

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерная технология

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_  
(Ф.И.О., ученая степень, звание)  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Проектирование и реализации системы управления по технологии "Умный дом"

Специальность 5B0700400 - Вычислительная техника и программное обеспечение

Выполнил (а) Сабунов М. М. BT-10-3  
(Фамилия и инициалы) группа

Научный руководитель старший преподаватель Зуба Е. А.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Ермеева З. Д., с.н.с. преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
Ермеева « 27 » 05 2014 г.  
(подпись)

по безопасности жизнедеятельности:

Приходько И. Г., д.т.н. профессор  
(Фамилия и инициалы, ученая степень, звание)  
Приходько « 21 » 05 2014 г.  
(подпись)

по применению вычислительной техники:

с.н.с. пр.п. К.ТТ Зуба Е. А.  
(Фамилия и инициалы, ученая степень, звание)  
Зуба « 12 » июня 2014 г.  
(подпись)

\_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

Нормоконтролер: с.н.с. пр.п. К.ТТ Зуба Е. А.  
(Фамилия и инициалы, ученая степень, звание)  
Зуба « 12 » июня 2014 г.  
(подпись)

Рецензент: \_\_\_\_\_  
(Фамилия и инициалы, ученая степень, звание)  
« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий  
Специальность Вычислительная техника и программное обеспечение  
Кафедра Компьютерные технологии

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Сабиров Ильяс Каратович  
(фамилия, имя, отчество)

Тема проекта Проектирование и реализация системы управления по технологии "Умный дом"

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

Срок сдачи законченной работы «14» июня 2014 г.

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

Разработка и реализация системы управления по технологии "Умный дом" по заказу компании NT Company

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

1. Разработка программного обеспечения для системы
2. Использование материалов и комплектующие для макета
3. Сбор материалов и каркаса
4. Проведение работ всей системы управления
5. Используемой модуль для передачи сигнала на микроконтроллер.

Перечень графического материала (с точным указанием обязательных чертежей)

1. Видя подписании в пульты ВУ
2. Видя сигналов пульты ВУ при удержании кнопки
3. Видя передатчика RF-модуля
4. Видя приемника RF-модуля
5. Микроконтроллер Atmel AVR
6. Схема устройства микроконтроллера AVR
7. Общая схема архитектуры микроконтроллера AVR
8. Пример ПК-диспетч.

Рекомендуемая основная литература

1. Е. А. Теев. Умный дом своими руками. Строим интеллектуальную цифровую систему в своей квартире - СПб - 2008 - 216 с.
2. В. Н. Гололобов. Умный дом своими руками. - М. - КТ пресс 2007. - 416 с.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БЖКД	Дроздов М. Г.	11.04 - 21.05.14	<i>Дроздов</i>
Экономическая	Зрешнева З. Д.	15.04 - 21.05.14	<i>Зрешнева</i>
Основная часть	Зуба Е. А.	15.05 - 12.06.14	<i>Зуба</i>
Контроль	Зуба Е. А.	12.06.14	<i>Зуба</i>

**Г Р А Ф И К**  
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1.	Сбор материалов для макета	20.02 - 2.03.14	
2.	Сбор каркаса макета и установка на него всех материалов	3.03 - 20.03.14	
3.	Монтаж и присоединение проводов к устройствам	21.03. - 27.03.14 30.03.14 - 15.04.14	
4.	Программное обеспечение и аппаратные средства для программирования микроконтроллеров	30.03 - 15.04.14	
5.	Отладка программы	15.04. - 22.04.14	
6.	Безопасность жизнедеятельности	22.04 - 28.04.14	
7.	Выпуск план	30.04. - 10.05.14	

Дата выдачи задания « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

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

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

Задание принял к исполнению студент \_\_\_\_\_  
(подпись) (Фамилия и инициалы)

## **Аннотация**

В работе исследована область реализации системы управления по технологии «Умный дом».

Показан сам принцип работы системы и устройств непосредственно при помощи пульта дистанционного управления, описано взаимодействие аппаратуры с микроконтроллером. В работе использовано программное обеспечение типа САПР (автоматизированного проектирования) и интегрированная среда разработки для программирования микроконтроллера.

В заключительных главах рассматриваются вопросы безопасности жизнедеятельности, приводится технико-экономическое обоснование, рассчитывается стоимость разработки проекта.

## **Андатпа**

Зерттеу жұмысында «Ақылды үй» технологиясы бойынша басқару жүйесін іске асыру саласы зерттелген.

Тікелей қашықтан басқару пультінің көмегімен жүйе мен құрылғылардың жұмыс қағидаттары көрсетілген. Сонымен қатар, жұмыста жүйенің әрбір құрылғысы қарастырылған және сипаттама берілген: шамшырақ, желдеткіш, розетка және моторлы жалюзи және олардың микроконтроллермен өзара әрекеттестігі. Жұмыста САПР (автоматтандырылған жобалау) типіндегі бағдарламалық жасақтама және микроконтроллерді бағдарламалауға арналған әзірлеменің ықпалдастырылған ортасы пайдаланылған.

Сонымен қатар, екі қорытынды бөлімдерде тіршілік әрекетінің қауіпсіздігі мәселелері қарастырылған, техника-экономикалық негіздеме берілген, жобаны әзірлеу құны есептелген.

## **Abstract**

The sphere of implementation of management system by technology “Clever house” was investigated in work.

The functioning principle of the system and devices itself is shown directly with the help of remote control panel. Each device of the system is also considered and shown in work: a light, a ventilator, sockets and a Venetian blind with engine. Their interaction with microcontroller. Such software as computer-aided design system and integrated development environment for programming of microcontroller were applied in work.

Also apart from this, life safety issues are considered, feasibility study is given and development cost of project is calculated in two concluding chapters.

## Содержание

Введение.....	9
1 Аналитика .....	10
1.1 Анализ предметной области .....	10
1.2 Комплектующие макета «Умный дом» .....	10
1.2.1 Пульт ДУ .....	10
1.2.2 Микроконтроллер AVR.....	16
1.2.3 Каркас макета .....	21
1.2.4 LCD-дисплей.....	24
1.2.5 Электронное реле.....	26
Вывод .....	27
2 Технологии и программное обеспечение .....	28
2.1 Технология взаимодействия пульта с микроконтроллером .....	22
2.2 Технология работы микроконтроллера с LCD-дисплеем.....	39
2.3 Технология подсоединения сторонних устройств к плате.....	32
2.4 Windows 7 .....	33
2.5 ProteusPro .....	34
2.6 CodeVisionAVR .....	35
2.7 Photoshop.....	36
Вывод .....	37
3 Программная реализация .....	38
3.1 Схемотехническое моделирование .....	38
3.2 Компонентная программная реализация .....	38
3.3 Библиотеки .....	41
3.4 Печатная плата и её изготовление .....	42
3.5 Бытовые устройства на макете.....	47
3.6 Общий концепт макета.....	50
3.7 Представление кода .....	50
Вывод .....	51
4 Технико-экономическое обоснование проекта.....	52
4.1 Описание работы и обоснование необходимости.....	52
4.2 Трудовые ресурсы, используемые в работе.....	52
4.3 Расчет стоимости работы по проектированию и разработке.....	53
4.4 Расчет затрат на амортизацию.....	59
4.5 Цена программного продукта.....	62
5 Безопасность жизнедеятельности.....	64
5.1 Анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал.....	64
5.2 Расчет пожарной безопасности проектируемого объекта .....	69
5.3 Расчет уровня шума.....	72
Заключение .....	73
Список использованной литературы .....	74
Приложение А .....	75

## Введение

Цель дипломного проекта – создание специальной системы автоматизированного жилого комплекса, в которой можно будет управлять работой некоторых устройств (освещение, жалюзи, энергоснабжение и т.д.), при помощи установленных контроллеров и исполнительных устройств. При этом система управления должна согласованно управлять всей работой инженерных установок и электрооборудованием. Также необходимо провести анализ имеющихся методов конструкции систем автоматизации, выбрать подходящее программное обеспечение, создать оптимальный набор компонентов системы, придумать своеобразные алгоритмы, которые произведут управление работой устройств и реализуют их в программном обеспечении для функционирования контроллера.

«Умный дом» – это жилой дом современного типа, созданный для комфортного проживания людей с помощью современных высокотехнологичных компонентов. Сама дефиниция «Умный дом» была придумана институтом интеллектуального здания в Вашингтоне, в 1970-х годах: «здание, обеспечивающее продуктивное и эффективное использование рабочего пространства». Также любое здание будь то административное, производственное или жилое состоит из определенной вариативности подсистем, отвечающих за осуществление определенных функций, которые определяют свои задачи в процессе функционирования этого здания. При усложнении этих систем и увеличения количества, выполняемых ими функций, управление ими становилось все сложнее. Также стремительно растут расходы на содержание обслуживающего персонала, ремонт и обслуживание этих подсистем. Впервые эти проблемы встали при эксплуатации больших административных и производственных комплексов. Современное здание такого типа - это город в миниатюре. Фактически в нем действуют все службы, являвшиеся ранее непременными атрибутами городского хозяйства. В таких зданиях обычно существует административная служба или администратор, которые используют и обслуживают эту систему практически круглосуточно.

Важным отличием «Умного дома» от его остальных способов организации жизненного помещения является то, что это самая развивающаяся концепция взаимодействия человека с помещением, когда человек единственной командой определяет желаемую обстановку, в то время автоматика в соответствии с внутренними и внешними условиями определяет и отлаживает принцип работы всех интегрированных систем и устройств. В помещении, оборудованном системой "Умный дом" можно единственным нажатием на установке (или пульте ДУ, сенсорной панели и т. д.) выбрать один из разновидностей возможности. Всё по-своему настроится на работу нужной из систем в соответствии с вашим желанием, временем суток, расположением в помещении, погодой, освещённостью и т. п.

## **1 Аналитика**

### **1.1 Анализ предметной области**

Системы автоматического управления и контроля за функционированием бытовых приборов называются системами «Умный дом». «Умный дом» предусматривает объединения всех инженерных систем и устройств в одну единую сеть с установкой диспетчерского пульта дистанционного управления для контроля и мониторинга параметрами их работы.

Также если происходят или достигаются параметры критических значений, происходит оповещение владельца об этом посредством вывода информации на ЖК-дисплей. Система «Умный дом» позволяет полностью контролировать освещение в помещении, задавать нужную температуру и проводить проветривание, включать и выключать розетки и приборы для питания, производить мониторинг траты энергии, воды и позволяет сэкономить немалое количество денег. То есть «Умный дом» предусматривает рациональное использование всех подсистем помещения в зависимости от заложенной программы, времени суток, наличия хозяина, либо его отсутствия. При помощи системы предоставляется возможность удаленного управления и контроль над помещением.

«Умные дома» в РК на протяжении некоторого периода набирают популярность. Услуги автоматизации и контроля над приборами в помещениях предоставляют такие компании, как «Digis», «AdvaconSystems», «BuildingControl», «SmartHouseCompany». Некоторые из них предоставляют свои услуги также для бизнеса, образования, рекламы или конференц системы.

### **1.2 Комплектующие макета «Умный дом»**

#### **1.2.1 Пульт ДУ**

Пульт дистанционного управления (ДУ) широко применяется во многих электронных устройствах: аудио и видеотехника, акустические устройства, системы освещения, кондиционеры и т.п. Его основной принцип действия основан на транспортировке команд управления при помощи модулированного потока излучения от пульта управления к приемнику (приемный модуль).

В создании устройства с пультом ДУ можно воспользоваться имеющимся на рынке готовым приемо-передающим модулем, комплектами микросхем для их изготовления или произвести разработку собственной установки.

В случаях когда система или устройство разрабатывается на основе микроконтроллера, можно воспользоваться готовым пультом, например, от ТВ, а приемный модуль попробовать реализовать на том микроконтроллере, подключив лишь к нему фото приемный модуль. Такой подход осуществляет



получение желаемого результата с минимальными расходами времени и средств. В разработке программного обеспечения декодируемого сигнала ДУ необходимо лишь знать определенный протокол транспортировки данных того или иного пульта.

Пульт ДУ после нажатии определенной кнопки формирует своеобразную кодовую последовательность, в которой создается световой поток, испускаемый инфракрасным светодиодом. На практике в пультах дистанционного управления производится три разновидности модуляции (Рисунок 1.1).



Рисунок 1.1 – Виды модуляции в пультах ДУ

Отправляемый пультом своеобразный пакет световых импульсов основан из информационного поля и заголовка. Заголовок, как правило, представляет импульс. Длительность импульса и за ним следующей паузы позволяют понять и определить тот самый тип используемого протокола. В информационном поле содержатся биты команды и адреса. Количество битов команды и адреса, а также порядок размещения в поле зависит только от разновидности протокола. Адрес, модулированный пультом, всегда является постоянным и не может зависеть от нажатой кнопки. Устройства разных типов, как правило, имеют различающиеся адреса, что в итоге позволяет исключить их срабатывание в

один промежуток времени от одного пульта при одном и том же протоколе обмена. Код команды напротив выявляет нажатую кнопку и для различных систем может быть одинаковым. Таким образом, взаимоисключение пультов ДУ от разных устройств обусловлена либо различием протоколов, либо различием адресов при одних и тех же протоколах. В таблице 1.1 показаны структуры информационных полей разновидных протоколов систем ДУ.

Т а б л и ц а 1 . 1 – Структуры информационных полей протоколов ДУ

Протокол, компания	Структура информационного поля																								
NEC, standart	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>A0</td><td>...</td><td>A7</td><td>A0</td><td>...</td><td>A7</td><td>C0</td><td>...</td><td>C7</td><td>C0</td><td>...</td><td>C7</td> </tr> <tr> <td colspan="6" style="text-align: center;">адрес</td> <td colspan="6" style="text-align: center;">команда</td> </tr> </table>	A0	...	A7	A0	...	A7	C0	...	C7	C0	...	C7	адрес						команда					
A0	...	A7	A0	...	A7	C0	...	C7	C0	...	C7														
адрес						команда																			
NEC, extended	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>A0</td><td>...</td><td>A7</td><td>A8</td><td>...</td><td>A16</td><td>C0</td><td>...</td><td>C7</td><td>C0</td><td>...</td><td>C7</td> </tr> <tr> <td colspan="6" style="text-align: center;">адрес</td> <td colspan="6" style="text-align: center;">команда</td> </tr> </table>	A0	...	A7	A8	...	A16	C0	...	C7	C0	...	C7	адрес						команда					
A0	...	A7	A8	...	A16	C0	...	C7	C0	...	C7														
адрес						команда																			
JVC	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>A0</td><td>...</td><td>A7</td><td>C0</td><td>...</td><td>C7</td> </tr> <tr> <td colspan="3" style="text-align: center;">адрес</td> <td colspan="3" style="text-align: center;">команда</td> </tr> </table>	A0	...	A7	C0	...	C7	адрес			команда														
A0	...	A7	C0	...	C7																				
адрес			команда																						
SAMSUNG	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>A0</td><td>...</td><td>A7</td><td>A0</td><td>...</td><td>A7</td><td>C0</td><td>...</td><td>C7</td><td>C0</td><td>...</td><td>C7</td> </tr> <tr> <td colspan="6" style="text-align: center;">адрес</td> <td colspan="6" style="text-align: center;">команда</td> </tr> </table>	A0	...	A7	A0	...	A7	C0	...	C7	C0	...	C7	адрес						команда					
A0	...	A7	A0	...	A7	C0	...	C7	C0	...	C7														
адрес						команда																			
SONY	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>C0</td><td>...</td><td>C6</td><td>A0</td><td>...</td><td>A4</td> </tr> <tr> <td colspan="3" style="text-align: center;">команда</td> <td colspan="3" style="text-align: center;">адрес</td> </tr> </table>	C0	...	C6	A0	...	A4	команда			адрес														
C0	...	C6	A0	...	A4																				
команда			адрес																						
PANASONIC	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>A0</td><td>...</td><td>A16</td><td>A0</td><td>...</td><td>A16</td><td>C0</td><td>...</td><td>C7</td> </tr> <tr> <td colspan="3" style="text-align: center;">адрес 0</td> <td colspan="3" style="text-align: center;">адрес 1</td> <td colspan="3" style="text-align: center;">команда</td> </tr> </table>	A0	...	A16	A0	...	A16	C0	...	C7	адрес 0			адрес 1			команда								
A0	...	A16	A0	...	A16	C0	...	C7																	
адрес 0			адрес 1			команда																			
PHILIPS RC5	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>A4</td><td>...</td><td>A0</td><td>C5</td><td>...</td><td>C0</td> </tr> <tr> <td colspan="3" style="text-align: center;">адрес</td> <td colspan="3" style="text-align: center;">команда</td> </tr> </table>	A4	...	A0	C5	...	C0	адрес			команда														
A4	...	A0	C5	...	C0																				
адрес			команда																						

В тех случаях, когда кнопка пульта уже нажата, вслед за главным пакетом отправляются последовательные повторы (Рисунок 1.2), которые могут представлять:

- 1 повторение главного пакета;
- 2 только заголовок главного пакета с изменяемой длительностью паузы;
- 3 только команда (или информационное поле) главного пакета, исключая заголовок.

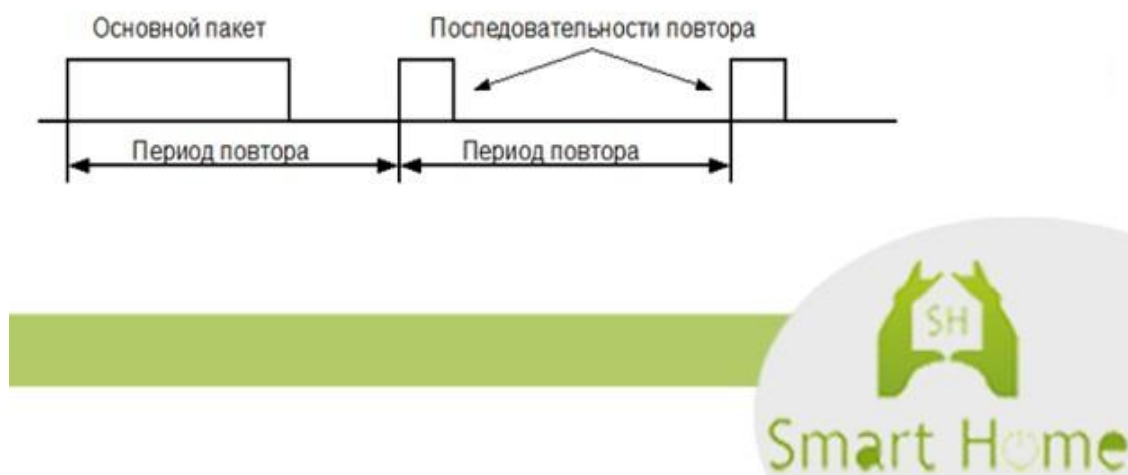


Рисунок 1.2 – Вид сигнала пульта ДУ при удержании кнопки

Временами требуется установить беспроводное соединение между устройствами и для этой цели чаще всего применяется Bluetooth и Wi-Fi. Но ведь совсем одно - передавать определенный видео материал и огромные файлы, а другое - править устройством или механизмом на 10 команд. Радиолюбители частенько производят, обеспечивают и переделывают заново модули приемников и передатчиков для работы с уже готовыми шифраторами/дешифраторами команд. В обоих случаях мы имеем возможность воспользоваться доступными RF-модулями.

#### *Разновидности модулей*

RF-модули для транспортировки данных осуществляют работу в диапазоне УКВ и пользуются стандартными частотами, 868МГц, 2,4ГГц либо 433МГц (реже используются 315МГц, 450МГц, 490МГц, 915МГц и др.) Чем больше несущая частота, тем с высшей скоростью возможна передача информации.

Как правило, изготавливаемые RF-схемы определены для работы с определенным протоколом транспортировки данных. Чаще остальных это SPI (RS-232) или UART. Обычно SPI модули обходятся дешевле и позволяют воспользоваться нестандартными (пользовательскими) протоколами передачи. Также существуют модели НМ-\*315 и НМ-\*433 различающиеся главной (несущей) частотой (315МГц и 433МГц соответственно). Помимо этого, существуют модули похожие по способу работы, поэтому передаваемая информация может оказаться полезной владельцу и других модулей.

#### *Передатчик*

Практически все RF-модули подразумевают под собой маленькую печатную плату с контактами для присоединения питания, транспортировки данных и главных управляющих сигналов. Его вид представлен на Рисунке 1.3. В передатчике НМ-Т868 имеется 3-х контактный разъем: GND(общий),

VCC(+питания) и DATA(данные), а также кусочек наконечника для припайки самой антенны.



Рисунок 1.3 – Вид передатчика RF-модуля

#### *Приемник*

Приёмник HM-R868 (Рисунок 1.4) внешне похож на соответствующий ему передатчикно на его разьеме существует дополненный четвертый контакт это ENABLE, при отправления на него питания приемник функционирует.



Рисунок 1.4 – Вид приёмника RF-модуля

По документации рабочим напряжением является 2,5-5В и чем больше напряжение, тем выше «дальность» работы. На Рисунке 1.5 представлена схема соединения приёмника и передатчика. Радио-удлинитель при подаче напряжения на вход DATA передатчика, на выходе DATA приемника также будет давать такое же напряжение (при том, что на ENABLE также будет

обеспечено напряжение). Также есть несколько нюансов. Во-первых: частота транспортировки данных (в работе - 600-4800 бит/с). Во-вторых: если на входе DATA нет сигнала больше чем 70мс, то передатчик уходит в сонный режим (отключается). В-третьих: если на месте приема ресивером нет работы передатчика, то на его выходе возникает некий шум. Из вышесказанного следует, что если при входе сигнал на трансмиттере будет отсутствовать менее 70мс и осуществляться в верном диапазоне частот, то модули будут как обычный провод (на остальные помехи и другие сигналы не обращаем внимания).

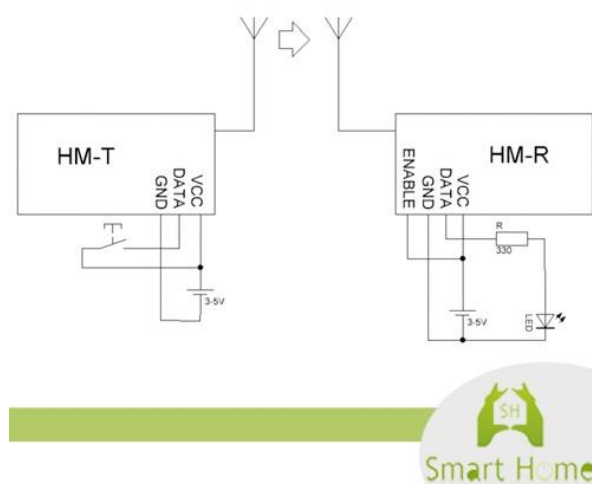


Рисунок 1.5 – Вид схемы соединения приёмника и передатчика

Имеется возможность подключения RF-модуля напрямую к вычислительно-аппаратному UART или компьютеру с помощью MAX232. В собственных целях пользуются модулями и пакетами видов: стоп-бит и старт-биты, байты с информацией, контрольный байт(или несколько). Первый старт-бит нужно сделать более длинным, это добавит времени для того, чтобы передатчик очнулся, приемник построился под него, а принимающий микроконтроллер осуществил прием. До этого времени информацию лучше считывать и верифицировать отдельными битами, даже если один из них неверный — завершаем прием и начинаем слушать эфир снова. Дальше транспортируемую информацию считывают сразу по байтам, вставляя в соответствующие регистры/переменные. В конце приема выполняем контрольное выражение, если его итог равен главному байту, то выполняем нужные по требованию действия с полученной информацией, по-другому заново прослушиваем эфир. Взамен арифметических операций используются логические биты: AND, NOT, OR и, в частности, XOR.

### 1.2.2 Микроконтроллер AVR

Микроконтроллер - это специальная микросхема, предназначенная для управления различными электронными устройствами. Микроконтроллеры впервые появились в 1971 году, кстати, в том же году, что и микропроцессоры общего назначения. Создатели микроконтроллеров разработали гениальную идею совместить процессор, ПЗУ, память и периферию внутри единого корпуса, внешне напоминая простую микросхему. С тех пор производство микроконтроллеров ежегодно во много раз больше производства процессоров, а их потребление не снижается. На данный момент их выпускают десятки компаний, причем производятся не только современные 32-битные микроконтроллеры, но и 16, и даже 8-битные (как i8051 и аналоги). Внутри каждого семейства часто можно встретить почти одинаковые модели, различающиеся скоростью работы ЦПУ и объемом памяти. Всё дело в том, что микроконтроллеры применяются преимущественно во встроенных системах, в игрушках, в станках, в массовой домашней технике, в домашней автоматике – там, где нужна не мощность процессора, а, скорее, баланс между ценой и достаточной функциональностью. Именно поэтому самые старые типы микроконтроллеров еще до сих пор в ходу – они многое могут: от автоматического открывания дверей и включения полива газонов до интеграции в систему «Умный дом». При этом существуют и более мощные микроконтроллеры, способные выполнять сотни миллионов операций в секунду и обвязанные периферией «до зубов». У них и задачи соответствующие. Таким образом, разработчик сначала оценивает задачу, а уж потом выбирает под нее подходящее «железо». Сегодня существует более 200 модификаций микроконтроллеров, совместимых с i8051, выпускаемых двумя десятками компаний, и большое количество микроконтроллеров других типов. Популярностью у разработчиков пользуются 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel, 16-битные MSP430 фирмы TI, а также 32-битные микроконтроллеры, архитектуры ARM, которую разрабатывает фирма ARM Limited и продаёт лицензии другим фирмам для их производства (Рисунки 1.5-1.7).



Рисунок 1.5—16-битный 28-pin PDIP PIC24 микроконтроллер

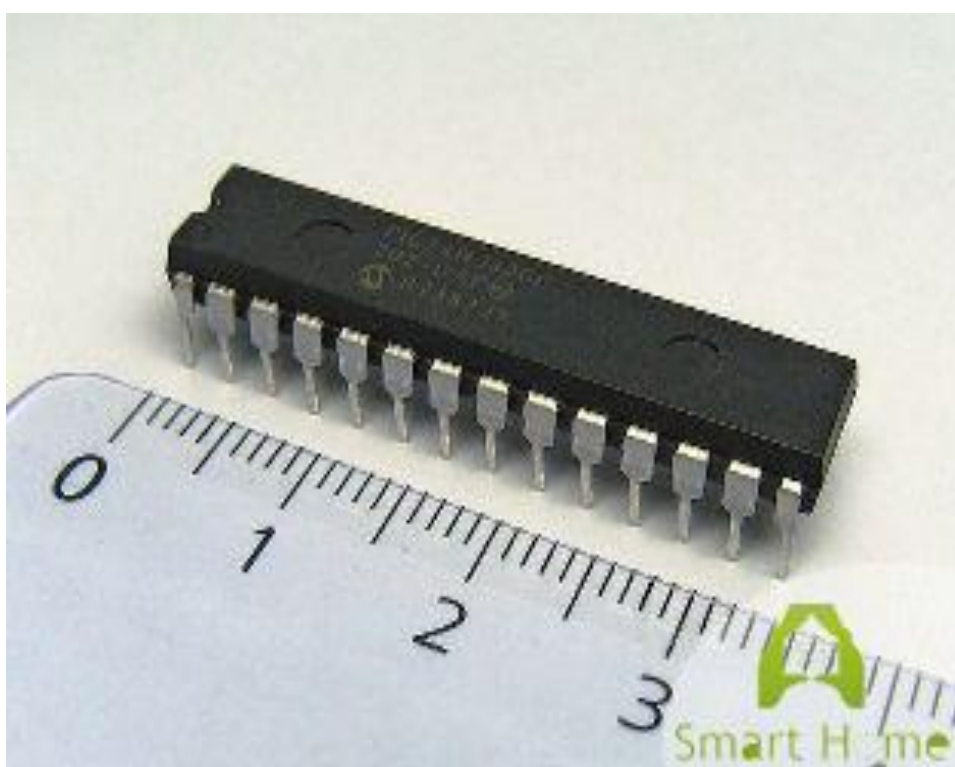


Рисунок 1.6—Микроконтроллер Atmel AVR ATmega8 в корпусе DIP

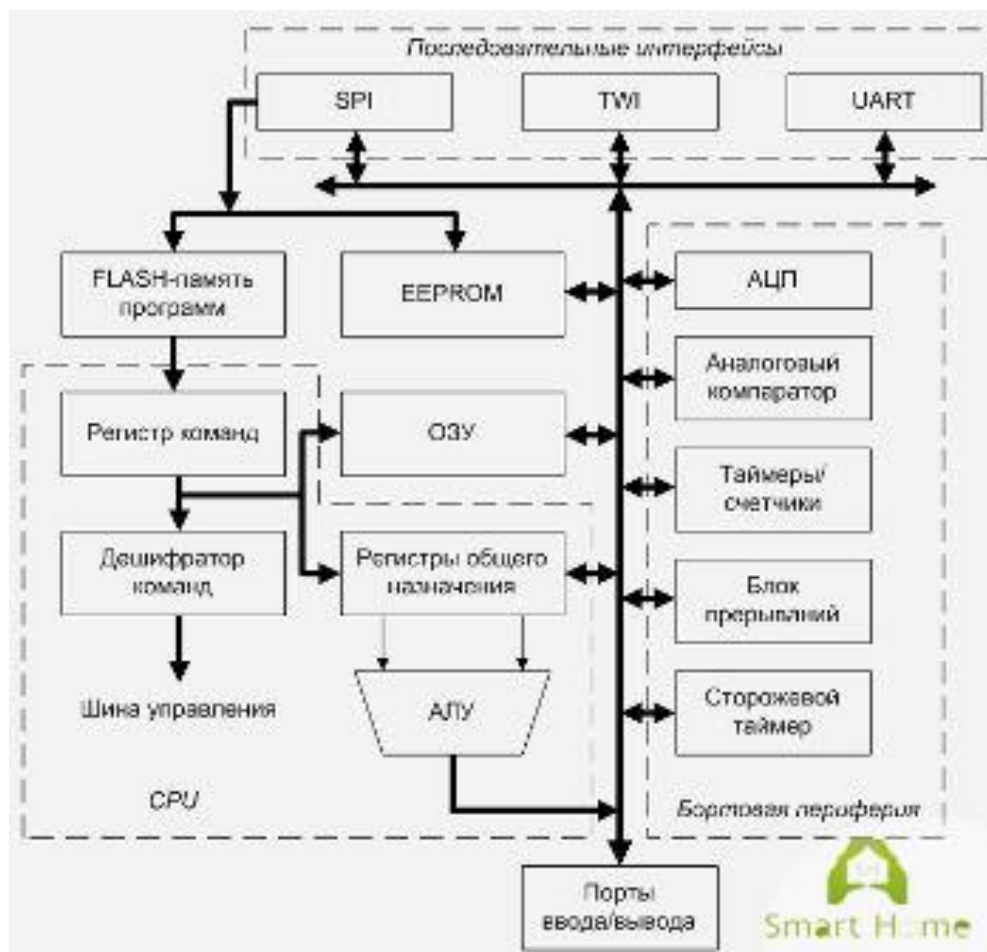


Рисунок 1.7– Схема устройства микроконтроллера AVR

### *Устройство микроконтроллера AVR.*

Микроконтроллер характеризуется числами параметров, так как он единожды является сложным программно-управляемым устройством и электронным прибором (микросхемой). Приставка "микро" в названии микроконтроллера означает, что выполняется он по микроэлектронной технологии. В ходе выполнения работы он считывает команды из памяти или порта ввода и исполняет их. Что означает каждая команда, определяется системой команд микроконтроллера. Система команд определена в архитектуре, и выполнение кода команды описывается в проведении внутренними элементами микросхемы определенных микроопераций. Микроконтроллеры помогают гибко управлять различными электронными и электрическими компонентами. Некоторые модели микроконтроллеров настолько объемны, что могут переключать реле (к примеру, на елочных гирляндах). Микроконтроллеры не работают в одиночку, а запаиваются в схему, где, кроме него, подключаются экраны, клавиатурные входы, различные датчики и т.п. Софт для микроконтроллеров может привлечь внимание тех, кто обожает «гоняться за битами», так как обычно память в микроконтроллерах составляет от 2 до 128 Кб. Если меньше, то писать приходится на ассемблере или Форте, если есть возможность, то используют специальные версии



Бейсика, Паскаля, но в основном Си. Прежде чем уже запрограммировать микроконтроллер, его тестируют в специальных эмуляторах аппаратных или программных. Чип, который содержит всё, чтобы этого хватало для построения законченного изделия и есть пример типового микроконтроллера. Например, наручные электронные часы или калькулятор имеют внутри микроконтроллер, который осуществляет все функции такого устройства. Отдельные периферийные устройства подключаются непосредственно к ножкам микросхемы микроконтроллера, либо совместно используются дополнительные элементы или микросхемы малой либо средней степени интеграции. Микроконтроллеры широко используются в изделиях, которые содержат всю систему целиком исключительно в одной миниатюрной микросхеме, часто называемой микросборкой. Например «чиповая» кредитная карточка содержит микроконтроллер внутри в пластиковой основе. Таблетка домофона так же внутри себя содержит микроконтроллер. И примеров использования и применения микроконтроллеров настолько обширен в современном мире, что легко обнаружить наличие контроллера в любом мало-мальски интеллектуальном устройстве от детской игрушки до беспроводной гарнитуры сотового телефона.

Схема состоит из четырех частей: центральное процессорное устройство (ЦПУ) или просто центральный процессор, память, шины данных и периферические устройства. На Рисунке 1.8 представлена общая схема архитектуры микроконтроллера AVR/.

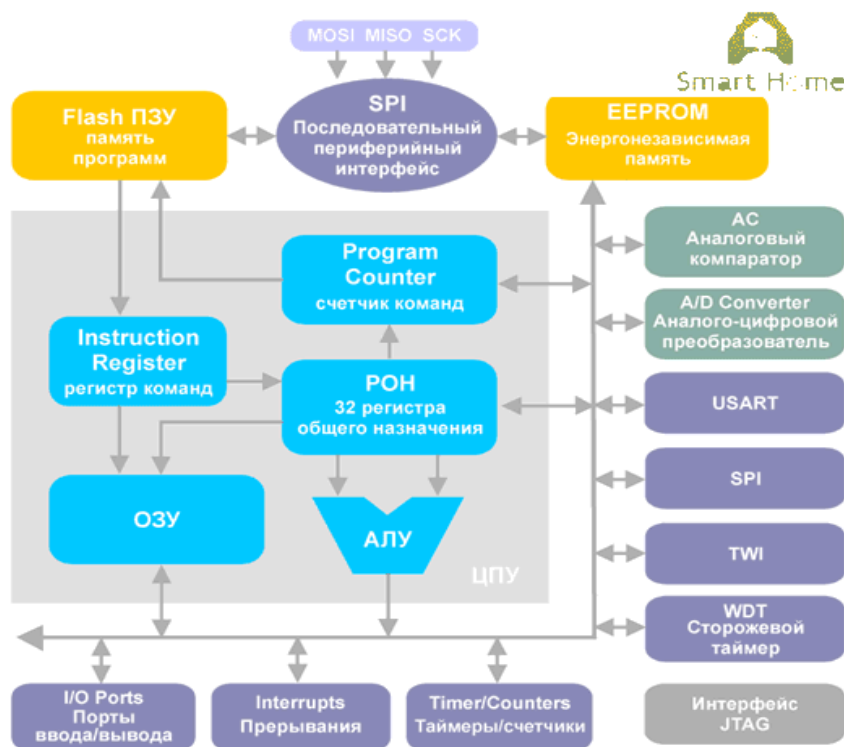


Рисунок 1.8 –Общая схема архитектуры микроконтроллера AVR

### *Центральный процессор*

Центральный процессор микроконтроллеров AVR является 8-битным и построен на принципах RISC архитектуры. Основным достоинством этого типа архитектуры является выполнение каждой инструкции (за исключением инструкций работы с памятью и условных переходов) процессора за один такт, что позволяет достичь производительности 1MIPS на 1МГц.

Регистр команд – это внутренняя область памяти, в которой хранится инструкция для процессора.

Программный счетчик – это счетчик, который указывает на инструкцию, выполняемую в данный момент времени. В соответствии с этим счетчиком инструкции выбираются из памяти программ и переносятся в регистр команд.

Регистры общего назначения – это регистры, в которых хранятся данные, необходимые для выполнения команд. Это наиболее быстродействующая память микроконтроллера. Микроконтроллеры AVR содержат 32 регистра общего назначения, имеющих байтовый формат то есть каждый регистр состоит из 8-ми бит.

Оперативно запоминающее устройство (ОЗУ) – это энергозависимая быстродействующая память микроконтроллера. Начало адресного пространства ОЗУ занимают регистры общего назначения, но физически не являются ее частью. Такое ухищрение позволяет обращаться к регистрам собственно как к регистрам, а также как к памяти. Это является особенностью микроконтроллеров AVR, повышающей эффективность и производительность.

Арифметико-логическое устройство (АЛУ) – это сердце центрального процессора. Это устройство собственно и производит вычисления, и логические операции.

### *Память*

Принципы реализации взаимодействия памяти и ЦПУ в микроконтроллерах AVR базируются на гарвардской архитектуре. Согласно этой архитектуре память данных и память программ имеют отдельное адресное пространство и отдельные шины. Эта архитектура существенно увеличивает производительность по сравнению с устройствами, построенными по фон-неймановской архитектуре, в которой память данных и память программ находится в одном адресном пространстве и имеет одну шину (примером таких систем является ПК на базе архитектуры x86).

Память программ – это Flash(флэш) память, предназначенная для хранения последовательности команд, управляющих работой ЦПУ. Память программ является 16-битовой, а также построена на принципах электрического перепрограммирования, что позволяет многократно ее перезаписывать. Последовательность команд записывается в программную память во время операции программирования микроконтроллера, через внешний интерфейс. Кроме того микроконтроллеры серии Mega имеют возможность само программирования, что позволяет обновлять программу этих устройств без программатора либо изменять алгоритм программы в зависимости от внешних или внутренних условий. Стоит отметить, что

гарантированное число циклов перезаписи Flash-памяти ограничено и указывается в документации.

Память данных – это память, в которой хранятся данные, необходимые для выполнения инструкций микроконтроллера. Память данных в микроконтроллерах AVR разделена на три типа:

1. Регистровая память – это регистры общего, а также служебные регистры ввода вывода. Эти регистры находятся в адресном пространстве ОЗУ, но физически не являются его частью. В области памяти регистров ввода/вывода расположены различные служебные регистры микроконтроллера. Служебные регистры полностью контролируют и управляют работой микроконтроллера и периферийными устройствами.

2. Оперативная память (ОЗУ) – это внутренняя энергозависимая память, которая имеет байтовый формат и используется для хранения данных. ОЗУ не может принимать участия в выполнении команд, сначала данные из ОЗУ должны быть загружены в регистры общего назначения. Число циклов перезаписи этой памяти не ограничено.

3. Энергонезависимая память данных (EEPROM) – эта память предназначена для долговременного хранения данных. Идеально подходит для хранения различных констант и серийных номеров, имеет возможность электрической перезаписи и доступна программе непосредственно в ходе ее выполнения. Эта память может быть записана извне во время программирования микроконтроллера. Несмотря на все достоинства, EEPROM является довольно медленным и имеет ограниченное число циклов перезаписи, что необходимо учитывать при составлении программ.

### *Периферия*

Периферия микроконтроллеров AVR включает в себя порты ввода вывода, причем каждый вывод порта программируется отдельно и может быть настроен как на вход, так и на выход. Кроме портов ввода вывода есть целый ряд устройств, каждое из которых заслуживает отдельного рассмотрения. Давайте, их пока просто перечислим: контроллер прерываний (Interrupts), таймеры счетчики (Timer/Counters), сторожевой таймер (WDT), аналоговый компаратор (AC), аналогово-цифровой преобразователь (ADC), универсальный последовательный приёмопередатчик (USART), последовательный периферийный интерфейс (SPI), двухпроводный последовательный интерфейс (TWI или I2C), интерфейс JTAG, тактовый генератор (System Clock), часы реального времени (RTC) и многое другое.

## **1.2.3 Каркас макета**

В наше время древесно-стружечная плита достаточно один из самых популярных материалов (Рисунок 1.9). Основные достоинства ДСП это низкая стоимость и простота в обращении. ДСП изготавливается путем горячего прессования крупнодисперсной стружки, происходящей из отходов деревообработки и древесины любых пород. На первом этапе производства

происходит переработка сырья. Осуществления круглой специальной древесины сокращается из-за использования таких материалов, как щепы, опилки и вторичная древесина. Достаточно часто в производство идут все виды сырья одновременно, или в смешанных видах. Полученную стружку сортируют, очищают и сушат. Затем из просмоленной стружки формируется так называемый ковер, из которого после прессования получается плита. Далее плиты кромкуются и подвергаются различным формам конечной обработки - шлифованию, нанесению покрытий и другим. Качество ДСП зависит от степени обработки поверхности. Первосортные плиты должны быть отшлифованы, на поверхности не должно быть царапин и других механических дефектов, пятен различного происхождения, края должны быть ровными, их толщина 10-26 мм. Из таких плит делают мебель.



Рисунок 1.9 – ДСП

Сорт ДСП определяется качеством поверхности. Различают ДСП 1 сорта, 2 сорта и несортную плиту используемую, как правило, в строительных целях. Согласно ГОСТ 10632-89, плиты первого сорта не должны иметь углублений (выступов) или царапин, парафиновых, смоляных пятен, сколов кромок, краски углов, не до шлифовки, волнистости поверхности. На ДСП второго сорта допускаются сколы кромок в пределах отклонений по длине или ширине плиты. Поверхность второсортной плиты может содержать дефекты шлифования не более 10% от площади. Так же на ДСП 2 сорта допускаются большие, в сравнении с 1 сортом, включения коры и крупной фракции стружки.

#### *Крепеж*

Сегодня большое внимание достается качеству и надежности построенных конструкций, поэтому крепеж обязан быть изготовлен добросовестно. На Рисунке 1.10 представлены виды крепежей. К элементам

крепежа выдвигаются крайне жесткие требования. Он должен быть гарантом того, что конструкция прослужит долгие годы, а соединение будет надежным. Дольше по времени в строительстве используется перфорированный крепеж. Он проверен годами, и к нему прибегают многие строители. При использовании такого вида крепежа характеристики прочности балочных или деревянных конструкций в значительной мере увеличиваются. Также это один из быстрых вариантов крепежа, потому время на этот процесс уходит куда меньше, чем при использовании других видов. Не потребуются специальные инструменты, но будут нужны следующие материалы: прямой подвес, перфолента и перфорированный уголок. Цинковое покрытие даст возможность использовать крепёжные элементы для внешней и внутренней отделки в равной мере. Широко применяется перфорированный крепеж при креплении полок, строительстве перегородок, при реставрации, для мелкого ремонта.



Рисунок 1.10 –Крепежи

Прочностные характеристики у перфорированного крепежа очень качественные и высокие. Такие характеристики дают возможность применения перфорированного крепежа для многих видов соединений. Например, конструкция кровли, крепления балок и прочее. Высокую надежность и качество конструкции гарантирует оцинкованная сталь, используемая для создания элементов крепежа. Такое покрытие не подвергается коррозии. Нагрузка на соединение распределяется равномерно, это предотвращает возможное возникновение трещин. При всей прочности перфорированного крепежа, сам он очень легкий. Для того чтобы выполнить такой крепеж, вам не нужно будет приобретать специальные навыки или оборудование. Снижаются затраты на материалы, потому что дерева для крепежа нужно значительно меньше, чем при использовании любого другого вида крепежа. Вы увидите это, когда подсчитаете конечную сумму за выполненную работу. С каждым годом

использование перфорированного крепежа все больше и больше приобретает популярность. Универсальность и простота в использовании, широкий выбор крепежных деталей и демократичная цена дают возможность обустроить в жилище именно тот интерьер, который вам по душе.

### 1.2.4 LCD-дисплей

Жидкокристаллический дисплей, он же ЖК-дисплей, ЖКД, жидкокристаллический индикатор (ЖКИ), это гладкий дисплей плоской поверхности, в основе которых располагаются жидкие кристаллы, и далее устройство (монитор или телевизор) на основе такого дисплея. ЖК-дисплеи имеют свои собственные ограничения; они могут отображать только символы определенных размеров (Рисунок 1.11).



Рисунок 1.11 –Пример обычного ЖК-дисплея

Графические ЖК-дисплеи используются для отображения пользовательских символов и изображений. ЖК-дисплеи находят применение во многих приложениях; они используются в видео играх, мобильных телефонах, лифтах и т.д., как витрины. Различные графические ЖК-дисплеи доступны на рынке с различными размерами. Матричный ЖК-дисплей имеет формат отображения 128x64 точки имеет желто-зеленый цвет подсветки. Каждый ЖК нуждается в контроллере для выполнения своих внутренних операций. Этот ЖК использует два KS0108 контроллеров. ЖК-дисплеи включают схему подкачки, т.е. вся она делится поровну на страницы. Схема поискового вызова из графического дисплея можно легко понять из Рисунка 1.12.

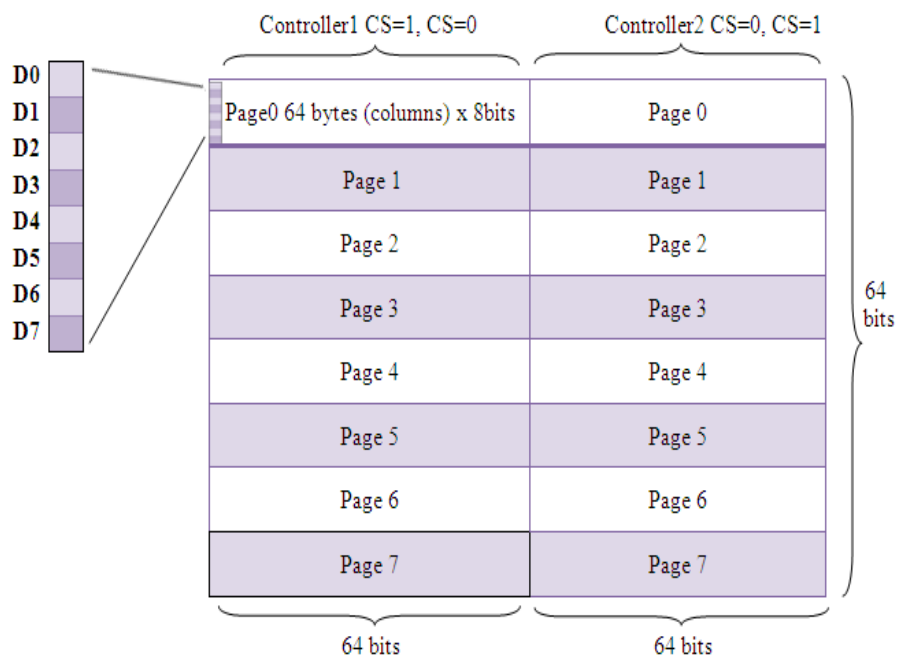


Рисунок 1.12 – Структура работы ЖК-дисплеев

1 Разрешение в общей сложности  $128 \times 64 = 1024$  пикселей.

2  $128 \times 64$  ЖК делится поровну на две половины. Каждая половина управляется отдельным контроллером и состоит из 8 страниц. В диаграмме выше, CS расшифровывается как контроллер «Select».

3 Каждая страница состоит из 8 строк и 64 колонок. Так две горизонтальные страницы содержат 128 ( $64 \times 2$ ) столбцов и 8 вертикальных страниц содержат 64 строк ( $8 \times 8$ ). На Рисунке 1.13 представлены контакты ЖК-дисплея.

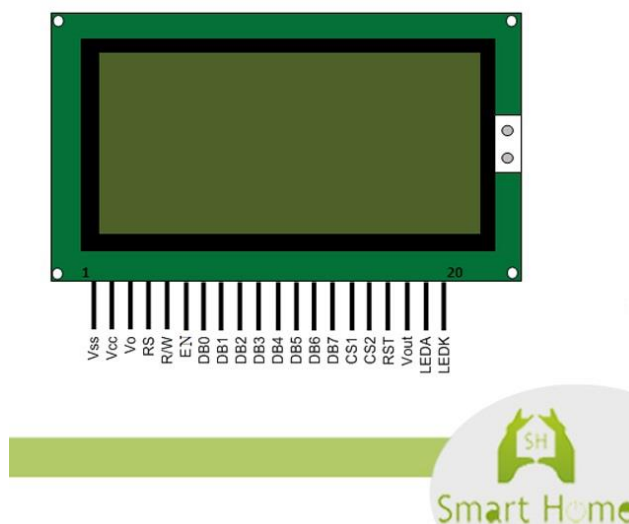


Рисунок 1.13 – Контакты ЖК-дисплея

В таблице 1.2 JHD12864E приводится описание контактов ЖК-дисплея.

Т а б л и ц а 1.2 – Описание контактов

Pin	Функция	Название
1	Первый (0 В)	V <sub>ss</sub>
2	Напряжение питания; 5V	V <sub>cc</sub>
3	Регулировка контрастности	Bo
4	Высокая для отображения данных; Низкий для кода команды	Регистрация выберите (RS)
5	От низкого до записи в реестре; Высокая читать из реестра	Чтение / запись (R / W)
6	Читает данные при высокой; Записывает данные в высокой к низкой перехода (падение края)	Включить (RU)
7	8-битные выводы данных	DB0
8		DB 1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Выбор чип для IC1; Активный высокий	CS1
16	Выбор чип для IC2; Активный высокий	CS2
17	Сигнал Reset; Активный низкий	RST
18	Выходное напряжение для ЖК вождения	Vout
19	Подсветка V <sub>cc</sub> (5 В)	LED
20	Подсветка Земля (0 В)	Светодиодные К

### 1.2.5 Электронное реле

Реле – это электрическое устройство (выключатель), предназначенное для замыкания и размыкания различных участков электрических цепей при заданных изменениях электрических или неэлектрических входных величин. Типы реле могут различаться по управляющему сигналу и по исполнению. Наибольшее распространение получили электрические (электромагнитные) реле.

Реле предназначено для коммутации больших токов нагрузки или является переключателем. Принцип работы реле заключается при малом токе (сигналом кнопки) включения цепи с большим током. Используют реле в тех



случаях, которых исполнительное устройство (стартер, генератор, вентилятор, обогрев зеркал, клаксон и т.д.) потребляет большой ток(до 30-40 ампер).

Электромагнитное реле состоит из:

1 Электромагнита (представляет собой электрический провод, намотанный на катушку с сердечником из магнитного материала).

2 Якоря (пластина из магнитного материала, через толкатель управляющая контактами).

3 Переключателя (могут быть замыкающими, размыкающими, переключающими).

При пропускании электрического тока через обмотку электромагнита возникающее магнитное поле притягивает к сердечнику якорь, который через толкатель смещает и тем самым переключает контакты (Рисунок 1.14).

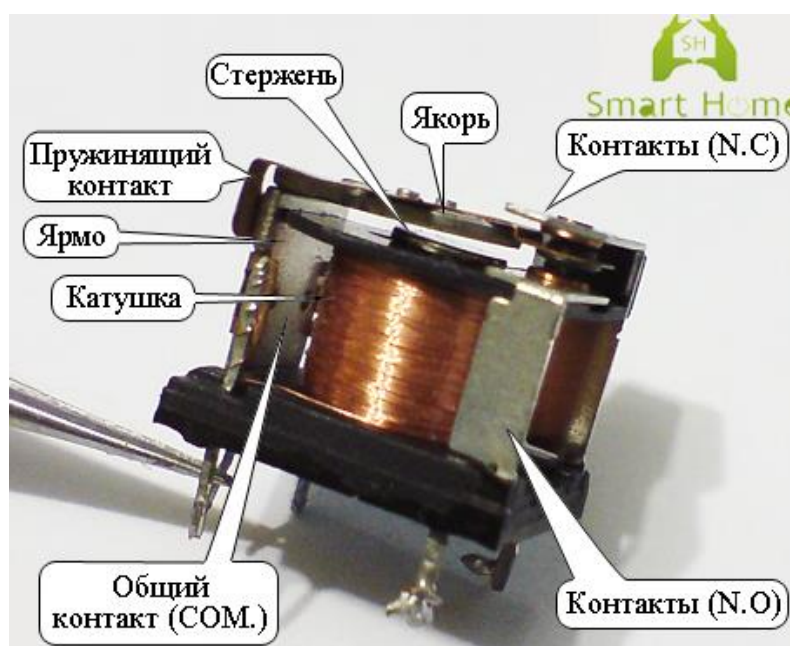


Рисунок 1.14 –Структура реле

Характеристики устройства реле:

- 1 Номинальное напряжение: 12В.
- 2 Диапазон электропитания: 8...16В.
- 3 Ток управления: не более 0,2А.
- 4 Активное сопротивление обмотки:  $80 \pm 10$  Ом.
- 5 Напряжение срабатывания: не менее 8,0В.
- 6 Напряжение отпускания: 1,5...5,0В.
- 7 Максимальный ток в силовой цепи: 30А.

## Вывод

В главе был проведён анализ предметной области. Представлено краткое описание возможностей «Умного дома» и его каждой составляющей. Также

изучена сфера системы в РК и компании, которые занимаются установкой таких систем.

## 2 Технологии и программное обеспечение

### 2.1. Технология взаимодействия пульта с микроконтроллером

Используем радио модули Норе НМ-Т433иНоре НМ-R433, на передачу и на приём, соответственно. Само устройство изображается маленькой платой 15x25 мм с выдвинутым из неё разъёмом. У транспортировщика разъём трех контактный – VCC, GND и DATA, у ресивера также существует вход ENABLE при токе, на приёмнике включается приём данных (Рисунок 2.1).

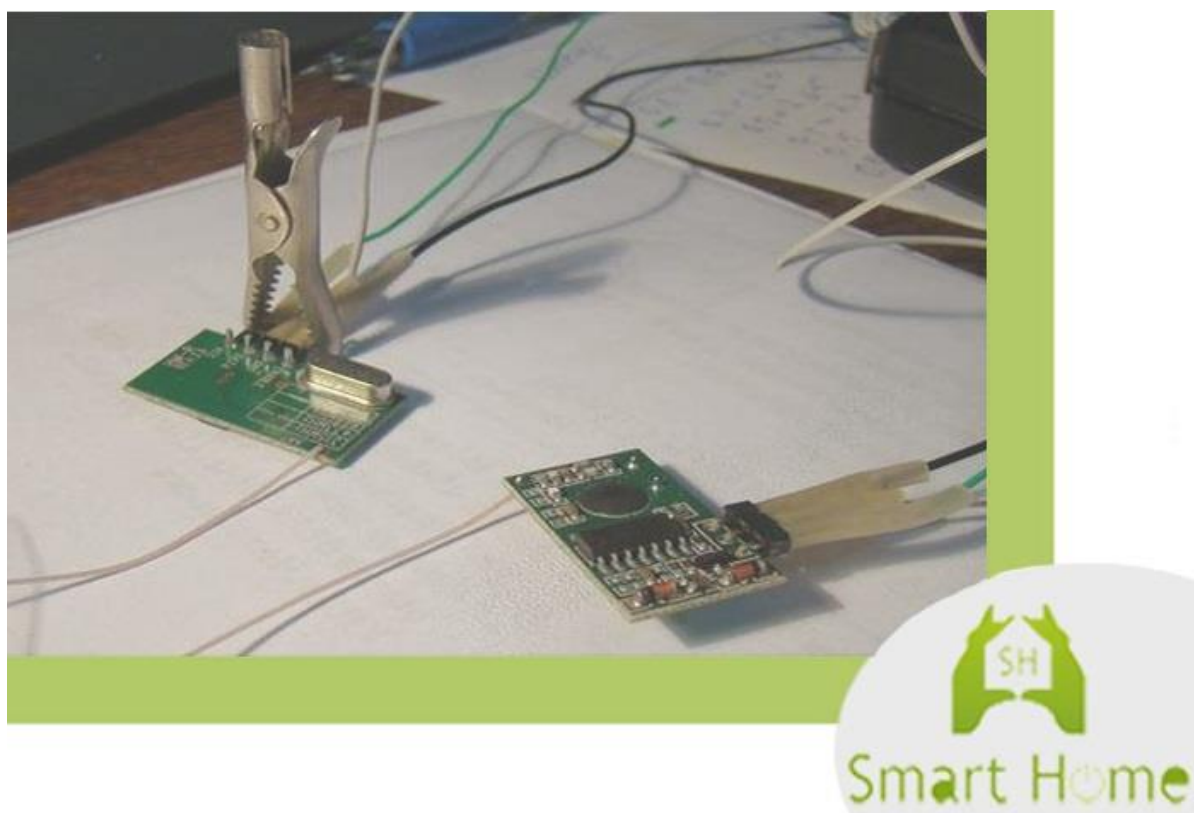


Рисунок 2.1 –Пример присоединения передатчика и ресивера

Данное устройство приходится «радио проводом». Что послали на вывод DATA на передатчике, то и придет в DATA на ресивере. Помех при передаче не обнаружено. Всё работает чётко, как положено, управляется без помех. Частота передачи радиосигнала осуществляется на волне в 433МГц.

Характеристики передачи:

- 1 Напряжение питания: от 2.5 до 5 вольт.
- 2 Частота: 315,433,868 и 915 МГц в зависимости от модели.
- 3 Скорость передачи: от 600 до 9600 бод.

- 4 Температурный диапазон:  $-30 \dots 80^{\circ}\text{C}$ .
- 5 Употребляемый ток: 30мА у отправителя и 10мА у ресивера.
- 6 Дальность или размах сигнала: производителем гарантировано передача радиосигнала до 250-ти метров.

В итоге, пульт дистанционного управления посылает сигналы через RF-модуль на микроконтроллер, а он в свою очередь отправляет конечный сигнал по заложенной в нём программе. На Рисунке 2.2 показана структура передачи сигнала в системе.



Рисунок 2.2–Схема передачи сигнала в системе

## 2.2 Технология работы микроконтроллера с LCD-дисплеем

Atmega8A может работать от 3,3V при условии, что частота не выше 8 МГц. Непосредственно перед дисплеем не надо подавать питание на электролит большей емкости. В начале функционирования контроллер даёт на дисплей команды инициализации. Для заряда конденсатора требуется определенное время. Пока он заряжается, и дисплей начнёт функционировать протечёт определенное время, и команды инициализации он не получит. Конечно это несколько малое время, но в любом таком случае эффект очень ощутим (Рисунок 2.3).

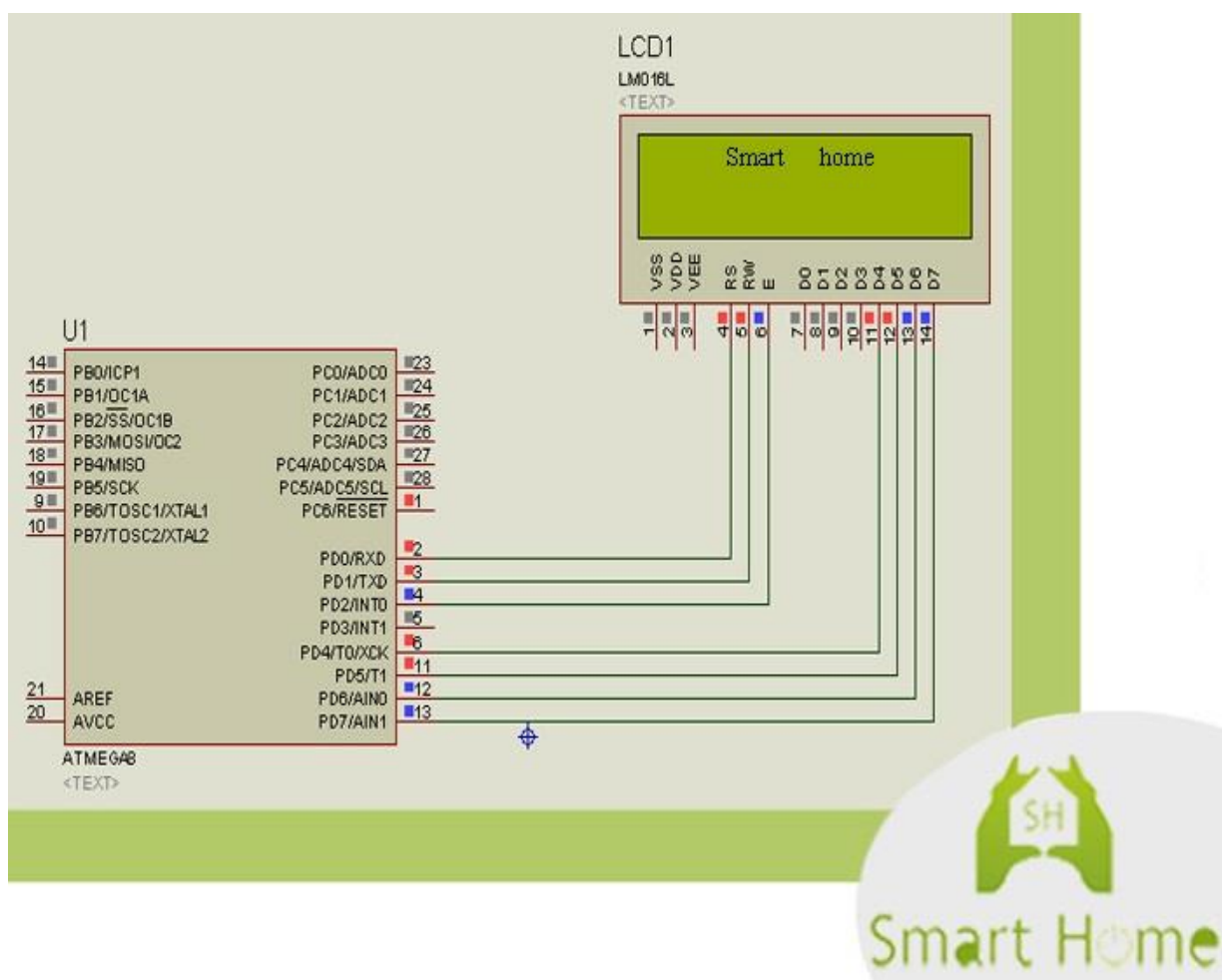


Рисунок 2.3 –Присоединения микроконтроллера к ЖК-монитору

Описание некоторых элементов:

- 1 Заземление (GND) 0В.
- 2 Питание (VCC) +5В.
- 3 Корректировка контрастности. Присоединяется с помощью переменного резистора.

В итоге, микроконтроллер посылает сигналы через разные порты на контакты ЖК-дисплея, при этом отправляется уже конечный результат, который появляется на экране. На Рисунке 2.4 показана структура передачи сигнала из микроконтроллера на ЖК-дисплей.

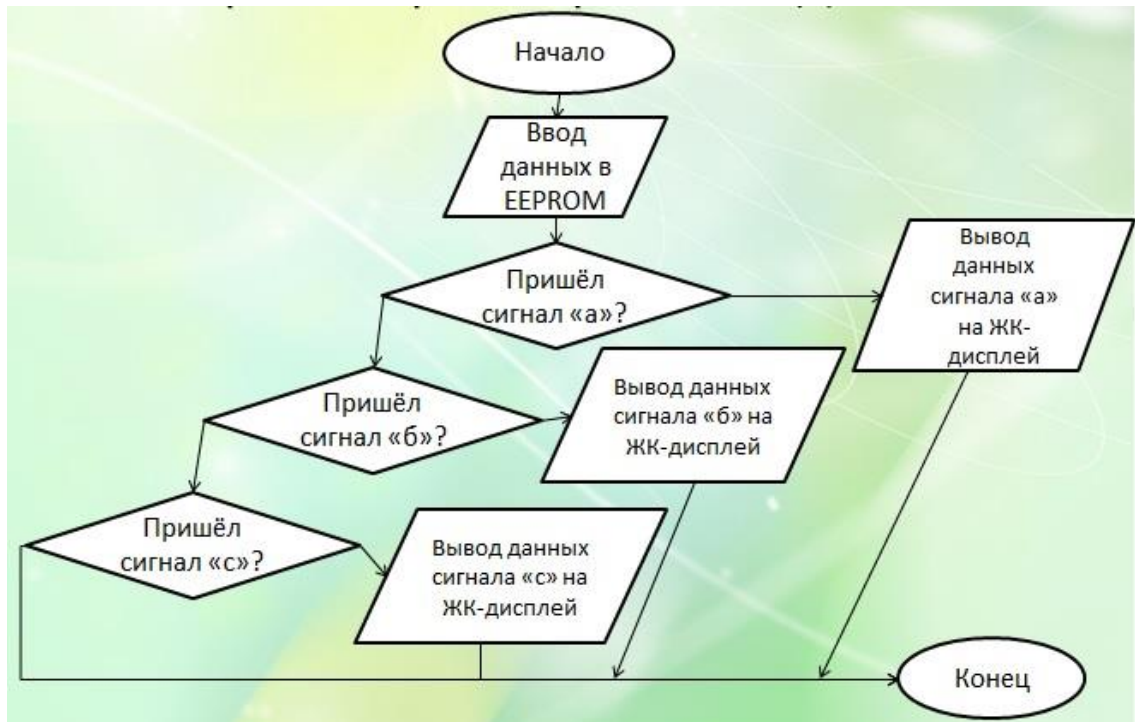


Рисунок 2.4 –Схема передачи сигнала из микроконтроллера на ЖК-дисплей

В свою очередь с помощью ЖК-дисплея мы выбираем настройки пользователя или включаем устройства системы. На Рисунке 2.5 изображена структура меню ЖК-дисплея.

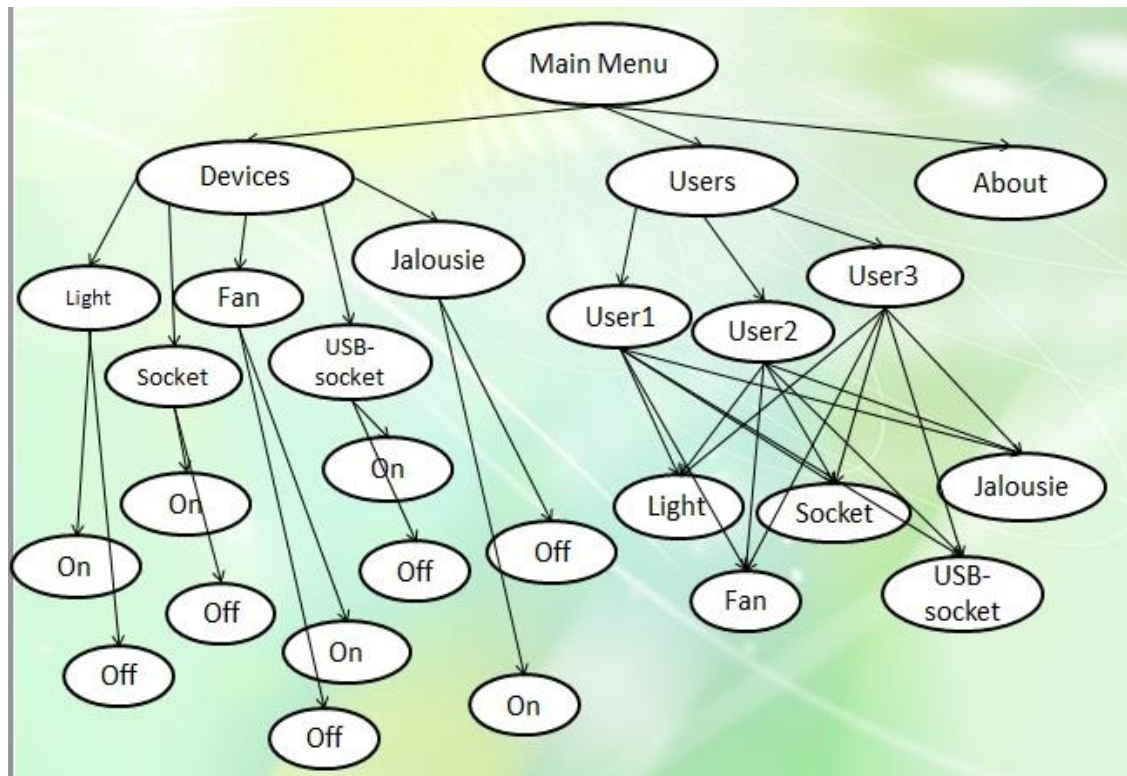


Рисунок 2.5 –Наглядное отображение принципа работы ЖК-дисплея

### MainMenu

Главное и начальное меню, в котором изображены окна: устройства, пользователи, о программе.

### Device

Окно, в котором представлен список всех устройств макета, и имеется возможность выбора конкретного устройства.

### Users

Окно юзеров, в котором выбираем определенного пользователя.

### About

Окно описания макета.

### Light

В данном окне имеется возможность включить (On) свет или выключить (Off).

### Socket

В окне имеется возможность включить (On) розетку или выключить (Off).

### Fan

В данном окне имеется возможность включить (On) вентилятор или выключить (Off).

### USB-socket

В данном окне имеется возможность включить (On) USB-розетку или выключить (Off).

### Jalousie

В данном окне имеется возможность включить (On) жалюзи или выключить (Off).

### User1

Окно первого пользователя, в котором мы задаём определенную нужную последовательность включения устройств.

### User2

Окно второго пользователя, в котором мы задаём определенную нужную последовательность включения устройств.

### User3

Окно третьего пользователя, в котором мы задаём определенную нужную последовательность включения устройств.

## 2.3 Технология подсоединения сторонних устройств к плате

Я использовал и подключил такие устройства, как бра, вентилятор, розетки, usb-выход, рулонные жалюзи путем соединения с платой при помощи тиристорных реле и напрямую к плате (Рисунок 2.6):

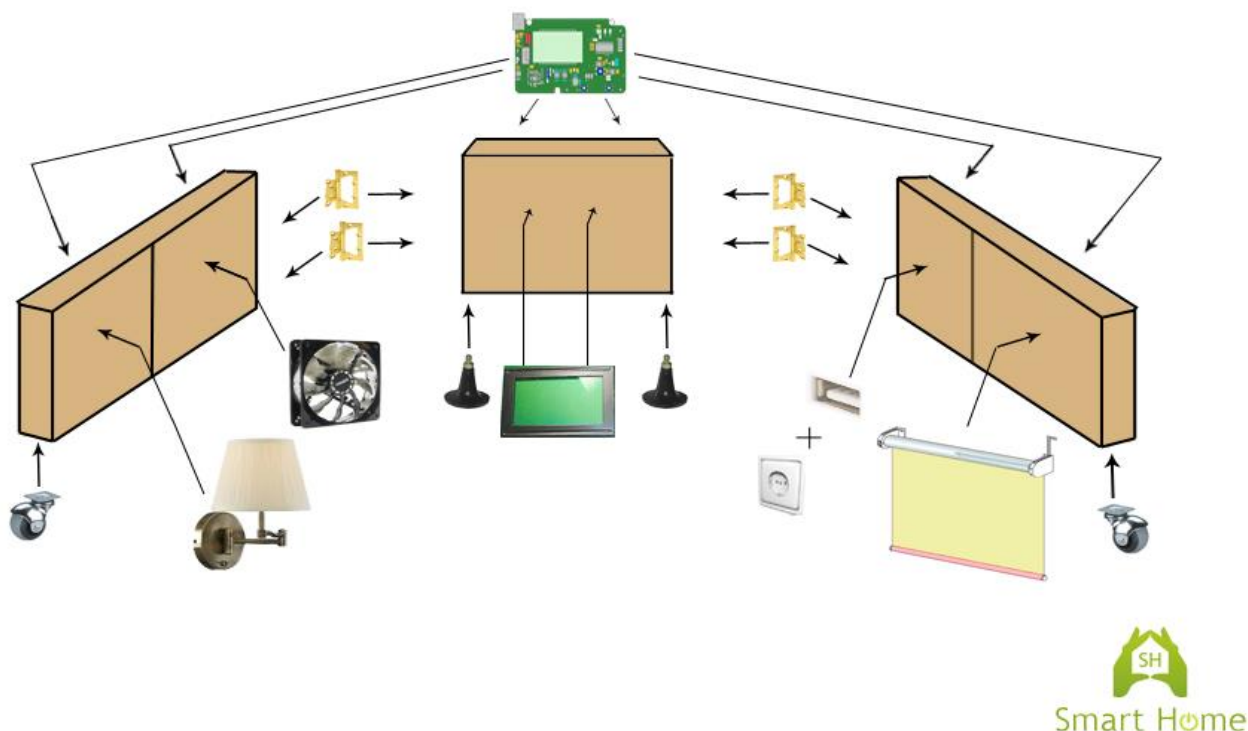


Рисунок 2.6 –Внешний вид макета

Также подключение бытовых устройств осуществляется по методам:

- 1 Бра через плавное регулирование напряжения.
- 2 Вентилятор через три ступени дискретного управления.
- 3 USB–выход через дискретное управление.
- 4 Мотор со шторкой через реверсивное управление без фиксации.

## 2.4 Windows 7.

В работе использовалась операционная система Windows 7 при учёте удачной конфигурации данной ОС, а также обширной поддерживаемости её с другими программами. «Windows 7»–это своеобразная пользовательская операционная система семейства Windows NT, следующая по времени выхода за Windows Vista и предшествующая Windows 8. В линейке Windows NT система имеет номер версии 6.1 (Windows 2000— 5.0, Windows XP– 5.1, Windows Server 2003 — 5.2, Windows Vista и Windows Server 2008– 6.0) (Рисунок 2.7–2.8).



Рисунок 2.7 –пример панели задач ОС Windows 7

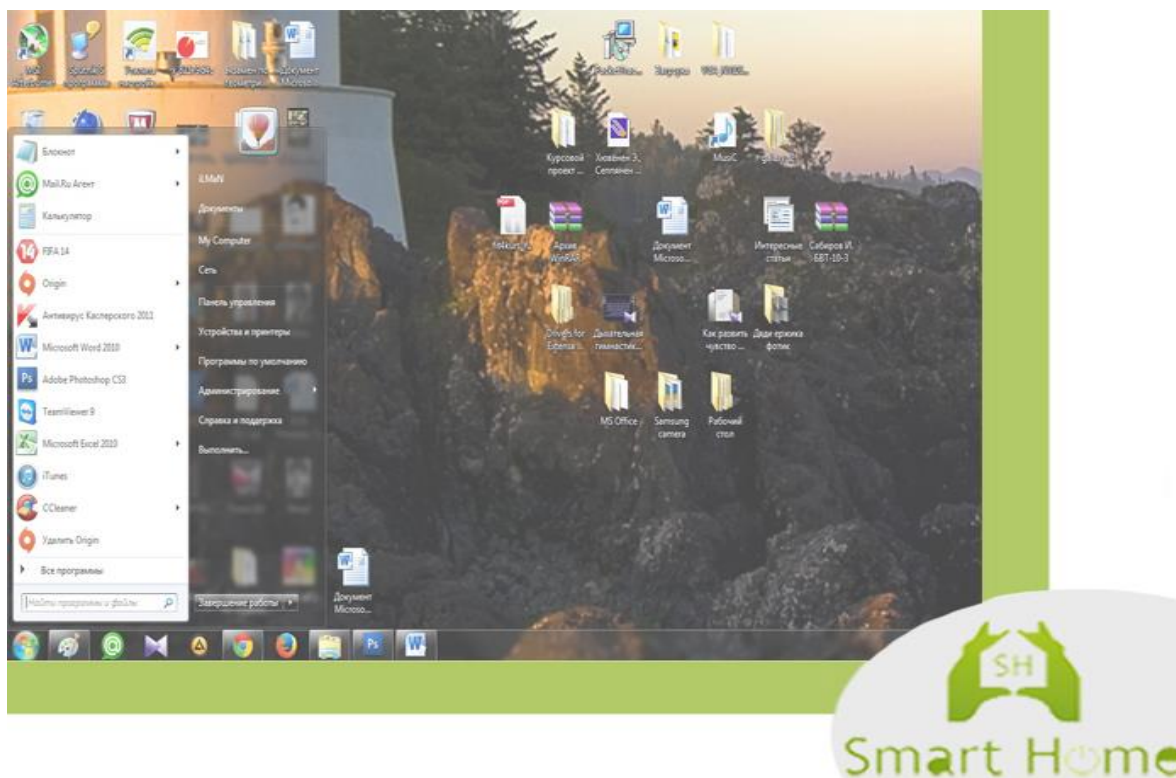


Рисунок 2.8 –Интерфейс ОС Windows 7

Операционная система вышла в продажу 22 октября 2009 года—меньше, чем через три года после выпуска предыдущей операционной системы, Windows Vista. Партнёрам и клиентам, обладающим лицензией Volume Licensing, доступ к RTM был предоставлен 24 июля 2009 года. В интернете оригинальные установочные образы финальной версии системы были доступны с 21 июля 2009 года.

По данным веб-аналитики от W3Schools, на январь 2014 года доля Windows 7 среди используемых в мире операционных систем для доступа к сети Интернет составила около 55,3 %. По такому принципу отбора она находится на первом месте, обгоняя в августе 2011 предыдущего лидера – Windows XP.

## 2.5 Proteus Pro.

PROTEUS VSM—пакет программ для автоматизированного проектирования (САПР) электронных схем. Создание компании Labcenter Electronics, базирующейся в Англии (Рисунок 2.9).

Пакет программы представляет структуру схемотехнического моделирования, базирующуюся на основе специальных моделей электронных деталей, принятых в PSpice. Различной чертой пакета PROTEUS VSM является то, что можно моделировать работы программируемых устройств: микропроцессоров, микроконтроллеров, DSP и т.д. Библиотека компонентов имеет справочные данные. Также в пакете PROTEUS VSM имеется система проектирования печатных плат.



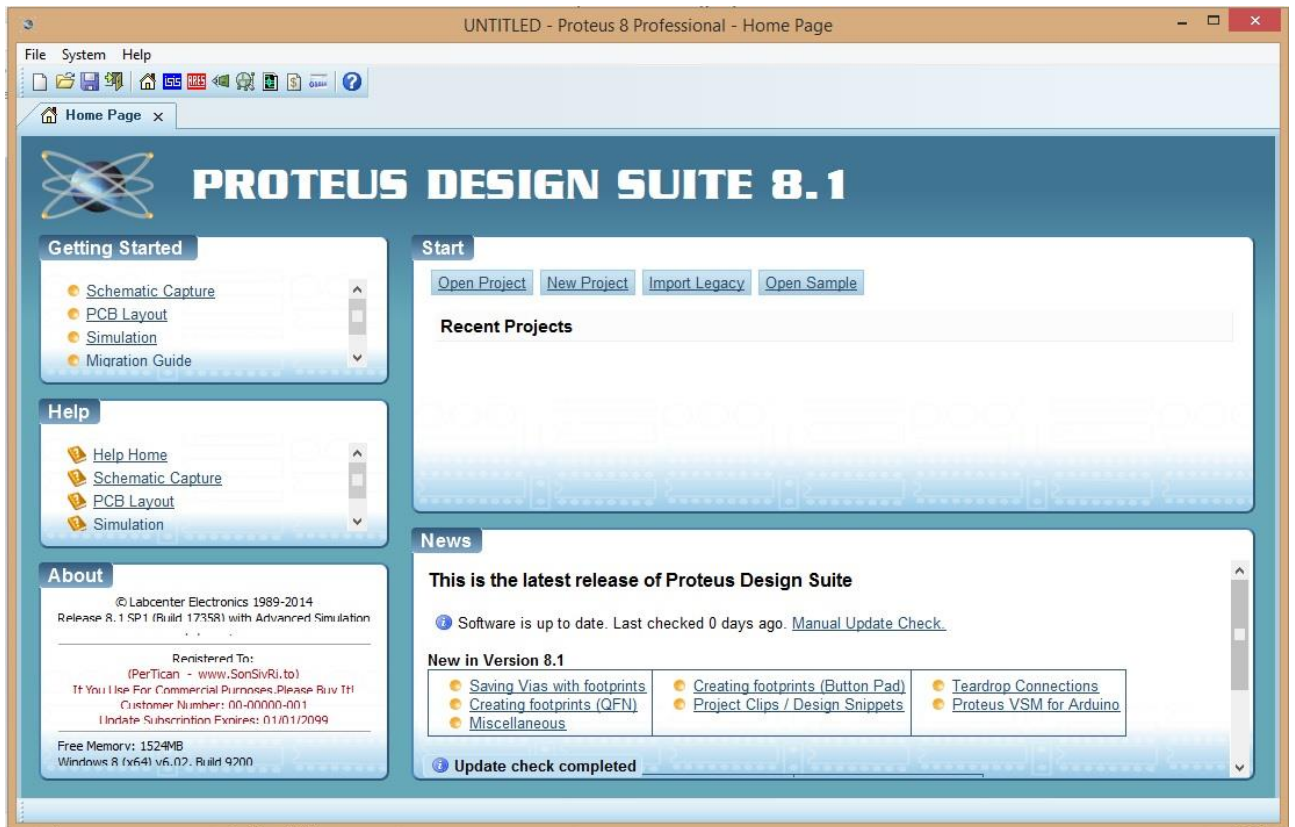


Рисунок 2.9 –Интерфейс работы на Proteus

Пакет Proteus состоит из нескольких частей, двух подпрограмм: ISIS это программа синтеза и моделирования электронных схем и ARES это программа разработки печатных плат. Вместе с программой устанавливается набор ознакомительных проектов на определенное время.

## 2.6 Code Vision AVR.

Code Vision AVR (CAVR)– это интегрированная среда разработки программного обеспечения для микроконтроллеров семейства AVR, разработанной фирмой Atmel.

CAVR имеет следующие компоненты:

- компилятор языка ассемблер для AVR;
- компилятор Си-подобного языка для AVR;
- генератор базового кода программы, допускающего сделать инициализацию зависимых устройств;
- модуль взаимодействия с программатором;
- модуль взаимодействия с отладочной платой STK-500;
- редактор исходного кода с подсветкой синтаксиса;
- терминал;
- Выходными файлами CAVR;

- HEX, BIN или ROM-файл для загрузки в микроконтроллер при помощи программатора;
- COFF файл, имеющий информацию для отладчика;
- OBJ файл, в котором хранится промежуточный код компиляции или объектный код;

CAVR по своей сути коммерческое программное обеспечение. Есть бесплатная временная версия с ограничением ряда возможностей для ознакомления, в частности, размер программного кода имеет ограничение 4-мя килобайтами и не имеет некоторые библиотеки(Рисунок 2.10).

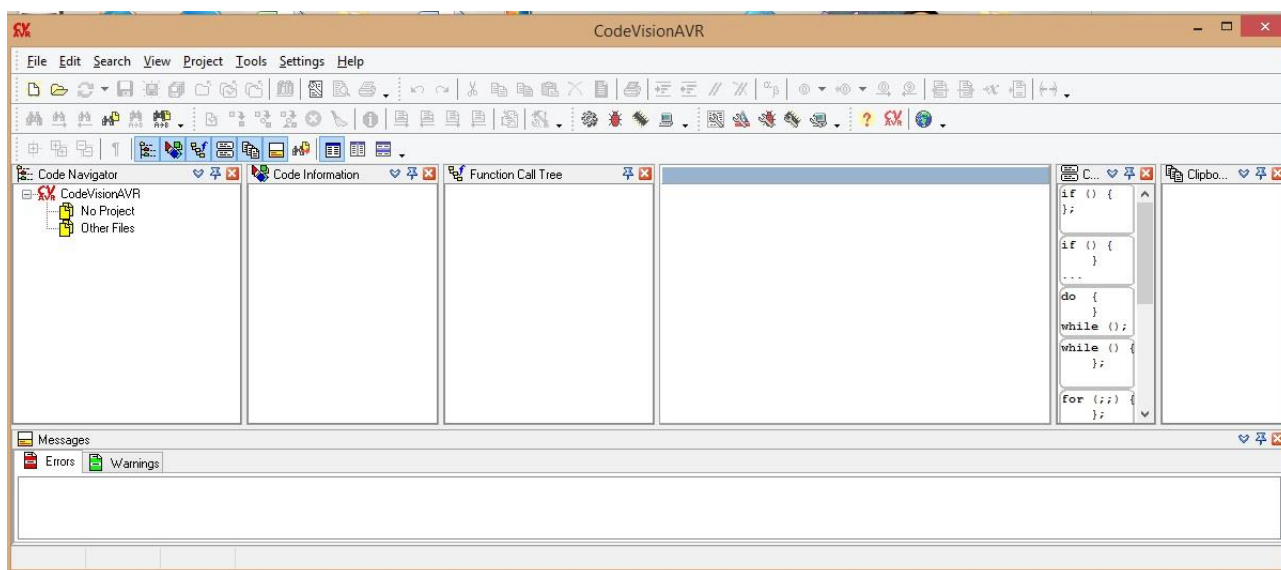


Рисунок 2.10 –Интерфейс работы на CAVR

## 2.7 Photoshop.

Сперва программа разрабатывалась для редактирования изображений и полиграфии, но в данное время она широко используется и везде. Photoshop имеет связи и с другими программами для обработки анимации и т.д. Совместно с такими программами, как Adobe ImageReady, Adobe Illustrator, Adobe Premiere, Adobe After Effects и Adobe Encore DVD, он используется для создания профессиональных DVD, осуществляет компоненты нелинейного монтажа и выявления специальных спецэффектов, как слои, фоны, текстуры и т. д. для телевидения, кинематографа и всемирной паутины. Photoshop также вводится в кругах разработчиков компьютерных игр (Рисунок 2.11).

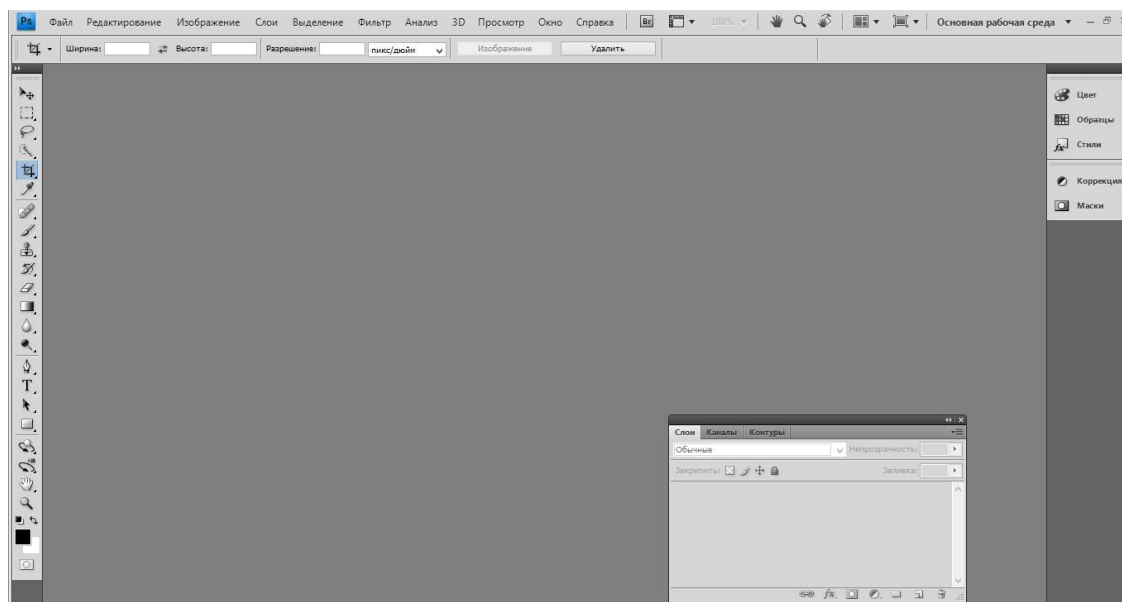


Рисунок 2.11 –Интерфейс работы на Photoshop

Главный формат Photoshop, это PSD, может быть экспортирован и импортирован всеми доступными продуктами, перечисленными выше. Photoshop поддерживает создание меню для DVD. Совместно с Adobe Encore DVD, Photoshop умеет создавать меню или кнопки DVD. Photoshop CS3 в версии Extended осуществляет также работу с 3-ми слоями. Из-за своей популярности Photoshop введена поддержка специфического формата PSD во многих графических программах, таких как Corel PHOTO-PAINT, GIMP, Corel aint Shop Pro и т.д.

Имеется обработка рисунков, с глубиной цвета 8 бит (256 градаций на один канал), 16 бит (используется 15 битов плюс один уровень, то есть 32769 уровней) и 32 бит (используются числа одинарной точности с плавающей запятой). Также имеется сохранение в файле различных элементов, как то: направляющих (Guide), каналов (например, канала прозрачности — Alpha channel), путей обтравки (Clipping path), слоёв, содержащих векторные и текстовые объекты. Файл может включать цветовые профили (ICC), функции преобразования цвета (transfer functions). Допускаются неквадратные пиксели (Pixel Aspect Ratio).

## Вывод

Во главе представлена схема действующего оборудования, управляемого пультом устройствами: бра, вентилятор, ЖК-дисплей, розетка, жалюзи. Дальность сигнал от пульта до макета может достигать до 250 метров. Представлено описание взаимодействия между оборудованием макета, показан принцип передачи и получения сигналов. Их обработка в микроконтроллере и отправка на ЖК-дисплей. Структура работы ЖК-дисплея и описание каждого окна. В главе приведен схематичный рисунок,

отображающий расположение устройств в корпусе. Также был проведён анализ используемого программного обеспечения.

Представлено теоретическое описание каждого программного продукта и рисунки с его использованием. Также качественный продукт отвечает самым современным стандартом.

## 3 Программная реализация

### 3.1 Схемотехническое моделирование

Использовалась система автоматизированного проектирования, позволяющая виртуально смоделировать работу большого количества аналоговых и цифровых устройств.

Я собрал схему системы (Рисунок 3.1) и виртуально симулировал в Proteus. Также данное программное обеспечение подготовило и распечатало мне рисунок для платы.

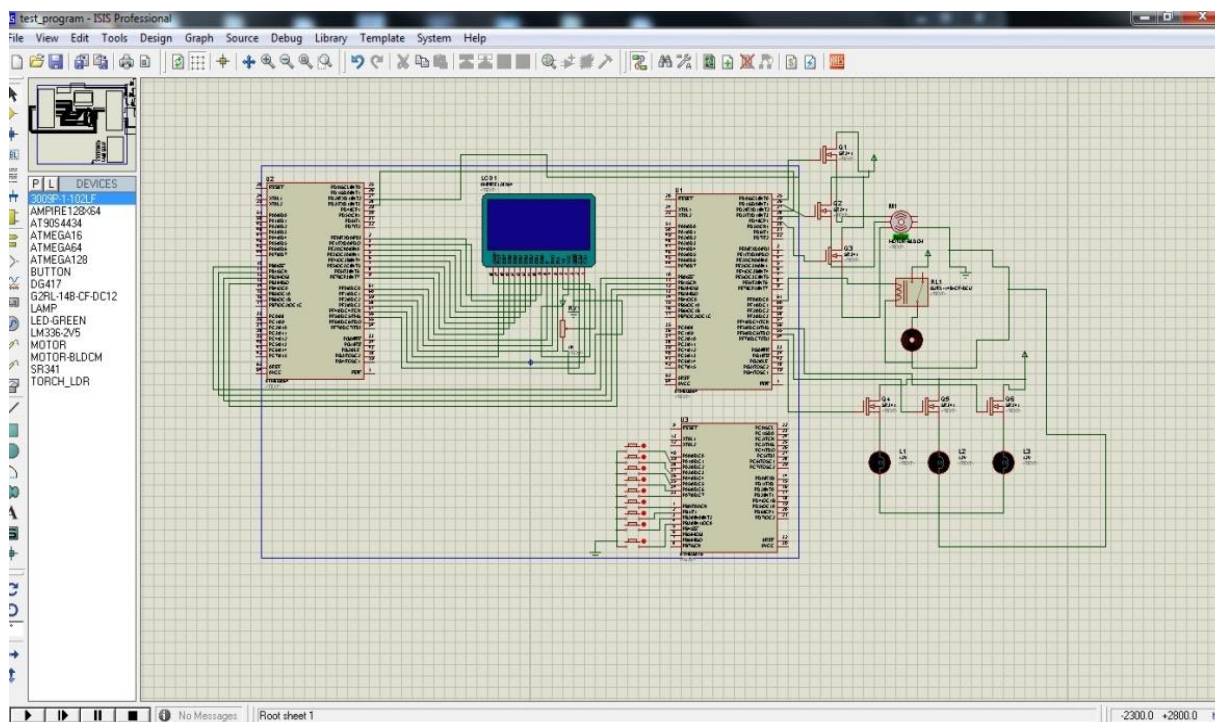


Рисунок 3.1 –Работа в Proteus

### 3.2 Компонентная программная реализация

Компонентная программная реализация производилась в интегрированной среде разработки программного обеспечения

CodeVisionAVR для микроконтроллеров AVR. Начало работы в среде CVAVR представлено на рисунке 3.2

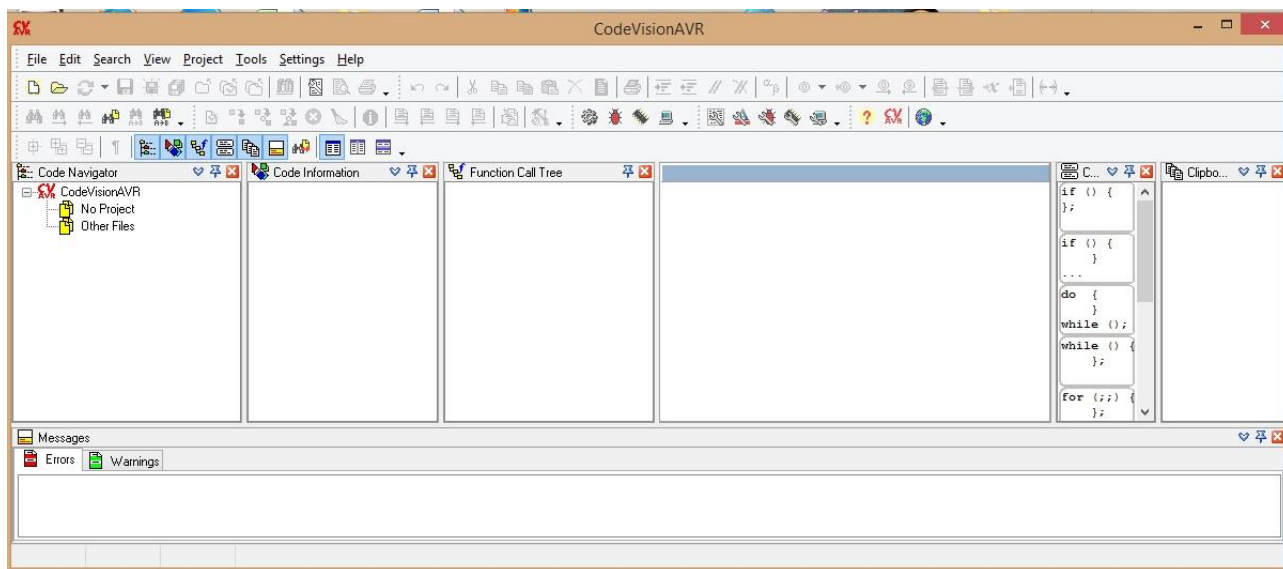


Рисунок 3.2 –Начало работы на CAVR

Создавал специальный проект и начинал программировать на языке C++. В первую очередь программировал микроконтроллер от пульта (Рисунок 3.3).

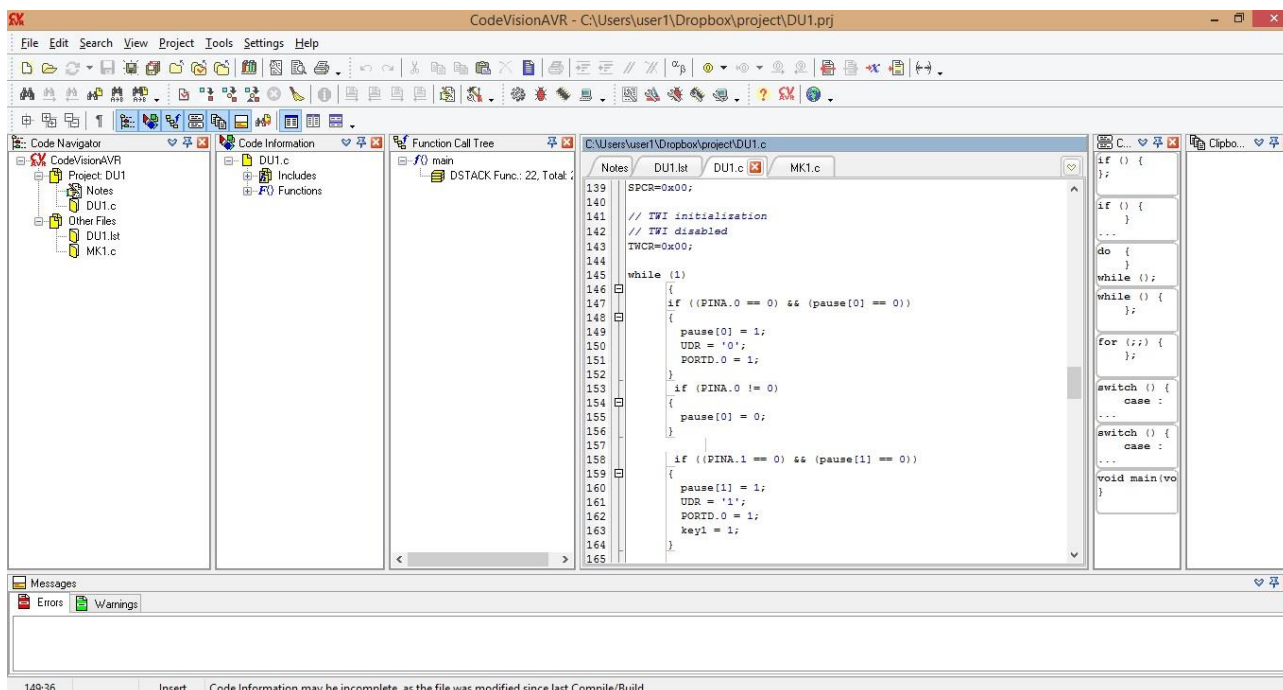


Рисунок 3.3 –Программирование пульта на CAVR

Далее программировал главный микроконтроллер, который управляет оборудованием на макете (Рисунок 3.4).

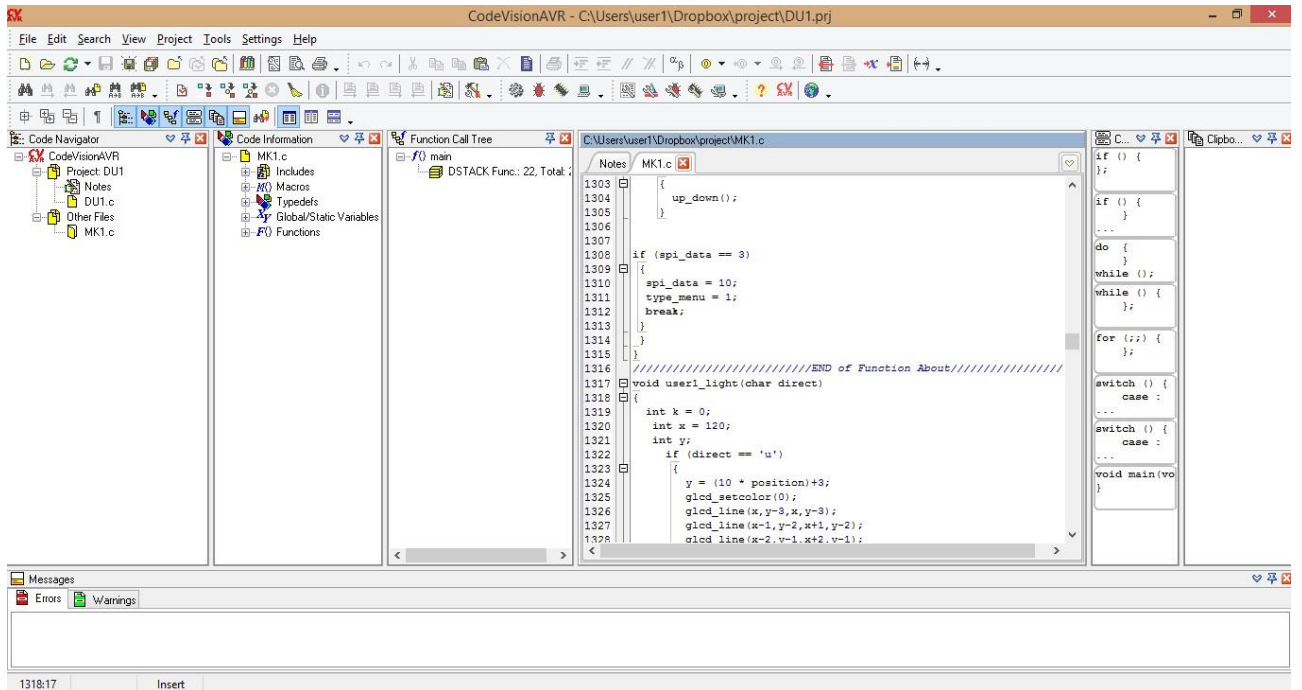
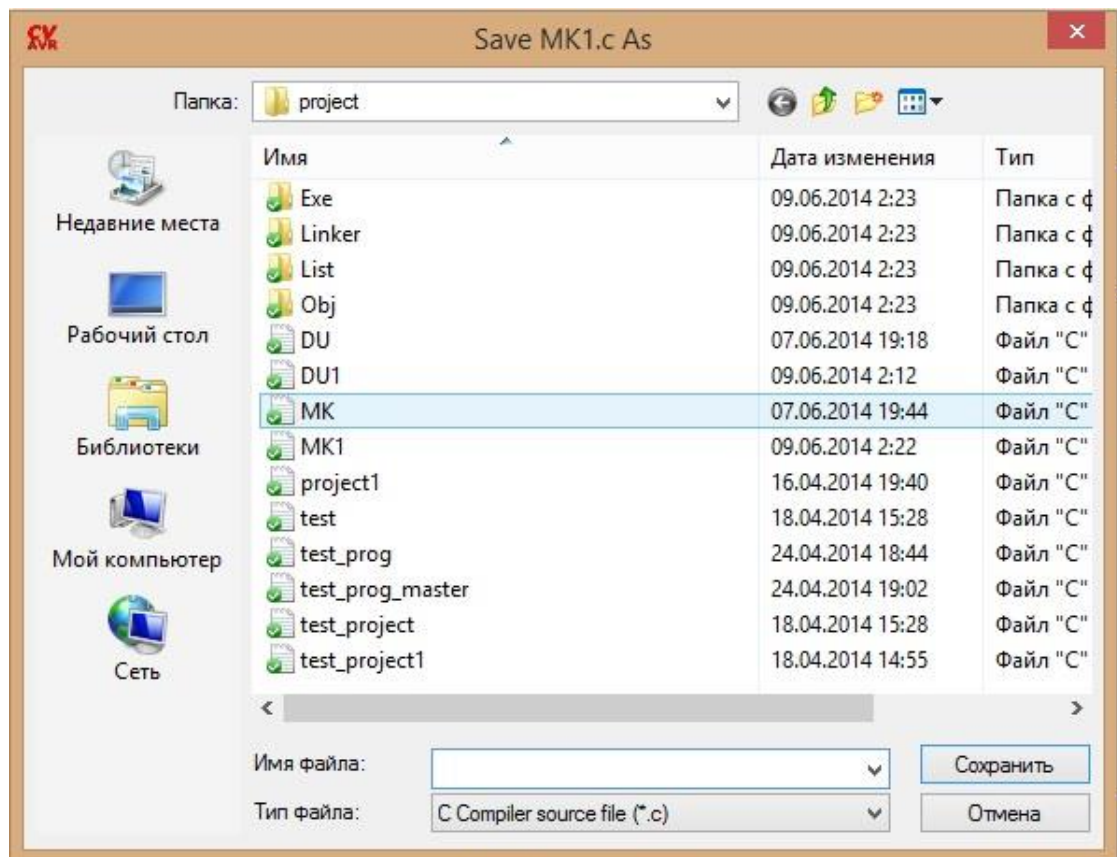


Рисунок 3.4 – Программирование микроконтроллера САVR

Сохраняем все проекты и готовые программы для программирования микроконтроллеров (Рисунок 3.5).



### 3.3 Библиотеки

#### *Библиотечные файлы \*.lib и \*.h в CVAVR*

В корневой папке CodeVisionAVR имеется каталог **lib**, в котором находятся файлы с расширением \*.lib ( например - LCD.LIB, DS1621.LIB и т.д. ), входящие в комплект поставки программы. Это Объектные файлы библиотеки (Object File Library). Они содержат определения функций – собственно реализацию. В каталоге **inc** находятся файлы с расширением \*.h (например LCD.H, DS1621.H и т.д.) входящие в комплект поставки программы. Это заголовочные файлы ( Header ); в них содержатся прототипы функций.

Header файлы собственно и подключаются в основном проекте:

```
#include <LCD.H>
```

Взаимосвязь этих, двух файлов происходит посредством директивы #pragma library name.lib:

*Заголовочный файл LCD.H*

```
#ifndef _LCD_INCLUDED_
#define _LCD_INCLUDED_
```

```
#pragma used+
```

*//прототипы функций.*

```
void _lcd_write_data( unsigned char data );
```

....

```
void lcd_putsf( char flash *str );
```

```
unsigned char lcd_init( unsigned char lcd_columns );
```

....

```
#pragma used
```

```
#pragma library lcd.lib //подключениеобъектногофайлабиблиотеки
```

```
#endif
```

#### *Библиотечные файлы \*.c и \*.h*

Это связка библиотечных файлов в языке C является общепринятой. \*.h – заголовочный файл, который содержит прототипы функций. Header файл может также содержать любое разнообразие макросов, структур и всевозможных объявлений:

```
extern unsigned char Buffer_UART[255]; //буферприема/передачи UART
```

```
extern unsigned int ADC_Channel[7]; //каналыАЦП
```

```
struct{
```

```
unsigned char control;
```

```
unsigned char dataOut;
```

```
unsigned char current;
```

```
unsigned ErrRcv;
```

```
unsigned ErrCRC;
```

```

unsigned DataRcv:1; } device;
#define setbit ( port,bits ) port.bits=1;
#define clearbit( port,bits ) port.bits=0;
void port_init( void );
void adc_init( void );

```

Файл с расширением \*.C содержит определение функции т.е. её полную реализацию:

```

void port_init( void )
{
  DDRD=0x84;
  PORTD=0xEF;
  DDRC=0x04;
  PORTC=0xf0;
}
void adc_init( void )
{
  //ИнициализацияАЦП
  ADMUX=MUX0; //ADCO
  ADCSRA=0xCE;
  SFIOR&=0xEF; }

```

Также CodeVisionAVR включает следующие библиотеки:

- Alphanumeric LCD modules for up to 4x40 characters.
- Philips I<sup>2</sup>C Bus.
- National Semiconductor LM75 Temperature Sensor.
- Maxim/Dallas Semiconductor DS1621 Thermometer/Thermostat.
- Philips PCF8563 and PCF8583 Real Time Clocks.
- Maxim/Dallas Semiconductor DS1302 and DS1307 Real Time Clocks.
- Maxim/Dallas Semiconductor 1 Wire protocol.
- Maxim/Dallas Semiconductor DS1820/DS18B20/DS1822 1 Wire

Temperature Sensors.

- Maxim/Dallas Semiconductor DS2430/DS2433 1 Wire EEPROMs.
- SPI.
- MMC/SD/SD HC FLASH Memory Card drivers and FAT12, FAT16, FAT32 access libraries.
- Power management.
- Delays.

### 3.4 Печатная плата и её изготовление

Процесс создания каркаса платы осуществляется четырьмя этапами:

1 Я создал прозрачный шаблон, на котором помечены контура соединения.

2 Просветил созданный шаблон.



3 Положил плату в «проявитель»: на плате размечены только те места, которые размечены на шаблоне.

4 Опускал плату в «травитель»: он съедает всю медь, исключив окрашенную.

#### *Создание схемы и её распечатка*

Первым шагом является создание печатной платы: создал схему того, что планируется на плате. Можно выбрать много сторонних программ, но в нашем случае выбирал Proteus. Нажимаю кнопку «печать» и подготавливаю всё к печати. Я использую лазерный принтер HP 1001. Далее я подготовил специальную плёнку и распечатал на неё схему. (Рисунок 3.6)



Рисунок 3.6 – Вид плёнки с макетом схемы

#### *Подготовка платы*

Для фоторезистивного процесса необходимо специальная плата, покрытая определенным веществом. Вырезаю из платы кусочек, подходящий параметрам схемы. Пользуюсь мини-дрелью для вырезания (Рисунок 3.7).

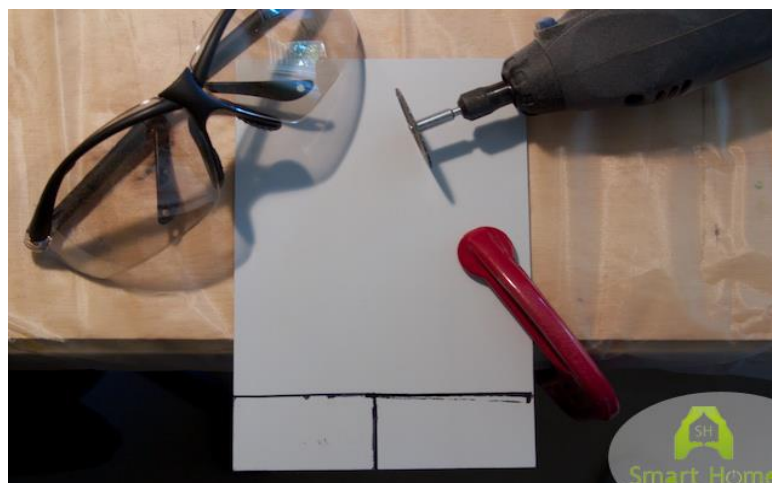


Рисунок 3.7 – Вид материала для платы и его нарезка

Снимаю с платы защитный слой (белая тонкая плёнка) и кладу её на основу. Плату укрываю плёнкой с распечаткой и прижимаю всё стеклом (Рисунки 3.8-3.9).

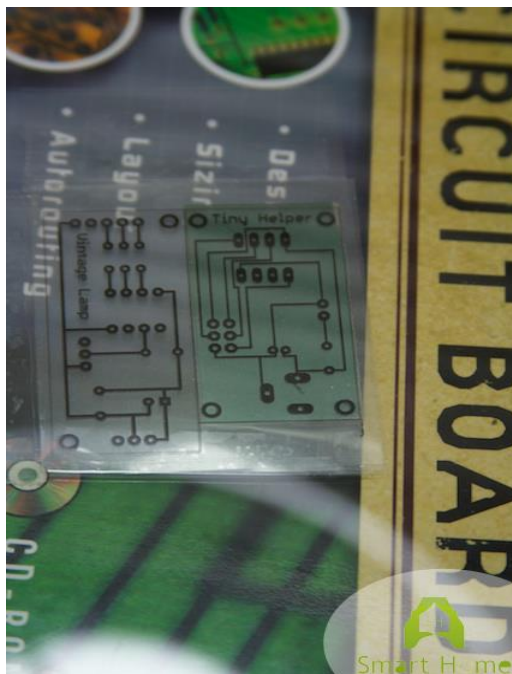


Рисунок 3.8 – Наложение схемы на плату



Рисунок 3.9 – Построение, лежащее под лампой накаливания

### Проявка

Во время экспонирования платы, разводится проявитель. На коробке проявителя расписана должная консистенция рекомендуемой температуры. На Рисунке 3.10 показан результат подготовки для травления платы.

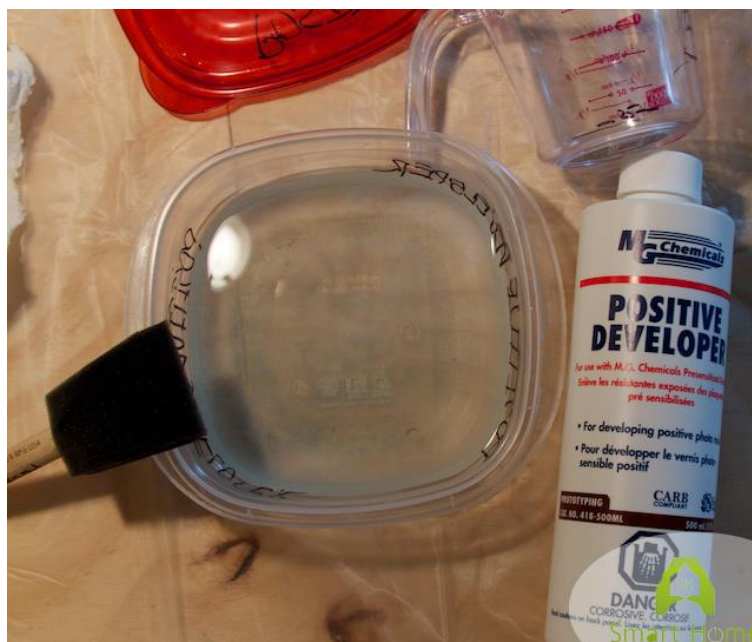


Рисунок 3.10 – Подготовка для травления

Проявитель подготавливается в течении десяти минут. Затем беру плату и опускаю её в проявитель. На Рисунке 3.11 показано всё необходимое для травления

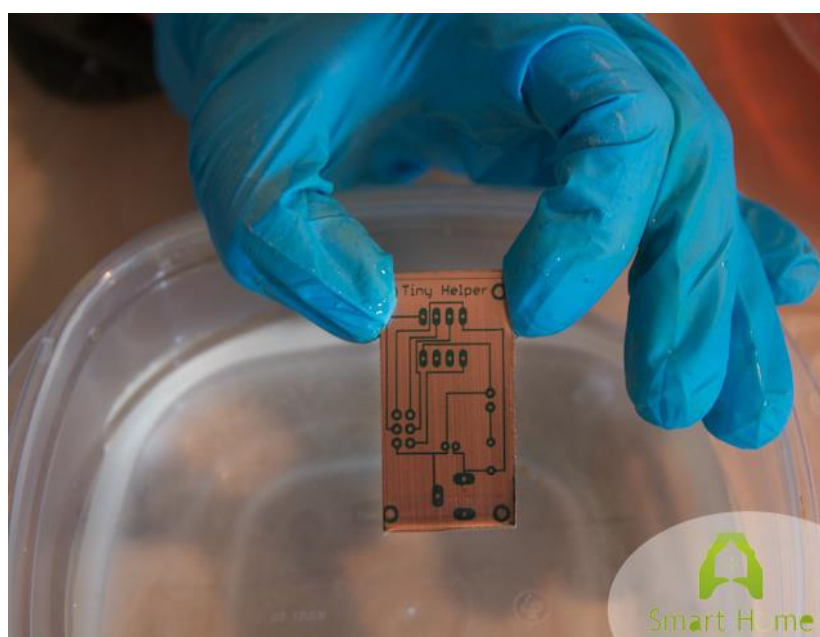


Рисунок 3.11 – Все необходимое для травления

### *Травление*

Прополаскиваю плату в воде и опускаю её в «травитель». Рекомендуемая температура – 50° С, но возможно травить в комнатной температуре. Травилось в течении минут 25 (Рисунок 3.12).



Рисунок 1.12 – Результат травления

### *Зачистка*

Остатки красителя легко удаляются специальными спиртованными салфетками. Беру все необходимые компоненты и припаиваю их к медной основе. Далее начинаю сверлить отверстия в плате и проделываю лужение. Все результаты операции видны на Рисунках 3.13-3.15.

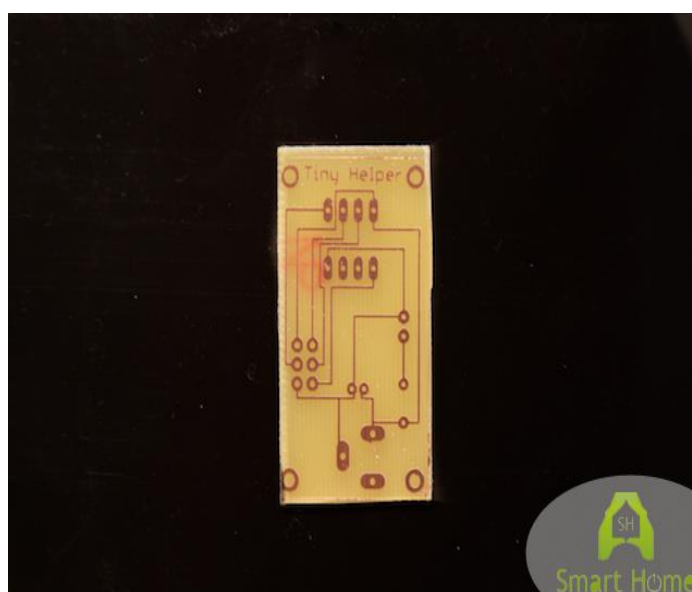


Рисунок 3.13 – Вид платы после зачистки

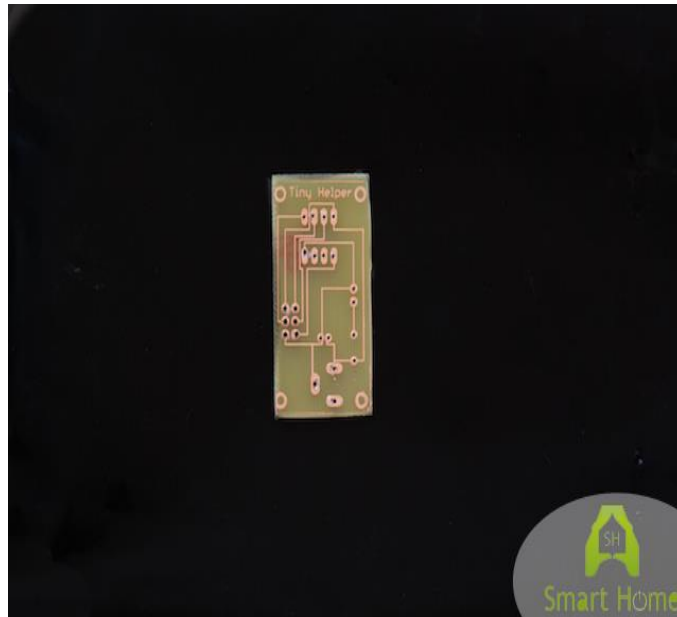


Рисунок 3.14 – Вид платы после сверления

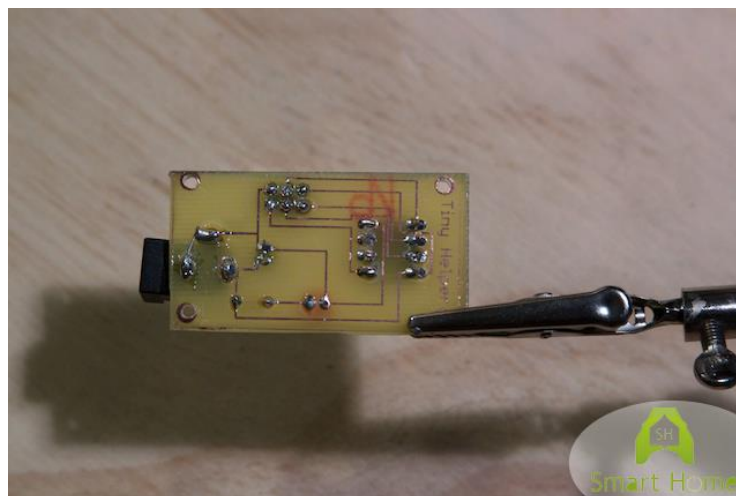


Рисунок 3.15 – Вид платы после лужения

### **3.5 Бытовые устройства на макете «Умный дом»**

В макете используется четыре бытовых прибора:

Первый прибор – бра (Рисунок 3.16). Бра крепится к ДСП с левой ближней стороны макета. Присоединяется устройство к микроконтроллеру через тиристор, который будет управлять подаваемой нагрузкой.

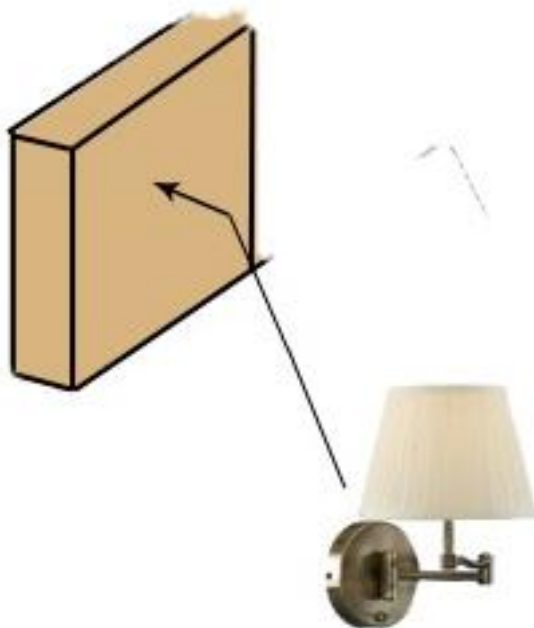


Рисунок 3.16 –Расположение бра на макете

Второй прибор – вентилятор (Рисунок 3.17). Он крепится к левой дальней стороне ДСП и присоединен при помощи электромагнитного реле.

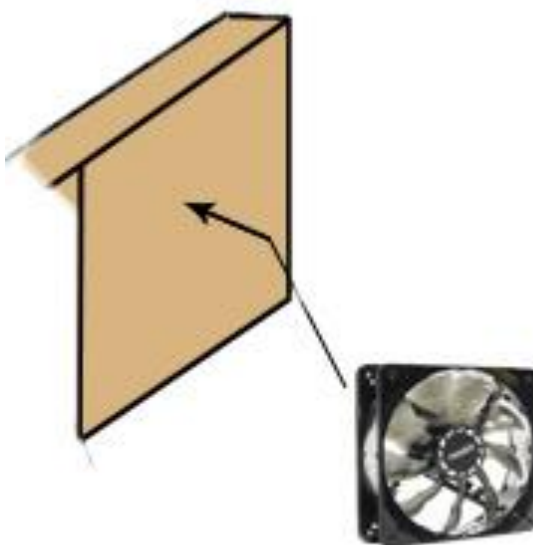


Рисунок 3.17 –Расположение вентилятора на макете

Третий прибор – это розетка (Рисунок 3.18). Она крепится к правой дальней стороне ДСП и присоединена при помощи электромагнитного реле.

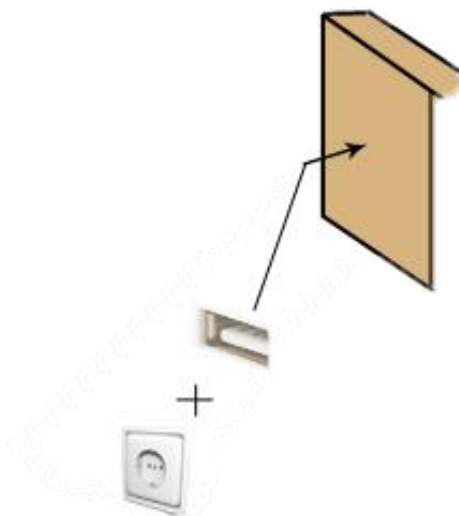


Рисунок 3.18 – Расположение розетки на макете

Четвертый прибор – жалюзи с электромотором (Рисунок 3.19). Оно крепится к правой ближней стороне ДСП и соединен с микроконтроллером напрямую.

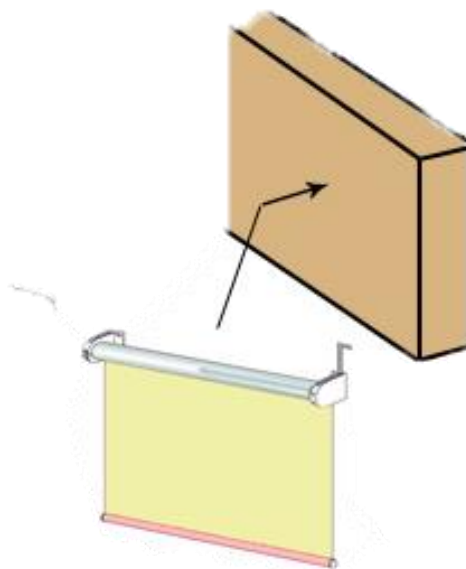


Рисунок 3.19 – Расположение жалюзи на макете

Пятый прибор – ЖК-дисплей (Рисунок 3.20). Он крепится к центральной стороне ДСП и соединен с микроконтроллером напрямую.

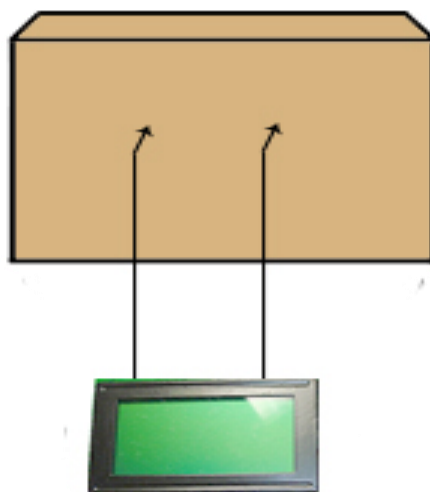


Рисунок 3.20 –Расположение ЖК-дисплея на макете

### 3.6 Общий концепт макета

Весь макет изображается на Рисунке 3.21. На нём видно, что все приборы установлены с передней части макета, а все связующие шнуры и плата с микроконтроллером на задней стороне.

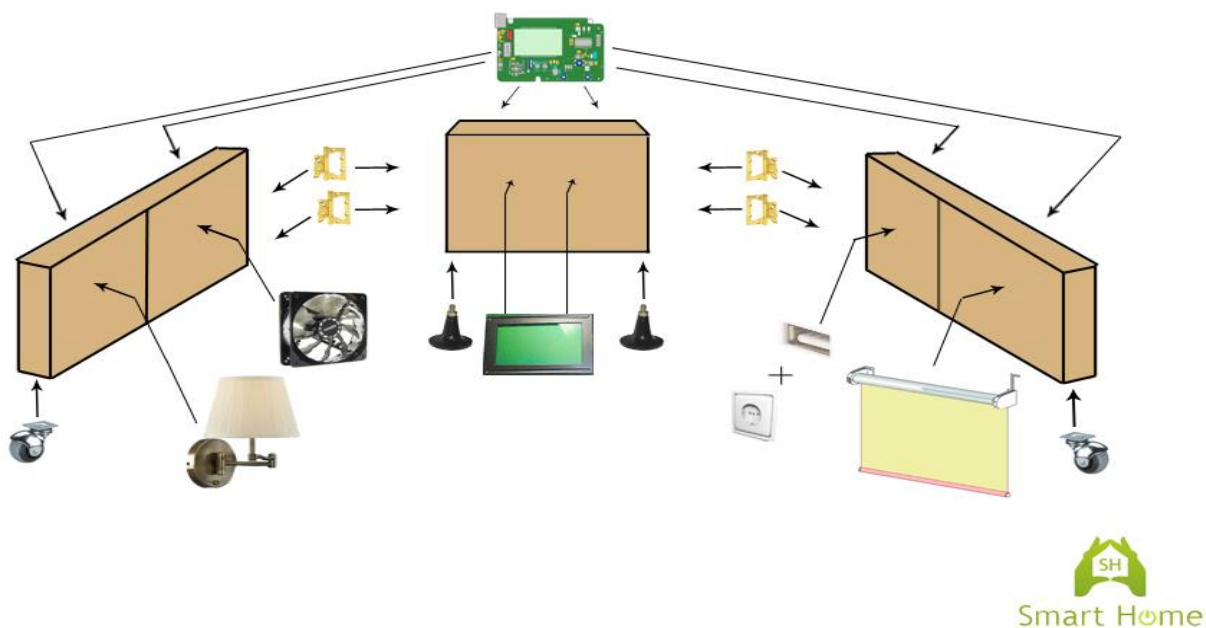


Рисунок 3.21 –Общий концепт макета

### 3.7 Представление кода



Как я уже писал, среда программирования в моей работе является CAVR. Примеры программирования кода рассмотрены в Рисунках 3.22 и 3.23.

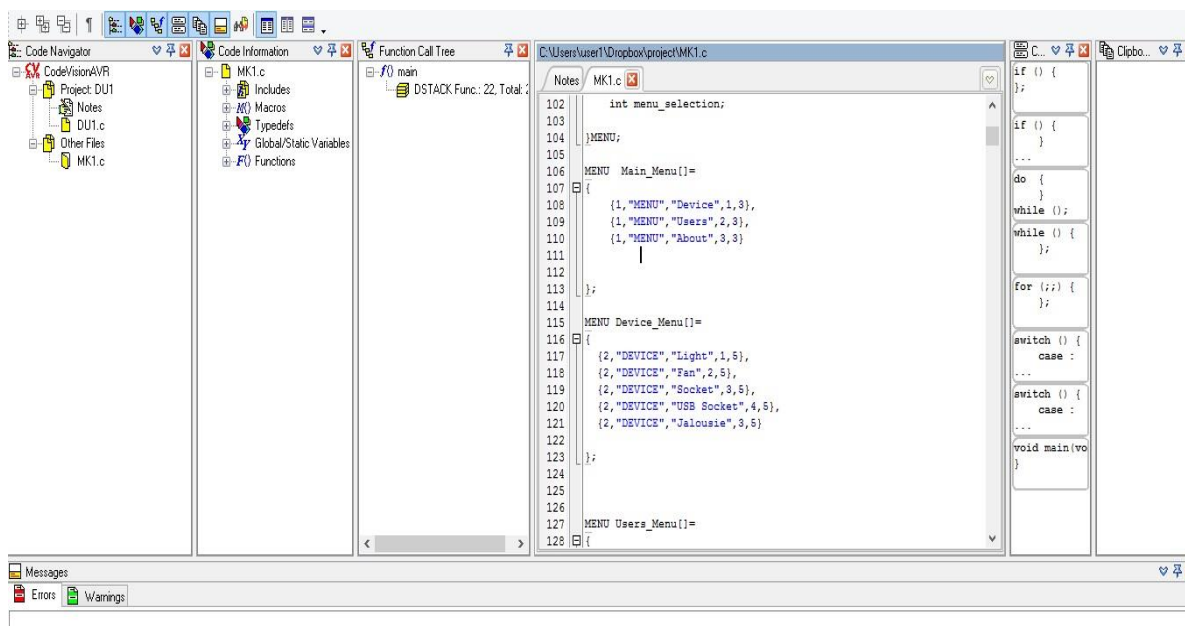


Рисунок 3.22 –Программирование в средеCAVR (Часть 1)

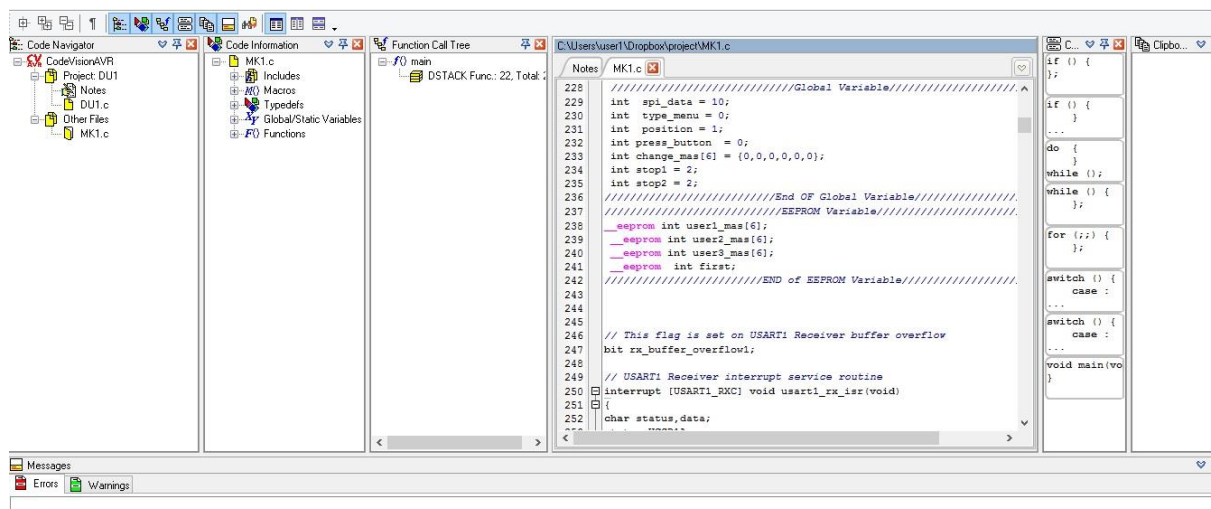


Рисунок 3.23 –Программирование в средеCAVR (Часть 2)

## Вывод

Во главе представлено краткое описание моделирования и компонентная программная реализация. Показан метод травления и лужения платы, подсоединения устройств к микроконтроллеру и общее представление макета. Также приведены примеры библиотек среды разработки программирования и непосредственно примеры использования.

## 4 Технико-экономическое обоснование проекта

### 4.1 Описание работы и обоснование необходимости

Тема данного дипломного проекта – «Разработка интерактивной обучающей системы для курса «Проектирование баз данных».

Цель данного проекта – разработать обучающую систему, которая упростила бы процесс изучения баз данных в университете.

Так как этот предмет является одним из самых сложных в курсе программ курса, разработка данной системы создаст принципиально новый подход к обучению, упростит контроль преподавателя над успеваемостью студентов, позволит более доступным и интересным языком изложить сложный материал.

В данном разделе приводится рассмотрение экономической составляющей реализации данного продукта, отражающее временные, трудовые и финансовые затраты на проект.

### 4.2 Трудовые ресурсы, используемые в работе

В работе задействованы:

- руководитель – постановка задачи, разработка основных бизнес правил для работы проекта;
- инженер-разработчик – разработка сайта, проектирование базы данных;
- консультанты по экономической части и по части ОБЖД;

Общее количество сотрудников и их заработная плата представлены в таблице 4.1.

Т а б л и ц а 4 . 1 – Количество задействованных в проекте работников, и их заработная плата

Исполнитель	Количество, человек	Месячная заработная плата, тенге
Руководитель	1	100000
Консультант по части «Экономика»	1	70000
Консультант по части «Безопасность жизнедеятельности»	1	75000
Инженер-разработчик	1	150000
Итого	4	395000

### 4.3 Расчет стоимости работы по проектированию и разработке

Разработка многомодульного программного продукта – сложный и трудоемкий процесс, требующий наряду с интеллектуальными, техническими затратами и финансовыми затратами. Поэтому необходимым является произведение расчета стоимости этой разработки. Затраты на разработку данного программного комплекса определяется по формуле

$$C = \Phi OT + C_H + A + \mathcal{E} + C_{np} + H \quad (4.1)$$

где  $\Phi OT$  – фонд оплаты труда;

$C_H$  – социальный налог;

$A$  – амортизационные отчисления;

$\mathcal{E}$  – затраты на электроэнергию;

$C_{np}$  - прочие расходы;

$H$  – накладные расходы.

Необходимый фонд оплаты труда рассчитывается по формуле

$$\Phi OT = Z_{осн} + Z_{доп} \quad (4.2)$$

где:  $Z_{осн}$  - основная заработная плата;  $Z_{доп}$  - дополнительная заработная плата;

В процессе работы задействованы:

– Руководитель – постановка задачи, разработка основных бизнес правил для работы проекта.

– Инженер-разработчик – разработка приложений, проектирование дизайна, сопровождение системы.

– Консультанты по экономической части и по части ОБЖД.

Общее количество сотрудников и их заработная плата представлены 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	Месячная заработная плата, тенге
Руководитель	1	80000
Консультант по части “Экономика”	1	75000
Консультант по части “Безопасность жизнедеятельности”	1	60000
Инженер-разработчик	1	125000
Итого	4	340000

Труд сотрудников института, задействованных в работе, оплачивается согласно положению о заработной плате АУЭС, труд программиста-разработчика принят условно, на договорной основе в размере 120000 тенге.

Базовый показатель для определения составляющих затрат труда вычисляется по формуле

$$Q = q * c \quad (4.3)$$

где Q – условное число команд,  
 q=4900 - коэффициент, учитывающий условное число команд в зависимости от типа задачи,  
 c=1,26- коэффициент, учитывающий новизну и сложность программы.

$$Q = 4900 * 1.26 = 6174.$$

Выбор значения коэффициента q был произведен из таблицы 3.3

Т а б л и ц а 4 . 3 – Значения коэффициента q

Тип задачи	Пределы изменений коэффициента
Задачи учета	От 1400 до 1500
Задачи оперативного управления	От 1500 до 1700
Задачи планирования	от 3000 до 3500
Много вариантыные задачи	от 4500 до 5000
Комплексные задачи	от 5000 до 5500

По степени новизны программные продукты могут быть отнесены к одной из 4-х групп:

- группа А - разработка принципиально новых задач;
- группа Б - разработка оригинальных программ;
- группа В - разработка программ с использованием типовых решений.
- группа Г - разовая типовая задача.

Выбор коэффициента с производится из таблицы 3.4, на пересечении групп сложности и степени новизны.

Т а б л и ц а 4 . 4 – Коэффициенты расчета трудоемкости

Язык программ.	Группа сложности	Степень новизны				Коэффициент В
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	0,69	1,2
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,5

Низкого уровня	1	1,58	1,45	1,32	0,79	1,2
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,5

При разработке многомодульного программного продукта используется техническое задание, согласно которому выполнение работ происходит последовательно по заданным пунктам, с учетом сроков их исполнения. График выполнения работ по разработке программного продукта представлен в таблице 4.5

Т а б л и ц а 4 . 5 – График выполнения работ по разработке программного продукта

Код работы	Наименование работы	Ожидаемая длительность, дни	Обозначение
1	Подготовка описания задачи	4	T <sub>ПЗ</sub>
2	Описание задачи	3	T <sub>ОЗ</sub>
3	Разработка алгоритма	5	T <sub>А</sub>
4	Установка и запуск ПО	3	T <sub>ЛС</sub>
5	Проектирование и создание макета	6	T <sub>СМ</sub>
6	Разработка и создание платы с микроконтроллером	6	T <sub>М</sub>
7	Разработка и создание интерфейса системы	5	T <sub>ИН</sub>
8	Разработка основной части программы	6	T <sub>ОСН</sub>
9	Тестирование приложения и основных модулей с целью выявления ошибок	2	T <sub>ТЕСТ</sub>
10	Составление технической документации	3	T <sub>ТЕХ</sub>
11	Подготовка раздела “Экономика”	9	T <sub>Э</sub>
12	Подготовка раздела “Безопасность жизнедеятельности”	9	T <sub>БЖ</sub>
Всего		61	

Так как участники, задействованные в проекте, работают в различные промежутки времени, в течение которого реализуется проект, необходимо произвести расчет дневной и почасовой оплаты труда.

Заработная плата каждого работника за один рабочий день рассчитывается по формуле

$$D = \frac{O}{n} \quad (4.3)$$

где, O – оклад работника в тенге;

n – количество дней в рабочем месяце(это 24 дня – шестидневная рабочая неделя)

для руководителя

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

для консультанта по части “Экономика”

$$D = \frac{75000}{24} = 3125 \text{ тенге/день}$$

для консультанта по части “БЖД”

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

для инженера разработчика

$$D = \frac{125000}{24} = 5200 \text{ тенге/день};$$

Заработная плата за один час рассчитывается по формуле

$$H = \frac{D}{z} \quad (4.4)$$

где:  $D$  – заработная плата работника за один рабочий день,  
 $z$  – количество часов рабочего дня (8 часов):

для руководителя

$$H = \frac{3333}{8} = 416,6 \text{ тенге/час};$$

для консультанта по части “Экономика”

$$H = \frac{3125}{8} = 390,6 \text{ тенге/час};$$

для консультанта по части “БЖД”

$$H = \frac{2500}{8} = 312,5 \text{ тенге/час};$$

для инженера разработчика

$$H = \frac{5200}{8} = 650 \text{ тенге/час};$$

Время рассчитывается в человеко-часах, причем  $T_{ПЗ}$  берется по фактически отработанному времени, а время остальных этапов определяется расчётно, по условному числу команд  $Q$ .

Определяем время, затраченное на каждый этап создания программного продукта

$T_{ПЗ}$  (время на подготовку описания задачи), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ПЗ} = 32 \text{ чел/час}$$

$T_{ОЗ}$  (время на описание задачи) определяется по формуле

$$T_{ОЗ} = \frac{Q * B}{50 * K} \quad (4.5)$$

где  $B=1,2$  - коэффициент учета изменений задачи, коэффициент  $B$  зависимости от сложности задачи и числа изменений выбирается в интервале от 1,2 до 1,5 (таблица 4.6)

$K=1,2$  - коэффициент, учитывающий квалификацию программиста.

$$T_{ОЗ} = \frac{6174 * 1,5}{50 * 1,2} = 148,2 \text{ чел/час.}$$

Т а б л и ц а 4 . 6 – Коэффициенты квалификации программиста

Опыт работы	Коэффициент квалификации
До двух лет	0.8
2-3 года	1
3-5 лет	1.1 – 1.2
5-7 лет	1.3 – 1.4
более 7 лет	1.5 – 1.6

3  $T_A$  (время на разработку алгоритма) рассчитываем по формуле

$$T_A = \frac{Q}{50 * K} \quad (4.6)$$

$$T_A = \frac{6174}{50 * 1,2} = 123,5 \text{ чел/час.}$$

4  $T_{ЛС}$  (Установка и запуск ПО) определяется аналогично  $T_A$

$$T_{ЛС} = 123,5 \text{ чел/час}$$

5.  $T_{CM}$  (время на проектирование и создание макета) определяется по формуле

$$T_{CM} = \frac{3500 \cdot 1,5}{50 \cdot 1,2} = 87,5 \text{ чел/час}$$

6.  $T_M$  (время на разработку и создание платы с микроконтроллером) определяется по формуле

$$T_M = \frac{3500 \cdot 1,5}{50 \cdot 1,2} = 87,5 \text{ чел/час}$$

1  $T_{IH}$  (Разработка и создание интерфейса программы) определяется по формуле

$$T_{IH} = \frac{6174 \cdot 1,5}{50 \cdot 1,2} = 154,4 \text{ чел/час}$$

2  $T_{OCH}$  (время на разработку основной части), определяется по формуле

$$T_{OCH} = \frac{4500 \cdot 1,5}{50 \cdot 1,2} = 112,5 \text{ чел/час}$$

3  $T_{ТЕСТ}$  (время на тестирование), определяется по формуле

$$T_{ТЕСТ} = \frac{3000 \cdot 1,5}{50 \cdot 1,2} = 75 \text{ чел/час}$$

4  $T_{ТЕХ}$  (время на составление технической документации), берется по факту и составляет (от 3-х до 5-ти дней по 8 часов)

$$T_{ТЕХ} = 24 \text{ чел/час}$$

5  $T_{Э}, T_{БЖ}$  (время на подготовку разделов Экономика и БЖ), берется по факту и составляет (от 7-и до 10-ти дней по 8 часов)

$$T_{Э} = T_{БЖ} = 72 \text{ чел/час}$$

Суммарные затраты труда рассчитываются как сумма составных затрат труда по формуле 4.10

$$\begin{aligned} T_{СТ} &= 32 + 148,2 + 123,5 + 123,5 + 87,5 + 87,5 + 154,4 + 112,5 + 75 + 24 + 72 + 72 \\ &= 1094 \text{ чел/час} \end{aligned}$$



Таким образом, суммарная основная заработная плата составит

$$З_{осн} = З_{инж} + З_{рук} \quad (4.8)$$

$$З_{осн} = 1094 * 650 + 72 * 312,5 + 72 * 390,6 + 80 * 416,6 = 759896 \text{ тенге}$$

Дополнительная заработная плата в среднем определяется в размере 10% от основной заработной платы и рассчитывается по формуле

$$З_{доп} = З_{осн} * 10\% \quad (4.9)$$

$$З_{доп} = 759896 * 10\% = 75990 \text{ тенге}$$

Общий фонд оплаты труда согласно формуле 4.2

$$ФОТ = 759896 + 75990 = 835886 \text{ тенге};$$

Социальный налог составляет 11% от ФОТ и рассчитывается по формуле

$$С_H = (ФОТ - ПО) * 11\% \quad (4.10)$$

где: ПО (пенсионные отчисления) составляют 10% от ФОТ и рассчитываются по формуле

$$ПО = ФОТ * 10\% \quad (4.11)$$

Размер пенсионных отчислений согласно формуле 4.11 составляет

$$ПО = 835886 * 10\% = 83589 \text{ тенге};$$

Отчисления по социальному налогу согласно формуле 4.10

$$С_H = (835886 - 83589) * 11\% = 82752,7 \text{ тенге}$$

#### **4.4 Расчет затрат на амортизацию**

Амортизационные отчисления производятся по установленным нормам амортизации, выражаются, в процентах к балансовой стоимости оборудования и рассчитываются по формуле

$$A = \frac{C_{обор} * H_A * N}{100 * 12 * t} \quad (4.12)$$

где:  $H_A$  - норма амортизации;

$C_{обор}$  - первоначальная стоимость оборудования;

$N$  - количество дней на выполнение работ;

$t$  - общее время использования персонального компьютера;

Норма амортизации  $H_A$ , рассчитывается по формуле

$$H_A = \frac{C_{ОБОР} - C_{ЛИКВ}}{T_{НОРМ} * C_{ОБОР}} * 100\% \quad (4.13)$$

где:  $C_{ЛИКВ}$  - ликвидационная стоимость, составляет 5.61% от стоимости оборудования;  $T_{НОРМ}$  - нормативный срок службы (для персонального компьютера – 4 года) в таблице 4.7.

Т а б л и ц а 4 . 7 – Оборудование, используемое для разработки ПП:

Наименование	Модель	Стоимость,тг
Процессор	<b>HP P6-2419ER D7B84EA</b>	20 910
Монитор	<b>Samsung S22C150NS</b>	23 660
Клавиатура	<b>A4tech KRS-8572</b>	1 810
Компьютерная мышь	<b>HP H2C21AA</b>	1500
Модем	D-link DSL-2500	1 810
Макет с оборудованием	IliasPro limited	4230.5
Итого		53920.5

Ликвидационная стоимость рассчитывается по формуле

$$C_{ЛИКВ} = 0,056 * C_{ОБОР} \quad (4.14)$$

$$C_{ЛИКВ} = 0,056 * 161050 = 9034,9 \text{ тг.}$$

Общее время использования персонального компьютера учитывает лишь время работы на компьютере и рассчитывается по формуле:

$$t = 123,5 + 87,5 + 87,5 + 154,4 + 112,5 + 75 = 640,4 \text{ чел/час} = 80 \text{ дней}$$

$$H_A = \frac{161050 - 9034,9}{4 * 161050} = 23,5 \%$$

$$A = \frac{53920,5 * 23,5 * 80}{100 * 12 * 137} = 616,61 \text{ тенге}$$

Затраты на электроэнергию вычисляется по формуле

$$\mathcal{E} = M * K_3 * T * C_{кВт\cdotч} \quad (4.15)$$

где:  $M$  – мощность ЭВМ (600 Вт=0,6 кВт);

$K_z$  - коэффициент загрузки (0.8);

$C_{кВт\cdotч}$  - 14,935 тг– стоимость 1 кВт-час электроэнергии;

$T$  - время работы;

$$\mathcal{E} = 0,6 * 0,8 * 14,935 * 640,4 = 4591 \text{ тенге}$$

Расходы на материалы и комплектующие, используемые в процессе написания программного продукта ( $C_{Мик}$ ), а также затраты на техническое обслуживание и ремонт ( $C_{ТО}$ ) составляют, соответственно, 2,06% и 2,266% от стоимости оборудования – формулы(4.16 – 4.17)

$$C_{мик} = 0,0206 * C_{обор} \text{ тенге} \quad (4.16)$$

$$C_{то} = 0,02266 * C_{обор} \text{ тенге} \quad (4.17)$$

$$C_{Мик} = 0,0206 * 161050 = 3318 \text{ тенге}$$

$$C_{ТО} = 0,02266 * 161050 = 3649 \text{ тенге}$$

$$C_{пр} = C_{мик} * C_{то} \text{ тенге} \quad (4.18)$$

$$C_{ПП} = 3318 + 3649 = 6967 \text{ тенге}$$

Накладные расходы, связанные с управлением и обслуживанием, содержанием и эксплуатацией оборудования и прочими дополнительными затратами на обеспечение процессов производства и обращения, составляют 50% от всех затрат, вычисляются по формуле

$$H = (\Phi OT + C_H + A + \mathcal{E} + C_{пр}) * 0,5 \quad (4.19)$$

$$H = (835886 + 82752,7 + 616,61 + 4591 + 6967) = 465715 \text{ тенге}$$

Таким образом, затраты на разработку данного программного комплекса согласно формуле 4.1 равны

$$C = 835886 + 82752,7 + 616,61 + 4591 + 6967 + 465715 = 1397144,8 \text{ тенге}$$

Сводные результаты расчета себестоимости программного продукта предоставлены в таблице 4.8 и на рисунке 4.1:

Т а б л и ц а 4 . 8 – Себестоимость разработки программного обеспечения умного дома:

Статья расходов	Сумма, тенге.	В процентах от общей суммы, %
ФОТ	835886	60
С <sub>Н</sub>	82752,7	6,7
А	616,61	0,1
Э	4591	0,5
С <sub>пр</sub>	6967	0,7
Н	465715	33
Итого:	1397144,8	100



Рис 4.1 - Структура себестоимости программного обеспечения

#### 4.5 Цена программного продукта

Минимальная цена программного продукта рассчитывается по следующей формуле

$$C_{min} = (C_{пп} + C_{мд} + C_3)(1 + P/100) \text{ д.ед.} \quad (21)$$

где  $C_{пп}$ , д.ед – себестоимость разработки программного продукта;  $C_{мд}$ , д.ед - цена магнитного диска;  $C_3$ , д.ед - стоимость;  $P, \%$  - планируемый процент рентабельности по отношению к себестоимости.

Тогда по формуле 21

$$C_{min} = (1397144,8 + 193,8 + 142,8) * (1 + 0,1938) = 1668314 \text{ тг.}$$

Далее определяется цена реализации с учётом налога на добавленную стоимость (НДС) (формула 22)

$$C_p = C_{min} + \text{НДС}. \quad (22)$$

Ставка НДС на 2013 год, согласно статьи 268 Налогового кодекса РК составляет 12 %.

$$C_p = 1668314 + 200526 = 1868840 \text{ тг.}$$

### **Вывод**

Себестоимость разработки программного обеспечения умного дома, составляет 1868840 тг. Себестоимость данного продукта сформирована в основном за счёт фонда оплаты труда и накладных расходов, что составляет 835856 тенге (60%) и 465715 тенге (33%) соответственно. Средняя цена на рынке на аналогичные продукты составляет от 1200000 до 2000000 тенге. При таких условиях цена 1868840 тг является приемлемой для рынка программного обеспечения.

## 5 Безопасность жизнедеятельности

### 5.1 Анализ потенциально опасных и вредных производственных факторов проектируемого объекта, воздействующих на персонал

Источниками повышенной опасности могут служить следующие элементы:

- распределительный щит;
- источники питания;
- блоки ПЭВМ и печати, находящиеся в ремонте.

В соответствии с СНиП 4.02–42–2006[20] к легкой физической работе относятся все виды деятельности, производимые сидя и не требующие физического напряжения. Работа пользователя разработанного пакета программ относится к категории 1а.

Согласно СНиП 4.02–42–2006помещение для ЭВМ по степени опасности поражения человека электрическим током относится к помещениям без повышенной опасности (нет токопроводящих полов, сырости, повышенной температуры, возможности одновременного прикосновения к корпусам оборудования с «землей» и к токонесущим частям).

В соответствии с СНиП 4.02–42–2006при обслуживании ЭВМ персонал может подвергаться воздействию потенциально опасных физических и психофизиологических опасных и вредных производственных факторов:

- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;

В соответствии с СНиП 4.02–42–2006на рабочем месте оператора допускаются следующие уровни напряжений прикосновения и токов, значения которых приведены в таблице 5.1.

Т а б л и ц а 5.1 – Предельно допустимые значения напряжений прикосновения и токов

Род тока	U (не более), В	I, мА
Переменный, 50 Гц	2.0	0.3
Постоянный	8.0	1.0

- повышенный уровень статического электричества;

Электризация материалов часто препятствует нормальному ходу технологических процессов производства, а также создает дополнительную пожарную опасность вследствие искрообразования при разрядах при наличии в помещениях, резервуарах и ангарах горючих паро- и газо-воздушных смесей.

В ряде случаев статическая электризация тела человека и затем последующий разряд с человека на землю или заземленное производственное оборудование, а также электрический разряд с незаземленного оборудования

через тело человека могут вызвать болевые и нервные ощущения и быть причиной непроизвольного резкого движения в результате которого человек может получить травму (падения, ушибы и т.д.).

– повышенный уровень электромагнитных излучений;

В СанПиН энергетическая экспозиция за рабочий день (рабочую смену) не должна превышать значений, указанных в таблице 5.2.

Т а б л и ц а 5.2 – Предельно допустимые значения энергетической экспозиции

Ввиду того, что технические средства не позволяют воспроизвести обозначения степени, далее в тексте они обозначены показателем степени в скобках Диапазоны частот	Предельно допустимая энергетическая экспозиция		
	По электрической составляющей, (В/м)(2) x ч	По магнитной составляющей, (А/м)(2) x ч	По плотности потока энергии, (мкВт/кв.см) x ч
30 кГц – 3 МГц	20000,0	200,0	
3 – 30 МГц	7000,0	Не разработаны	
30 – 50 МГц	800,0	0,72	
50 – 300 МГц	800,0	Не разработаны	
300 МГц – 300 ГГц			200,0

– повышенная или пониженная температура воздуха рабочей зоны;

– повышенная или пониженная подвижность воздуха;

Для поддержания оптимальных параметров микроклимата предусматривается кондиционирование воздуха второго класса, что позволяет достичь нормируемой чистоты и метеорологические условия воздуха, для чего используется автоматическое регулирование установок кондиционирования воздуха, которых в ВЦ установлено две<sup>9</sup>. Подача воздуха для охлаждения ЭВМ предусматривается для каждой машины по собственному воздуховоду, что исключит возможность распространения пожара с одной машины на другую. Для обогрева помещений в холодные периоды года предусматривается система отопления, которая должна быть пожаро и взрывобезопасна. В качестве системы отопления можно использовать систему центрального водяного отопления, достоинствами которой являются ее гигиеничность, надежность в эксплуатации и возможность регулирования температуры в широких пределах (Таблица 5.3).

Т а б л и ц а 5.3 – Оптимальные величины показателей микроклимата на рабочих местах офиса

Период года	Категория работ	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный и переходный (t<8°C)	Іб	21-23	60-40	0.1

Теплый ( $t^38^{\circ}\text{C}$ )	Іб	22-24	60-40	0.1
-----------------------------------	----	-------	-------	-----

- повышенная или пониженная влажность воздуха;
- отсутствие или недостаток естественного света;

Причина возникновения заключается в несоответствии естественного и искусственного освещения установленным нормам. Слабое освещение приводит к напряжению глаз, что при длительном воздействии ведет к ухудшению зрения. Также возникает головная боль, нервное напряжение.

- повышенная пульсация светового потока;
- недостаточная освещенность рабочего места;

В помещениях офиса предусматривается естественное и искусственное освещение в соответствии со СНиП 23-05-95 [21]. Естественное освещение в офисе применяют одностороннее боковое с  $\text{keo} = 1\%$ , светопроемы ориентированы преимущественно на север и северо-восток, освещенность  $E=300$  лк. Рабочие места операторов, работающих с дисплеями, располагают на удалении от окон 1.2 м и таким образом, чтобы окна находились слева. Искусственное освещение в помещениях осуществляется системой общего равномерного освещения. В случаях преимущественной работы с документами допускается применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения для освещения зоны расположения документов). Для исключения засветки экранов дисплеев прямыми световыми потоками светильники общего освещения располагают сбоку от рабочего места, параллельно линии зрения оператора и стене с окнами. В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы ЛБ-80. В светильниках местного освещения допускается применение ламп накаливания.

- повышенный уровень шума на рабочем месте;

Основным источником шума является компьютерное оборудование. Воздействие шума отражается как на органах слуха, так и на общем психологическом состоянии человека. Возможны глухота, нервные расстройства.

- умственное перенапряжение;
- эмоциональные нагрузки.

Эмоциональные нагрузки – это способность работника влиять на результат собственного труда при различных уровнях сложности осуществляемой деятельности.

Характеристика работы по показателю «эмоциональные нагрузки» в зависимости от класса условий труда подразделяется на:

- «степень ответственности за результат собственной деятельности. Значимость ошибки» – указывает, в какой мере работник может влиять на результат собственного труда при различных уровнях сложности осуществляемой деятельности. С возрастанием сложности повышается степень ответственности, поскольку ошибочные действия приводят к дополнительным



усилиям со стороны работника или целого коллектива, что соответственно приводит к увеличению эмоционального напряжения;

– «степень риска для собственной жизни»- вероятность наступления нежелательного события в случае наличия травмоопасных факторов (короткое замыкание, удар током, самовозгорание и т.д.);

– «степень ответственности за безопасность других лиц» – возможность возникновения нежелательных событий в случае наличия травмоопасных факторов при коллективном выполнении работ. Учитывается только прямая ответственность за безопасность других лиц, предусмотренная должностной инструкцией;

– «количество конфликтных ситуаций в офисе, обусловленных профессиональной деятельностью за день» – определяется на основании хронометражных наблюдений;

– снижение уровня зрения.

В результате постоянной работы за компьютером могут развиваться проблемы со зрением, близорукость. Необходимо делать постоянные перерывы в работе с компьютером. Полезным будет использование специальных компьютерных очков.

Соблюдение безопасности при работе с электрооборудованием. Весь инструмент, подключаемый к сети, должен подключаться гибким несъемным кабелем, оборудованным штепсельной розеткой. В случае если электроинструмент 1-го класса, в этом кабеле должна быть отдельная заземляющая жила.

До начала работы нужно определить класс электроинструмента по его документам, проверить надежность крепления штепсельной вилки и кабеля, прочность корпуса, наличие и исправность кожухов. При необходимости проверяется тестирование устройства.

Для инструмента первого класса проверяется надежность работы заземляющей жилы. Запрещается подключать к общей сети электроинструмент, такой, как аккумуляторная дрель шуруповерт, в случае, если рабочее напряжение их менее 42 В. Вспомогательное оборудование для работы может подключать к сети персонал с группой электробезопасности не менее 3.

Запрещается использование неисправного оборудования или оборудования с дефектами.

Запрещается удаление опилок или стружки руками. Для их удаления используются специальные щетки или крючки. Провода, идущие к дрели, должны подвешиваться, чтобы избежать контакта с поверхностями из металла, а также влажными или горячими поверхностями.

Запрещается в процессе работы:

- Передавать электроинструмент в управление другим рабочим.

- Разбирать и ремонтировать инструмент.

- Поднимать инструмент за провод, дотрагиваться до его вращающихся частей или удалять стружку до момента полной остановки вращения.

- Также запрещается до полной остановки вращения производить замену сверл.

- Устанавливать режущий инструмент в патрон или вынимать из него без выключения электродрели из сети.

- Вносить преобразователь частоты или специальный трансформатор внутрь емкостей или резервуаров из металла.

- Выполнять работы по сверлению или другие с электродрелью с приставных лестниц. Для выполнения работ на высоте должны применяться прочные подмости или леса.

- Запрещается работать электродрелью в рукавицах.

Одежда рабочего должна облегать тело плотно, полы застегнуты, а волосы убраны и закрыты головным убором.

Мероприятия по защите от поражения электрическим током

Недоступность токоведущих частей электроустановок устраняет возможность случайного прикосновения человека к ним и обеспечивается рядом способов: ограждением токоведущих частей, размещением их на недоступной высоте, изоляцией и другими средствами. Двойная изоляция – это электро изоляция, состоящая из рабочей и дополнительной изоляции. Рабочая изоляция предназначена для изоляции токоведущих частей электроустановки, обеспечивая ее нормальную работу и защиту от поражения электрическим током. Дополнительная изоляция предусматривается дополнительно к рабочей для защиты от поражения током в случае повреждения рабочей изоляции. Двойную изоляцию широко применяют при конструировании ручных электрических машин, ручного механизированного инструмента

Разветвленная цепь большой протяженности имеет значительную емкость и небольшое активное сопротивление изоляции относительно земли. Если единую, сильно разветвленную сеть разделить на ряд небольших сетей такого же напряжения, которые будут обладать незначительной емкостью и высоким сопротивлением изоляции, то опасность поражения резко снизится. Разделение электрической сети на отдельные, электрически не связанные между собой, участки производят с помощью специальных разделяющих трансформаторов, преобразователей частоты, выпрямительных установок.

При работе с переносным ручным электроинструментом – дрелью, гайковертом, шабером, шлифовальными машинками и др., а также ручными переносными лампами – человек имеет длительный контакт с корпусами этого оборудования. В результате чего для него резко повышается опасность поражения током в случае повреждения изоляции и появления напряжения на корпусе, особенно, если работа производится в помещении с повышенной опасностью поражением электрическим током, особо опасном или вне помещения. Для устранения этой опасности необходимо питать ручной инструмент и переносные лампы напряжением не выше 42 В.

Безопасность жизнедеятельности при работе с паяльником

Главное требование правил пожарной безопасности при использовании разными электрическими нагревательными и обогревательными печами,

рефлекторами определяется в запрете пользования их для осушения одежды, белья и т. д. Взаимодействуя с электропаяльниками, электро-выжигательными и похожими им устройствами, следует у помещения розетки иметь постоянную подставку для паяльника, произведенную из специальных материалов. Все труды по изготовлению детишками разных поделок при помощи выжигания и пайки требуется проводить под контролем. Электроприборы убирают на специальное место хранения по окончании труда только после пересмотра их взрослыми.

## 5.2 Расчет пожарной безопасности проектируемого объекта

В большинстве случаев пожары возникают из-за неисправностей и неправильной эксплуатации электротехнических установок и устройств, по причине коротких замыканий в электрических сетях, при перегреве и воспламенении стораемых веществ и материалов. Вычислительный центр относится к классу II по степени огнестойкости. По степени пожарной опасности данное производство может быть отнесено к категории В.

Причинами пожара в производстве могут стать короткое замыкание в цепях питания электрооборудования; значительные перегрузки проводки; плохие контакты в местах соединения проводников, приводящие к увеличению переходного сопротивления, на котором выделяется большое количество тепла; небрежное обращение с огнем; удары молнии и др.

Так как на производственном участке имеется большое количество электрооборудования, предполагается использовать установку газового объемного пожаротушения, В качестве огнегасительного вещества используется комбинированный углекислотно-хладоновый состав (85% двуокиси углерода, 15% хладона 111В 2).

Рассчитаем необходимую массу огнегасительного вещества. Производственный участок – помещение размером 5х 10 метров, высота потолков – 3 м

1 Требуемая масса комбинированного углекислотно-хладонового состава  $m_d$ , кг, для объемного пожаротушения определяется по формуле 4.1

$$m_d = k_6 q_n V \quad (4.1)$$

где  $k_6$  – коэффициент компенсации не учитываемых потерь углекислотно-хладонового состава, принимается равным 1,2;

$q_n$  – нормативная массовая огнетушащая концентрация углекислотно-хладонового состава, принимается 0,27 кг/м<sup>3</sup> при времени заполнения помещения, равным 30 сек;

$V$  – объем защищаемого помещения, м<sup>3</sup>.

$$m_d = 1,2 * 0,27 * 10 * 5 * 3 = 48,6 \text{ кг}$$

2 Количество  $\xi_1$  баллонов определяется из расчета вместимости в 40-литровый баллон 25 кг углекислотно-хладонового состава

$$\xi_1 = m_d / 25 = 48,6 / 25 = 2 \text{ полных баллона}$$

3 Внутренний диаметр магистрального трубопровода  $d_1$ , мм, определяется по формуле

$$d_1 = d_1 \sqrt{\xi_2} \quad (4.2)$$

где  $d_1$  – диаметр сифонной трубки баллона, мм (30 мм);  
 $\xi_2$  – число одновременно разряжаемых баллонов.

$$d_1 = 30 \sqrt{2} = 43 \text{ мм}$$

4 Эквивалентная длина магистрального трубопровода  $l_2$ , м, определяется по формуле

$$l_2 = k_7 \quad (4.3)$$

где  $k_7$  – коэффициент увеличения длины трубопровода для компенсации не учитываемых местных потерь (принимается равным 1,1);  
 $l$  – длина трубопровода по проекту, м (принимается равной 30 м).

$$l_2 = 1,1 * 30 = 33 \text{ м}$$

5 Площадь сечения выходного отверстия оросителя  $A_3$ , мм<sup>2</sup>, определяется по формуле

$$A_3 = S / \xi_1 \quad (4.4)$$

где  $S$  – площадь сечения магистрального трубопровода, мм<sup>2</sup>;  
 $\xi_1$  – число оросителей.

$$A_3 = 3,1415 * 2 * 33 / 8 = 26 \text{ мм}^2$$

6 Расход углекислотно-хладонового состава  $Q$ , кг/с, в зависимости от эквивалентной длины и диаметра трубопровода

$$Q = 5,6 \text{ кг/с} \quad (4.13)$$

7 Расчетное время подачи углекислотно-хладонового состава  $t$ , мин, определяется по формуле

$$t=m_d/60Q \quad (4.14)$$

где  $m_d$  – расчетная масса углекислотно-хладонового состава, кг;  
 $Q$  – расход углекислотно-хладонового состава, кг/с.

$$t=48,6/5,6=8,7 \text{ мин}$$

8 Масса основного запаса углекислотно-хладонового состава,  $m$ , кг, определяется по формуле

$$m = 1,1m_d(1+k_8/k_6) \quad (4.15)$$

где  $k_8$  – коэффициент, учитывающий остаток углекислотно-хладонового состава в баллонах и трубопроводах, равен 0.2.

$$m = 1,1*48,6*(1+0.2/1.2)=62,4 \text{ кг}$$

Насадки расположены на потолке в два ряда по четыре штуки в ряду на расстоянии 1.5 м от стен и 2м друг от друга. Они соединены последовательно магистральной трубой диаметром 33 мм, баллоны с газом расположены в соседнем помещении. План помещений представлен на рисунке 4.1.

Определим технические и организационные мероприятия. К техническим мероприятиям относятся противопожарные меры, применяемые при строительстве вычислительного центра. В частности, при строительстве необходимо соблюсти следующее:

1 территорию производства необходимо постоянно содержать в чистоте, горючий мусор должен систематически удаляться на специально отведенные участки и по мере накопления вывозиться;

2 все токоведущие части, распределительные устройства, рубильники и другие пусковые аппараты монтируются на негорючих основаниях (мрамор, текстолит, гетинакс, асбест, и т.п.);

3 измерение сопротивления изоляции электросети проводится не реже двух раз в год. Неисправные участки обесточиваются и заменяются новыми;

4 отопление аккумуляторного помещения делается централизованным (водяным или паровым) в виде целых сварных труб без фланцев и вентиляй;

5 курение в помещении строго воспрещается;

6 на случай возникновения пожара необходимо предусмотреть возможность эвакуации людей. Эвакуационные пути должны обеспечивать эвакуацию всех людей, находящихся в здании в течение необходимого времени. Число эвакуационных путей не менее двух;

7 двери на путях эвакуации навешиваются так, чтобы открывались по направлению выхода из здания;

- 8 устройство раздвижных и подъездных дверей на путях эвакуации не допускается;
- 9 минимальная ширина дверей на путях эвакуации не менее 0,8м;
- 10 высота перехода на путях эвакуации не менее 2 м;
- 11 устройство винтовых лестниц и забежных ступеней на путях эвакуации не допускается;
- 12 схема эвакуации людей тщательно разрабатывается и вывешивается на видных местах;
- 13 весь трудовой коллектив проходит обучение мерам противопожарной безопасности.

### 5.3 Расчет уровня шума

Одним из неблагоприятных факторов производственной среды в ИВЦ является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников

$$L_{\Sigma} = 10 \lg \sum_{i=1}^{i=n} 10^{0,1L_i}, \quad (4.16)$$

где  $L_i$  – уровень звукового давления  $i$ -го источника шума;  $n$  – количество источников шума.

Полученные результаты расчета сравниваются с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу, получим

$$L_{\gamma} = 10 \cdot \lg(10^4 + 10^{4,5} + 10^{1,7} + 10^1 + 10^{4,5} + 10^{4,2}) = 49,5 \text{ Дб}$$

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (СНиП 12.1.003-83).

## **Заключение**

В данной дипломной работе мною было спроектировано и реализована система управления «Умный дом». Для того чтобы спрогнозировать направление развития технологии, проанализируем доступные нам факты. Концепция «умного дома» достаточно интересна и перспективна. На данный момент небольшое количество компаний, в том числе в Казахстане, предоставляют свои услуги по созданию таких «чудо» домов. Сама технология реализуется достаточно дёшево (беспроводное или с использованием существующих силовых кабелей), а вот настройка такой системы, особенно если она управляется при помощи программ с компьютера, вещь достаточно сложная для обывателя, как и любые новые технологии, к которым люди долго привыкают, и обойдется не так уж и дешево ее владельцам. Кроме того, наличие ряда таких решений необходимо учитывать при разработке дизайна помещений. Идеальное место применения таких технологий - частные дома и коттеджи, и крупные офисы. В принципе, учитывая, что хозяины загородных домов тратят большие деньги на их содержание, стоимость такого решения будет относительно небольшой.

В «Умном доме» все устройства и бытовая техника - от климатических систем до телевизоров - управляется чрезвычайно сложными компьютерными структурами и системами. «Умный дом» включает свет и музыку, когда гости и близкие входят в дом и перемещаются по многочисленным комнатам. При этом световое и музыкальное сопровождение по мере передвижения посетителя по «умным» апартаментам изменяется согласно желаниям владельца, сохранённые в настройках. Людям не нужно задавать температурный режим в помещениях или настраивать освещение - установленная «интеллектуальная» система по состоянию хозяина распознает, какая температура и освещение необходимы ему в данный момент для полного комфорта. Для обеспечения удобства в квартире могут использоваться разнообразные технологии, начиная от самодельных устройств и заканчивая высокоинтеллектуальными компьютерными АСУ.

## Список используемой литературы

- 1 Е.А. Тесля. «Умный дом» своими руками. Строим интеллектуальную цифровую систему в своей квартире / Тесля Е.А. - Санкт Петербург, 2008. - 224с.
- 2 М.Э. Сопер. Практические советы и решения по созданию « Умного дома » / Сопер М. Э. - М.: НТ Пресс, 2007. - 432 с
- 3 В.Н. Харке «Умный дом. Объединение в сеть бытовой техники и систем коммуникаций в жилищном строительстве» / Харке В.Н. - М.: Техносфера, 2006. - 292с
- 4 Т. Р. Элсенпитер, Дж. Велт. «Умный Дом строим сами» / Элсенпитер Т. Р., Велт Дж / КУДИЦ-ОБРАЗ. 2005. – 384с.
- 5 В.Н. Гололобов. «Умный дом» своими руками. / Гололобов В.Н. – М.: НТ Пресс, 2007. – 416 с.
- 6 Сайт <http://www.engineer-oht.ru>
- 7 Сайт <http://php.ru/manual/>
- 8 Кузнецов М., Симдянов И. РНР . Практика создания Web-сайтов. – 2-е изд. – Спб.: БХВ-Петербург, 2008. – 1264 с.
- 9 Стешин А. Изучаем windows. – СПб: Питер, 2009. – 209 с.
- 10 Лукасьян Г.М. Экономика от «А» до «Я»: Тематический справочник. – М.: ИНФРА-М, 2009. – 480 с.
- 11 Иванов И.Н. Экономика промышленного предприятия: Учебник. – М.: ИНФРА-М, 2011. – 395 с.
- 12 Рофе А.И. Экономика труда: Учебник. – М.: КноРус, 2010. – 400 с.
- 13 Скляренко В.К., Прудников В.М. Экономика предприятия: Учебник. – М.: ИНФРА-М, 2006. – 528 с.
- 14 Белов С.В., Ильницкая А.В., Козьяков А.Ф. Безопасность жизнедеятельности. Учебник для вузов. – М.: Высшая школа, 2005. – 448 с.
- 15 СНиП РК 4.02–42–2006 Отопление, вентиляция и кондиционирование. – Астана: Издательство стандартов, 2007.
- 16 Хакимжанов Т.Е. Сборник задач по охране труда и безопасности жизнедеятельности: Учебное пособие для вузов. – Алматы: Эверо, 2007. – 274
- 17 СНиП РК 2.02–15–2003 Пожарная автоматика зданий и сооружений. Астана: Издательство стандартов, 2004.



## Приложение А

### Акт внедрения на предприятие

**«NJ Company» LTD**  
Business Center «Real Invest Group»  
31, Pirogova str., of.109  
050040, Almaty, Kazakhstan  
Tel/Fax: 8 (727) 376-38-35  
Web: www.njcompany.kz  
E-mail: njc.seminar@gmail.com,  
[info@njcompany.kz](mailto:info@njcompany.kz)



**ТОО «NJ Company»**  
050040, Казахстан, г.Алматы  
Бизнес-центр «Real Invest Group»  
ул. Пирогова, 31, офис 109  
Тел./Факс: 8 (727) 376-38-35  
Сайт: www.njcompany.kz  
Эл.адрес: njc.seminar@gmail.com,  
[info@njcompany.kz](mailto:info@njcompany.kz)

исх. № 44 от 29 мая 2014 г.

#### АКТ

#### о приеме работы разработки системы по технологии умный дом.

В период с 19.04.2014 по 23.04.2014 комиссия проводила прием структуры и описания макетного устройства (МУ) «Разработка системы по технологии умный дом», произведенный программистом-разработчиком Сабировым И.М.

Комиссия провела проверку готовности к работе данной системы и признала, что данная идея полностью готова для введения в эксплуатацию.

В процессе испытаний была проверена работоспособность системы на:

- работоспособность программного модуля;
- работоспособность структуры макета и отдельных элементов;
- непрерывную эксплуатацию структуры макета;

Комиссии представлена структура и описание системы по технологии «Умный дом».


Решение комиссии – принять данную разработку и предоставить его заказчику.

#### СДАЛ:


Программист-разработчик  
Сабиров И. М.


#### ПРИНЯЛИ:

##### Председатель комиссии:

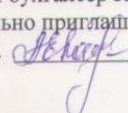
Директор «NJ Company» Нұржігіт А.С. 

##### Члены комиссии:

Главный менеджер Нуртаева А.Н. 

Главный бухгалтер Курганская Е.Б. 

Специально приглашенный гость комиссии

Пак А.Е. 



## Листинг программы

```
#include <mega64.h>

// Graphic LCD functions
#include <glcd.h>
#include <delay.h>
// Font used for displaying text
// on the graphic LCD
#include <font5x7.h>

// Function used for reading image
// data from external memory
unsigned char read_ext_memory(GLCDMEMADDR_t addr)
{
    unsigned char data;
    // Place your code here

    return data;
}

// Function used for writing image
// data to external memory
void write_ext_memory(GLCDMEMADDR_t addr, unsigned char data)
{
    // Place your code here

}

#ifndef RXB8
#define RXB8 1
#endif

#ifndef TXB8
#define TXB8 0
#endif

#ifndef UPE
#define UPE 2
#endif

#ifndef DOR
#define DOR 3
#endif

#ifndef FE
#define FE 4
#endif
```

*Продолжение приложения А*

```

#ifndef UDRE
#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART1 Receiver buffer
#define RX_BUFFER_SIZE1 8
char rx_buffer1[RX_BUFFER_SIZE1];

#if RX_BUFFER_SIZE1 <= 256
unsigned char rx_wr_index1, rx_rd_index1, rx_counter1;
#else
unsigned int rx_wr_index1, rx_rd_index1, rx_counter1;
#endif

// This flag is set on USART1 Receiver buffer overflow
bit rx_buffer_overflow1;

// USART1 Receiver interrupt service routine
interrupt [USART1_RXC] void usart1_rx_isr(void)
{
char status, data;
status=UCSR1A;
data=UDR1;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{
rx_buffer1[rx_wr_index1++]=data;
#if RX_BUFFER_SIZE1 == 256
// special case for receiver buffer size=256
if (++rx_counter1 == 0) rx_buffer_overflow1=1;
#else
if (rx_wr_index1 == RX_BUFFER_SIZE1) rx_wr_index1=0;
if (++rx_counter1 == RX_BUFFER_SIZE1)
{
rx_counter1=0;
rx_buffer_overflow1=1;
}
}
#endif
}
}

#pragma used+

```

*Продолжение приложения А*

```

chargetchar1(void)
{
char data;
while (rx_counter1==0);
data=rx_buffer1[rx_rd_index1++];
#if RX_BUFFER_SIZE1 != 256
if (rx_rd_index1 == RX_BUFFER_SIZE1) rx_rd_index1=0;
#endif
#asm("cli")
--rx_counter1;
#asm("sei")
return data;
}
#pragma used-
// USART1 Transmitter buffer
#define TX_BUFFER_SIZE1 8
char tx_buffer1[TX_BUFFER_SIZE1];

#if TX_BUFFER_SIZE1 <= 256
unsigned char tx_wr_index1,tx_rd_index1,tx_counter1;
#else
unsigned int tx_wr_index1,tx_rd_index1,tx_counter1;
#endif

// USART1 Transmitter interrupt service routine
interrupt [USART1_TXC] void usart1_tx_isr(void)
{
if (tx_counter1)
{
--tx_counter1;
UDR1=tx_buffer1[tx_rd_index1++];
#if TX_BUFFER_SIZE1 != 256
if (tx_rd_index1 == TX_BUFFER_SIZE1) tx_rd_index1=0;
#endif
}
}

// Write a character to the USART1 Transmitter buffer
#pragma used+
void putchar1(char c)
{
while (tx_counter1 == TX_BUFFER_SIZE1);
#asm("cli")
if (tx_counter1 || ((UCSR1A & DATA_REGISTER_EMPTY)==0))
{
tx_buffer1[tx_wr_index1++]=c;
#if TX_BUFFER_SIZE1 != 256
if (tx_wr_index1 == TX_BUFFER_SIZE1) tx_wr_index1=0;
#endif
++tx_counter1;
} else

```

*Продолжение приложения А*

```

UDR1=c;
#asm("sei")
}
#pragma used-
eeprom int change_menu;
eeprom int type_menu;
eeprom int startpage;
eeprom int press_button;

// SPI interrupt service routine
interrupt [SPI_STC] void spi_isr(void)
{
unsigned char data;
data=SPDR;
if (data == 'a')
{
    press_button = 1;
}
    if (data == 'b')
{
    press_button = 2;
}
if (data == 'c')
{
    press_button = 3;
}
if (data == 'd')
{
    press_button = 4;
}
}

///// //////////////////////////////////MENU//////////////////////////////////////
typedef struct menu
{
    int ID;
    char name_menu[20];
    char name_selection[20];
    int position;
    int menu_selection;

}MENU;

MENU MainMenu[]=
{
    {1,"MainMenu","Users",1,3},
    {1,"MainMenu","Tools",2,3},
    {1,"MainMenu","About",3,3}

};

MENU Tools_Menu[]=
{

```

*Продолжение приложения А*

```
{2, "Tools", "Light", 1, 3},
{2, "Tools", "Fans", 2, 3},
{2, "Tools", "Something", 3, 3}

};
////////////////////////////////////endMENU////////////////////////////////////

////////////////////////////////////eeprom////////////////////////////////////

////////////////////////////////////endeeprom////////////////////////////////////

////////////////////////////////////          //////////////////////////////////////
/*void test(void)
{
glcd_clear();
glcd_outtextxy(0,0,"OK");

}
*/

////////////////////////////////////

////////////////////////////////////FCursor////////////////////////////////////
void cursor(int s,int k,int l)
{int x,y,color;
  x = 121;
  if (s == 0){

    y = 10;
    color = 15;
    glcd_putpixel(x+3,y,color);
    glcd_putpixel(x+2,y+1,color);
    glcd_putpixel(x+1,y+2,color);
    glcd_putpixel(x,y+3,color);
    glcd_putpixel(x+1,y+4,color);
    glcd_putpixel(x+2,y+5,color);
    glcd_putpixel(x+3,y+6,color);
    glcd_putpixel(x+2,y+3,color);
    glcd_putpixel(x+3,y+3,color);
    glcd_putpixel(x+4,y+3,color);
    glcd_putpixel(x+5,y+3,color);
    glcd_putpixel(x+6,y+3,color);
    glcd_putpixel(x,y+3,color);
  }

  if (s == 1)
  {
```

*Продолжение приложения А*

```

y = 10 * l;
color = 0;
glcd_putpixel(x+3,y,color);
    glcd_putpixel(x+2,y+1,color);
    glcd_putpixel(x+1,y+2,color);
    glcd_putpixel(x,y+3,color);
    glcd_putpixel(x+1,y+4,color);
    glcd_putpixel(x+2,y+5,color);
    glcd_putpixel(x+3,y+6,color);
    glcd_putpixel(x+2,y+3,color);
    glcd_putpixel(x+3,y+3,color);
    glcd_putpixel(x+4,y+3,color);
    glcd_putpixel(x+5,y+3,color);
    glcd_putpixel(x+6,y+3,color);
    glcd_putpixel(x,y+3,color);
y = 10 * k;
color = 15;
glcd_putpixel(x+3,y,color);
glcd_putpixel(x+2,y+1,color);
glcd_putpixel(x+1,y+2,color);
glcd_putpixel(x,y+3,color);
glcd_putpixel(x+1,y+4,color);
glcd_putpixel(x+2,y+5,color);
glcd_putpixel(x+3,y+6,color);
glcd_putpixel(x+2,y+3,color);
glcd_putpixel(x+3,y+3,color);
glcd_putpixel(x+4,y+3,color);
glcd_putpixel(x+5,y+3,color);
glcd_putpixel(x+6,y+3,color);
glcd_putpixel(x,y+3,color);

    } }
////////////////////////////////////endCursor////////////////////////////////////
////////////////////////////////////
void start_page()
{

glcd_outtextxy(35,30,"StartPage");
glcd_outtextxy(45,55,"MENU");
startpage = 1;

}
void tools(void)
{
int k,l;
change_menu = 0;
type_menu = 0;
press_button = 0;
l = 1;

```

*Продолжение приложения А*

```

while(1)
{
if (change_menu == 0)
{
glcd_clear();
glcd_outtextxy(45,0,Tools_Menu[1].name_menu);
glcd_line(0,8,128,8);
glcd_outtextxy(5,10,Tools_Menu[0].name_selection);
glcd_outtextxy(5,20,Tools_Menu[1].name_selection);
glcd_outtextxy(5,30,Tools_Menu[2].name_selection);

cursor(0,0,0);
change_menu = 1;
}

if(press_button == 1)
{press_button = 0;
if (l == 1)
{}
if (l == 2)
{

}
if (l == 3)
{}

}
if(press_button == 2)
{ press_button = 0;

if ( l == 3)
{k = 1;}
else {
k=1;
k=k+1;
} cursor(1,k,l);
l=k;
}
if(press_button == 3)
{ press_button = 0;

if (l == 1)
{k =3;}
else {
k=1;
k= k -1;
}
cursor(1,k,l);
l=k;
}
}
}

```

*Продолжение приложения А*



```

    }

if (press_button == 4)
    {
    press_button = 0;
    glcd_clear();
    change_menu = 0;
    if (Tools_Menu[1].ID == 2)
    {
    type_menu = 1;
    break;
    }
    }
}

/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
void select_menu(void)
{
    int k,l;
    change_menu = 0;
    type_menu = 0;
    press_button = 0;
    l = 1;

    while(1)
    {
        if (change_menu == 0)
        {
            glcd_clear();
            glcd_outtextxy(45,0,MainMenu[1].name_menu);
            glcd_line(0,8,128,8);
            glcd_outtextxy(5,10,MainMenu[0].name_selection);
            glcd_outtextxy(5,20,MainMenu[1].name_selection);
            glcd_outtextxy(5,30,MainMenu[2].name_selection);

            cursor(0,0,0);
            change_menu = 1;
        }

        if (press_button == 1)
        {
            press_button = 0;
            if (l == 1)
            {}
            if (l == 2)
        }
    }
    type_menu = 3;
    break;
}

```

*Продолжение приложения А*

```

}
if (l == 3)
    {}

    }
    if(press_button == 2)
    {
    press_button =0;

    if ( l == 3)
        {k = 1;}
        else {
        k=1;
        k=k+1;
        }
    cursor(1,k,l);
    l=k;
    }
    if(press_button == 3)
    { press_button =0;

    if (l == 1)
        {k = 3;}
        else {
        k=1;
        k= k -1;
        }
    cursor(1,k,l);
    l=k;
    }

    if(press_button == 4)
    {press_button =0;
    glcd_clear();
    change_menu = 0;
    startpage = 0;
    break;
    }

    }
}
////////////////////////////////////

// Declare your global variables here

void main(void)
{ char mas[20];
inti = 0;
intnum = 0;

```

*Продолжение приложения А*

```

// Declare your local variables here
// Graphic LCD initialization data
GLCDINIT_t glcd_init_data;
press_button = 0;
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=Out Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=0 State2=T State1=T
State0=T
PORTB=0x00;
DDRB=0x08;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTE=0x00;
DDRE=0x00;

// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T
State0=T
PORTF=0x00;

```

*Продолжение приложения А*

```

DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=0xFF
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;

```

*Продолжение приложения А*

```

OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer3 Stopped
// Mode: Normal top=0xFFFF
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

ETIMSK=0x00;
// USART0 initialization
// USART0 disabled
UCSR0B=0x00;

// USART1 initialization

```

*Продолжение приложения А*

```

// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART1 Receiver: On
// USART1 Transmitter: On
// USART1 Mode: Asynchronous
// USART1 Baud Rate: 9600
UCSR1A=0x00;
UCSR1B=0xD8;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI Type: Slave
// SPI Clock Phase: Cycle Half
// SPI Clock Polarity: Low
// SPI Data Order: MSB First
SPCR=0xC6;
SPSR=0x00;

// Clear the SPI interrupt flag
#asm
    in    r30,spsr
    in    r30,spdr
#endasm

// TWI initialization
// TWI disabled
TWCR=0x00;
// Graphic LCD initialization
// The KS0108 connections are specified in the
// Project|Configure|C Compiler|Libraries|Graphic LCD menu:
// DB0 - PORTE Bit 0
// DB1 - PORTE Bit 1
// DB2 - PORTE Bit 2
// DB3 - PORTE Bit 3
// DB4 - PORTE Bit 4
// DB5 - PORTE Bit 5
// DB6 - PORTE Bit 6
// DB7 - PORTE Bit 7
// E - PORTF Bit 0
// RD /WR - PORTF Bit 1

```

*Продолжение приложения А*

```

// RS - PORTFBit 2
// /RST - PORTF Bit 3
// /CS1 - PORTF Bit 4
// /CS2 - PORTF Bit 5

// Specify the current font for displaying text
glcd_init_data.font=font5x7;
// No function is used for reading
// image data from external memory
glcd_init_data.readxmem=NULL;
// No function is used for writing
// image data to external memory
glcd_init_data.writexmem=NULL;

glcd_init(&glcd_init_data);

// Global enable interrupts
#asm("sei")

glcd_outtextxy(35,30,"SMART HOME");
glcd_line(14,40,114,40);
glcd_line(14,40,14,45);
glcd_line(14,45,114,45);
glcd_line(114,40,114,45);

while(num <= 100)
{
    glcd_putpixel(num+14,41,15);
    glcd_putpixel(num+14,42,15);
    glcd_putpixel(num+14,43,15);
    glcd_putpixel(num+14,44,15);
    num ++;
    delay_ms(1);
}
startpage = 0;
glcd_clear();
while (1)
{
    if (startpage == 0 )
    {
        start_page();
    }
    if (press_button == 1)
    {
        press_button = 0;
        select_menu();
    }
    if (type_menu == 3)
{
tools();

```

*Окончание приложения А*

```
    }  
if (type_menu == 1)  
{  
    select_menu();  
}  
}}  
  
\
```