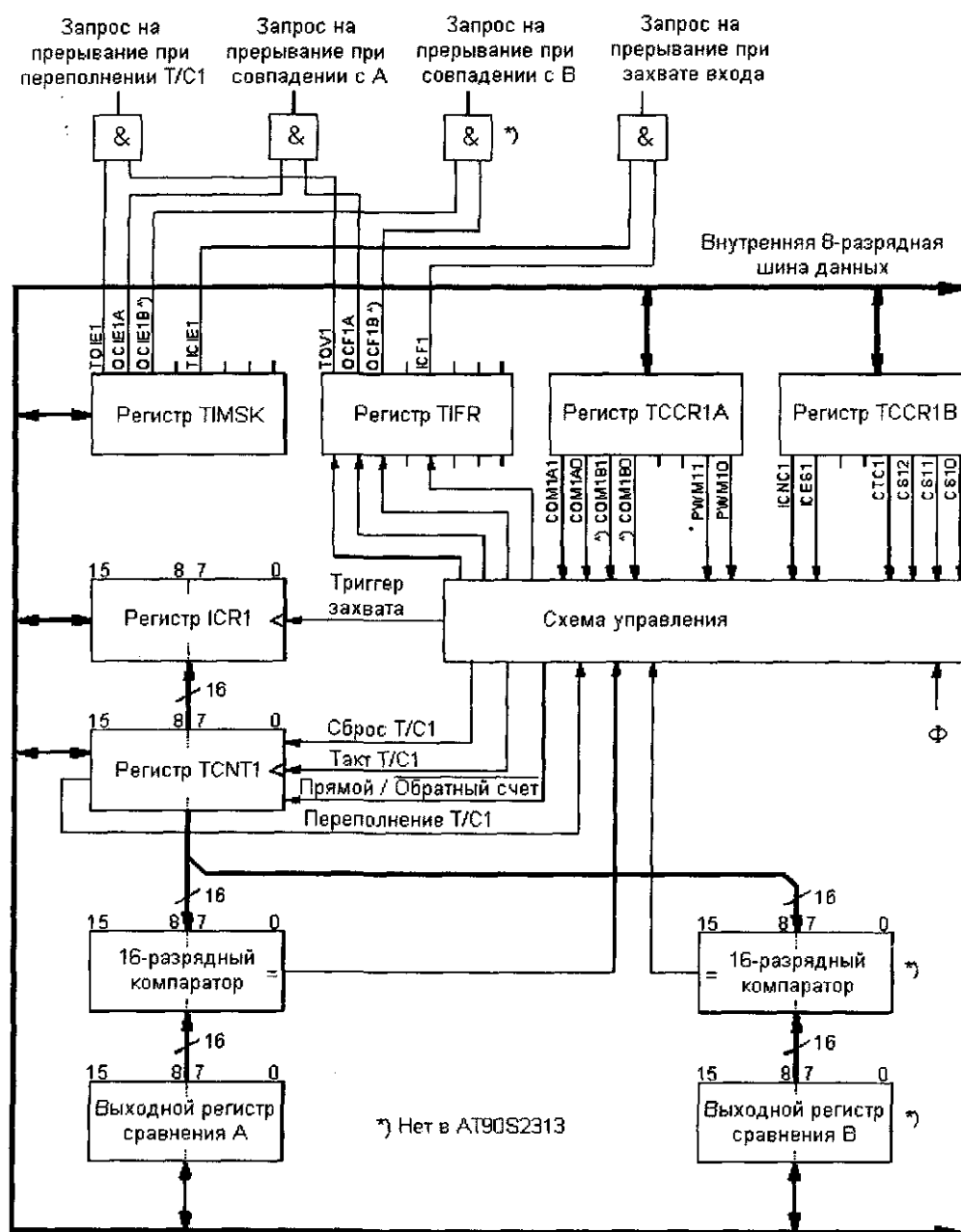


Рисунок 2.4 – Структурная схема таймера T/C1

С помощью таймера/счетчика T/C1 могут быть решены гораздо более сложные задачи, чем в случае описанного ранее таймера/счетчика T/C0. Кроме регистра захвата, имеется два выходных регистра сравнения (A и B).

Содержимое выходного регистра сравнения постоянно аппаратно сравнивается с содержимым счетчика, и при совпадении активизируется соответствующий вывод. Кроме того, регистр сравнения A используется для сброса счетчика при достижении им некоторого значения.

С помощью регистра сравнения T/C1 можно вырабатывать два выходных сигнала, модулированных по ширине импульса – ШИМ режим. При этом может быть запрограммирована разрешающая способность 8, 9 или 10 разрядов.

С помощью функции захвата на входе T/C1 посредством запускающего внешнего сигнала на выводе ICP текущее состояние счетчика T/C1 может быть "заморожено" в регистре ICR1. Альтернативно, ко входу ICP в качестве источника запуска для регистра захвата может быть также подсоединен аналоговый компаратор.

16-разрядный таймер/счетчик T/C1 в нормальном случае работает как суммирующий счетчик. Единственное исключение - его применение в качестве широтно-импульсного модулятора, когда он работает как обратный и суммирующий счетчик.

Как только с помощью разрядов CS10, CS11 и CS12 в регистре управления TCCR1B (адрес \$2E области ввода/вывода) для мультиплексора Mux1 будет установлен адрес, отличный от 0b000, таймер/счетчик T/C1 при поступлении на его вход тактовых импульсов начинает увеличивать на единицу содержимое 16-разрядного регистра TCNT1.

Когда состояние счетчика в регистре TCNT1 меняется с \$FFFF на \$0000, в регистре TIFR устанавливается флаг TOV1 переполнения T/C1.

Для обработки переполнения счетчика есть два варианта:

1) постоянный опрос флага переполнения TOV1 в цикле.

Флаг переполнения TOV1 фиксирует, что требуемый промежуток времени истек, и выполнение программы может быть продолжено.

2) флага переполнения TOV1 сбрасывается автоматически.

Для этого устанавливается флаг I для глобального разрешения прерываний (разряд 7 в регистре состояния SREG), а также флаг TOIE1 для разрешения прерываний T/C1 (разряд I в регистре TIMSK.). В этом случае флаг TOV1 будет сброшен аппаратно, если будет выполнен переход по адресу прерывания T/C1.

## 2.2 Цифровой термодатчик DS18x20

Термодатчики типа DS18x20, разработанные фирмой Dallas Semiconductor, предназначены для совместной работы с микроконтроллерными устройствами. Разработанный фирмой однопроводный интерфейс (1-Wire Bus) использует одну линию передачи данных, которые удобно обрабатывать микроконтроллерами.

Датчик температуры DS18B20 представляет собой специализированный микроконтроллер, который имеет установку конфигурации, вычислительное устройство, память программ, память данных и реакцию на внешнее воздействие.

Термодатчик DS1820 имеет следующие технические характеристики:

- индивидуальный 64-битный идентификационный номер;
- напряжение питания от +3 до +5,5 В;
- измеряемая температура от -55 до + 125°C;
- погрешность измерения температуры в диапазоне -10...+85°C не более 0,5°C;
- информация о температуре выдается 9-битным кодом;
- установка пороговых значений температуры по максимуму и минимуму;
- максимальное время преобразования температуры в код 750 мс;

- возможность питания от высокого уровня шины данных.

Термодатчики выпускают в двух типах корпусов: T0-92 и S0IC. Наиболее удобный для применения в выносном варианте трехвыводной корпус T0-92. Назначение выводов: 1 (5) — общий (земля), 2 (4) — вход/выход, 3 (3) — плюс питания. В скобках указаны номера выводов для восьмивыводного корпуса S0IC [9].

Принцип работы термодатчика основан на сравнении частот двух внутренних генераторов. Один генератор выдает постоянную частоту независимо от температуры, а частота второго генератора изменяется в зависимости от температуры корпуса термодатчика. Путем вычислений формируется выходной код, который включает и информацию о знаке температуры.

Для начала работы с термодатчиком управляющий микроконтроллер должен инициализировать его посылкой необходимых команд. Перечень основных команд термодатчика:

F0h - поиск ROM (Search ROM) - команда выдается управляющим микроконтроллером для определения числа и типа термодатчиков, подключенных к одной линии.

33h - чтение ROM (Read ROM) - команда инициализирует термодатчик для генерации в линию идентификационного номера. Эту команду нельзя посылать, если к одной линии связи подключено несколько термодатчиков. Прежде чем подключить несколько датчиков на одну линию, необходимо для каждого датчика определить его личный номер с использованием данной команды.

55h - идентификация ROM (Match ROM) - команда выдается перед 64-битным идентификационным номером и подтверждает обращение именно к этому термодатчику. Все последующие команды будут восприниматься только одним датчиком до команды обнуления линии.

CCh - пропуск ROM (Skip ROM) - команда может использоваться, когда необходимо обратиться ко всем датчикам, расположенным на одной линии, или когда к линии подключен только один датчик. Общей для многих

датчиков может быть команда начала преобразования температуры. При обращении к одному термодатчику команда позволяет упростить программу за счет того, что пропускается громоздкая подпрограмма идентификации кода и вычисления кода четности.

ECh - поиск аварии (Alarm Search) - действие команды аналогично команде «Поиск ROM», но отвечает на нее термодатчик, если измеренная температура выходит за пределы предварительных установок по максимуму и минимуму.

44h - начало преобразования температуры (Convert T) - команда разрешает преобразование температуры и запись результата в блокнот. От подачи этой команды до считывания необходимо выдержать паузу, необходимую для преобразования с установленной точностью.

BEh - чтение блокнота (Read Scratchpad) - в блокноте содержится 8 байт информации (рис.2.7). Если нужна информация только о температуре, то считывается 9 бит. Термодатчик будет выдавать информацию до тех пор, пока управляющий микроконтроллер не выдаст в линию нулевой импульс.

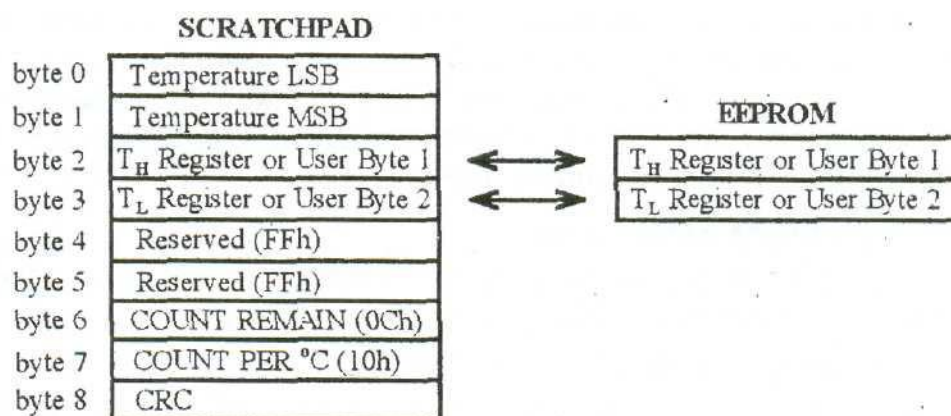


Рисунок 2.5 – Карта памяти термодатчика DS 1820

4Eh - запись в блокнот (Write Scratchpad) - после этой команды управляющий микроконтроллер должен послать два байта для записи в блокнот максимальной T<sub>H</sub> и минимальной T<sub>L</sub> температуры ограничения по максимуму и минимуму. Все 16 бит необходимо передавать непрерывно без обнуления линии.

48H - копирование блокнота (Copy Scratchpad) - после этой команды минимальная (TL) и максимальная (TH) установленные значения температур переписываются в энергонезависимую память (EEPROM). После отключения напряжения питания записанные значения сохраняются в памяти.

B8h - восстановление (Recall B2) - команда необходима для копирования значений температуры из EEPROM в рабочую зону блокнота. При выполнении восстановления термодатчик выдает в линию низкий уровень, а после окончания записи — высокий.

B4h - питание от линии (Read Power Supply) - после этой команды термодатчик переходит к питанию от линии. В составе термодатчика имеется конденсатор, который заряжается от высокого уровня линии.

Перед опросом термодатчика управляющим микроконтроллером необходимо выдержать время, необходимое для заряда конденсатора.

Микроконтроллерный термодатчик тактируется внутренним генератором частотой 4 МГц и рассчитан на работу с управляющим микроконтроллером, который должен тактироваться такой же частотой.

Термодатчик работает на пониженных частотах из-за большого разброса длительности импульсов принятого в интерфейсе формата обмена данными между термодатчиком и управляющим микроконтроллером.

Передача данных по однопроводной шине выполняется импульсами нулевого уровня, но различной длительности. Импульс воспринимается как логическая 1, если его длительность не превышает 15 мкс (рис.2.8). Если длительность импульса больше 15 мкс, то он воспринимается как логический 0.

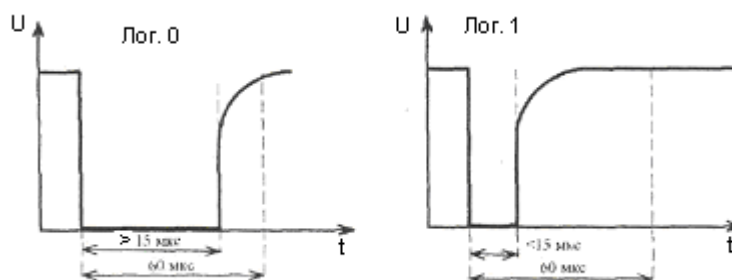


Рисунок 2.6 – Импульсы логического 0 и 1



Длительность одного бита информации принята равной 60 мкс. Отсюда и разброс возможных длительностей импульсов: лог. 0 — 1...15 мкс, лог. 1 — 15...60 мкс. Для того чтобы термодатчик подготовить к приему информации, управляющий микроконтроллер должен послать в линию импульс обнуления (Reset) (рис.2.9).

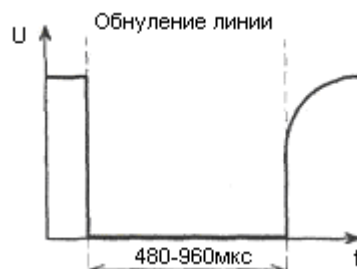


Рисунок 2.7 – Импульс обнуления термодатчика

Импульс обнуления должен иметь длительность 480...960 мкс. В ответ на обнуление линии термодатчик посылает импульс присутствия (Presence). Если в устройстве не предусмотрено отключение датчика, то импульс присутствия для упрощения программы можно не проверять, а заполнить это время (около 100 мкс) паузой. Длительность импульса Presence может быть в пределах 60...240 мкс.

Если к одной линии подключено несколько термодатчиков, то проверка импульса присутствия обязательна. Перед приемом каждого бита информации с датчика, микроконтроллер управления AVR должен послать короткий импульс готовности (запроса) длительностью 1...3 мкс.

Обмен данными и командами начинается с младшего бита. Формат регистров термодатчика показан на (рис.2.8).

Младший регистр температуры LS

26	25	24	23	22	21	20	2 <sup>-1</sup>
----	----	----	----	----	----	----	-----------------

Старший регистр температуры MS (знаковый)

S	S	S	S	S	S	S	S
---	---	---	---	---	---	---	---

Регистры установки ограничения температуры TH, TL

S	26	25	24	23	22	21	20
---	----	----	----	----	----	----	----

Рисунок 2.8 – Формат регистров термодатчика DS 1820

Младший регистр температуры (LS) несет информацию о температуре. В нулевой бит записана информация о десятых долях температуры. Если нулевой бит единичный, то десятые доли равны  $0,5^{\circ}\text{C}$ .

Старший регистр температуры (MS) содержит информацию о знаке температуры, Если значения регистра нулевые, то знак температуры положительный, и наоборот.

Поскольку все биты регистра MS одинаковы, то достаточно считать только младший бит, т.е. длина слова от термодатчика составляет 9 бит = 8 бит (температура LS) + 1 бит знаковый (MS).

Регистры установки ограничения температуры TH, TL несут информацию только о целых значениях температуры. В седьмом бите записывают информацию о знаке температуры. При отрицательной температуре в седьмой бит необходимо записать 1 [9].

В таблице 2.4 приведен пример принимаемых кодов при различных температурах. Для того чтобы получить значение температуры при минусовых температурах, необходимо принятую информацию перевести в дополнительный код. Для перевода в дополнительный код необходимо принятый код инвертировать и прибавить единицу. Например, для значения 1111 1111 получим 0000 0001, т. е.  $0,5^{\circ}\text{C}$ , но со знаком минус.

Таблица 2.4 – Значения кодов термодатчика при различной температуре

Температура	двоичный код	16-й код
$+85,0^{\circ}\text{C}$	0000 0000 1010 1010	00AAh
$+25,0^{\circ}\text{C}$	0000 0000 0011 0010	0032h
$+0,5^{\circ}\text{C}$	0000 0000 0000 0001	0001h
$0^{\circ}\text{C}$	0000 0000 0000 0000	0000h

-0,5° C	1111 1111 11111111	FFFFh
-25,0° C	1111 1111 1100 1110	FFCEh
-55,0° C	1111 1111 1001 0010	FF92h

### 2.2.1 Считывание температуры с датчика

Последовательность действий управляющего микроконтроллера PIC при считывании температуры с одного термодатчика DS 1820:

- 1) послать сигнал обнуления линии (480...960 мкс);
- 2) принять импульс присутствия или заполнить паузой (60...240 мкс);
- 3) послать команду пропуска идентификации CCh;
- 4) послать команду начала преобразования 44h;
- 5) пауза не менее 500 мкс для завершения процесса преобразования;
- 6) обнулить линию;
- 7) послать команду пропуска идентификации CCh;
- 8) послать команду считывания блокнота BEh;
- 9) принять 9 байт;
- 10) проанализировать бит знака;
- 11) если знак отрицательный, то перевести значение температуры в дополнительный код.

### 2.3 Программатор

Из множества программаторов доступных для микроконтроллеров серии AVR был выбран USBasp, ввиду надежности и большого количества положительных отзывов. Программатор работает по интерфейсу USB, загрузка микропрограммы выполняется софтом из среды Windows.

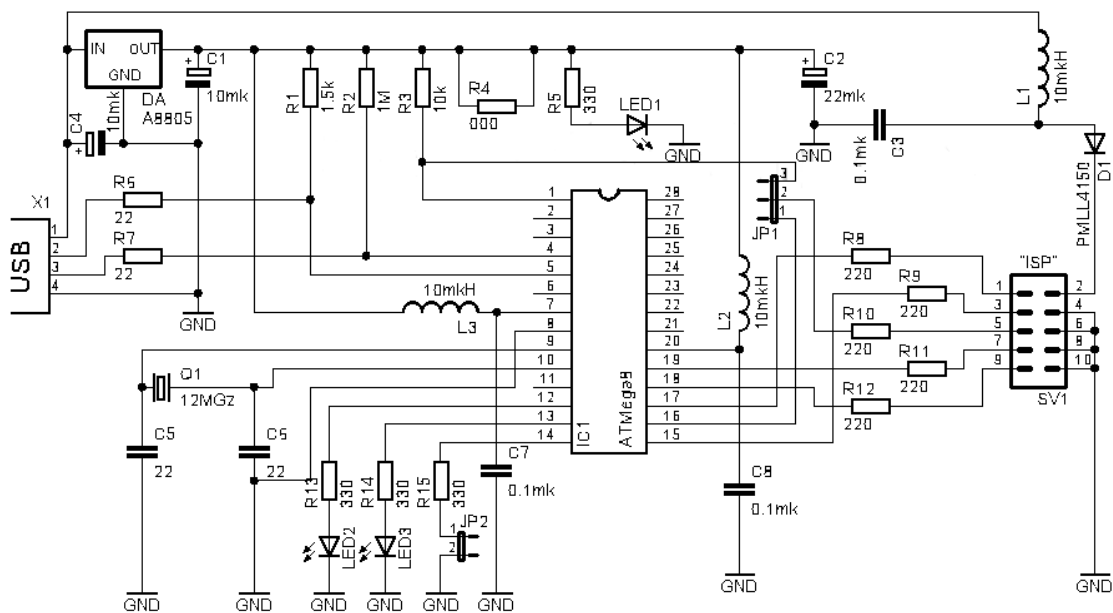


Рисунок 2.9 – Схема программатора USBasp

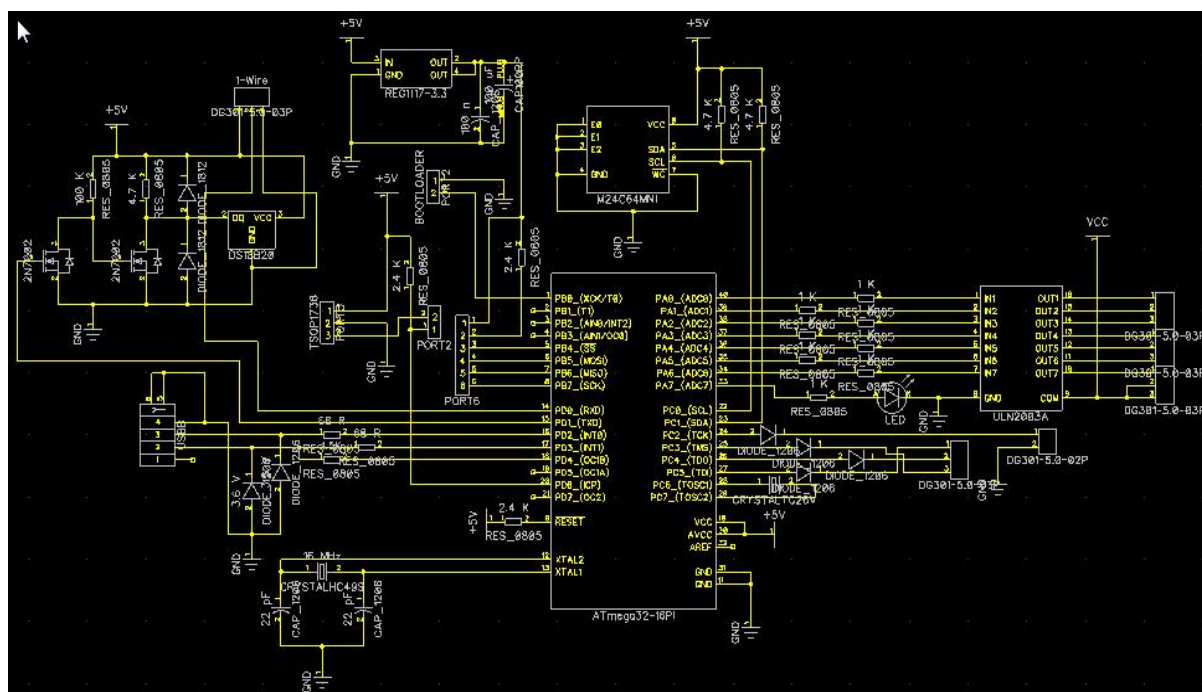


Рисунок 2.10- Схема (основная часть)

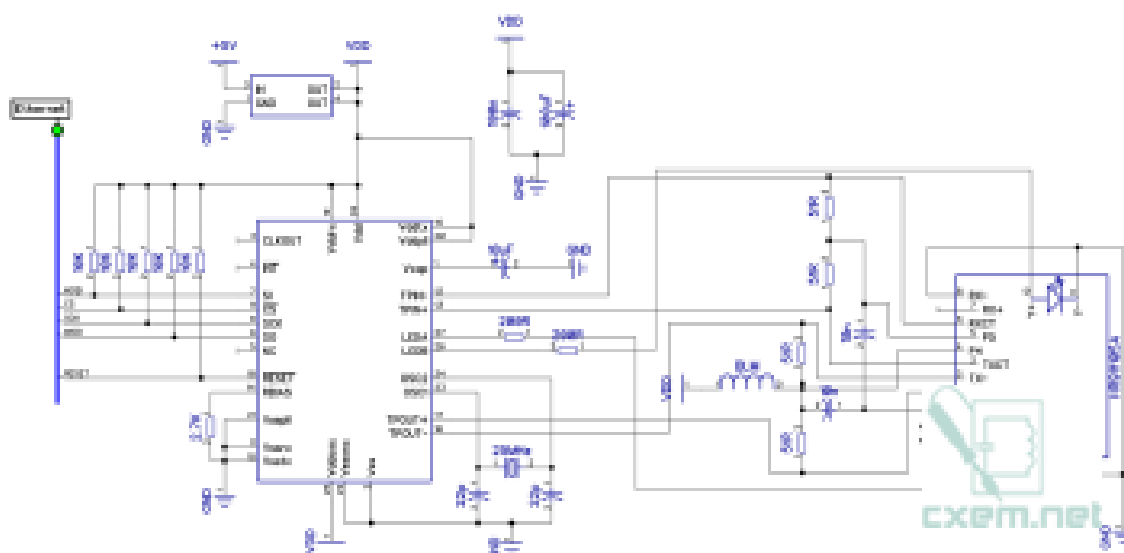


Рисунок 2.11- Схема (сетевой модуль)

Микроконтроллер семейства AVR Atmega32, с памятью AT24C64, в качестве управления выходными каналами используется микросхема ULN2003. Сам сетевой модуль требует напряжение 3,3В, поэтому, нужно установить для него стабилизатор REG1117-3.3. Два транзистора 2N7002 используются для 1-wire, рядом 2 диода защищают линию от напряжения, выше напряжения питания, еще установлен один датчик DS18B20. PORT6 для подключения сетевого модуля. Также разъем USB для подключения к ПК. Перемычка BOOTLOADER предназначена для перевода устройства в режим обновления прошивки.

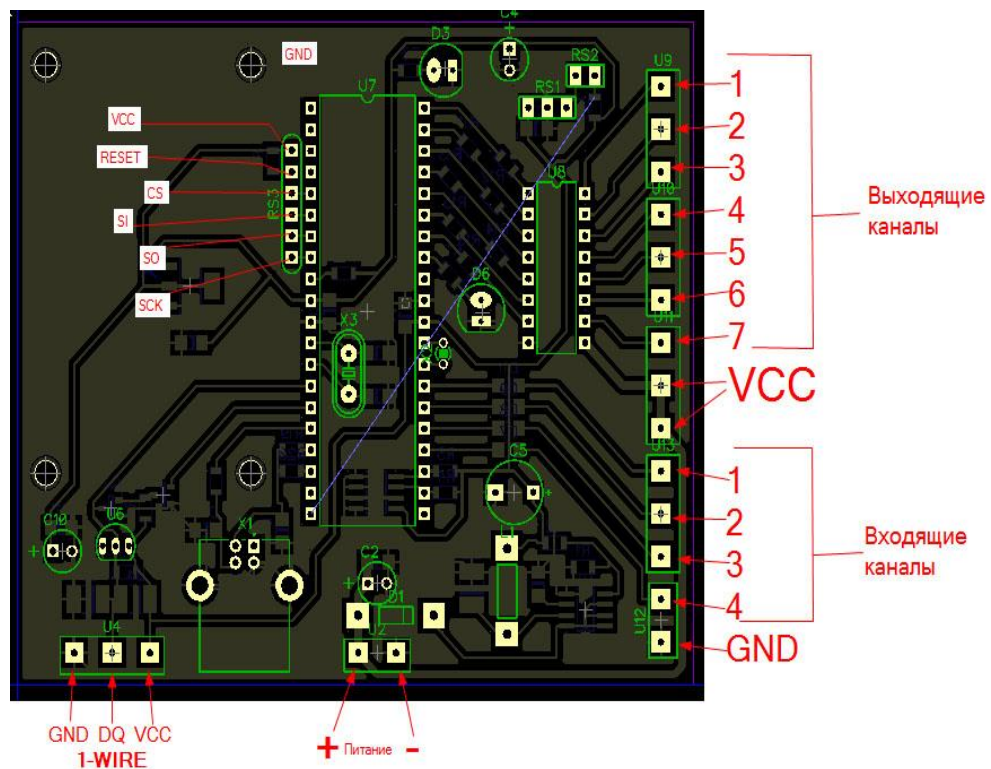


Рисунок 2.12- Основная печатная плата

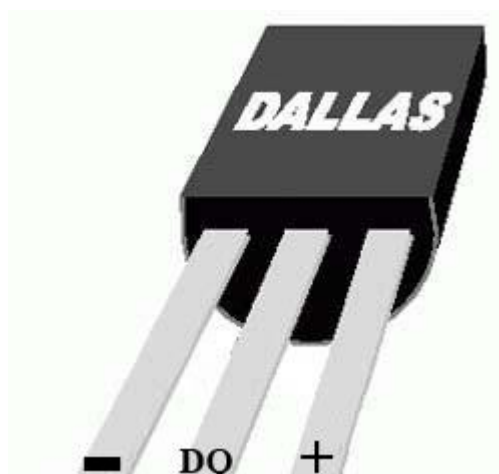


Рисунок 2.13- Распиновка датчика

### 3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

#### 3.1 Среда разработки

Написание программы проводилось в программе Atmel AVR Studio.

Atmel Studio 6 – единая интегрированная среда разработки для микроконтроллеров ARM и AVR, позволяющая значительно упростить процесс разработки и отладки приложений на микроконтроллерах

Корпорация Atmel представила обновленную версию популярной интегрированной среды разработки Atmel Studio 6, которая теперь поддерживает проекты на микроконтроллерах с ядром ARM Cortex-M и проекты на 8/32-разрядных микроконтроллерах AVR. В первый раз инженеры и разработчики, использующие 8/32-разрядные микроконтроллеры AVR и 32-разрядные микроконтроллеры с архитектурой Cortex-M, получают все необходимые для разработки и отладки инструменты в единой интегрированной среде [14].

Интегрированная среда разработки Atmel Studio 6 значительно сократит затраты на создание новых проектов, т.к. она бесплатна, предоставляет отладочные инструменты профессионального уровня и поставляется совместно с Atmel Software Framework (ранее AVR Software Framework). С Atmel Software Framework разработчики получают большое количество проверенных библиотек с открытым исходным кодом, использование которых освобождает от написания многих подпрограмм и кодов для проектов, значительно сокращая время разработки без ущерба для качества решений. Программные структуры включают в себя полный набор драйверов встроенной в микроконтроллер периферии и внешних компонентов, стеки протоколов проводных и беспроводных интерфейсов, библиотеки декодирования звука, прорисовки графики и библиотеки математических функций с фиксированной и плавающей точкой. Для микроконтроллеров Atmel с архитектурой ARM Cortex-M библиотеки обеспечивают полную поддержку Cortex Microcontroller Software Interface Standard (CMSIS) –

программный интерфейс к периферийным модулям ядра и периферии различных микроконтроллеров.

Быстрая разработка устройств в сочетании с появлением новых быстрых микроконтроллеров, включение в них большого объема Flash-памяти подталкивает разработчиков писать код на Си и Си++, а не на ассемблере. Atmel Studio 6 объединяет редактор кода с подсветкой синтаксиса, мастер создания новых проектов, Си/Си++ компилятор, мощный симулятор и инструменты визуализации для всех программаторов и внутрисхемных отладчиков микроконтроллеров ARM Cortex-M и AVR. Симулятор, который пока доступен для AVR микроконтроллеров, не только моделирует CPU и прерывания, но также встроенные в микроконтроллер устройства ввода/вывода, позволяя провести полную разработку приложения без дополнительного аппаратного обеспечения [4].

### 3.2 Листинг программы на языке C

```
#define ENC28J60_SPI_DDR  DDRB
#define ENC28J60_SPI_PORT  PORTB
#define ENC28J60_SPI_CS    (1<<PB4)
#define ENC28J60_SPI_MOSI  (1<<PB5)
#define ENC28J60_SPI_MISO  (1<<PB6)
#define ENC28J60_SPI_SCK   (1<<PB7)

#define ENC28J60_BUFSIZE   0x2000
#define ENC28J60_RXSIZE    0x1A00
#define ENC28J60_MAXFRAME  1500

#define ENC28J60_RXSTART   0
#define ENC28J60_RXEND     (ENC28J60_RXSIZE-1)
#define ENC28J60_TXSTART   ENC28J60_RXSIZE
```



```

#define ENC28J60_BUFEND      (ENC28J60_BUFSIZE-1)

uint16_t enc28j60_rxdpt = 0;

void enc28j60_init(uint8_t *macadr)
{
    // Настраиваем SPI
    ENC28J60_SPI_DDR |=
ENC28J60_SPI_CS|ENC28J60_SPI_MOSI|ENC28J60_SPI_SCK;
    ENC28J60_SPI_DDR &= ~ENC28J60_SPI_MISO;
    enc28j60_release();

    SPCR = (1<<SPE)|(1<<MSTR);
    SPSR |= (1<<SPI2X);

    // Выполняем сброс
    enc28j60_soft_reset();

    // Настраиваем размер буфера для приёма пакетов
    enc28j60_wcr16(ERXST, ENC28J60_RXSTART);
    enc28j60_wcr16(ERXND, ENC28J60_RXEND);

    // Указатель для чтения принятых пакетов
    enc28j60_wcr16(ERXRDPT, ENC28J60_RXSTART);
    enc28j60_rxdpt = ENC28J60_RXSTART;

    // Настраиваем MAC
    enc28j60_wcr(MACON2, 0); // очищаем сброс
    enc28j60_wcr(MACON1, MACON1_TXPAUS|MACON1_RXPAUS| //
включаем управление потоком

```

```

    MACON1_MARXEN); // разрешаем приём данных
enc28j60_wcr(MACON3, MACON3_PADCFG0| // разрешаем паддинг
    MACON3_TXCRCEN| // разрешаем расчёт контрольной суммы
    MACON3_FRMLNEN| // разрешаем контроль длины фреймов
    MACON3_FULDPX); // включаем полный дуплекс
enc28j60_wcr16(MAMXFL, ENC28J60_MAXFRAME); //
устанавливаем максимальный размер фрейма
enc28j60_wcr(MABBIPG, 0x15); // устанавливаем промежуток между
фреймами
enc28j60_wcr(MAIPGL, 0x12);
enc28j60_wcr(MAIPGH, 0x0c);
enc28j60_wcr(MAADR5, macadr[0]); // устанавливаем MAC-адрес
enc28j60_wcr(MAADR4, macadr[1]);
enc28j60_wcr(MAADR3, macadr[2]);
enc28j60_wcr(MAADR2, macadr[3]);
enc28j60_wcr(MAADR1, macadr[4]);
enc28j60_wcr(MAADR0, macadr[5]);

// Настраиваем PHY
enc28j60_write_phy(PHCON1, PHCON1_PDPXMD); // включаем
полный дуплекс
enc28j60_write_phy(PHCON2, PHCON2_HDLDIS); // отключаем
loopback
enc28j60_write_phy(PHLCON, PHLCON_LACFG2| // настраиваем
светодиодики
    PHLCON_LBCFG2|PHLCON_LBCFG1|PHLCON_LBCFG0|
    PHLCON_LFRQ0|PHLCON_STRCH);

// разрешаем приём пакетов
enc28j60_bfs(ECON1, ECON1_RXEN);

```

```

        // "Правильное" значение ERXRDPT
uint16_t enc28j60_rxdpt = 0;

uint16_t enc28j60_recv_packet(uint8_t *buf, uint16_t buflen)
{
    uint16_t len = 0, rxlen, status, temp;

    // Есть ли принятые пакеты?
    if(enc28j60_rcr(EPKTCNT))
    {
        // Считываем заголовок
        enc28j60_wcr16(ERDPT, enc28j60_rxdpt);

        enc28j60_read_buffer((void*)&enc28j60_rxdpt, sizeof(enc28j60_rxdpt));
        enc28j60_read_buffer((void*)&rxlen, sizeof(rxlen));
        enc28j60_read_buffer((void*)&status, sizeof(status));

        // Пакет принят успешно?
        if(status & 0x80)
        {
            // Выбрасываем контрольную сумму
            len = rxlen - 4;

            // Читаем пакет в буфер (если буфера не хватает, пакет обрезается)
            if(len > buflen) len = buflen;
            enc28j60_read_buffer(buf, len);
        }

        // Устанавливаем ERXRDPT на адрес следующего пакета - 1
        temp = (enc28j60_rxdpt - 1) & ENC28J60_BUFEND;
    }
}

```

```

    enc28j60_wcr16(ERXRDPT, temp);

    // Уменьшаем счётчик пакетов
    enc28j60_bfs(ECON2, ECON2_PKTDEC);
}

return len;
}

#include <avr/io.h>
#include <avr/pgmspace.h>
#include "lan.h" // цепляем стек

void udp_packet(eth_frame_t *frame, uint16_t len)
{
    ip_packet_t *ip = (void*)(frame->data);
    udp_packet_t *udp = (void*)(ip->data);

    // обрабатываем пакеты, приходящие на порт 1990
    if(udp->to_port == htons(1990))
    {
        // если нам приходит "preved", ...
        if(!memcmp_P(udp->data, PSTR("preved"), 6))
        {
            // ... то отвечаем "medved"
            memcpy_P(udp->data, PSTR("medved"), 6);
            udp_reply(frame, 6);
        }

        // а если "value_<байт>"

```

```

else if(!memcmp_P(udp->data, PSTR("value_"), 6))
{
    // ... то пишем этот байт в порт
    PORTA = udp->data[6];
}
}
}

```

```

int main()
{
    // настраиваем порт на выход
    DDRA = 0xff;

    // инициализируем стек
    counter_init();
    lan_init();
    sei();

    // ловим пакеты
    while(1)
        lan_poll();

    return 0;
}

```

## 4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

### 4.1 Анализ условий труда

Для размещения всего необходимого оборудования необходимо помещение с площадью не менее 50 м<sup>2</sup>.

Рассматриваемое помещение имеет следующие характеристики:

- Размеры помещения – 8,9 х 6 х 3,2 м;
- число окон - 5;
- количество рабочих мест – 3.

Интерьер:

- потолок покрашен гидроэмульсионной краской;
- стены оклеены бежевыми обоями под покраску;
- пол бетонный, покрыт линолеумом серого цвета толщина 2,4 мм.

План рабочего помещения представлен на рисунке 4.1.

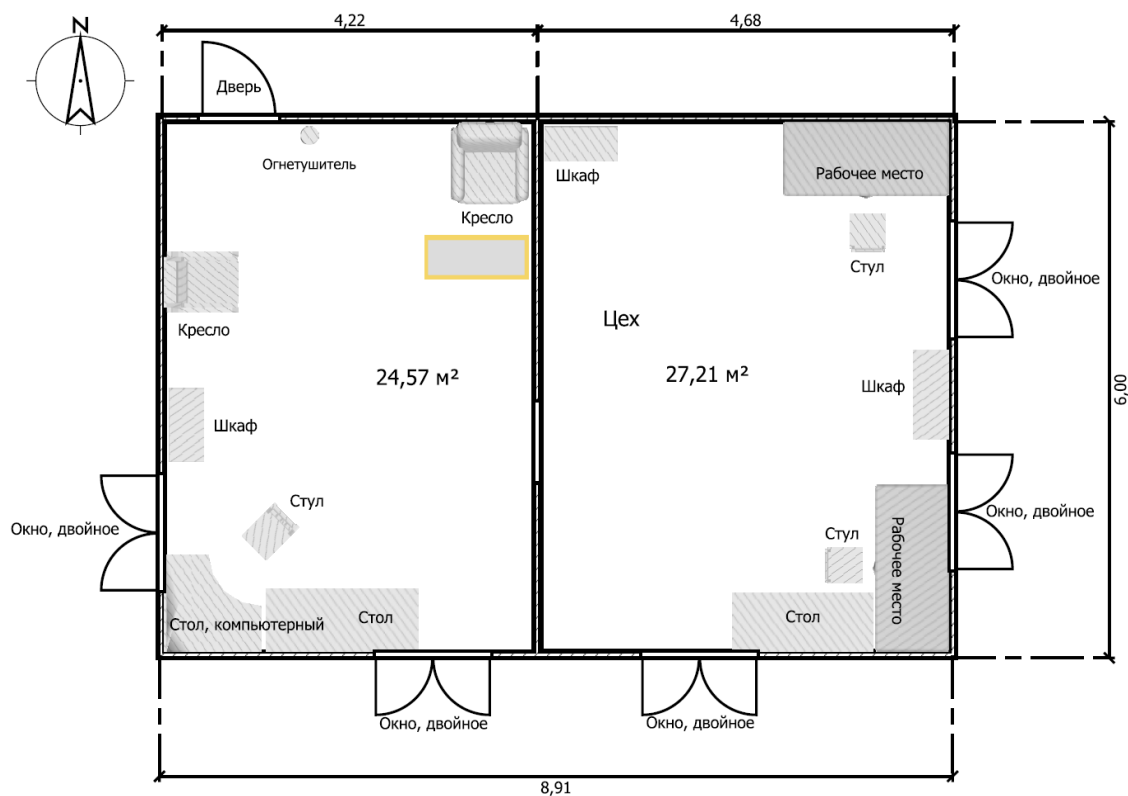


Рисунок 4.1 – План помещения.

Помещение по опасности поражения электрическим током можно отнести к 1 классу, т.е. это помещение без повышенной опасности (сухое, с нормальной температурой воздуха, изолированными полами и малым числом заземленных приборов).

В цехе трудиться три работника. Для работы требуются следующие оборудования:

- Персональный компьютер Lenovo H430 - 2шт.
- Ноутбук HP Pavilion dv7- 2шт.
- Генератор Г4-102 - 2шт.
- Осциллограф Agilent N300
- Паяльные станции Lukey -852D - 2 шт.

Лабораторные источники питания - 2шт.

Все оборудование является источником тепла, а также потенциально опасным оборудованием.

Микроклимат помещения. В летний период времени температура в помещении составляет 24 С, в зимнее 22 С, Относительная влажность воздуха в помещении 40%, скорость движения воздуха 0.1 м/с. Что соответствует ГОСТ 12.3.018-79.

Сборка устройств подразумевает работу с мелкими деталями размером менее 1 мм [11]. Для работы необходимо обеспечить освещение достаточное для такой работы. При пайке могут выделяться вредные вещества от канифоли. Вентиляция должна обеспечить воздухообмен для удаления излишней теплоты и веществ возникающих в процессе пайки. Для этого кондиционер должен быть с режимом принудительной вентиляции.

Рабочую зону нужно разделить на сектора: сборки устройств, проверки готовых, и программирования программного обеспечения. Оборудование, используемое в данных процессах различное, и для наиболее рационального использования их нужно разнести в отдельные места.

Место где будут собираться устройства нужно располагать возле окон, для того что бы обеспечить естественное освещение и при необходимости

проветривать рабочее место. В рабочем месте для тестирования и проверки готовых устройств, стоит громоздкое оборудование и нужно обеспечить свободный проход в случае эвакуации.

Высота рабочего стола 77 см, что соответствует ГОСТ 12.3.013-77 в котором для нерегулируемых рабочих поверхностей стола предусмотрена высота не менее 72,5 см.

Технические характеристики персональных компьютеров. Мониторы жидкокристаллические, рабочая диагональ монитора 30 дюймов (76,2 см). В помещении используется естественное и искусственное освещение. Окна имеют следующие параметры: высота окна – 1.4 м, ширина окна – 2.1м. Под окнами установлены два радиатора отопления по семь секций. На окнах имеются вертикальные жалюзи голубого цвета, что соответствует ГОСТ Р54863.

На горизонтальной поверхности стола в зоне работы общая освещенность должна быть 300 лк (III разряд работы) [8]. Таким образом, объем освещения в помещении удовлетворяет нормам. Соответствует ГОСТ 12.1.046.

Анализ искусственного освещения включает то, что мы имеем лампы ПВЛМ 2х40 и соответствующие светильники, исходя из этого, необходимо рассчитать количество ламп для достижения минимального уровня освещённости. Для этого воспользуемся методом светового потока.

Анализ системы кондиционирования, включает расчет тепловыделения оборудования и прочих теплопоступлений и на основе этих данных выбор и расположение системы кондиционирования.

Тепловыделение персонального компьютера в среднем составляет 200 Вт. У нас используется 2 шт. Паяльные станции выделяют до 50 Вт, Осциллограф до 20 Вт. Прочее оборудование выделяет до 150 Вт, во время работы.



Суммарно получится  $Q = (2 \cdot 200) + (2 \cdot 50) + 20 + 150 = 670 \text{ Вт}$

тепловыделения от оборудования.

Теплопоступления от осветительных приборов

$$Q_{\text{осв}} = F_n \cdot N_{\text{осв}} \cdot \eta, \quad (4.1)$$

где  $\eta = 0.5 - 0.6$  - для люминесцентных ламп,

$N_{\text{осв}}$  – мощность лампы.

$F_n$  – площадь пола.

$$F_n = 6 \cdot 8.9 = 53.28, \text{ м}^2.$$

$$Q_{\text{осв}} = 53.28 \cdot 80 \cdot 0.6 = 2556, \text{ Вт}.$$

Теплопоступление от людей

$$Q_{\text{л}} = q_{\text{явн}} \cdot n, \quad (4.2)$$

Летом  $q_{\text{явн}}(\text{муж}) = 67 \text{ Вт}$

Зимой  $q_{\text{явн}}(\text{муж}) = 82 \text{ Вт}$

$n$  – число людей.

Теплопоступление летом:

$$\text{От мужчин } Q_{\text{л}} = 67 \cdot 3 = 201, \text{ Вт}.$$

Теплопоступление зимой:

$$\text{От мужчин } Q_{\text{л}} = 82 \cdot 3 = 246, \text{ Вт}.$$

Теплопоступления от солнечного излучения через остекление

$$Q_{\text{р}} = m \cdot F(q^I + q^{II})\beta \cdot K_1 \cdot K_2 \quad \text{Вт/м}^2, \quad (4.3)$$

где  $m$  – количество окон в помещении,

$F$  – площадь окна,  $\text{м}^2$

$q^I, q^{II}$  - тепловые потоки от прямой и рассеянной принимаются для расчетного числа суток в зависимости от географической широты.  $q^I = 73$  и  $q^{II} = 77$  для Западной стороны,  $q^I = 288$  и  $q^{II} = 85$  для Южной стороны.

$\beta$  – коэффициент тепло пропуска [12].  $\beta=0.15$

$K_1$  – коэффициент затемнения световых проемов,  $K_1=0.65$

$K_2$  – коэффициент загрязнения остекления для вертикального остекления.  
 $K_2=0.95$

Для западной стороны

$$Q^3 p = 3 \cdot 3 \cdot (73 + 77) \cdot 0.15 \cdot 0.65 \cdot 0.95 = 125.07, \text{ Вт.}$$

Для южной стороны

$$Q^o p = 2 \cdot 3 \cdot (288 + 85) \cdot 0.15 \cdot 0.65 \cdot 0.95 = 207.29, \text{ Вт.}$$

Теплопоступления и теплопотери в результате разности температур

Количество тепла  $Q_{отд} = 0$ , для теплого периода года

Для холодного периода

$$Q_{отд} = V_{пом} \cdot X_0 (t_{Нрасч} - t_{Врасч}), \text{ Вт} \quad (4.4)$$

$V_{пом}$  – объем помещения.

$X_0$  – удельная тепловая характеристика,

$$X_0 = 0.42 \text{ Вт/м}^3 \text{ } ^\circ\text{C}.$$

$t_v$  – температура внутри помещения.  $T_v = 22^\circ\text{C}$ .

$t_n$  – температура снаружи помещения.  $T_n = -31^\circ\text{C}$ .

$$V_{пом} = 6 \cdot 8,9 \cdot 3 = 160 \text{ м}^3.$$

$$Q_{отд} = 160 \cdot 0.42(-31 - 22) = -3561, \text{ Вт.}$$

#### 4.2 Расчет системы кондиционирования цеха

ГОСТ 12.1.005-88 Общие санитарно-гигиенические требования к воздуху рабочей зоны. При обеспечении оптимальных показателей микроклимата температура внутренних поверхностей конструкций, ограждающих рабочую зону, или устройств (экранов и т.п.), а также температура наружных поверхностей технологического оборудования не

должны выходить более чем на 2 °С за пределы оптимальных величин температуры воздуха, установленных для отдельных категорий работ. При температуре поверхностей ограждающих конструкций ниже или выше оптимальных величин температуры воздуха, рабочие места должны быть удалены от них на расстояние менее 1 м.

#### 4.2.1 Общее тепло в помещении

$$Q_{\text{изб}} = Q_{\text{об}} + Q_{\text{осв}} + Q_{\text{л}} + Q_{\text{р}} + Q_{\text{отд}}, \text{ Вт.} \quad (4.5)$$

Летом

$$Q_{\text{изб}} = 670 + 2556 + 201 + 332 + 0 = 3759, \text{ Вт.}$$

Зимой

$$Q_{\text{изб}} = 670 + 2556 + 246 + 332 - 3561 = 243, \text{ Вт.}$$

4.2.2 Количество тепла, необходимого для удаления (или поступления) воздуха в помещение

$$L = \frac{Q_{\text{изб}} \cdot 860}{C \cdot \Delta t \cdot \gamma}, \text{ ккал м}^3/\text{ч}, \quad (4.6)$$

где, С – теплоемкость воздуха.

$$C = 0.24 \text{ ккал/кг } ^\circ\text{C}.$$

$\Delta t$  - выбираем в зависимости тепло напряжённости воздуха  $Q_n$ .

ГОСТ 12.1.014-84 Воздух рабочей зоны.

$$Q_n = \frac{Q_{\text{изб}} \cdot 860}{V_n}, \text{ ккал/м}^3, \quad (4.7)$$

где  $Q_n \leq 20$  ккал/м<sup>3</sup>,  $\Delta t = 6$  °C

$Q_n > 20$  ккал/м<sup>3</sup>,  $\Delta t = 8$  °C.

$\gamma$  - удельная масса приточного воздуха [12].

$$\gamma = 1.206 \text{ кг/м}^3.$$

$$Q_n = \frac{3.759 \cdot 860}{180} = 17.959, \text{ ккал/м}^3, \text{ летом}$$

тогда  $\Delta t = 6 \text{ }^{\circ}\text{C}$

$$L = \frac{3.759 \cdot 860}{0.24 \cdot 6 \cdot 1.206} = 1861.5, \text{ м}^3/\text{ч} \quad \text{летом}$$

$$Q_H = \frac{0.243 \cdot 860}{180} = 1.161, \text{ ккал}/\text{м}^3, \quad \text{зимой}$$

тогда  $\Delta t = 6 \text{ }^{\circ}\text{C}$

$$L = \frac{0.243 \cdot 860}{0.24 \cdot 6 \cdot 1.206} = 120.3, \text{ м}^3/\text{ч} \quad \text{зимой}$$

#### 4.2.3 Кратность воздухообмена

Формула для расчета кратности воздухообмена.

$$n = \frac{L}{V_n}, \quad (4.8)$$

$$n = \frac{1861.5}{180} = 10.34, \text{ 1/ч.} \quad \text{летом}$$

$$n = \frac{120.3}{180} = 0.66, \text{ 1/ч.} \quad \text{зимой}$$

#### 4.2.4 Выбор кондиционера

Заданным требованиям по расходу воздуха и холодопроизводительности соответствует модель Gree GJC24AC-E3RNC2A. Это оконный кондиционер с режимами охлаждения, вентиляции и обогрева, а также снабженным пультом дистанционного управления.

Приведем его параметры:

- Максимальный воздушный поток:  $1000 \text{ м}^3/\text{ч}$
- Мощность в режиме охлаждения: 6600 Вт
- Потребляемая мощность при охлаждении: 2430 Вт
- Режим приточной вентиляции: есть
- Дополнительные режимы: режим вентиляции (без охлаждения и обогрева), самодиагностика неисправностей.
- Габариты: 66x42.8x77 см

План помещения измениться т.к. необходимо поставить кондиционер, размеры которого известны из спецификаций. Над каждым рабочим местом с паяльниками будет установлен кожух с вытяжкой, подключенной к раструбу кондиционера и выводом наружу.

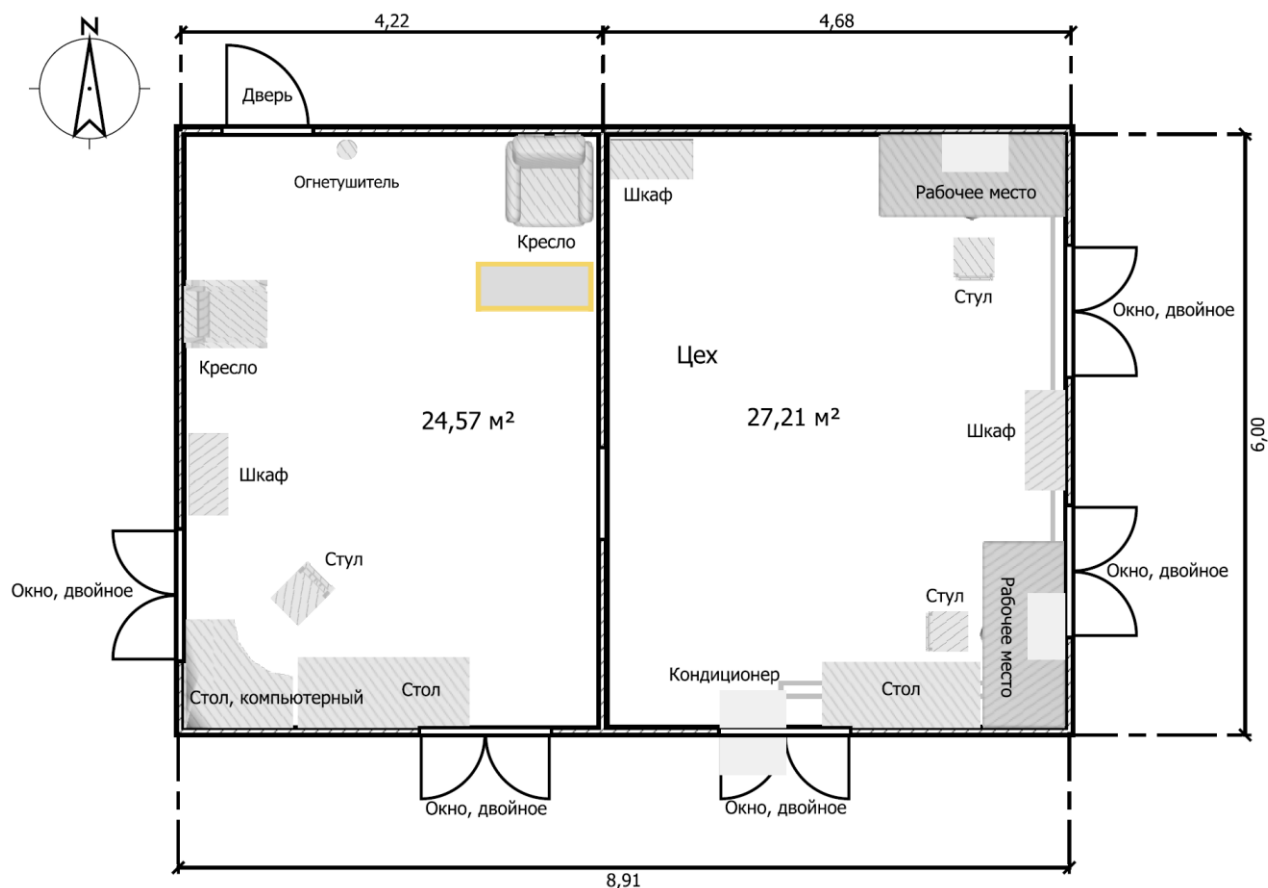


Рисунок 4.2 – План помещения с кондиционером

#### 4.3 Расчет искусственного освещения в помещении

Задачей расчета является определение потребной мощности электрической осветительной установки для создания в производственном помещении заданной освещенности.

Для расчета общего равномерного освещения при горизонтальной рабочей поверхности основным является метод светового потока (коэффициента использования), учитывающий световой поток, отраженный от потолка и стен. Световой поток лампы  $\Phi_{\text{л}}$  (лм) рассчитывается по формуле:

$$\Phi_{\text{л}} = \frac{E_{\text{н}} \cdot S \cdot z \cdot K_3}{N \cdot \eta}, \quad (4.9)$$

где  $E_{\text{н}}$  – нормированная минимальная освещенность, лк (для работ высокой точности не менее 300 лк);

$S$  – площадь освещаемого помещения,  $53,28 \text{ м}^2$ ;

$z$  – коэффициент минимальной освещенности, равный отношению  $\frac{E_{\text{ср}}}{E_{\text{мин}}}$ , значение которого для люминесцентных ламп  $z=1,1$ ;

$K_3$  – коэффициент запаса (для помещений: диспетчерские, пульта операторов, контрольно-измерительные приборы; составляет общее 1.5);

$N$  – число светильников в помещении;

$\eta$  – коэффициент использования светового потока ламп, зависящий от КПД и кривой распределения света светильника, коэффициента отражения потолка  $\rho_{\text{п}}$  и стен  $\rho_{\text{с}}$ , высоты подвески светильников и показателя помещения  $i$ .

#### 4.3.1 Показатель помещения

Показатель помещения  $i$  определяется по формуле:

$$i = \frac{A \cdot B}{H_{\text{р}} \cdot (A + B)}, \quad (4.10)$$

где,  $A$  и  $B$  ширина и длина помещения, м (размеры производственного помещения  $6 \times 8.9$ );

$H_{\text{р}}$  – высота светильника над рабочей поверхностью, (и равняется 3м).

$$i = \frac{6 \cdot 8.9}{3(6 + 8.9)} = 1.19$$

По ГОСТ 2239-79 подбираем стандартную лампу и определяем электрическую мощность всей осветительной системы. В практике допускается отклонение светового потока выбранной лампы от расчетного до  $-10\%$  и  $+20\%$ , в противном случае выбирают другую схему расположения светильников.

Отражения от потолка  $\rho_{\text{п}}$  и стен  $\rho_{\text{с}}$ , примем минимальное значение 30 и 10 соответственно. Выберем светильник ПВЛМ 2х40, для которого значение  $\eta$  при  $i = 1,19$  (округляем до 1,25) равняется 34%. [8]

#### 4.3.2 Выбор ламп производственного помещения

Световой поток одной лампы ЛБ 40 известен и равен 3120 Лм.

Необходимо рассчитать количество ламп для достижения минимальной освещённости в 300 лк. Для этого воспользуемся формулой

#### 4.3.3 Необходимое количество светильников:

$$N = \frac{E_n \cdot K_z \cdot S \cdot z}{\Phi_{\text{л}} \cdot \eta}, \quad (4.11)$$

где  $E_n$  - минимальная освещенность

$K_z$  - коэффициент запаса

$S$ - освещаемая площадь, м<sup>2</sup>

$z$  – Коэффициент неравномерности освещения

Подставим значения в формулу 4.11:

$$N = \frac{300 \cdot 1.5 \cdot 53.28 \cdot 1.1}{2 \cdot 3120 \cdot 0.34} = 12,43$$

Количество светильников округляем до 12

Светильник ПВЛМ 2х40 имеет длину 1325 мм. При длине помещения в 8900 мм, в него может поместиться в один ряд 6 светильников.

Т.к. количество ламп принято 12, то легко распределить 4 светильника в 3 ряда. В каждом светильнике ПВЛМ 2х40 физически две лампы ЛБ мощностью 40 Вт. Схема расположения ламп освещения, их количество приведено на рисунке 4.3.

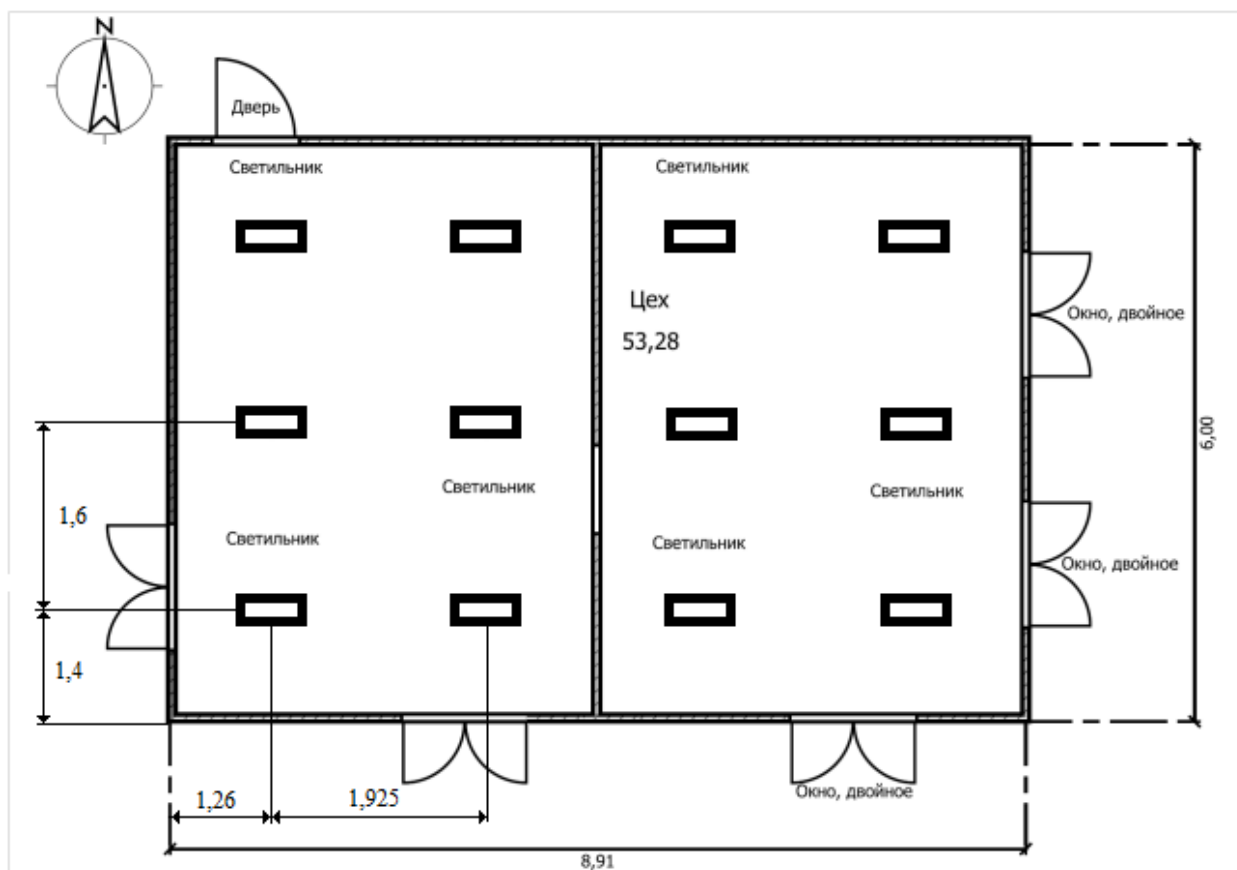


Рисунок 4.3 – Схема размещения светильников

Данное расположение светильников оптимально, свет будет распределяться равномерно по всему помещению, работать в таком помещении будет комфортно.



## 5 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

### 5.1 Определение затрат

В данной работе речь идет о разработке цифровой системы управления удаленными объектами на базе ip-сетей.

Для определения себестоимости разработки нужно найти и включить в нее все затраты. Калькуляция плановой себестоимости проекта составляется по следующим статьям:

- материалы,
- спецоборудование для научных (экспериментальных) работ,
- основная заработная плата,
- дополнительная заработная плата,
- отчисления на социальные нужды,
- расходы на потерю электроэнергии,
- накладные расходы,
- арендная плата.

#### 5.1.1 Расчет трудоемкости и продолжительности работ

Трудоемкость выполнения работ по созданию устройства управления удаленным оборудованием на каждой из стадий определяется в соответствии с (5.2).

Трудоемкость выполнения работ по созданию устройства на базе микроконтроллера по сумме трудоемкости этапов и видов работ, оцениваемых экспертным путем в человеко-днях, и носит вероятностный характер, так как зависит от множества трудно учитываемых факторов.

Трудоемкость каждого вида работ определяется по формуле

$$T_i = \frac{3 \cdot T_{\min} + 2 \cdot T_{\max}}{5}, \quad (5.1)$$

где  $T_{\min}$  – минимально возможная трудоемкость выполнения отдельного вида работ (5.1);

$T_{\max}$  – максимально возможная трудоемкость выполнения отдельного вида работ.

Продолжительность каждого вида работ в календарных днях ( $t_i$ ) определяется в днях по формуле:

$$t_i = \frac{T_i}{\bar{C}_i} \cdot K_{\text{вых}}, \quad (5.2)$$

где  $T_i$  – трудоемкость работ, человек-дней;

$\bar{C}_i$  – численность исполнителей, человек;

$K_{\text{вых}}$  – коэффициент, учитывающий выходные и праздничные дни:

$$K_{\text{вых}} = \frac{K_{\text{кал}}}{K_{\text{раб}}}, \quad (5.3)$$

где  $K_{\text{кал}}$  – число календарных дней;

$K_{\text{раб}}$  – рабочие дни;

Согласно производственному и налоговому календарю на 2014 год, количество рабочих дней составляет 246 дней, количество праздничных дней – 10, таким образом:  $K_{\text{вых}}=1,4$  [3].

Полный список этапов работ по созданию ПО, экспертные оценки и расчетные величины их трудоемкости, а также продолжительность каждого вида работ, рассчитанные по формулам (5.1) и (5.2), представлены в таблице 5.1

Таблица 1.1 - Расчет трудоемкости работ по созданию ПО

№ работы	Стадии разработки	Трудоемкость, чел. дни			Количество работников, чел.	Продолжительность работ, календарные
		$T_{\min}$	$T_{\max}$	$T_i$	$\bar{C}_i$	$t_i$
1	2	3	4	5	6	7

Техническое задание						
Продолжение таблицы 1.1						
1	постановка задачи	1	1	1	1	1
2	сбор материалов и анализ существующих разработок	1	2	1	1	2
3	подбор литературы	2	5	5	1	6
4	определение требований к системе	2	3	2	1	2
5	определение стадий, этапов и сроков разработки электронной библиотеки	2	3	2	1	2
Эскизный проект						
6	анализ программных средств схожей тематики	3	5	5	1	5
7	разработка функциональной схемы программы	3	5	4	1	5
8	разработка структуры программы управления по подсистемам	3	5	5	1	6
9	документирование	1	3	2	1	1
Технический проект						
10	определение требований к программе управления	1	2	2	1	3
11	выбор инструментальных средств	1	1	1	1	1
12	определение свойств и требований к аппаратному обеспечению	2	3	2	1	3
Рабочий проект						
13	программирование	5	15	15	1	25
14	тестирование и отладка программы управления	2	5	5	1	8
15	согласование и утверждение работоспособности системы	2	3	2	1	3

## Внедрение

Продолжение таблицы 5.1

16	опытная эксплуатация	7	8	7	1	10
17	анализ данных, полученных в результате эксплуатации	3	3	3	1	4
18	корректировка технической документации по результатам	2	3	2	1	3
	Общая трудоемкость разработки	-	-	66	-	90

Таким образом, общая продолжительность проведения единовременных работ составит 90 рабочих дней, а трудоемкость 66. при последовательном выполнении всех перечисленных в таблице 1.1 этапов работы. Для последующего производства устройства данная трудоемкость уже учитываться не будет, т.к. разработка требует разового вложения труда. Необходимо лишь учитывать затраты на сборку и тестирование данного устройства.

### 5.1.2 Расчет затрат на разработку программного продукта

Основная заработная плата исполнителей определяется по формуле:

$$C_{осн} = C_{ср} \cdot T, \quad (5.4)$$

где  $C_{осн}$  – заработная плата исполнителей (тг.);

$C_{ср}$  – средняя дневная оплата труда работника организации-разработчика программного продукта (5000 тг./чел.дн.);

$T$  – трудоемкость разработки программного продукта (чел.дн.).

$$C_{осн} = 5000 \cdot 66 = 330000 \text{ тг.}$$

Оклад за месяц усреднено за 22 рабочих дня.

Таблица 5.2 – Расчет основной заработной платы

Исполнитель	Оклад, тг/мес.	Оклад, тг./дн.	Трудоемкость, чел.-дн.	Сумма Тг.
Разработчик	110000	5000	66	330000
Основная заработная плата исполнителя $З_{осн}$				330000

Дополнительная заработная плата исполнителей, учитывающая потери времени на отпуска и болезни (принимается в среднем 15% от основной заработной платы);

$$З_{доп} = 0,15 \cdot З_{осн}, \quad (5.5)$$

$$З_{доп} = 0,15 \cdot 330000 = 49500 \text{ тг.}$$

Отчисления на социальные нужды состоят из единого социального налога. Ставка налога рассчитывается, исходя из зарплаты сотрудника.

- Пенсионный фонд Республики Казахстан — 10 %

- Социальный налог—11%.

Фонд оплаты труда складывается из основной и дополнительной заработной платы:

$$ФОТ = З_{осн} + З_{доп} = 330000 + 49500 = 379500 \text{ тг.}$$

Пенсионные отчисления

$$С_{п} = ФОТ \cdot 10\% = 37950 \text{ тг.}$$

Социальный налог высчитывается после пенсионных отчислений

$$С_{соц} = (ФОТ - С_{п}) \cdot 11\% = (379500 - 37950) \cdot 11\% = 37570 \text{ тг.}, \quad (5.6)$$

В статью «Материалы» включаются затраты на основные и вспомогательные материалы, покупные полуфабрикаты и комплектующие изделия, необходимые для выполнения конкретной разработки [7]. Затраты по этой статье определяются по действующим ценам с учетом транспортно-заготовительных расходов, устанавливаемых в процентах от стоимости материалов, комплектующих и т.п.

Таблица 5.3 – Материальные расходы

Материалы и другие материальные ресурсы	Единица измерения	Количество	Цена за единицу тг.	Сумма тг.
Гетинакс	Листов шт.	3	500	1500
Резисторы	Шт.	40	20	800
Конденсаторы	Шт.	50	30	1500
Припой	Гр.	120	1500	1500
Канифоль	Шт.	1	200	200
Кислота паяльная	Шт.	1	400	400
Транзисторы	Шт.	15	50	750
Разъёмы	Шт.	10	50	500
Итого				7150

Расходы на транспортировку составят 10% от итоговой суммы.

$$C_m = 7150 + 715 = 7865 \text{ тг.}$$

Таблица 5.4 – Расходы на полуфабрикаты и комплектующие

Наименование	Тип	Количество	Цена за единицу тг.	Сумма тг.
DS18B20	датчик	1	1300	1300
E10801	сетевой адаптер	1	1500	1500
ATmega32 AU	микроконтроллер	1	1000	1000
Кнопки		4	50	200
Батарейки		4	100	400
Корпус		1	1000	1000
Кабели		1	100	100
Итого				5700

Расходы на транспортировку составят 13% от итоговой суммы.

$$C_k = 5700 + 741 = 6441 \text{ тг.}$$

Таблица 5.5 – Расходы на специальное оборудование

Оборудование	Цена за единицу т.	Количество	Сумма тг.
Паяльная станция	7000	1	7000
Штангенциркуль	3500	1	3500
ЛС метр	8000	1	8000
Мультиметр	3500	1	3500
Программатор	25000	1	25000
Персональный компьютер	150000	1	150000
Программное обеспечение	100000	1	100000
Итого			297000

### 5.1.3 Амортизация

Амортизационные отчисления  $C_a$  определяются по формуле:

$$C_a = (H \cdot \Phi) / 100\%, \quad (5.7)$$

где  $H$  — среднегодовая норма амортизации, %

$\Phi$  — первоначальная стоимость оборудования, тенге.

Среднегодовая норма амортизации составляет 15 процентов от балансовой стоимости оборудования, т.к. срок эксплуатации в среднем превышает 6 лет [7].

$$C_a = (15 \cdot 297000) / 100 = 44550 \text{ тг.}$$

Т.к. проект изначально разрабатывался как научно исследовательский, время, в течение которого будет использоваться оборудование 90 дней, округляя 3 месяца.

$$C_a = 44550/3 = 14850 \text{ тг.}$$

Стоимость электроэнергии  $C_э$  определяется по формуле:

$$C_э = \Delta A \cdot W, \quad (5.8)$$

где  $\Delta A$  - расходы энергии в день, кВт/ч

$W$  — цена 1 кВт/ч энергии, тенге.

$$C_э = 66 \cdot 24 \cdot 0.5 \text{ кВт/ч} \cdot 15 = 11880 \text{ тг.}$$

Расходы на аренду помещения зависят от текущих цен на недвижимость в качестве арендной платы в г. Алматы возьмём 55000 тенге в месяц. Данная цена включает коммунальные услуги.

Формула для расчета в предполагаемый период разработки 3 месяца:

$$C_{арн} = 4 \cdot \Pi_{арн} = 55000 \cdot 3 = 165000 \text{ тг}, \quad (5.9)$$

#### 5.1.4 Общая сумма затрат разработки

$$C = \Phi OT + C_{соц} + C_a + C_m + C_k + C_э + C_{арн} + H, \quad (5.10)$$

где  $\Phi OT$  – фонд оплаты труда

$C_{соц}$  – социальный налог

$C_a$  – амортизация основных фондов

$C_m$  – расходы на материалы

$C_k$  – расходы на комплектующие

$C_э$  – расходы на электроэнергию

$C_{арн}$  – арендная плата

$H$  – накладные расходы 25% от суммы всех затрат

$$C = 379500 + 37570 + 14850 + 7865 + 6441 + 11880 + 165000 + 155776 = \\ = 778882 \text{ тг.}$$



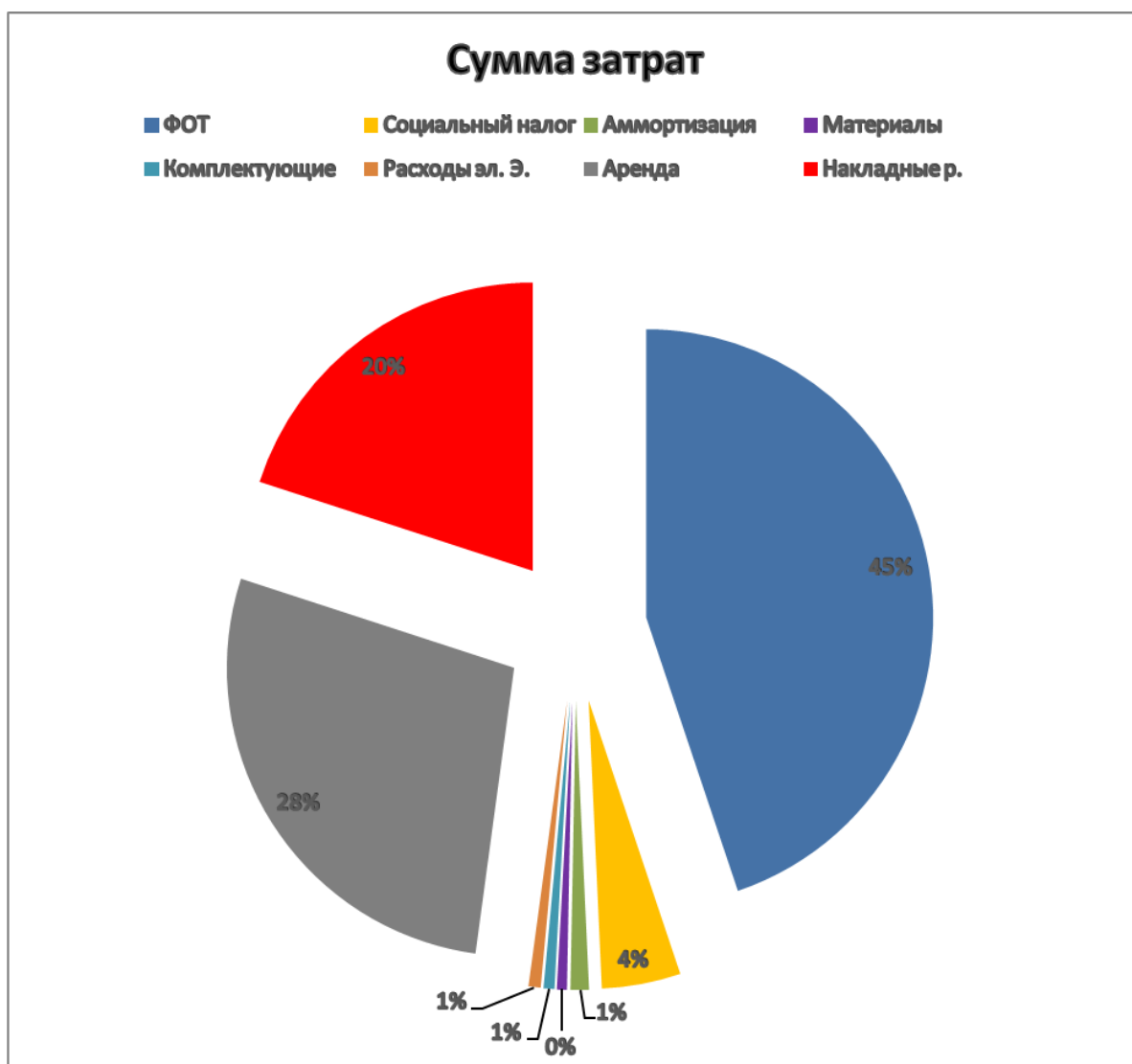


Рисунок 5.1 – Общая сумма затрат в процентном соотношении

## 5.2 Расчет себестоимости

Предполагаемый объем производства 40 шт. в месяц. Расчет себестоимости за месяц будет состоять из расходов на аренду, ФОТ, социальный налог, расходов на материалы и комплектующие на ед. продукции и накладных расходов.

ФОТ за месяц 40000 тг. Социальный налог 11%, будет рассчитываться после вычета, 10% на пенсионные отчисления. Социальный налог равен 4455 тг. Накладные расходы примем 30% от суммы всех затрат. Остальные расходы рассчитанные ранее сохраняются.

$$C = \frac{\Phi OT + C_{\text{соц}} + C_a + C_{\text{э}} + C_{\text{арн}} + H + 40 * (C_m + C_k)}{40}, \quad (5.11)$$

$$C = \frac{40000 + 4455 + 44550 + 1415 + 55000 + 34993 + 40 * (7865 + 6441)}{40} = 18816 \text{ тг.}$$

### 5.2.1 Первоначальная цена

Для определения первоначальной цены нужно определиться с уровнем рентабельности. Для новой разработки не нужно задавать слишком большой уровень рентабельности, постепенно повышать его следует лишь после старта продаж. Зададимся уровнем рентабельности в 30% [3].

$$Ц_n = C(1 + P/100), \quad (5.12)$$

где P - уровень рентабельности в процентах

$$Ц_n = 18816 + 5644 = 24460 \text{ тг.}$$

С учетом НДС который на сегодня равен 12% цена вырастет.

$$Ц_p = Ц_n + \text{НДС}$$

$$Ц_p = 24460 + 2935 = 27395 \text{ тг.}$$

### 5.2.2 Срок окупаемости

Прибыль с каждой проданной единицы равна 5644 тг. Корпоративный налог равный 20%, вычитается из суммы прибыли, налог составляет 1130 тг. если в месяц будут продаваться предполагаемые 40 штук, то прибыль составит 180560 тг. в месяц. Ранее рассчитанные расходы на разработку составляют 778882 тг. следовательно, разработка окупится через 4 месяца, далее она будет окупать свое производство и давать доход.

$$T_{\text{окп}} = П / C, \quad (5.13)$$

$$П = (5644 - 1130) * 40 = 180560 \text{ тг.}$$

$$T_{\text{окп}} = 180560 / 778882 = 4 \text{ месяца.}$$

## ЗАКЛЮЧЕНИЕ

В результате выполнения дипломного проекта, была разработана система управления удаленными объектами на базе ip-сетей. Система выполнена на основе микроконтроллера Atmega32, а также на сетевом модуле enc28j60 и датчика DS18B20. Разработана прошивка программы которая позволяет системе подключаться к ПК через локальную сеть или интернет. Техническое задание проекта полностью выполнено.

Во второй главе были рассмотрены основные элементы системы, их конструктивная характеристика и электрическая схема устройств. Особенности проектирования и реализации опытного образца с работающими основными функциями с предоставленными эскизами печатных плат.

Третья глава посвящена программному обеспечению и написанию корректно работающего кода прошивки микроконтроллера на языке C с возможностью модернизации и отладки в программе Proteus.

В четвертой главе были рассмотрены вопросы охраны труда, а именно: анализ условий труда, выявление опасных и вредных производственных факторов, анализ существующего искусственного освещения и расчет системы кондиционирования помещения в котором будет производиться прибор, а также тестирование готовых устройств.

В пятой главе произведен расчет себестоимости проекта, а также экономический эффект от внедрения новой системы. Также рассчитан приблизительный срок окупаемости при промышленном масштабе производства. Выгоду от внедрения новой системы, можно оценить по следующим критериям: Низкая цена устройств, простота внедрения, возможность изменения конфигурации, точность измерения достаточная для большинства технологических процессов.

## СПИСОК ЛИТЕРАТУРЫ

1. Atmega8A Datasheet, [http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A\\_datasheet.pdf](http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A_datasheet.pdf)
2. Баранов В.Н. Применение микроконтроллеров AVR. Схемы, алгоритмы, программы – М.: Додэка, 2004. – 287 с.
3. Базылов К.Б., Алибаева С.А., Бабич А.А. Выпускная работа бакалавров. Экономический раздел. – А.: АИЭС, 2008. – 20 с.
4. Белов А.В. Разработка устройств на микроконтроллерах AVR. – М.: Наука и Техника, 2012. – 530 с.
5. Геращенко О.А. Гордов А.Н. Температурные измерения. Справочник./; АН УССР Ин-т проблем энергосбережения. – К.: Наукова думка, 1989. – 704 с.
6. Голубцов М.С., Кириченкова А.В. Микроконтроллеры AVR: от простого к сложному. – М.: СОЛОН– Пресс, 2006. – 304с
7. Грязнова А.Г. Юданова А.Ю. Микроэкономика: практический подход. –М.: Финансы и кредит, 2007. – 653 с.
8. Дюсебаев М.К. Безопасность жизнедеятельности: методические указания к выполнению раздела дипломных проектов – А.: АИЭС, 2003. – 27 с.
9. DS18B20 Datasheet, <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
10. Евстифеев А.В. Микроконтроллеры AVR семейства Mega. – М.: Додэка, 2008. – 558 с.
11. Князевский Б.А. Охрана труда. – М.: Высшая школа, 2002. – 365 с.
12. Хакимжанов Т.Е. Безопасность жизнедеятельности. Расчет аспирационных систем. – А.: АИЭС, 2002. – 29 с.
13. Хоровиц П., Хилл У. Искусство схемотехники. – М.: Мир, 1993. – 418 с.
14. Шпак Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров. – М.: МК-Пресс, 2011. – 546 с.
15. TCP/IP, [http://network.xsp.ru/3\\_5.php](http://network.xsp.ru/3_5.php)