

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Инфокоммуникационные технологии»

РАБОТА ПРОВЕРЕНА

Рецензент

_____ Д.И. Ефимов
_____ 2020 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____ С.Н. Даровских
_____ 2020 г.

**Эмулятор источника питания для проверки устойчивой
работы радиоизмерительной системы ИТ-19С**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–Д.11.05.01.2019.367.00 ПЗ ВКР

Консультанты:

По экономике

_____ Р.Ш. Закиров
_____ 2019 г.

Руководитель работы:

_____ А.Ю. Конорев
_____ 2019 г.

По охране труда

_____ И.С. Окраинская
_____ 2019 г.

Автор работы:

студент группы КЭ-678

_____ В.Ю. Егоров
_____ 2019 г.

Нормоконтролер:

_____ В.Д. Спицына
_____ 2020 г.

Челябинск
2020

РЕФЕРАТ

Егоров В.Ю. Эмулятор источника питания. – Челябинск: ЮУрГУ, ВШЭКН, 2019, 109 с., 18 ил., 7 таб., библиогр. список 23 наим., 1 прил., 6 листов чертежей ф. А1.

В дипломной работе разрабатывается устройство, которое эмулирует источник питания для проверки систем зарядки радиоизмерительных станций, выпускаемых ООО «ПЛАНАР».

В техническом задании были оговорены условия, в соответствии с которыми, обязательным было применение микроконтроллера и программируемых цифровых резисторов.

Разрабатываемое устройство должно осуществлять измерение характеристик зарядной системы в зависимости от выбранного типа аккумулятора.

В разделе безопасности жизнедеятельности проанализированы условия эксплуатации спроектированного устройства. В организационно-экономическом разделе приведены расчеты технико-экономических показателей разработки данного изделия.

					ЮУрГУ–Д.11.05.01.2019.367.00 ПЗ						
<i>Изм</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>	<i>Эмулятор источника питания для проверки устойчивой работы радиоизмерительной системы ИТ-19С</i>			<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>	
<i>Разраб.</i>		Егоров						<i>в</i>		3	109
<i>Пров.</i>		Конорев						ЮУрГУ Кафедра ИКТ			
<i>Рецензент</i>											
<i>Н.контр.</i>		Спицына									
<i>Утв.</i>		Даровских									

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1 Анализ технического задания.....	8
1.1 Содержание работы.....	8
1.2 Описание изделия.....	8
1.2.1 Основные понятия.....	8
1.2.2 Назначение законченных устройств.....	9
1.3 Назначение разрабатываемого устройства.....	10
2 Теоретический раздел.....	12
2.1 Принцип работы батареи.....	12
2.2 Технология литий-ионных аккумуляторов.....	12
2.2.1 Безопасность.....	13
2.3 Зарядка литий-ионных аккумуляторов.....	14
2.3.1 Безопасность при зарядке.....	14
2.3.2 Первоначальная зарядка и интервалы зарядки.....	15
2.3.3 Стадии зарядки.....	15
3 Анализ существующих эмуляторов.....	17
3.1 Эмулятор N6785A.....	17
3.2 Эмулятор 2281S.....	19
4 Схемотехнический раздел.....	24
4.1 Разработка структурной схемы.....	24
4.2 Назначение и функционал управляющей части устройства.....	24
4.3 Обоснование выбора элементов цифровой части прибора.....	25
4.3.1 Выбор микроконтроллера.....	25
4.3.2 Выбор дисплея.....	28
4.3.3 Выбор цифрового резистора.....	30
4.3.4 Выбор ОУ для измерения тока и напряжения зарядки.....	31
4.3.5 Выбор ОУ для измерения тока утечки.....	32

5	Конструкторский раздел.....	34
5.1	Разработка и трассировка плат.....	34
5.2	Выбор материала печатных плат.....	34
5.3	Выбор элементной базы.....	35
5.4	Графическое представление плат.....	38
6	Программный раздел.....	40
6.1	Разработка программы для микроконтроллера.....	41
6.2	Разработка программы для клавиатуры.....	43
6.3	Разработка программы для дисплея.....	44
6.4	Разработка программы для цифрового резистора.....	45
6.5	Разработка программы для пересчета кода АЦП в цифровое значение.....	46
7	Безопасность жизнедеятельности.....	49
7.1	Электробезопасность эмулятора источника питания.....	49
7.2	Анализ потенциально опасных и вредных производственных факторов (ОВПФ).....	50
7.3	Нормирование ОВПФ и мероприятия по их ликвидации.....	51
7.3.1	Освещение.....	51
7.3.2	Пожарная безопасность.....	52
7.3.3	Микроклимат.....	53
7.3.4	Требования к шумам и вибрационному воздействию.....	54
7.4	Требования к организации рабочего места.....	56
7.4.1	Организация рабочего места.....	57
8	Организационно-экономический раздел.....	60
8.1	Анализ аналогов, сравнение технических параметров.....	60
8.2	Понятия параметров сетевого планирования.....	61
8.3	Анализ этапов разработки, построение сетевого графика.....	63
8.4	Расчет стоимости работы.....	72
8.4.1	Расчет себестоимости готового изделия.....	72

8.5 Расчет окупаемости прибора.....	79
ЗАКЛЮЧЕНИЕ.....	80
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	81
ПРИЛОЖЕНИЕ А. Исходный код программы.....	83

ВВЕДЕНИЕ

Целью данной работы является разработка устройства для проверки системы зарядки переносных измерителей с автономным источником питания. В ходе работы: разработан эмулятор источника питания; реализован алгоритм проверки; написано программное обеспечение с возможностью добавления характеристик других источников питания; проведено тестирование на переносной радиотехнической системе ИТ-19С; сделан анализ и вывод об эффективности разработанного устройства на основе полученных результатов.

Для реализации алгоритма и тестирования устройства, было написано программное обеспечение на языке программирования: Си.

Данная работа предназначена для проверки систем зарядки измерительных измерителей с автономным источником питания. Разработанное ПО вместе с устройством позволяет имитировать большое множество аккумуляторных батарей.

1 Анализ технического задания

1.1 Содержание работы

Для реализации требований, описанных в техническом задании, необходимо рассмотреть следующие пункты:

- а) проведение сравнительного анализа существующих реализаций и планируемой реализации устройства, обоснование целесообразности;
- б) на основе анализа имеющейся информации разработать принцип работы устройства и ПО;
- в) разработка структурной схемы;
- г) подобрать элементную базу;
- д) выбрать программную среду для реализации ПО и его тестирования;
- е) протестировать устройство на эффективность решения поставленной задачи, обозначить критерии эффективности, по которым происходит сравнение результатов.

1.2 Описание изделия

1.2.1 Основные понятия

Разрабатываемое устройство предназначено для проведения измерения характеристик зарядного оборудования, находящегося внутри переносной измерительной системы семейства ИТ, разрабатываемые компанией ООО «ПЛАНАР».

Полное наименование ИТ, в зависимости от вариантов исполнения:

- «ИТ-19С» – Анализатор сигналов цифрового кабельного телевидения.

Сокращенное наименование – «ИТ-19С»;

- «ИТ-09Т» – Анализатор сигналов DVB-T. Сокращенное наименование – «ИТ-09Т»;

- «ИТ-11» – Анализатор сигналов DVB-S. Сокращенное наименование – «ИТ-11»;

- «ИТ-089» – Анализатор сигналов цифрового вещательного телевидения. Сокращенное наименование – «ИТ-089»;

- «ИТ-100» – Анализатор ТВ сигналов мультисистемный. Сокращенное наименование – «ИТ-100».

Разрабатываемое устройство может применяться практически ко всему оборудованию семейству «ИТ».

1.2.2 Назначение законченных устройств

Анализаторы сигналов семейства «ИТ» предназначены для измерения параметров телевизионного радиосигнала, параметров элементов построения сети и других радиоэлектронных устройств.

Область применения анализаторов: контроль и настройка сетей распределительных приемных систем телевидения и радиовещания, отдельных элементов сети и других радиоэлектронных устройств.

Спектр производимых приборов способен удовлетворить требования практически любого специалиста, занимающегося монтажом и обслуживанием сетей кабельного телевидения и систем индивидуального приема телевидения.

ИТ-19С – обновленная версия одного из самых популярных анализаторов сигнала стандарта DVB-C ИТ-09С. Для аналоговых каналов может измерять отношение сигнал/шум, а также отношение видео/аудио. Для цифровых каналов измеряет мощность канала. Производит измерение радиочастотного спектра, напряжения дистанционного питания, MER, PreBER, PostBER.

ИТ-09Т – для аналоговых каналов анализатор измеряет уровень канала, отношения сигнал/шум, видео/аудио. Для цифровых каналов измеряет мощность канала и отношение сигнал/шум. Для DVB-T каналов производит измерение BER, MER до и после декодера Витерби, имеет счетчик неисправленных пакетов и запас по помехоустойчивости MARGIN.

ИТ-11 – измеряет показатели качества приема MER, BER до и после декодера Витерби, напряжение на входном разъеме в диапазоне от 3 до 30 В, имеет режим измерения «на слух», и режим настройки антенны с двойным конвертором. Также

имеет автоматическое определение параметров настройки такие как частота транспондера, символьная скорость, относительная скорость кодирования, поляризация.

ИТ-089 – анализатор сигналов цифрового вещательного телевидения в составе базового модуля МБ-08 и сменного модуля МС-089 предназначен для измерения параметров сигналов цифрового вещательного телевидения стандартов DVB-T/T2 с модуляцией CODFM, а также аналогового телевидения. Анализатор может быть использован при контроле и настройке сетей распределительных приемных систем телевидения и радиовещания, отдельных элементов построения сети и других радиоэлектронных устройств. Для аналоговых каналов может производить измерения уровня напряжения радиосигнала, отношения сигнал/шум и отношения видео/аудио. Для цифровых каналов измеряет фактический уровень напряжения радиосигнала и отношение сигнал/шум. Также измеряет показатели приема канала с цифровой модуляцией такие как: MER, preVBER, postVBER, декодера Витерби, preLDPC, postLDPC декодера LDPC, эхо измерения.

ИТ-100 – анализатор ТВ сигналов предназначен для измерения параметров цифрового и аналогового телевидения. Он может использоваться при настройке и контроле кабельных коаксиальных, оптоволоконных или IP сетей приема телевидения и радиовещания, отдельных элементов построения сети и прочих радиоэлектронных устройств. В аналоговом телевидении поддержка стандартов PAL, SECAM, NTSC. В кабельном цифровом телевидении поддержка стандартов DVB-C, ITU-T J.83 Annex B, C. В наземном вещательном цифровом телевидении поддерживает стандарты DVB-T/T2. В спутниковом цифровом вещательном телевидении поддержка стандартов DVB-S/S2. В IP цифровом телевидении поддержка UDP и RTP протокола, прием транспортного потока через unicast и multicast (IGMPv2).

1.3
устройства

Назначение разрабатываемого

В данный момент компания ООО «ПЛАНАР» занимается не только созданием новых измерительных систем, но и производит модернизацию выпускаемой продукции. В процессе модернизации измерителей, появилась необходимость в разработке прибора, способного быстро и точно производить измерение характеристик зарядной системы измерительных устройств.

Раньше все измерения проводились вручную и занимали достаточное количество времени, а при использовании разработанного эмулятора процесс проверки займет не более 10 секунд, что позволит оптимизировать работу на этапе проверки измерительной системы.

2 Теоретический раздел

В данном разделе проводится описание процесса зарядки, рассматриваются явления, возникающие в результате этого процесса и как с ними бороться, а также меры безопасности.

2.1 Принцип работы батареи

Зарядка аккумуляторов возможна благодаря обратимой химической реакции, которая возобновляет энергию в химической системе. В зависимости от применяемых химикатов аккумулятор станет владеть различными характеристиками. Для того, чтобы не нанести большой вред аккумуляторам, требуется доскональное знание этих характеристик [1].

2.2 Технология литий-ионных аккумуляторов

Литий-ионные аккумуляторы имеют наилучшее соотношение энергии / веса и энергии / размера среди современных аккумуляторных батарей. В настоящее время это самые быстрорастущие аккумуляторные системы на рынке, которые используются в таких приложениях, как ноутбуки, мобильные телефоны, портативные медиаплееры, карманные компьютеры и медицинские устройства.

По сравнению с традиционными перезаряжаемыми аккумуляторами у литий-ионных аккумуляторов не имеется внутреннего сопротивления, но зато имеет длительный срок службы, быстрое время зарядки, низкий уровень саморазряда, низкую токсичность и не требует поддержки по сравнению с обычными аккумуляторными батареями. Например, элементы ионно-литиевых батарей с кобальтовыми катодами обеспечивают в два раза больше энергии, чем никелевые батареи, и в четыре раза больше энергии, чем свинцово-кислотные батареи. Литий-ионные аккумуляторы практически не требуют обслуживания - преимущество, которое не могут предложить другие химические аккумуляторы. У литий-ионных

аккумуляторов низкий саморазряд, и они безопасны для окружающей среды. Причиняют минимальный вред после окончания использования.

Недостатками литий-ионных аккумуляторов являются низкий уровень перезарядки и необходимость встроенной схемы защиты. Короткое замыкание может вызвать протекание большого тока, что приведет к повышению температуры и тепловому расхождению, при котором выделяются опасные газы.

2.2.1 Безопасность

При соблюдении определенных мер предосторожности литий-ионные аккумуляторы безопасны для зарядки и разрядки. Производители аккумуляторов также обеспечивают высокий уровень надежности с тремя уровнями защиты, такими как:

- а) количество активного материала ограничено достижением работоспособного соотношения между плотностью энергии и безопасностью;
- б) в каждом элементе аккумулятора содержатся различные механизмы защиты;
- в) внутри аккумулятора содержится электрическая защитная цепочка.

Устройства защиты элементов аккумуляторов работают следующим образом:

- а) устройство с ПТК/ОТК (положительный/отрицательный температурный коэффициент) работает как защита от высоких скачков тока;
- б) УПЦ (устройство прерывания цепи) разрывает электрическую цепь, если слишком высокое напряжение зарядки повышает давление внутри аккумулятора;
- в) Специальное отверстие позволяет управлять выпуском газа при резком возрастании давления внутри аккумулятора.

Электрическая защитная цепочка работает следующим образом:

- а) твердотельный переключатель открывается, если высокое напряжение зарядки в каком-либо элементе аккумулятора превышает заданный уровень;

- б) Перемычка останавливает протекание тока, если температура элемента приближается к 90°C (194°F);
- в) токовая цепь разрывается, когда напряжение элемента падает ниже заданного уровня. Это сделано, чтобы предотвратить глубокий разряд аккумулятора.

Сегодня литий-ионные аккумуляторы являются одними из самых успешных и безопасных среди других химических аккумуляторов, они производятся с миллионами элементов каждый год.

2.3 Зарядка литий-ионных аккумуляторов

Существует только один способ зарядки литий-ионных аккумуляторов. Производители литий-ионных аккумуляторов имеют очень строгие стандарты зарядки, и литий-ионные аккумуляторы должны заряжаться в соответствии с условиями, указанными производителем.

Литий-ионные аккумуляторы заряжаются постоянным напряжением (после того, как напряжение достигло номинального значения) с пределом по току, чтобы избежать перегрева на начальной стадии зарядки. Зарядка заканчивается, когда зарядный ток падает ниже уровня, установленного производителем. Во время зарядки контролируются различные параметры: время зарядки и температура аккумулятора. Аккумулятор повреждается во время перезарядке и может взорваться.

2.3.1 Безопасность при зарядке

Статическое электричество или неисправное зарядное устройство могут повредить цепь защиты аккумулятора и установить полупроводниковые переключатели в положение «включено». Это может произойти без ведома пользователя. Батарея с поврежденной защитной цепью может работать нормально, но не обеспечивает защиту от неправильного использования.

Литий-ионные аккумуляторы для потребителей нельзя заряжать при температуре ниже 0 °C (32 °F). Если батареи заряжаются при низких температурах, зарядка может быть выполнена в обычном режиме. Тем не менее, химические реакции в элементе могут вызвать долговременный дефект и поставить под угрозу безопасность батарей.

Аккумулятор становится более уязвимым к поломке, при ударе, разрушении или высокой скорости зарядки.

Аккумулятор должен оставаться прохладным. Аккумулятор, который нагревается во время зарядки, не должен использоваться.

2.3.2 Первоначальная зарядка и интервалы зарядки

В отличие от многих других типов аккумуляторов, литий-ионные аккумуляторы не требуют специальной начальной зарядки. Первоначальный заряд литий-ионных аккумуляторов ничем не отличается от 10-го или 100-го.

Литий-ионные аккумуляторы можно и нужно часто заряжать. Батареи остаются в эксплуатации дольше, если они частично разряжены, чем если они постоянно разряжены полностью. Во избежание износа батареи следует избегать полной разрядки.

Емкость аккумулятора уменьшается со временем в независимости от того, используется он или нет.

2.3.3 Стадии зарядки

Зарядка литий-ионных аккумуляторов состоит из трех этапов:

- a) Предварительная оценка тока: зарядка литий-ионных аккумуляторов начинается с проверки напряжения аккумулятора. Если напряжение ниже заданного уровня, зарядка начинается небольшим фиксированным током;

- б) Постоянный ток. Зарядка продолжается путем приложения к аккумулятору постоянного тока. Величина тока зарядки зависит от аккумулятора и устанавливается производителем;
- в) Постоянное напряжение. После достижения заданного уровня напряжения устройство зарядки переключится с подачи постоянного тока на подачу постоянного напряжения. Эта стадия завершается, когда ток зарядки становится ниже заданного производителем уровня.

Нижеприведенный рисунок 1 иллюстрирует напряжение и ток литий-ионного аккумулятора во время зарядки.

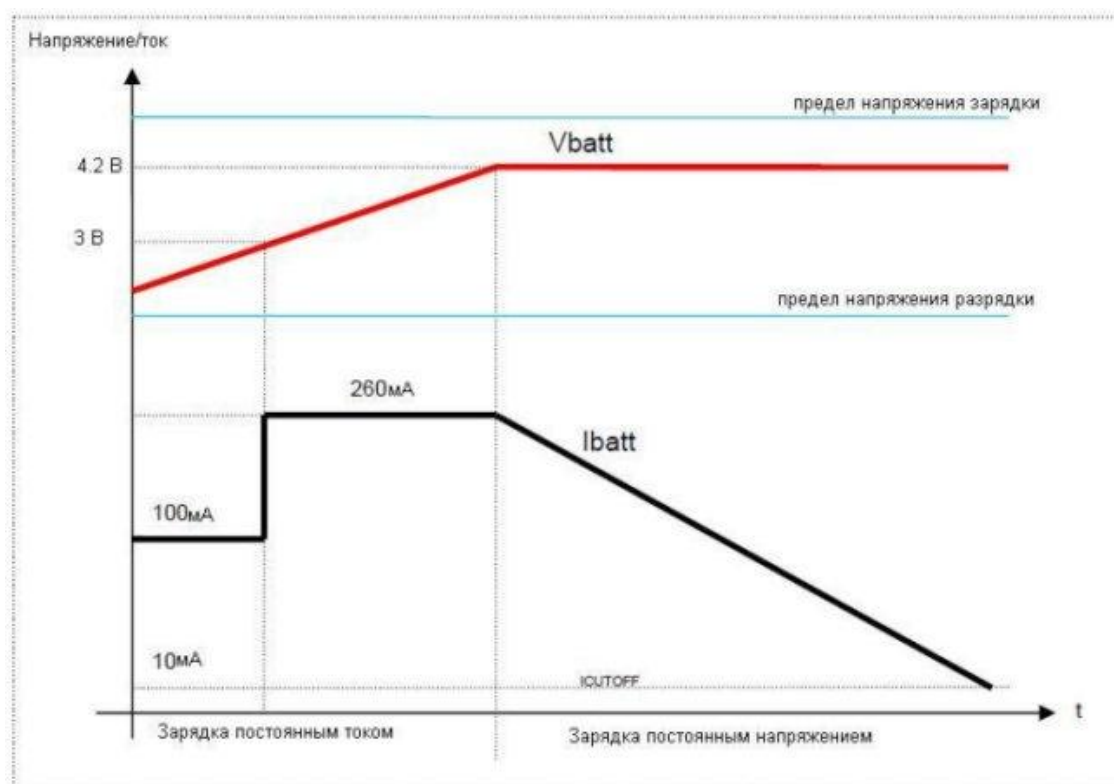


Рисунок 1 – Этапы зарядки и ограничения

На вышеприведенном рисунке «перезарядка» — это уровень, на котором схема защиты элемента включается, размыкает полупроводниковый переключатель и останавливает протекание тока. Впоследствии напряжение батареи, как правило, должно уменьшиться на несколько сотен милливольт, прежде чем возобновится протекание тока. «Чрезмерный разряд» - это уровень, при котором ток протекает, чтобы предотвратить глубокий разряд батареи.

3 Анализ существующих эмуляторов

В данный момент существует две компании, которые производят эмуляторы батарей. Это «Tektronix» под брендом «KEITHLEY» и «Keysight Technologies». Рассмотрим их подробнее.

3.1

Эмулятор N6785A

Компания Keysight Technologies выпустила прибор под именем N6785A. Как утверждает производитель, это отличный способ для решения проблем связанных с батареями. Внешний вид эмулятора представлен на рисунке 2.



Рисунок 2 – Эмулятор N6785A в составе анализатора N6705B

Анализатор N6705B позволяет устанавливать до четырех модулей питания постоянного тока, предоставляет непревзойденные возможности подачи и измерения постоянного тока и напряжения. Это достигается за счет интеграции источников питания высокого класса, а также функций цифрового мультиметра, осциллографа, генератора сигналов произвольной формы и регистратора данных. В результате вместо сложной измерительной схемы, включающей несколько приборов, токовые пробники и шунты, можно обойтись всего одним анализатором N6705B.

Сам же эмулятор N6785A в совокупности с анализатором N6705B предоставляет полностью интегрированное решение, которое включает источник постоянного напряжения и встроенные измерительные функции, упрощающие процесс анализа тока, потребляемого от батареи. Основные возможности включают безразрывное переключение измерительных диапазонов, программируемое выходное сопротивление и дополнительный цифровой мультиметр.

N6785A способен измерять токи от субмикросекундного диапазона до восьми ампер. Это позволяет легко регистрировать ток спящего режима, ток пробуждения, ток активного режима и импульсный ток при передаче.

При измерении тока диапазоны безразрывно переключаются между 8 А, 3 А, 100 мА и 1 мА с поддержанием частоты дискретизации 200 кВб/с и разрешения 18 бит². Конечный результат соответствует диапазону 3 ампера с разрешением 28 бит и погрешностью смещения всего 100 нА. Такая точность амплитуды и разрешение по времени позволяют детально исследовать профиль потребляемого тока.

N6785A может работать в различных режимах, которые легко настраиваются. Это помогает быстро адаптировать систему под конкретные условия применения. В режиме имитации батареи модуль ведет себя как аккумуляторная батарея, в которой можно задать напряжение, а также другие характеристики, в том числе и предельные значения положительного и отрицательного тока. При выходном токе ± 4 А, выходное напряжение будет 20 В, а при выходном токе в ± 8 А, напряжение будет 6 В. Выходное сопротивление программируется от 40 мОм до 1 Ом.

Применение аккумуляторной батареи позволяет проверить поведение устройства в реальных условиях – правильно ли оно работает при пониженном напряжении, какова продолжительность работы от одного заряда батареи и т.п.

На рисунке 3 тестируемое устройство (ТУ) питается от батареи, а N6785A включен последовательно в цепь питания для измерения потребляемого тока. В данном случае используется режим амперметра, в котором N6781A выступает только в роли измерительного прибора.

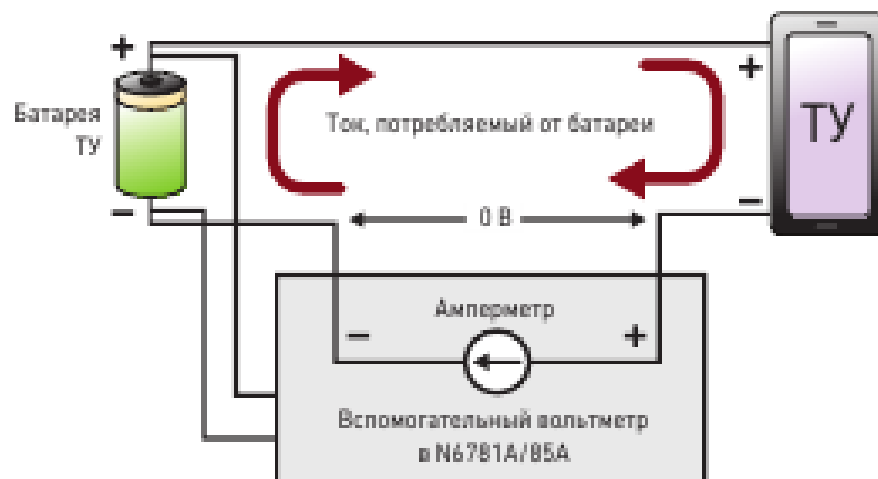


Рисунок 3 – Пример использования N6785A

Также источник питания/измеритель N6785A можно использовать в качестве электронной нагрузки, которая обеспечивает разряд тестируемой аккумуляторной батареи стабильным постоянным током. Кроме того, вы можете установить значение тока, и модуль будет потреблять от батареи именно этот ток.

В режиме тестирования зарядного устройства можно указать напряжение и диапазон заряда, а также предельное значение тока. Поскольку зарядное устройство может только подавать ток, напряжение и ток ограничены положительными значениями.

3.2 Эмулятор 2281S

Эмулятор батареи серии 2281S и прецизионный источник постоянного тока объединяют симуляцию батареи с функциями высокоточного источника питания. 2281S может анализировать потребление постоянного тока тестируемого устройства и генерировать модель аккумулятора на основе процесса зарядки аккумулятора, а также моделировать аккумулятор на основе модели батареи. 2281S способен выдавать мощность 120 Вт при 20 В и 6 А и потреблять ток до 1 А. Внешний вид эмулятора представлен на рисунке 4.



Рисунок 4 – Эмулятор 2281S

Прибор использует линейное регулирование для обеспечения низкого выходного шума и превосходной чувствительности измерения тока нагрузки. Цветной тонкопленочный транзистор (TFT) с высоким разрешением отображает широкий спектр информации об измерениях. Программные кнопки и колесо навигации в сочетании с TFT-дисплеем обеспечивает удобный пользовательский интерфейс, который ускоряет настройку и эксплуатацию прибора. Кроме того, встроенные функции построения графиков позволяют отслеживать такие тренды, как дрейф. Эти функции обеспечивают гибкость, необходимую как для настольных, так и для автоматизированных систем тестирования. Кроме того, 2281S предоставляет режим списка, триггеры и другие функции оптимизации скорости, чтобы минимизировать время тестирования в приложениях автоматического тестирования.

2281S способен имитировать выход батареи во время зарядки и цикла разряда, это позволяет делать следующие вещи:

- а) создание, редактирование, импорт и экспорт моделей;
- б) создание библиотеки моделей батарей, с помощью инструмента Source Measure;
- в) отображение изменений в реальном времени SOC, Voc и Vt для имитируемой батареи;

- г) вычисление емкости аккумулятора в ампер-часах и эквивалентном или последовательном сопротивлении;
- д) программирование батареи SOC, Voc, емкость и внутреннее сопротивление (ERS);
- е) обеспечение двух режимов моделирования – динамический и статический;
- ж) контроль заряда / разряда тока и напряжения;
- з) мощность до 120 Вт с низким уровнем шума, линейная регулируемая мощность;
- и) мониторинг токов нагрузки от 100 нА до 6 А с высокой точностью в режиме питания;
- к) измерение напряжения и тока с разрешением 6S разряд;
- л) ток нагрузки до 1 А и ток источника до 6А;
- м) встроенный графический способ упрощения;

В отличие от обычных источников питания, серия 2281S имеет выход до 120 Вт, 20 В и 6 А, а также может измерять с разрешением до 6,5 разрядов. Выходное напряжение измерения может быть измерено с точностью до 100 мкВ. Эти источники измеряют токи нагрузки от 100 нА до нескольких ампер. Четыре диапазона измерения тока нагрузки (10 А, 1 А, 100 мА и 10 мА) позволяют измерять ток полной нагрузки устройства, ток в режиме ожидания и малый ток в спящем режиме с точностью качества DMM. Высокое разрешение позволяет уверенно распознавать небольшие изменения токов нагрузки. Это также позволяет выполнять измерения в широком диапазоне измерений или в одном диапазоне, но с превосходной точностью как для низких, так и для высоких значений тока.

На рисунке 5 представлен дисплей эмулятора с тестом батареи.

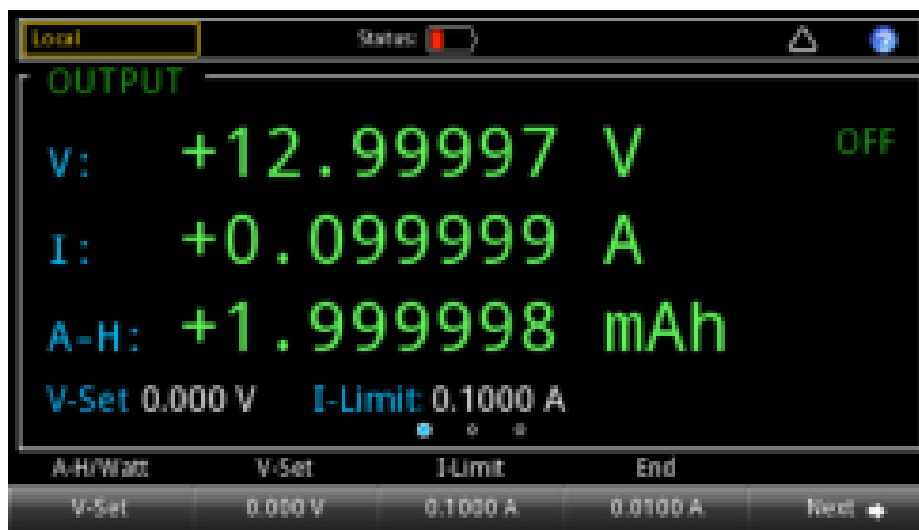


Рисунок 5 – Тест батареи

Серия 2281S настолько же мощна в системе производственных испытаний, насколько гибка на стенде R&D. Они предоставляют команды SCPI для всех функций измерения. Кроме того, некоторые другие функции могут помочь минимизировать время тестирования в автоматизированных системах. Например, внешний триггерный вход позволяет осуществлять аппаратную синхронизацию и управление другими приборами в тестовой системе. Для проверки батареи и функции имитатора батареи цифровой ввод/вывод может быть сконфигурирован как вход или выход. Если цифровой вход/выход сконфигурирован как вход, измерение может быть инициировано внешними сигналами, или если цифровой вход/выход сконфигурирован как выход, цифровой вывод отправит сигнал, когда измерение будет завершено. Кроме того, чтобы сократить время измерения, скорость считывания можно увеличить, сократив время сбора данных с 16,6 мс (или 20 мс) до 33 мкс (40 мкс).

Выбор клемм на передней или задней панели обеспечивает повышенную гибкость подключения. Для максимальной точности напряжения 4-проводное дистанционное зондирование гарантирует, что запрограммированное выходное напряжение фактически соответствует уровню, приложенному к нагрузке. Кроме того, шины считывания контролируются, чтобы обнаружить в них любые разрывы. Эти функции гарантируют, что любые производственные проблемы могут быть

быстро выявлены и устранены. Управление информацией в серии 2281S осуществляется через встроенные интерфейсы GPIB, USB или LAN. Интерфейс USB совместим с тестовой и измерительной системами. Интерфейс локальной сети, совместимый с LXI Core, поддерживает удаленное управление и мониторинг питания 2281S, поэтому инженеры-испытатели всегда могут получить доступ к источнику питания и просматривать измерения, даже если они расположены на другом континенте, чем тестовые системы.

4 Схемотехнический раздел

В данном разделе приведено обоснование выбора элементов разрабатываемого устройства и его частей, расчет рабочих характеристик управляющего микроконтроллера. Также разработана структурная схема всего прибора.

4.1 Разработка структурной схемы

Экспликацией отдельных видов процессов, протекающих в целостных функциональных блоках и цепях разрабатываемого устройства, является принципиальная схема. Она предназначена для отражения общей структуры устройства, его основных блоков, узлов, частей и главных связей между ними. Из принципиальной схемы выясняется назначение устройства и его основные режимы работы [2].

Принципиальная схема представлена на листе формата А1, прилагаемого к данной дипломной работе.

4.2 Назначение и функционал управляющей части устройства

Обвязка цифровой части устройства предназначена для выбора режима измерения и обеспечения рабочего режима управляющего микроконтроллера.

К основным функциям цифровой части прибора можно отнести:

- а) обеспечение рабочего режима микроконтроллера;
- б) обеспечение оптимальных параметров для дальнейшего удобства при программировании и отладке программы;
- в) обеспечение использования всех необходимых интерфейсов и элементов прибора;

Схемотехнически цифровую часть стенда можно разделить на две составляющие: часть установки и сопряжения управляющего контроллера и часть питания контроллера. При этом каждая часть располагается на своей плате и

проектируется с учетом принципов сохранения функционала и обеспечения эргономичности разработки.

4.3 Обоснование выбора элементов цифровой части прибора

С целью уменьшения габаритных размеров конечного устройства применяются готовые решения на основе интегральных микросхем, сочетающих в себе высокую плотность монтажа и показатели быстродействия.

4.3.1 Выбор микроконтроллера

Основным управляющим звеном стенда является микроконтроллер фирмы NXP Semiconductors LPC1765, предназначенный для решения вычислительных задач и управления прибором. Главным критерием выбора контроллера со схемотехнической точки зрения является достаточное количество выводов и интерфейсов, высокая производительность, компактность корпуса, а также удобство пайки и монтажа элемента в целом.

Внешний вид корпуса LQFP100 управляющего контроллера с указанием выводов представлен на рисунке 6.

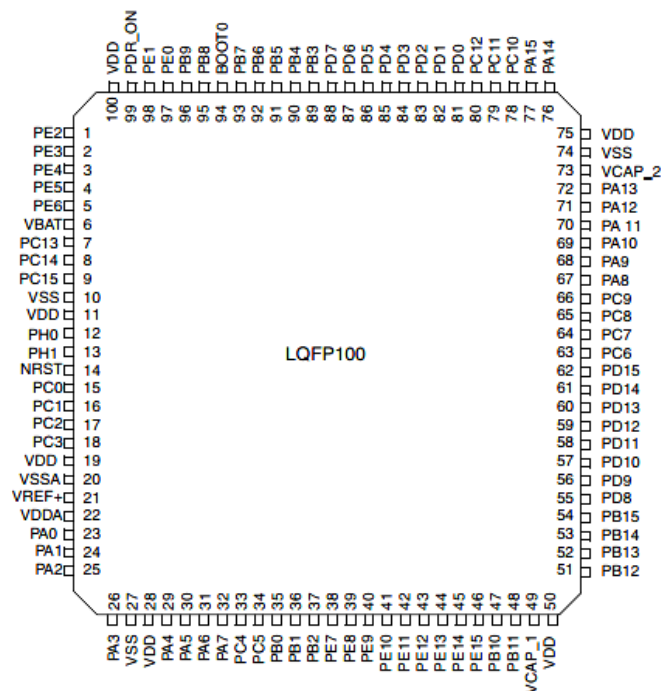


Рисунок 6 – Внешний вид микроконтроллера

При выборе микроконтроллера внимание уделялось на:

- а) низкую стоимость семейства;
- б) наличие необходимых интерфейсов в достаточном количестве;
- в) компактный корпус;
- г) высокую тактовую частоту и как следствие высокую производительность контроллера.

Характеристики выбранного микроконтроллера LPC1765:

- ядро : ARM Cortex-M3, 32-бит;
- flash-память: 256 КБайт;
- RAM-память: 64 КБайт;
- аппаратное управление дисплеем: интерфейсы 6800/8080;
- кол-во каналов АЦП: 8 шт.;
- разрешение АЦП: 12 бит;
- скорость АЦП: 1Мгц;
- кол-во каналов ЦАП: 1 шт.;
- разрешение ЦАП: 10 бит;
- разрешение ШИМ: 16 бит;
- сторожевой таймер: 4 шт.;
- аппаратный модуль RTC: 1 шт.;
- кол-во интерфейсов UART: 4 шт.;
- кол-во интерфейсов CAN: 2 шт.;
- кол-во интерфейсов USB-хост: 1 шт.;
- кол-во интерфейсов USB-устройство: 1 шт.;
- напряжение питания периферии: 2,4...3,6 В;
- рабочий диапазон температур: от минус 40 до плюс 85 °С;
- чувствительность к влажности: уровень MSL 3 ... 168 часов;
- вес – 2,243 г.

– напряжения питания ЦАП и АЦП равны питанию микроконтроллера. На плате проводники опорного напряжения данных модулей подведены к напряжению питания, равному 3,3 В.

На рисунке 7 приведена структурная схема микроконтроллера.

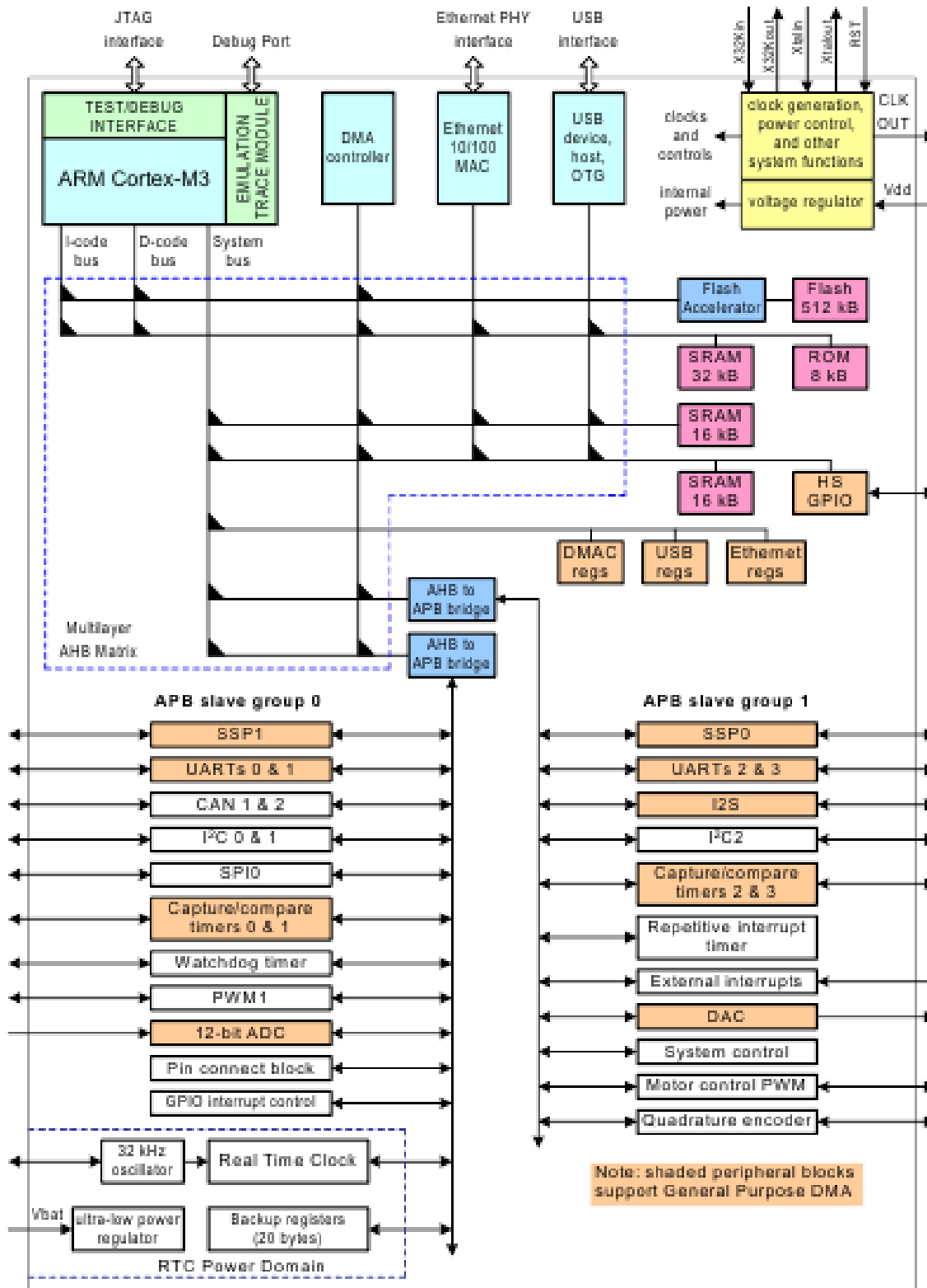


Рисунок 7 – Структурная схема микроконтроллера

4.3.2 Выбор дисплея

Для отображения информации и взаимодействия с пользователем был выбран дисплей компании Sitronix ST7565S. Внешний вид дисплея представлен на рисунке 8, а структурная схема дисплея на рисунке 9.

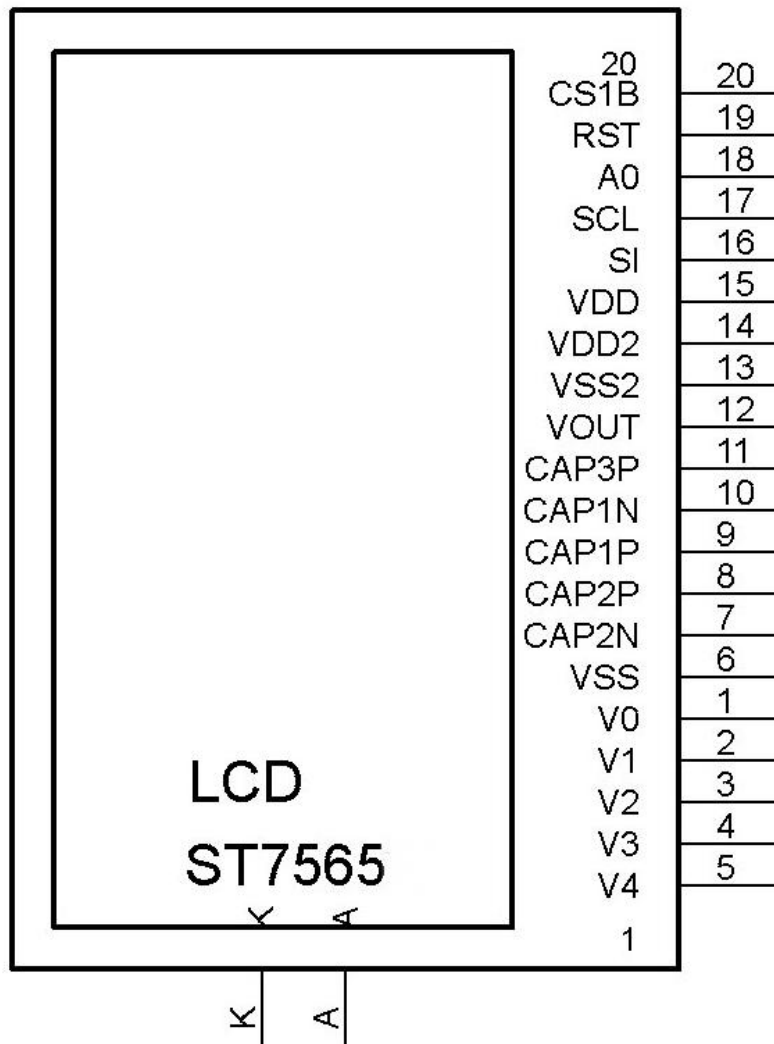


Рисунок 8 – Внешний вид дисплея с выводами

При выборе дисплея внимание уделялось на:

- а) доступность дисплея;
- б) низкая стоимость;
- в) простота подключения;
- г) удовлетворяющее разрешение дисплея.

Характеристики выбранного дисплея ST7565S:

- разрешение дисплея: 132×65 точек;
- рабочий диапазон температур: от минус 40 до плюс 85 °С;
- высокоскоростной интерфейс MPU: 8 бит;
- напряжение питания: 1,8...3,3 В;
- наличие процесса CMOS;
- температурный градиент: минус 0,05 %/°С.

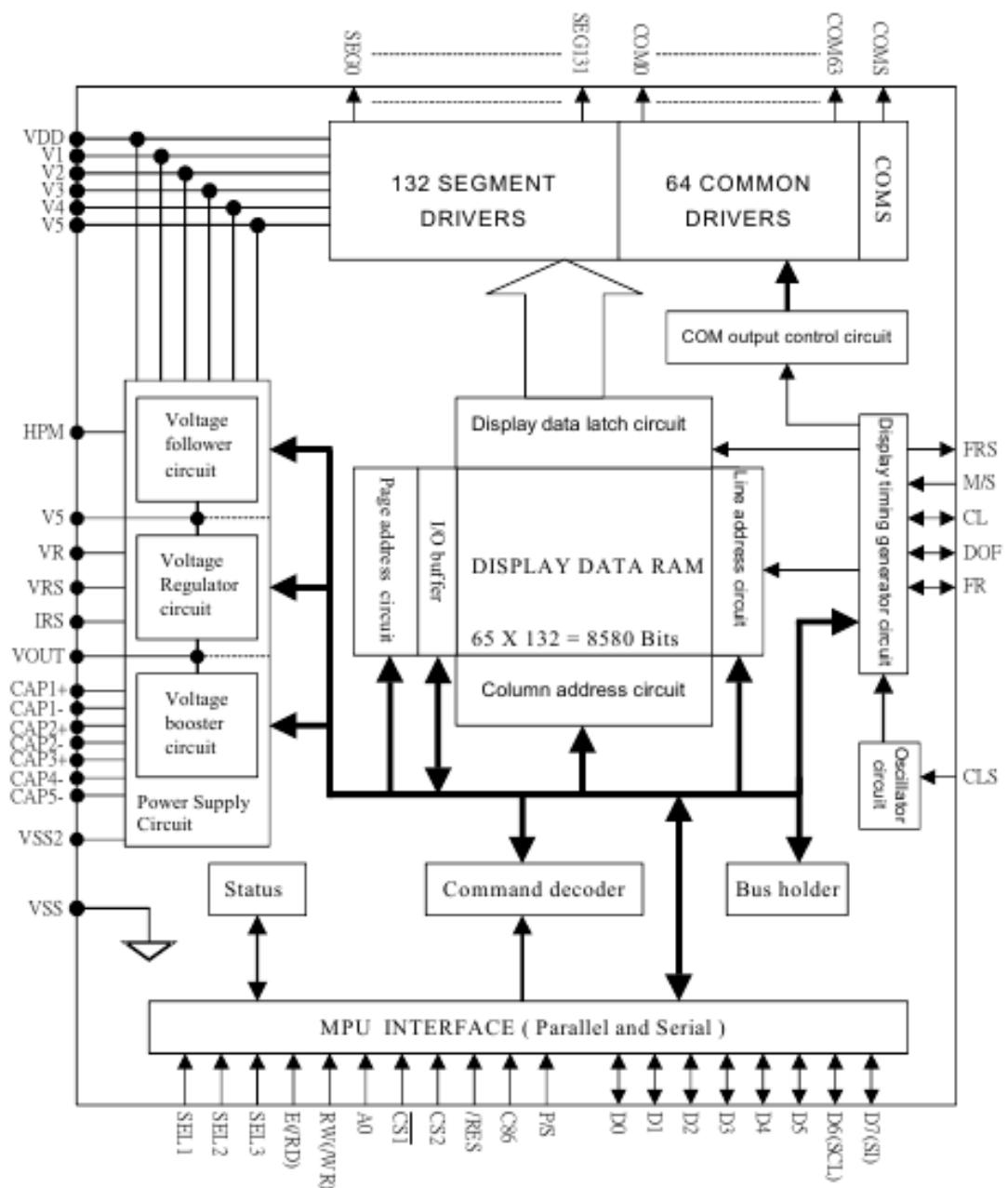


Рисунок 9 – Структурная схема дисплея

4.3.3 Выбор цифрового резистора

Цифровой резистор необходим для имитации процесса зарядки и контроля критической температуры. Для этих целей был выбран цифровой резистор компании Analog Device AD5247. Внешний вид резистора представлен на рисунке 10.

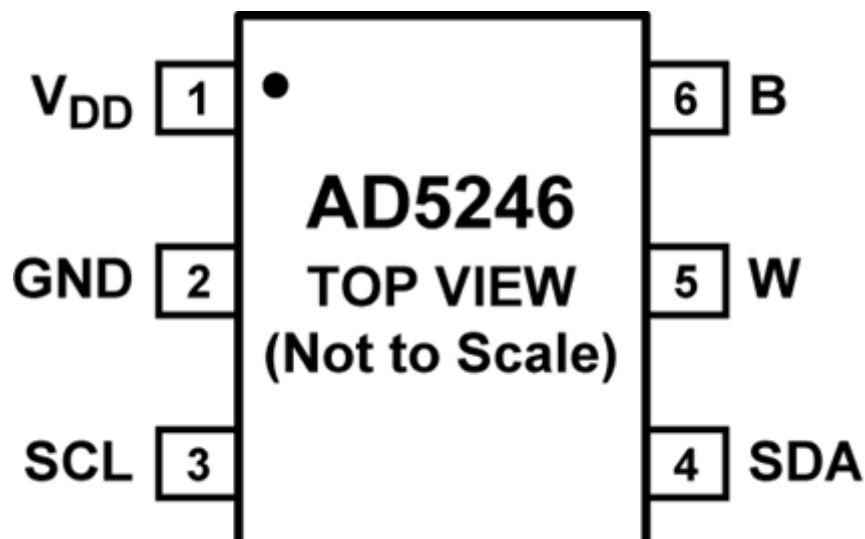


Рисунок 10 – Внешний вид резистора

При выборе цифрового резистора внимание уделялось на:

- а) размеры устройства;
- б) широкий диапазон установки номинала резистора;
- в) возможность пересчета сопротивления в температуру;
- г) широкий диапазон рабочих температур;
- д) наличие интерфейса I2C;

Характеристики выбранного цифрового резистора AD5247:

- количество позиций резистора: 128 позиций;
- рабочий диапазон температур: от минус 40 до плюс 125 °C;
- напряжение питания: 2,7...5,5 В;
- температурный коэффициент: 45 ppm/°C;
- полное чтение/запись регистра.

4.3.4 Выбор операционного усилителя для измерения тока и напряжения зарядки

Операционный усилитель нужен для снятия с его выводов значений тока зарядки и напряжения зарядки. Для этих целей был выбран операционный усилитель фирмы Analog Device AD822. Внешний вид операционного усилителя представлен на рисунке 12.

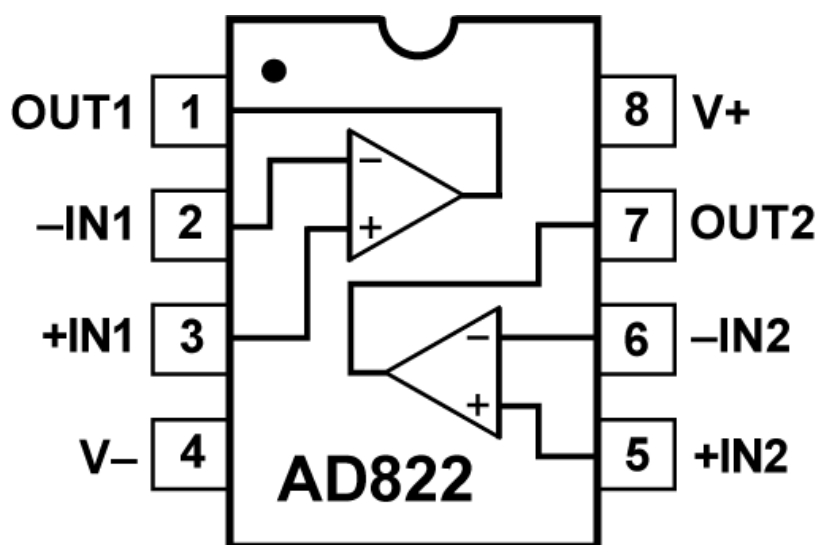


Рисунок 12 – Внешний вид операционного усилителя

При выборе операционного усилителя внимание уделялось на:

- а) маленький ток потребления;
- б) доступность;
- в) наличие особенности Rail-to-Rail.

Характеристики выбранного операционного усилителя AD822:

- напряжение питания: 5...30 В;
- напряжение двойного питания: ($\pm 2,5 \dots \pm 15$) В;
- минимальный выходной ток: 15 мА;
- частотная эффективность усиления: 1,8 МГц;

- ёмкостный привод: 350 пФ;
- максимальный входной ток смещения: 25 пА;
- максимальное входное напряжение смещения: 800 мкВ.

4.3.5 Выбор операционного усилителя для измерения тока утечки

Для измерения тока утечки в спящем режиме был выбран операционный усилитель компании Texas Instrument INA213. Внешний вид операционного усилителя INA213 представлен на рисунке 13.

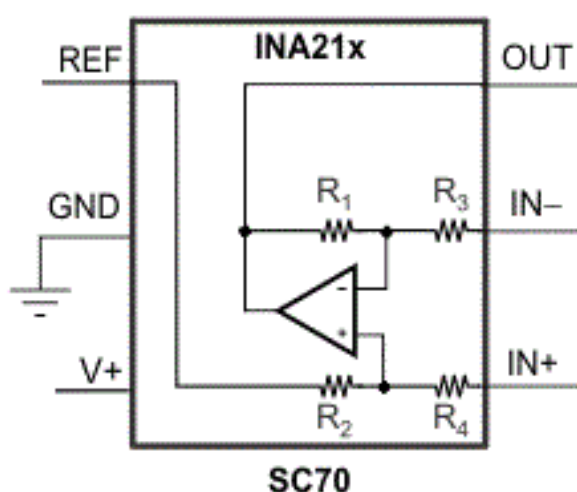


Рисунок 13 – Внешний вид INA213

Характеристики выбранного операционного усилителя INA213:

- напряжение питания: 2,7...26 В;
- максимальный дрейф смещения: 0,5 мкВ/°С;
- коэффициент усиления по напряжению: 50;
- рабочий диапазон температур: минус 40 плюс 125 °С;
- максимальное смещение: 100 мкВ;
- диапазон допустимых напряжений смещения: минус 0,3...26 В;
- ток собственного потребления: 65 мкА;
- ширина полосы сигнала: 14 кГц.

5 Конструкторский раздел

В данном разделе рассматривается метод изготовления и выбор материала печатных плат, приводится графическое их представление, обоснование выбора элементной базы. При этом учитывается, что весь процесс конструирования сводится к 3м основным задачам: микроминиатюризации, обеспечению необходимого для нормальной работы теплового режима и технологичности конструкции.

5.1 Разработка и трассировка плат

Проектирование цифровой части стенда и трассировка плат производились в ПО РСAD 2002 [6]. Данная программа позволяет автоматизировать процесс проектирования плат, соблюдая основные правила ГОСТ и стандартов проектирования ПП, составить необходимую спецификацию. При работе с программой были заданы начальные установки, которые впоследствии при работе использовались по умолчанию: шаг координатной сетки 2,54 мм, автоматическое измерение расположения элементов в мм, минимальное расстояние проводников - не менее 0,5 мм, расстояние между двумя соседними проводниками - не менее 0,6 мм. Данные условия выбираются исходя из последующей технологии изготовления ПП - фоторезистивного способа. При том учитывается, что центры отверстий и контактных площадок должны располагаться в узлах координатной сетки. По краям печатной платы и вокруг монтажных отверстий оставляются свободные зоны, в которых не должно быть проводников и прочих элементов.

5.2 Выбор материала печатных плат

Исходя из условия технического задания принято решение о создании цифровой части стенда имитации на двухслойной печатной плате [7]. Это позволяет уменьшить габариты и вес готового изделия. Кроме того, печатные платы обладают

большими преимуществами по сравнению с остальными видами проектирования. Основными их достоинствами являются:

- а) увеличение плотности монтажа и возможность микроминиатюризации изделий;
- б) гарантированная стабильность электрических характеристик;
- в) повышенная стойкость к климатическим и механическим воздействиям;
- г) унификация и стандартизация конструктивных изделий;
- д) возможность комплексной автоматизации монтажно-сборочных работ.

При разработке устройства за основу был взят нормативный документ ГОСТ Р 53432-2009 «Платы печатные. Общие технические требования к производству». Согласно этому документу, для производства печатных плат можно использовать способ фоторезистивного травления платы с последующей металлизацией отверстий. Данный метод основан на том, что слои между собой соединяются сквозными, металлизированными отверстиями, что позволяет создать плату с добавлением на другой стороне необходимых перемычек и переходных отверстий и увеличить плотность монтажа элементов при неизменном объеме платы. Материал платы выбран в соответствии с ГОСТ 10316-78- стеклотекстолит, который обладает высокой механической прочностью, химической прочностью, хорошими электроизоляционными свойствами и низким влагопоглощением. Номинальная толщина платы составляет $(1,5 \pm 0,20)$ мм в соответствии со 2-ым классом материала, напряжение пробоя составляет 30 Кв/мм, диапазон рабочих температур от минус 60 до плюс 150 °С.

5.3 Выбор элементной базы

При разработке печатных плат исходя из соображений экономии пространства используются элементы поверхностного монтажа типоразмера 0603. Это позволяет увеличить плотность, уменьшить размеры платы, распределить возникающие в процессе эксплуатации тепловые процессы.

Расстояние между элементами выбирается исходя из методики изготовления плат, а также с учетом размеров контактных площадок во избежание замыкания между собой.

В процессе разработки рассчитываются номинальные мощности элементов. Полный ток потребления цифровой части устройства составляет 50 мА и определен опытным путем. Напряжение цифровой части стенда не превышает 3,3 В. Из приведенных значений рассчитывается максимальная мощность элементов:

$$W = IU = 0,05 \cdot 3,3 = 0,165 \text{ Вт} . \quad (1)$$

Применение элементов, корпус которых отличен от SMD, обусловлено удобством их монтажа на плату, имеющимися на рынке номиналами и ценой.

Перечень элементной базы представлен в таблице 1

Таблица 1 – Элементная база

Наименование	Кол. шт.
Конденс.SMD-C0402-X7R-10n-10%-50V-Walsin	1
Конденс.SMD-C0402-X7R-100n-10%-50V-MURATA-GRM155R71H104KE14D	7
Конденс.SMD-C0603-NP0-39p-5%-50V-SAMSUNG	2
Конденс.SMD-C0603-NP0-56p-5%-50V-TDK	1
Конденс.SMD-C0603-NP0-1n-5%-50V-MURATA	2
Конденс.SMD-C0603-X7R-10n-10%-50V-SAMSUNG	1
Конденс.SMD-C0603-X7R-100n-10%-50V-SAMSUNG	8
Конденс.SMD-C0603-X7R-1u0-10%-50V-Walsin	2
Конденс.SMD-C0805-X5R-2u2-10%-50V-MURATA	3
Конденс.SMD-C0805-X7R-1u0-10%-50V-SAMSUNG	1
Резистор SMD-R0603-0,05W-470R0-1%	1
Резистор SMD-R0402-0,062W-5k10-1%	1
Резистор SMD-R0603-0,100W-5R1-5%	1
Резистор SMD-R0603-0,100W-100R0-1%	4
Резистор SMD-R0603-0,100W-100R-5%	8

Продолжение таблицы 1

Наименование	Кол. шт.
Резистор SMD-R0603-0,100W-1k00-1%	2
Резистор SMD-R0603-0,100W-1k0-5%	5
Резистор SMD-R0603-0,100W-2k00-1%	1
Резистор SMD-R0603-0,100W-4k70-1%	1
Резистор SMD-R0603-0,100W-10k-5%	11
Резистор SMD-R0603-0,100W-56k0-1%	1
Резистор SMD-R0603-0,100W-100k-5%	4
Резистор SMD-R0603-0,250W-0R10-1%-100ppm	1
Резистор SMD-R0603-0,250W-390R-5%	3
Предохранитель-1,1А-MF-MSMF110/16-2-BOURNS	1
Предохранитель-3,15А-ВП4-5	1
Дрос.SMD-BC54-470M-NL	1
Дрос.SMD-BLM18HG601SN1D	1
Кварц-SMD0503-12,000MHz-30/30ppm-20pF--10/+60C-FTX12.000M20SM5S	1
Электролит SMD-Tantal-100-10V-10%-Case_D-VISHAY	1
Электролит SMD-105C-KZ-56-25V-6,3*5,4	1
Транзистор-IRFZ24N	1
Транзистор-IRLML6402TR-PBF	1
Транзистор-PDTC114ET-SOT23	1
Кнопка тактовая-KS-00Q-01-E-Switch	3
Разъем-BWL-4_W7166-04PSTB00	1
Разъем-C3917-40BTI00R	1
Разъем-EWL-2_W2620-02PSTW00R	1
Разъем-PBD-12LP_C4828-12BDGHYT	1
Разъем-Питания-K375A(DJK-02A)	1
Диод-BAT854W-SOT323	1
Диод-US1NWF-7-SOD123F-Diodes	1
Микросхема-AD5247BKSZ10-2RL7	1
Микросхема-AD5247BKSZ50-RL7	1
Микросхема-AD820AR-SOIC8	1
Микросхема-AD822AR-SOIC8	1

Продолжение таблицы 1

Наименование	Кол. шт.
Микросхема-ADR390AUJZ	1
Микросхема-INA213AIDCKT	1
Микросхема-LM22672MRE-ADJ	1
Микросхема-LM317LDG-SOIC8	1
Микросхема-LMV331IDCKR	1
Микросхема-LPC1765FBD100	1
Резистор с выводами-2R2-5%-5W-SQP-5	2
Стабилитрон-SM15T22A-STMicroelectronics	1
Стойка-Пластик-TP-30 (h30mm)	4
Контроллер-ST7565S	1

5.4 Графическое представление плат

Графическое представление наглядно позволяет отобразить получившиеся результаты трассировки плат. При этом в дальнейшем возможно изменять параметры проектирования и вносить коррективы в разработку в реальном времени. Графическое объемное представление платы, на которой располагается микроконтроллер и дисплей, представлено на рисунках 14 и 15.

Основным элементом платы является микроконтроллер LPC1765. Его применение обусловлено большим ассортиментом данного типа контроллеров на рынке. Основное отличие между семействами состоит в количестве выводов и корпусе. Выбранная модель отличается количеством выводов - 100 и соответствующим типоразмером TQFP, а также компактностью корпуса.

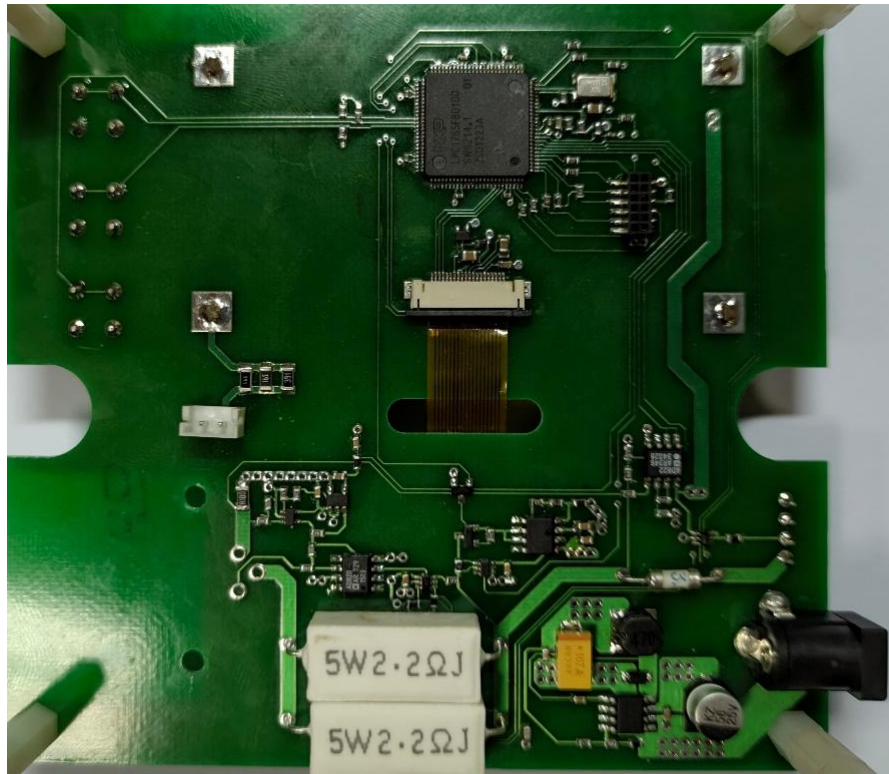


Рисунок 14 - Плата управляющего контроллера вид снизу

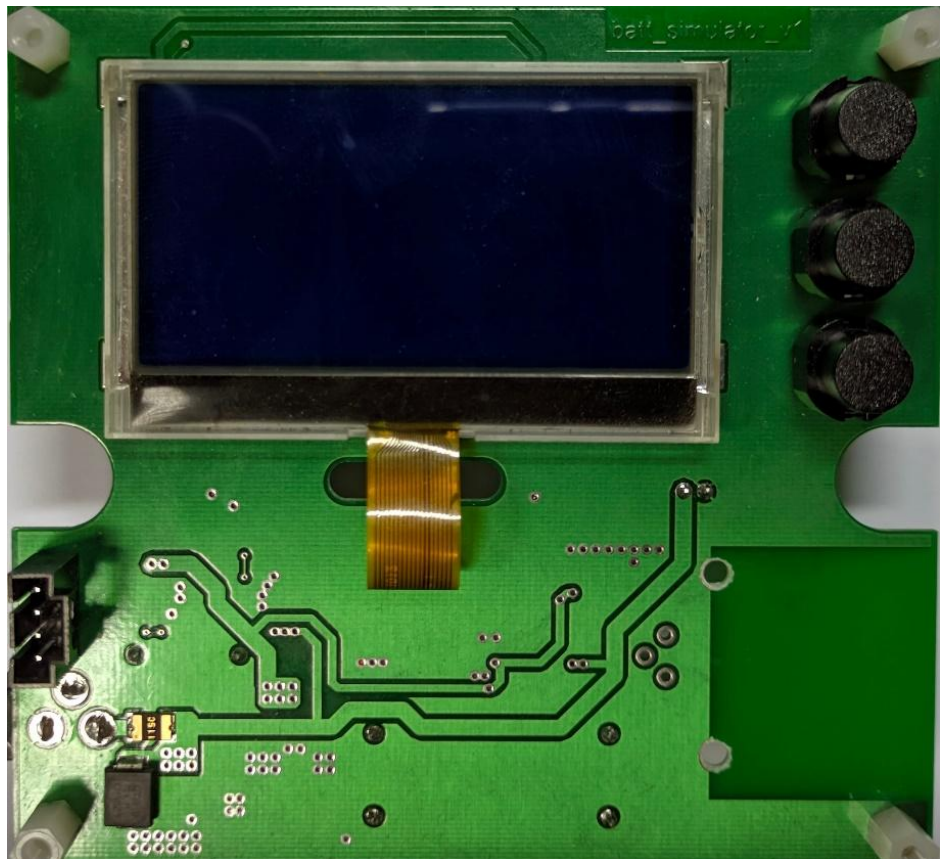


Рисунок 15 - Плата управляющего контроллера вид сверху

6 Программный раздел

В данном разделе приводится обоснование применения функций микроконтроллера, выбора периферийных модульных частей и языка программирования микропроцессорной и компьютерной программных сторон. Выделены основные этапы разработки и написания кода на основании составленных схемотехнических требований [3].

6.1 Разработка программы для микроконтроллера

Программа для микроконтроллера разрабатывается в программной среде Embedded Workbench IarIDE на основе платформы Windows 7. Выбор данной среды обусловлен ее бесплатным распространением и использованием, а также удобством интерфейса [4].

Основные функции программы представляются в виде отдельных файлов и дублируются сочетанием клавиш для быстрого доступа к основным действиям при разработке. В левой части программы имеется окно, где располагается структура проекта. Так как программа разрабатываемого прибора применяет встроенных функций среды и готовых решений программирования SPL и HAL, в окне также отображается используемое ядро системы CMSIS, включающее в себя адресацию всех основных регистров программируемого контроллера. Его применение позволяет сократить время на изучении дополнительной технической документации, а также адресных пространств регистров.

Основным окном служит окно ввода программного кода, занимающее большую часть интерфейса для удобства просмотра кода. Переключение между файлами при помощи вкладок позволяет быстро перенести или скопировать информацию между разрабатываемыми частями проекта. Язык программирования в данной среде – C. Большое распространение языка делает его универсальным средством программирования и переноса кода между платформами большинства устройств [5].

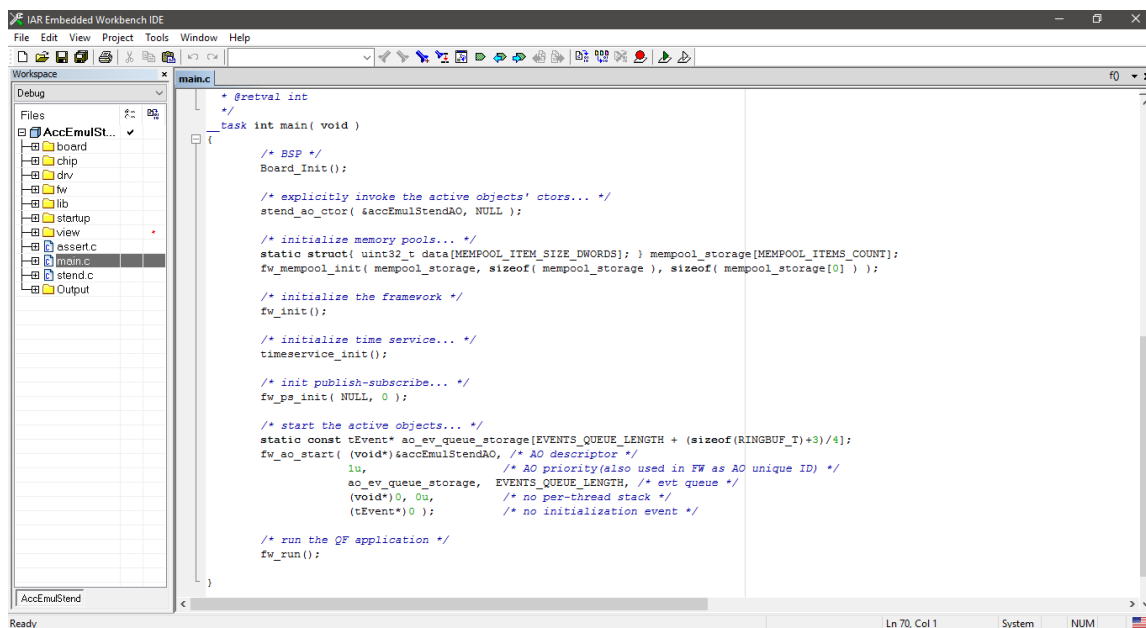


Рисунок 16 – Графический интерфейс программы Embedded Workbench IarIDE

Приведенный в приложении данной дипломной работы код управляющего микроконтроллера является общим алгоритмом всех применяемых функций контроллера, которые поочередно выполняются в одном бесконечном цикле. Для удобства весь процесс создания кода разделяется на несколько стадий:

- а) подключение необходимых файлов и библиотек;
- б) инициализация переменных;
- в) описание функций;
- г) основной цикл программы для вычислений.

Ниже представлен листинг программы для начальной настройки микроконтроллера и периферии.

```
void Board_Debug_Init(void) {
#if defined(DEBUG_ENABLE)
Chip_UART_Init(DEBUG_UART);
Chip_UART_SetBaud(DEBUG_UART, 115200);
Chip_UART_ConfigData(DEBUG_UART,          UART_LCR_WLEN8
UART_LCR_SBS_1BIT | UART_LCR_PARITY_DIS);
Chip_UART_TXEnable(DEBUG_UART);
#endif
}
```

```

    }
    void Board_UARTPutChar(char ch) {
    #if defined(DEBUG_ENABLE)
    while      ((Chip_UART_ReadLineStatus(DEBUG_UART)      &
UART_LSR_THRE) == 0) {}
    Chip_UART_SendByte(DEBUG_UART, (uint8_t) ch);
    #endif
    }
    int Board_UARTGetChar(void) {
    #if defined(DEBUG_ENABLE)
    if (Chip_UART_ReadLineStatus(DEBUG_UART) & UART_LSR_RDR) {
    return (int) Chip_UART_ReadByte(DEBUG_UART);
    }
    #endif
    return EOF;
    }
    void Board_UARTPutSTR(char *str) {
    #if defined(DEBUG_ENABLE)
    while (*str != '\0') {
    Board_UARTPutChar(*str++);
    }
    #endif
    }
    void Board_Init(void) {
    DEBUGINIT();
    Chip_GPIO_Init(LPC_GPIO);
    Chip_IOCON_Init(LPC_IOCON);
    Board_Kb_Init();
    Board_SPI_Init();
    Board_Glcd_Init();
    Board_I2C_Init();
    Board_UART_Init();
    Board_ADC_Init();
    Board_Debug_Init();
    }

```

6.2 Разработка программы для клавиатуры

В разработанном устройстве для взаимодействия пользователя с прибором находятся 3 кнопки. С помощью этих кнопок, в главном меню пользователь выбирает тип эмулируемой батареи, какой тест провести и запуск теста. В окне с

результатами измерений будут использоваться две кнопки, для возврата в главное меню и повторного проведение теста.

При нажатии на кнопку, создается сигнал, который попадет в обработчик прерываний. Далее, функция обработки нажатия ловит сигнал из обработчика прерываний и производит дальнейшую настройку аппаратуры. Ниже приведен код программы, обрабатывающий, какая кнопка была нажата.

```
void kb_handler(void) {
    tKbCtrl* kb = &kb_ctrl;
    unsigned char key_code;
    ++kb->counter;
    key_code = Board_Kb_GetKey();
    if (kb->kbstate.f_holded == false) {
        if (key_code) {
            kb->kbstate.f_holded = true;
            kb->kbstate.key_code = key_code;
            kb->last_counter = kb->counter;
        }
        } else { unsigned int duration = kb->counter - kb->last_counter;
        if (duration > UINT16_MAX)
            duration = UINT16_MAX;
        if (kb->kbstate.key_code == key_code) {
    if ( duration >= KB_SHORT_PRESS_MAX_MS
        && (duration % KB_HOLD_EVENTS_PERIOD_TICKS) == 0) {
        kb_handler_publish_event( &(const tKbEventData) { .state =kb->kbstate,
            .duration = (uint16_t)duration } );
        }
        } else { kb->kbstate.f_holded = false;
    kb_handler_publish_event( &(const tKbEventData) { .state =kb->kbstate,
        .duration = (uint16_t)duration } );
        memset( &kb->kbstate, 0, sizeof(kb->kbstate) );
        }
    }
}
```

Также в приведенном выше листинге обрабатывается не только простое нажатие, но и удержание клавиши, что создает другой сигнал для обработчика прерываний.

6.3 Разработка программы для дисплея

Выбранный дисплей имеет разрешение в 132×65 точки, но последний ряд пикселей не используется, так как дисплей разделен на 8 страниц по 8 пикселей в высоту. Поэтому разрешение дисплея для работы с ним составляет 132×64 точки.

Для корректного отображения символов был выбран размер одного символа 8×5 точек, и все символы были закодированы в отдельный файл, для удобной работы с ними.

Ниже приведен листинг программы для инициализации дисплея, полный листинг программы представлен в приложении.

```
void glcd_init(void) {
Board_Glcd_Rst_Clr();
Board_Glcd_Rst_Set();
reset_delay();
write_cmd(GLCD_BIAS_SET | 0x00); // 1/9 bias
write_cmd(GLCD_ADC_SELECT | 0x00); // normal (column adr inc: 0 - 83h)
write_cmd(GLCD_COM_SELECT | 0x08); // revers (row adr inc: 63 -> 0)
write_cmd(GLCD_RES_RATIO | 0x03); //
write_cmd(GLCD_SET_V5);
write_cmd(50);
write_cmd(GLCD_SET_MODE_NORMAL);
write_cmd(GLCD_BOOST_SET);
write_cmd(0);
write_cmd(GLCD_POWER_SET | 0x07);
write_cmd(GLCD_ON_DISPLAY | 0x01); // display on
glcd_clear_screen( cBLACK ); }
```

6.4 Разработка программы для цифрового резистора

Цифровой резистор используется для измерения выходного тока и напряжения, а также для регулировки температуры. Поэтому было написано две программы. В первой программе происходит измерение и регулировка температуры, а во второй программе измерение тока и напряжения на выходе устройства.

Ниже приведены две программы, где происходит настройка резистора, для измерения тока и напряжения, а также измерения температуры.

Первая программа для настройки резистора для измерения температуры:

```
void term_emul_init(void) {
    ad5247_ctor(&r,&(const
        tAD5247_Cfg){.adrR=AD5247BKSZ10_2RL7,.d=0,.id=I2C0});
}
void term_emul_set_resist_by_temp(int temp) {
    assert(temp >= 22 && temp <= 95);
    int resist_value = set_temp(temp);
    r.d = ad5247_convert_r_to_position(&r, resist_value);
    ad5247_set_position(&r);
}
static inline int set_temp(int temp) {
    tResistTemp res = {0};
    find_interval(temp, &res);
    int degree_resist = (res.last_res - res.next_res) / MEASUR_STEP;
    int resist = res.last_res - ((temp - res.last_temp) * degree_resist);
    return resist;
}
static inline void find_interval(int temp, tResistTemp* res) {
    for (int i = 1; i <= 33; i++) {
        if (temp <= t_r[i][0] && temp >= t_r[i-1][0]) {
            res->last_temp = t_r[i-1][0];
            res->next_temp = t_r[i][0];
            res->last_res = t_r[i-1][1];
            res->next_res = t_r[i][1];
            break;
        }
    }
}
```

Вторая программа для измерения тока и напряжения:

```
void charge_emul_init(void) {
    ad5247_ctor(&r,&(const tAD5247_Cfg){.adrR=AD5247BKSZ50_RL7,.d =
MIN,.id = I2C0});
}
void charge_emul_set_resist(int engine_posit) {
    assert(engine_posit >= MIN && engine_posit <= MAX);
    r.d = set_engine(engine_posit);
    ad5247_set_position(&r);
}
static inline int set_engine(int engine_posit) {
```

```

    int engine_value = (MAX - engine_posit);
    return engine_value;
}

```

6.5 Разработка программы для пересчета кода АЦП в цифровое значение

Все измерения снимаются с АЦП и приходят в виде кода. Данный код необходимо преобразовать и сделать пересчет снятых характеристик для отображения корректной информации.

Ниже приведен листинг программы, для пересчета значений из кода АЦП в значения тока и напряжения.

```

static int result_ready = 0;
static bool async_mode;
static tVaccMeas vaccMeas;
static tData acc_meas_data;
static tAccMeasParam param_meas;
static const tAccMeasParam result_done = {
    .super = { .sig = ACC_MEAS_DONE_SIG },
    .id = acc_measRESULT_COMPLETE
};
void ADC_IRQHandler(void) {
    static int counter = 0;
    static int done_flag = 0;
    uint32_t dr_chan0 = LPC_ADC->DR[ADC_CH0];
    uint32_t dr_chan1 = LPC_ADC->DR[ADC_CH1];
    uint32_t dr_chan2 = LPC_ADC->DR[ADC_CH2];
    uint32_t dr_chan3 = LPC_ADC->DR[ADC_CH3];
    if (ADC_DR_DONE(dr_chan0) == 1) {
        acc_meas_data.i_acc_code += ADC_DR_RESULT(dr_chan0);
        done_flag |= 1 << ADC_CH0;
    }
    if (ADC_DR_DONE(dr_chan1) == 1) {
        acc_meas_data.v_acc_code += ADC_DR_RESULT(dr_chan1);
        done_flag |= 1 << ADC_CH1;
    }
    if (ADC_DR_DONE(dr_chan2) == 1) {
        acc_meas_data.i_leak_acc_code += ADC_DR_RESULT(dr_chan2);
        done_flag |= 1 << ADC_CH2;
    }
    if (ADC_DR_DONE(dr_chan3) == 1) {

```

```

        acc_meas_data.v_acc_ref_code += ADC_DR_RESULT(dr_chan3);
        done_flag |= 1 << ADC_CH3;
    }
    if (done_flag == ((1 << ADC_CH0) | (1 << ADC_CH1) | (1 << ADC_CH2) |
(1 << ADC_CH3))) {
        done_flag = 0;
        counter++;
    }
    if (counter == LAST_MEAS) {
        LPC_ADC->CR &= ~ADC_CR_BURST;
        LPC_ADC->INTEN &= ~((1UL << ADC_CH0) |
            (1UL << ADC_CH1) |
            (1UL << ADC_CH2) |
            (1UL << ADC_CH3));
        acc_meas_data.i_acc_code /= 8;
        acc_meas_data.v_acc_code /= 8;
        acc_meas_data.i_leak_acc_code /= 8;
        acc_meas_data.v_acc_ref_code /= 8;
        counter = 0;
        if (async_mode == true) {
            acc_meas_data.v_acc_ref_mv =
convert_adc_code_to_correct_v(acc_meas_data.v_acc_ref_code);
            param_meas.v_acc =
convert_adc_code_to_v(acc_meas_data.v_acc_code);
            param_meas.i_acc =
convert_adc_code_to_i(acc_meas_data.i_acc_code);
            param_meas.i_leak_acc =
convert_adc_code_to_i_leak(acc_meas_data.i_leak_acc_code);
            on_result_done();
        } else {result_ready = 1;}
    }
}
void acc_meas_sync(tAccMeasParam* param) {
    async_mode = false;
    LPC_ADC->INTEN |= ((1UL << ADC_CH0) | (1UL << ADC_CH1) | (1UL
<< ADC_CH2) | (1UL << ADC_CH3));
    Chip_ADC_SetBurstCmd(LPC_ADC, ENABLE);
    while (!result_ready) continue;
    acc_meas_data.v_acc_ref_mv =
convert_adc_code_to_correct_v(acc_meas_data.v_acc_ref_code);
    param->v_acc = convert_adc_code_to_v(acc_meas_data.v_acc_code);
    param->i_acc = convert_adc_code_to_i(acc_meas_data.i_acc_code);
    param->i_leak_acc =
convert_adc_code_to_i_leak(acc_meas_data.i_leak_acc_code);

```

```
    }  
    void acc_meas_async(void) {  
        async_mode = true;  
        LPC_ADC->INTEN |= ((1UL << ADC_CH0) | (1UL << ADC_CH1) | (1UL  
<< ADC_CH2) | (1UL << ADC_CH3));  
        Chip_ADC_SetBurstCmd(LPC_ADC, ENABLE);  
    }  
}
```

7 Безопасность жизнедеятельности

7.1 Электробезопасность эмулятора источника питания

Общие требования к электробезопасности предусматривает ГОСТ 12.2.007.0-75 ССБТ «Изделия электротехнические. Общие требования электробезопасности.»

В данной дипломной работе разработано программное обеспечение для устройства, которое относится к классу III электрозащиты, так как не имеет ни внешних, ни внутренних электрических цепей с напряжением более 12 вольт относительно общего провода и более 24 В по размаху, а также получает питание непосредственно от источника напряжения 12 В. Поскольку устройство принадлежит к классу III по электрозащите, то оно не имеет заземления.

Тепловое излучение устройства невелико и определяется только рассеиваемой мощностью входящих в его состав радиоэлементов и составляет порядка 165 мВт, поэтому нет необходимости в применении средств защиты от теплового излучения.

Электрическая схема устройства исключает его самопроизвольное включение и отключение. На разъемах для подключения сочленяемых токоведущих частей устройства должна быть маркировка, на ответных разъемах сочленяемых токоведущих частей также должна быть маркировка для исключения неправильного подключения прибора к радиоэлектронной системе.

Элементы индикации и отображения информации располагаются интуитивно понятно, исходя из удобства функционирования, снабжаются поясняющими надписями и рисунками, указывающими их назначение или выполняемую функцию.

Для кнопок и разъемов выполняются поясняющие рисунки и соответствующие надписи. Для обозначения состояния питания предусмотрен индикатор, указывающий состояние устройства: «ВКЛЮЧЕН/ВЫКЛЮЧЕН».

Основным средством отображения информации служит дисплей, отображаемые символы и графические знаки на котором должны быть понятны любому работающему с устройством человеку. Для этого можно изменяться размер текста, если имеются соответствующие отклонения от нормы зрения у пользователя. Дисплей устанавливается в центре на горизонтальной линии и сагиттальной

плоскости, для того, чтобы при работе не возникало потребности в повороте головы и дополнительной концентрации зрения. Границы дисплея располагаются в вертикальной плоскости под углами, не превышающими $\pm 30^\circ$ от нормальной линии взгляда и в горизонтальной плоскости под углом $\pm 30^\circ$ от сагиттальной плоскости, что обеспечивает комфортные условия при работе с ним.

Все взаимодействие пользователя с устройством происходит с помощью трех кнопок. Внизу дисплея описано для чего используется каждая кнопка. При выборе теста измерения или типа аккумулятора при нажатии на соответствующие кнопки информация на дисплее обновляется. При неверном использовании интерфейса измерительные процессы проводятся не будут. Также после теста измерения, его результаты выводятся на дисплей с описанием каждого значения. При ошибке в измерении на дисплей выводится соответствующее сообщение.

7.2 Анализ потенциально опасных и вредных производственных факторов (ОВПФ)

В дипломной работе разработано программное обеспечение для платы эмулятора батареи для переносных радиоизмерительных станций. Работы по созданию программного обеспечения эмулятора проводились на предприятии ООО «ПЛАНАР». В данной главе будут рассмотрены вопросы соблюдения норм безопасности при разработке программы для данного устройства [11].

Проанализируем потенциально опасные и вредные производственные факторы. Все ОВПФ можно разделить на:

а) Физические:

- неблагоприятные параметры микроклимата;
- недостаточная освещенность рабочей зоны;
- шум;
- пожароопасность;
- статическое электричество.

б) Психофизиологические:

- нервно-психические перегрузки;
- монотонность труда.

Недостаточный уровень освещения вызывает у человека ослабление внимания, утомление глаз, резь в глазах. Шум повышает артериальное давление, ослабляет зрение, замедляет пищеварение, снижает активность сердечной деятельности. При прохождении через человека электрический ток вызывает электролиз биологических жидкостей, нарушения работы отдельных органов, нагрев кровеносных сосудов и мышечных тканей, судорожные сокращения мышц. Неправильная рабочая поза вызывает преждевременную усталость, возникновение различных профессиональных заболеваний.

Устройство не является источником оптического, рентгеновского, ультразвука и, соответственно, не имеет средств ограничения интенсивности указанных излучений. Также эмулятор не является источником шума.

7.3 Нормирование ОВПФ и мероприятия по их ликвидации

7.3.1 Освещение

Документом, регулирующим освещенность зоны, является СанПиН 2.2.2/2.2.4.1340-03 «Требования к освещению на рабочих местах, оборудованных ПЭВМ».

Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м².

Следует ограничивать отраженную блескость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м² и яркость потолка не должна превышать 200 кд/м².

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

7.3.2 Пожарная безопасность

Пожарная безопасность устройства определяется в соответствии ГОСТ 12.1.004-91 «Пожарная безопасность. Общие требования». Эмулятор источника питания должен быть сконструирован и изготовлен таким образом, что оно не представляет пожарной опасности в нормальных условиях эксплуатации и при аварийных режимах. При нормальной и аварийной работе устройства ни один из элементов конструкции не превышает температуру выше допустимых значений,

установленных ГОСТ 12.2.006-87 «Безопасность аппаратуры электронной сетевой и сходных с ней устройств, предназначенных для бытового и аналогичного общего применения. Общие требования и методы испытаний»

Для поддержания комфортных температурных условий внутри корпуса устройства предлагается наличие радиатора, для отвода тепла от операционного усилителя, чтобы не случилось выгорание элемента.

В помещении, где эксплуатируются установки под напряжением, для тушения пожара должны применяться вещества не проводящие электрический ток. К таким веществам относятся: углекислый газ, азот, песок, негорючие порошковые материалы. В качестве первичных средств тушения пожара применяются огнетушители ОУ-2 и ОП-5. Их количество определяется исходя из площади помещения. В нашем случае площадь составляет 45 м², поэтому достаточно двух огнетушителей.

7.3.3 Микроклимат

В соответствии с требованиями СанПиН 2.2.4.3359-16 "Гигиенические требования к микроклимату производственных помещений" необходимо создать оптимальные условия производственной среды в рабочей зоне в зависимости от времени года, категории работ по энергозатратам, избытка явного тепла.

Показателями, характеризующими микроклимат в производственных помещениях, являются:

- а) температура воздуха;
- б) температура поверхностей;
- в) относительная влажность воздуха;
- г) скорость движения воздуха;
- д) интенсивность теплового облучения.

Процесс работы относится к I категории – легкие физические работы, производимые сидя, стоя или связанные с ходьбой, но не требующие систематического напряжения или поднятия и переноски тяжестей. Оптимальные и

допустимые нормы температуры, относительной влажности, скорости движения воздуха с учетом периода года и категории работ приведены в таблице 2.

Таблица 2 – Нормированные параметры микроклимата

Период года	Температура, °С		Относительная влажность, %		Скорость ветра, м/с	
	Оптимальная	Допустимая	Оптимальная	Допустимая	Оптимальная	Допустимая
Холодный	22...24	21...25	40...60	75	0,1	<0,1
Теплый	23...25	22...28	40...60	70 (при 25°С)	0,1	0,1...0,2

Время года включает: холодный и переходный периоды с температурой наружного воздуха ниже плюс 10°С; теплый период с температурой плюс 10°С и выше. Параметры метеорологических условий должны контролироваться на рабочих местах.

7.3.4 Требования к шумам и вибрационному воздействию

Требования к шумовым воздействиям и вибрационной безопасности устанавливает СанПиН 2.2.4.3359-16 «Требования к уровням шума и вибрации на рабочих местах, оборудованных ПЭВМ». В соответствии с этим документом шум следует подразделять на:

- а) постоянный, уровень звука которого за 8-часовой рабочий день изменяется во времени не более чем на 5 дБА;
- б) непостоянный, уровень звука которого за 8-часовой рабочий день изменяется во времени более чем на 5 дБА.

В нашем случае предельно-допустимый уровень шума согласно СанПин 2.2.4.3359-16 не превышает 80 дБА.

Качественные и количественные критерии и показатели неблагоприятного воздействия вибрации на человека-оператора в процессе труда устанавливаются санитарными нормами, правилами и другими нормативными документами Минздрава.

В соответствии с ними вводятся следующие критерии оценки неблагоприятного воздействия шума и вибрации:

- а) критерий «безопасность», обеспечивающий сохранение здоровья оператора, оцениваемого по объективным показателям с учетом риска возникновения предусмотренных медицинской классификацией профессиональной болезни и патологий, а также исключающий возможность возникновения травмоопасных или аварийных ситуаций из-за воздействия шумов и вибрации;
- б) критерий «граница снижения производительности труда», обеспечивающий поддержание нормативной производительности труда оператора, не снижающейся из-за развития усталости под воздействием шумов и вибрации;
- в) критерий «комфорт», обеспечивающий оператору ощущение комфортности условий труда при полном отсутствии мешающего действия шумов и вибрации.

Эмулятор не создает колебаний в диапазоне воспринимаемых человеческим ухом длин волн, поэтому нет необходимости в расчетах вибрационных и шумовых воздействий. Устройство не имеет подвижных частей и других подобных элементов, создающих слышимые и воспринимаемые организмом вибрации. Уровень шума в помещении, где производится работа с эмулятором должна соответствовать всем необходимым нормам шумо- и виброзащиты. В процессе эксплуатации прибора за 8-часовой рабочий день уровень звука должен изменяться во времени не более чем на 5 дБА, следовательно, не вызывать неприятных и болевых ощущений у работника.

7.4 Требования к организации рабочего места

При размещении рабочих мест с ПЭВМ расстояние между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Рабочие места с ПЭВМ в помещениях с источниками вредных производственных факторов должны размещаться в изолированных кабинах с организованным воздухообменом. Рабочие места с ПЭВМ при выполнении творческой работы, требующей значительного умственного напряжения или высокой концентрации внимания, рекомендуется изолировать друг от друга перегородками высотой 1,5-2,0 м.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600-700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики. Поверхность рабочего стола должна иметь коэффициент отражения 0,5-0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ. Рабочий стул (кресло) должен быть подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего

края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию. Поверхность сиденья, спинки и других элементов стула (кресла) должна быть полумягкой, с нескользящим, слабо электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

7.4.1 Организация рабочего места

Площадь помещения (рисунок 17), в котором расположено рабочее место, составляет 45 м². Площадь рабочего места пользователя ПК с ЭЛТ-дисплеем составляет около 3 м², что ниже минимума, указанного в СанПиН 2.2.2/2.4.1340-03. Рабочих мест пятнадцать. В помещении проводится ежедневная влажная уборка и систематическое проветривание после каждого часа работы. Источниками шума в помещении являются печатающие устройства, сканеры, стационарные телефоны, персональные компьютеры, серверы и тому подобные.

Искусственное освещение происходит за счет, как общего равномерного освещения, выполненного в светильнике с четырьмя светодиодными лампами, расположенными параллельно окнам в три ряда. Каждый ряд имеет четыре светильника, так и дополнительных источников света – настольные лампы на каждом столе, что важно, т.к., чаще всего именно плохое освещение рабочего места оказывает более пагубное для зрения влияние, чем сам факт нахождения за компьютером.

Естественное боковое одностороннее освещение осуществляется посредством оконного проема, расположенного на северной и западной стороне здания. Окно оборудовано устройствами, регулирующими освещение – жалюзи.

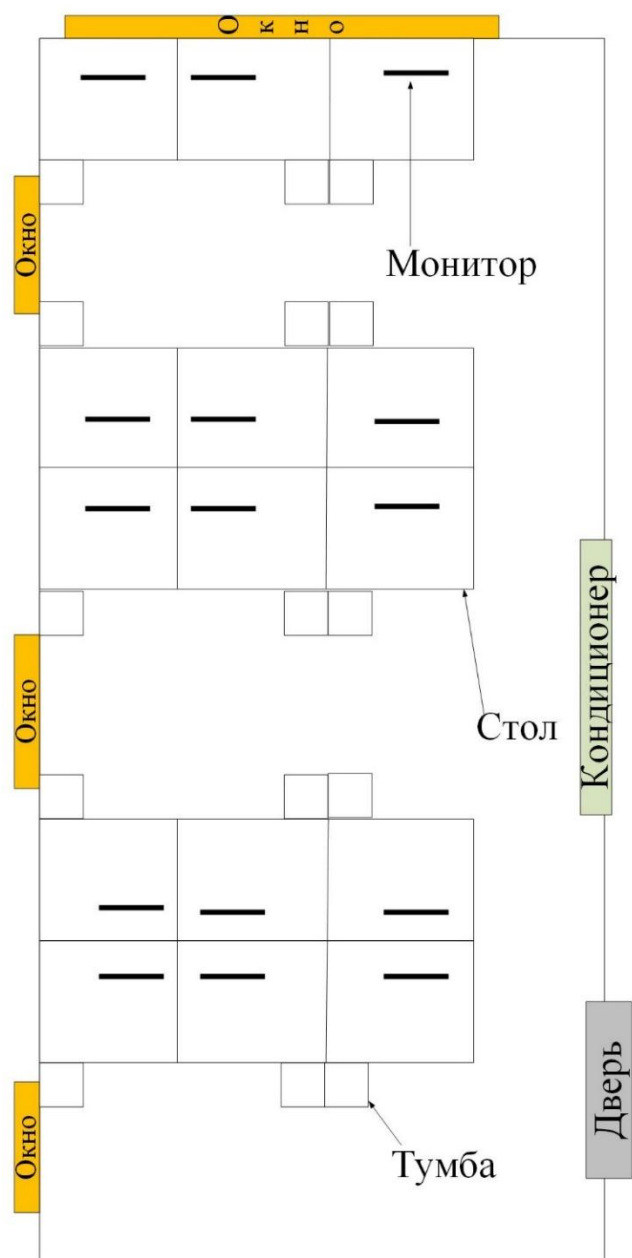


Рисунок 17 – Расположение рабочего места

Рабочее помещение оборудовано системой кондиционирования воздуха (кондиционер), что способствует поддержанию оптимальной для комфортной работы температуры помещения (СанПиН 2.2.4.3359-16). Рабочий стол обеспечивает оптимальное размещение на рабочей поверхности используемого оборудования. Во избежание поражения электрическим током выполнено зануление. Высота рабочей поверхности стола составляет 730 мм.

Непосредственно сама поверхность стола имеет ширину 800 мм, а также глубину 1000 мм. Столы имеют пространство для ног высотой 750 мм, шириной –

800 мм, глубиной на уровне колен – не менее 550 мм и на уровне вытянутых ног – не менее 750 мм. Рабочее место имеет специальную подставку для ног, имеющую ширину не менее 300 мм, глубину не менее 400 мм.

Источником шума является оргтехника, а именно сканер и принтер, расположенные на тумбе рядом с рабочим местом. Данная техника не оказывает негативного влияния на органы слуха, т.к. уровень шума не превышает 55 дБ.

Конструкция рабочего кресла для работника обеспечивает поддержание рациональной рабочей позы и позволяет изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины. Кресло имеет подъемно-поворотный механизм, позволяющий осуществлять регулирование его по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья. Клавиатура располагается на основной столешнице. Экран видеомонитора находится от глаз пользователя на расстоянии 700 мм.

Организация рабочего места, за исключением некоторых недочетов, соответствует требованиям СанПиН 2.2.2/2.4.1340-03; СанПиН 2.2.4.1294-03 и СанПиН 2.2.4.3359-16.

8 Организационно-экономический раздел

В данном разделе приводится экономический анализ и сетевое планирование разрабатываемого устройства. Оно заключается в выявлении оценки экономического эффекта от производства проектируемой единицы продукции. В процессе выполнения данного раздела выделены и рассчитаны основные этапы экономического планирования, по итогам которого построен сетевой график, наглядно отражающий время выполнения работ. Также проведен расчет сметы производства и себестоимости готового изделия.

В конце раздела подведены итоги экономического анализа, в которых отображены примерные сроки окупаемости и возможно удачные варианты для запуска в дальнейшем серийной продукции разработки.

8.1 Анализ аналогов, сравнение технических параметров

Эмулятор источника питания является внутренней разработкой ООО «ПЛАНАР», для проверки системы зарядки своих же выпускаемых радиоизмерительных станций.

Разработанный эмулятор позволяет работать в любом положении оператору цеха, а также работать и в полевых условиях, для выявления неполадок в системах. В настоящее время на рынке существует несколько аналогов разрабатываемого стенда. Одним из зарубежных аналогов является прибор эмулятор N6785A. Данный прибор относится к группе крупногабаритных, любая его перевозка - достаточно сложный и затратный процесс. Кроме того, закупка импортного оборудования - дорогостоящая и продолжительная по времени операция по сравнению с использованием собственной разработки. N6785A позволяет проводить измерения характеристик лишь на определенных модулях, которые не входят в комплект поставки и требуют дополнительных затрат.

Другим зарубежным аналогом является эмулятор 2281S. Он также относится к группе крупногабаритных, что доставляет проблемы при транспортировке. В

отличии от N6785A, для 2281S нет необходимости докупать дополнительные модули, но это не сильно уменьшает его цену.

Оба зарубежных аналога предназначены для проведения измерений в стационарных условиях.

8.2 Понятия параметров сетевого планирования

Сетевой анализ является методом анализа сроков начала и окончания частей проекта, который позволяет связать выполнение различных работ и процессов во времени, получив прогноз общей продолжительности реализации всего проекта. С его помощью можно выделить основные наиболее рациональные пути развития этапов производства от идеи до готовой единицы продукции и получить объективную оценку этих параметров при выбранном варианте структуры работ и распределения ресурсов [8].

Существуют различные методы построения сетевого планирования:

а) Детерминированные сетевые методы

- диаграмма Ганта с дополнительным временным люфтом 10...20 %
- метод критического пути (МКП).

б) Вероятностные сетевые методы

- Неальтернативные
 - 1) метод статистических испытаний (метод Монте-Карло);
 - 2) метод оценки и пересмотра планов (PERT).
- Альтернативные
 - 1) метод графической оценки и анализа (GERT).

Сетевое планирование и управление содержит три основных этапа: структурное планирование, календарное планирование и оперативное управление. Оно начинается с разбиения исследуемого проекта на определенные шаги, необходимые для достижения конечной цели разработки. Затем определяются

оценки продолжительности работ и строится сетевой график, который позволяет, прежде всего, оценить временные характеристики проекта и входящих в него работ.

Сетевой график состоит из элементов двух видов – работ и событий, и позволяет в наглядной форме представить временную структуру проекта. Он отображает взаимосвязи между работами внутри проекта и порядок их выполнения. График является направленным графом, в котором каждая работа представляется ориентированной дугой, а каждое событие – узлом. Событие в свою очередь определяется как момент времени, когда завершается одна работа и начинается другая.

В процессе разработки проекта, при построении сетевого графика учитываются основные правила его построения:

а) перед построением графика в технологической последовательности устанавливаются правила выполнения начала и окончания работ;

б) сетевая модель строится от начала к окончанию при прямом расчете и в обратном направлении при обратном расчете;

в) в сетевой модели отсутствуют "тупиковые" события;

г) в сетевой модели отсутствуют замкнутые контуры - пути, соединяющие события, связанные сами с собой.

Особое значение в сетевом графике имеют критические работы. Работа считается критической, если задержка ее начала приводит к задержке срока окончания проекта в целом. Некритическая работа отличается тем, что промежуток времени между ее ранним началом и поздним окончанием больше ее фактической продолжительности. Другими словами, любая не критическая работа имеет резерв времени, что позволяет максимально оптимизировать её выполнение. Поэтому в данной дипломной работе в экономических расчетах используется метод критического пути. Он исходит из того, что длительность операций можно оценить с достаточно высокой степенью точности и определенности. Основным достоинством метода критического пути является возможность манипулирования сроками выполнения задач, не лежащих на критическом пути.

Задачи, лежащие на критическом пути, имеют нулевой резерв времени выполнения и в случае изменения их длительности изменяются сроки всего проекта. В связи с этим при разработке критические задачи требуют более тщательного контроля, в частности, своевременного выявления проблем и рисков, влияющих на сроки их выполнения и, следовательно, на сроки выполнения проекта в целом.

Календарное планирование по выбранному МКП требует определенных входных данных. После их ввода производится процедура прямого и обратного прохода по сети и вычисляется выходная информация.

8.3 Анализ этапов разработки, построения сетевого графика

Перед построением модели сетевого графика рассчитываются временные параметры всех составляющих его компонентов.

Для полного построения необходимо произвести два вида расчетов: прямой и обратный. При прямом расчете вычисляются ранние сроки выполнения каждого события. При обратном вычислении они производятся в обратной последовательности, что позволяет вычислить поздние сроки событий. Также на сетевом графике отмечаются номера событий и разница во времени между ранними и поздними сроками.

Прямой расчет – определение минимально возможного времени реализации проекта начинается с работ, не имеющих предшественников. В ходе него определяется ES (ранний старт) и EF (ранний финиш). Ранние начала и ранние окончания работ определяются последовательно, слева направо по графику, то есть от исходного события сети к завершающему.

Основными формулами для расчетов являются:

$$ES_0 = 0, \quad (8.1)$$

$$EF = ES + CONT, \quad (8.2)$$

где: ES_0 - нулевой ранний старт,

ES - ранний старт события,

EF - ранний финиш события,

CONT -продолжительность события.

Ожидаемое время выполнения работы ($t_{\text{ожид.}}$) рассчитывается по двухоценочной методике, исходя из минимальной ($t_{\text{мин.}}$) и максимальной ($t_{\text{макс.}}$) оценок продолжительности работы. При этом предполагается, что минимальная оценка соответствует наиболее благоприятным, а максимальная - наиболее неблагоприятным условиям работы.

Ниже приведены основные формулы прямого расчета. Ожидаемая продолжительность каждой работы рассчитывается по формуле:

$$t_{\text{ожид.}} = 0,6 t_{\text{мин.}} + 0,4 t_{\text{макс.}}, \quad (8.3)$$

Среднеквадратическое отклонение (δ_{ij}) продолжительности выполнения работ в двухоценочной методике определяется по формуле:

$$\delta_{ij} = 0,2 \cdot (t_{ij\text{MAX}} - t_{ij\text{MIN}}), \quad (8.4)$$

Дисперсия определяется по формуле:

$$D_{ij} = 0,04 \cdot (t_{ij\text{MAX}} - t_{ij\text{MIN}})^2, \quad (8.5)$$

Если при прямом расчете возникают ситуации выполнения двух или более параллельных процессов, то ранний срок свершения следующего промежуточного события вычисляется по функции максимума:

$$t_p = \max(t_{pi} + t_{ij}), \quad (8.6)$$

где: t_{ij} - ожидаемая продолжительность работы,

t_{pi} - ранний срок свершения события, непосредственно предшествующего данному.

Обратный расчет. Определяются *LS* (поздний старт), *LF* (поздний финиш) и *R* (резерв). Поздние начала и поздние окончания определяются в обратном порядке – от завершающегося события графика к исходящему, то есть справа налево.

Основными формулами для расчетов являются:

$$EF_N = LF_N, \quad (8.7)$$

$$LS_i = LF_i - CONT, \quad (8.8)$$

где: LF_N - поздний финиш конечного события,

LS - поздний старт события,

LF - поздний финиш события,

$CONT$ - продолжительность события.

После прохождения по сетевому графику в прямом и обратном направлениях должно выполняться условие $ES_0=LS_0$. Это означает, что для завершающего события никакие резервы времени не планируются. Если при обратном расчете возникают ситуации выполнения двух или более параллельных процессов, то поздний срок свершения следующего промежуточного события вычисляется по функции минимума:

$$t_s = \min(t_{si} - t_{ij}), \quad (8.9)$$

где: t_{si} - ожидаемая продолжительность работы,

t_{ij} - поздний срок свершения события непосредственно следующего за данным промежуточным событием.

Резерв времени свершения события - это промежуток времени, на который может быть отсрочено событие без нарушения сроков разработки в целом. Он образуется у событий, для которых поздний срок свершения события больше раннего срока, то есть имеется свободный временной промежуток:

$$R_i = t_s - t_p, \quad (8.10)$$

Если $t_s = t_p$, то есть поздний срок свершения события равен раннему сроку свершения события, то такое событие не имеет резерва времени и это событие относится к критическому пути. Результаты расчетов временных параметров событий сетевого графика приведены в таблице 3.

Таблица 3 – Проведение работ по разработке эмулятора

Код работы	Наименование работы	Продолжительность, дн.			Исполнители, чел.		
		min	max	ожид.	Рук.	Инж.	Практ.
0,1	Анализ ТЗ	1	2	1	1	1	1
1,2	Выбор литературных источников	2	4	3	0	0	1
1,3	Определение стадий и этапов разработки эмулятора	2	4	3	1	1	1
2,4	Поиск и анализ существующих аналогов	4	6	5	1	1	1
3,4	Разработка структурной схемы	5	6	5	0	1	0
4,5	Разработка принципиальной схемы	4	5	4	1	1	0
5,6	Трассировка печатной платы	4	6	5	0	1	0
6,7	Изготовление фотошаблонов печатных плат	2	5	3	0	0	1
6,8	Заказ и поставка элементов для монтажа	10	14	12	1	0	0
7,9	Разработка структуры программы управляющего микроконтроллера	3	6	4	0	1	0
8,9	Поиск библиотек для микроконтроллера	2	4	3	0	0	1
9,10	Создание программы для микроконтроллера	12	15	13	0	1	1
9,11	Установка на печатную плату микросхем и элементов поверхностного монтажа	2	4	3	0	0	1
10,12	Отладка программы микроконтроллера	3	5	4	0	0	1

Продолжение таблицы 3

Код работы	Наименование работы	Продолжительность, дн.			Исполнители, чел.		
		min	max	ожид.	Рук.	Инж.	Практ.
11,12	Выявление плюсов и минусов алгоритмов, их оптимизация	4	6	5	0	1	0
12,13	Оформление программной документации	2	3	2	1	0	1
12,14	Анализ литературы организационно-экономического раздела	3	5	4	0	0	1
13,15	Разработка и анализ сетевого графика	2	3	2	0	0	1
14,15	Расчет прибыли и экономического эффекта	3	5	4	1	1	1
15,16	Оформление организационно-экономического раздела	4	5	4	0	0	1
15,17	Анализ литературы раздела БЖД	3	4	3	0	0	1
16,18	Разработка мер безопасности человека от негативных факторов	2	5	4	1	1	0
17,18	Оформление раздела БЖД	4	5	4	0	0	1
18,19	Оформление технической документации	3	5	4	1	1	1
18,2	Оформление пояснительной записки	2	6	5	0	0	1
19,21	Оформление графической документации дипломной работы	6	8	7	1	1	1
20,21	Проверка и сдача работы	5	7	6	1	0	1

После построения сетевого графика проводится его всесторонний анализ, с тем чтобы в дальнейшем принять меры по его оптимизации. Определить степень трудности выполнения в срок каждой группы работ не критического пути можно с помощью коэффициента напряженности работ. Коэффициентом напряженности K_H работы называется отношение продолжительности несовпадающих (заключенных между одними и теми же событиями) отрезков пути, одним из которых является путь максимальной продолжительности, проходящий через данную работу, а другим – критический путь:

$$K_H = \frac{t(L_{\max}) - t'_{кр}}{t_{кр} - t'_{кр}}, \quad (8.11)$$

где: $t(L_{\max})$ – продолжительность максимального пути, проходящего через работу;

$t_{кр}$ – продолжительность отрезка рассматриваемого пути, совпадающего с критическим путем;

$t'_{кр}$ – продолжительность критического пути.

Для работ, лежащих на критическом пути коэффициент напряженности K_{Hij} равен 1. Коэффициент напряженности всегда меньше либо равен 1, его величина подсчитывается как отношение суммы продолжительностей отрезков максимального пути, проходящего через данную работу, не совпадающих с критическим путём $t_{\max sij}$ к сумме продолжительностей отрезков критического пути, не совпадающих с максимальным путём, проходящим через эту работу $t_{кр ij}$.

В зависимости от коэффициента напряженности все работы попадают в одну из трёх зон напряженности:

- а) критическую, $K_{Hij} > 0,8$;
- б) промежуточную, $0,5 \leq K_{Hij} \leq 0,8$;
- в) резервную, $K_{Hij} < 0,5$.

Результаты расчётов представлены в таблице 4.

Таблица 4 – Параметры работ графика

Код работ ы	Ожидаем. продолжительн ость	Сроки начала		Сроки окончания		Резервы времени			Свободн ый	Коэф ф. напря жен ности
		Ран - ний	Позд - ний	Ранни й	Поздни й	Полны й	Частны й 1 рода	Частны й 2 рода		
0,1	1	0	0	1	1	0	0	0	0	1,00
1,2	3	1	1	4	4	0	0	0	0	1,00
1,3	3	1	4	5	7	0	0	0	0	1,00
2,4	5	4	4	9	9	0	0	0	0	1,00
3,4	5	4	4	9	9	0	0	0	0	1,00
4,5	4	9	9	13	13	0	0	0	0	1,00
5,6	5	13	13	18	18	0	0	0	0	1,00
6,7	3	18	18	21	29	0	0	0	0	1,00
6,8	12	18	18	30	30	0	0	0	0	1,00
7,9	4	21	29	25	33	8	0	0	-8	0,21
8,9	3	30	30	33	33	0	0	0	0	1,00
9,10	13	33	33	46	46	0	0	0	0	1,00
9,11	3	33	33	36	36	0	0	0	0	1,00
10,12	4	46	46	50	50	0	0	0	0	1,00
11,12	5	33	36	38	41	3	0	0	-3	0,74
12,13	2	50	50	52	52	0	0	0	0	1,00
12,14	4	50	50	54	54	0	0	0	0	1,00
13,15	2	52	54	54	56	2	0	0	-2	0,14
14,15	4	54	54	58	58	0	0	0	0	1,00
15,16	4	58	58	62	62	0	0	0	0	1,00
15,17	3	58	58	61	61	0	0	0	0	1,00
16,18	4	62	62	66	66	0	0	0	0	1,00
17,18	4	58	61	62	65	1	0	0	-1	0,34
18,19	4	66	66	70	70	0	0	0	0	1,00
18,20	5	66	66	71	71	0	0	0	0	1,00
19,21	7	70	70	77	77	0	0	0	0	1,00
20,21	6	71	71	77	77	0	0	0	0	1,00

Коэффициент сложности сетевого графика представляет собой отношение количества работ сетевого графика к количеству событий и определяется по формуле:

$$K_{\text{сложн}} = n_{\text{раб}} / n_{\text{соб}}, \quad (8.12)$$

где K_c – коэффициент сложности сетевого графика;

$n_{раб}$ – количество работ, ед.;

$n_{соб}$ – количество событий, ед.

Сетевые графики, имеющие коэффициент сложности от 1,0 до 1,5, являются простыми, от 1,51 до 2,0 – средней сложности, более 2,1 – сложными.

Для данного графика коэффициент сложности не превышает 1,5, график является простым:

$$K_{сложн} = 29/27=1,16, \quad (8.13)$$

Согласно приведенной выше таблице и расчетным данным, приводится сетевой график с рассчитанными значениями затрат по времени на каждую стадию разработки, он приведен ниже на рисунке 18, а также на отдельном листе А1, который прилагается к пояснительной записке.

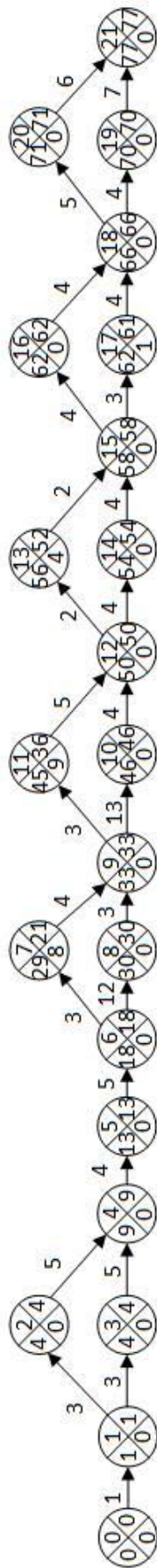


Рисунок 18 – Сетевой график

8.4 Расчет стоимости работы

В процессе расчета стоимости разработки учитываются все основные затраты, которые необходимы для полного цикла производства готовой конкурентно-способной продукции. Поэтому далее приводятся расчеты себестоимости стенда, прибыли от конечной разработки. Также определен примерный срок окупаемости [9].

8.4.1 Расчет себестоимости готового изделия

Одной из главных характеристик, которые используют экономический анализ для выявления эффективности деятельности предприятия, выступает себестоимость реализованной и готовой продукции.

Себестоимостью являются суммированные затраты денежных, трудовых, материальных ресурсов, затраченных на производство и реализацию продукции. Также она включает в себя затраты, связанные с предпринимательской деятельностью [10].

Способы расчета себестоимости продукции зависят от уровня готовности самой продукции, а также от ее вида и назначения. В данной работе необходимо учитывать затраты на производство всего стенда имитации сварки, а не отдельных его частей, так как данное устройство становится конкурентоспособным лишь в законченном собранном виде.

Объектом себестоимости является разрабатываемая продукция предприятия, поэтому составные части экономических показателей могут быть измерены:

- а) в натуральных показателях;
- б) в увеличенных натуральных единицах;
- в) в условных, или условно-натуральных измерителях;
- г) в стоимостных измерителях;
- д) в трудовых измерителях;
- е) в условных измерителях.

Формула расчета по затратам на производство готовой продукции выглядит следующим образом:

$$C = MЗ + A + Tr. + \text{прочие затраты}, \quad (8.14)$$

где: C – себестоимость;

$MЗ$ – материальные затраты;

A – амортизация;

$Tr.$ – затраты на оплату труда, а также отчисления в различные фонды.

Первой составляющей суммы в расчете себестоимости являются материальные затраты. Они включают в себя закупочную стоимость всех компонентов, необходимых для создания готовой единицы продукции. Так как исследованием данной работы является цифровая часть стенда, то расчеты приведены в соответствии с количеством используемых компонентов плат микроконтроллера и соединительной платы цифровой части. Смета затрат на электронные компоненты приведена в таблице 5.

Таблица 5 – Смета затрат на электронные компоненты

№ элем.	Наименование	Кол. шт.	Цена за штуку	Стоимость
1	Конденс.SMD-C0402-X7R-10n-10%-50V-Walsin	1	2	2,00
2	Конденс.SMD-C0402-X7R-100n-10%-50V-MURATA-GRM155R71H104KE14D	7	2,22	15,54
3	Конденс.SMD-C0603-NP0-39p-5%-50V-SAMSUNG	2	2	4,00
4	Конденс.SMD-C0603-NP0-56p-5%-50V-TDK	1	5	5,00
5	Конденс.SMD-C0603-NP0-1n-5%-50V-MURATA	2	16	32,00
6	Конденс.SMD-C0603-X7R-10n-10%-50V-SAMSUNG	1	0,26	0,26
7	Конденс.SMD-C0603-X7R-100n-10%-50V-SAMSUNG	8	1,38	11,04

Продолжение таблицы 5

№ элем.	Наименование	Кол. шт.	Цена за штуку	Стоимость
8	Конденс.SMD-C0603-X7R-1u0-10%-50V-Walsin	2	7	14,00
9	Конденс.SMD-C0805-X5R-2u2-10%-50V-MURATA	3	3,5	10,50
10	Конденс.SMD-C0805-X7R-1u0-10%-50V-SAMSUNG	1	0,9	0,90
11	Резистор SMD-R0603-0,05W-470R0-1%	1	5	5,00
12	Резистор SMD-R0402-0,062W-5k10-1%	1	0,9	0,90
13	Резистор SMD-R0603-0,100W-5R1-5%	1	0,5	0,50
14	Резистор SMD-R0603-0,100W-100R0-1%	4	0,33	1,32
15	Резистор SMD-R0603-0,100W-100R-5%	8	0,5	4,00
16	Резистор SMD-R0603-0,100W-1k00-1%	2	0,33	0,66
17	Резистор SMD-R0603-0,100W-1k0-5%	5	0,33	1,65
18	Резистор SMD-R0603-0,100W-2k00-1%	1	19,66	19,66
19	Резистор SMD-R0603-0,100W-4k70-1%	1	15,5	15,50
20	Резистор SMD-R0603-0,100W-10k-5%	11	0,25	2,75
21	Резистор SMD-R0603-0,100W-56k0-1%	1	13	13,00
22	Резистор SMD-R0603-0,100W-100k-5%	4	0,38	1,52
23	Резистор SMD-R0603-0,250W-0R10-1%-100ppm	1	0,84	0,84
24	Резистор SMD-R0603-0,250W-390R-5%	3	0,6	1,80
25	Предохранитель-1,1А-MF-MSMF110/16-2-BOURNS	1	9	9,00
26	Предохранитель-3,15А-ВП4-5	1	7	7,00
27	Дрос.SMD-BC54-470M-NL	1	173	173,00

Продолжение таблицы 5

№ элем.	Наименование	Кол. шт.	Цена за штуку	Стоимость
28	Дрос.SMD-BLM18HG601SN1D	1	9	9,00
29	Кварц-SMD0503-12,000MHz-30/30ppm-20pF--10/+60C-FTX12.000M20SM5S	1	7	7,00
30	Электролит SMD-Tantal-100-10V-10%-Case_D-VISHAY	1	20	20,00
31	Электролит SMD-105C-KZ-56-25V-6,3*5,4	1	6	6,00
32	Транзистор-IRFZ24N	1	206	206,00
33	Транзистор-IRLML6402TR-PBF	1	8	8,00
34	Транзистор-PDTC114ET-SOT23	1	5	5,00
35	Кнопка тактовая-KS-00Q-01-E-Switch	3	64	192,00
36	Разъем-BWL-4_W7166-04PSTB00	1	5	5,00
37	Разъем-C3917-40BTI00R	1	42	42,00
38	Разъем-EWL-2_W2620-02PSTW00R	1	1	1,00
39	Разъем-PBD-12LP_C4828-12BDGHYT	1	13	13,00
40	Разъем-Питания-K375A(DJK-02A)	1	76	76,00
41	Диод-BAТ854W-SOT323	1	30	30,00
42	Диод-US1NWF-7-SOD123F-Diodes	1	48	48,00
43	Микросхема-AD5247BKSZ10-2RL7	1	150	150,00
44	Микросхема-AD5247BKSZ50-RL7	1	186	186,00
45	Микросхема-AD820AR-SOIC8	1	210	210,00
46	Микросхема-AD822AR-SOIC8	1	137	137,00
47	Микросхема-ADR390AUJZ	1	70	70,00
48	Микросхема-INA213AIDCKT	1	160	160,00
49	Микросхема-LM22672MRE-ADJ	1	600	600,00
50	Микросхема-LM317LDG-SOIC8	1	10	10,00
51	Микросхема-LMV331IDCKR	1	44	44,00
52	Микросхема-LPC1765FBD100	1	390	390,00
53	Резистор с выводами-2R2-5%-5W-SQP-5	2	6	12,00
54	Стабилитрон-SM15T22A-STMicroelectronics	1	27	27,00
55	Стойка-Пластик-TP-30 (h30mm)	4	16	64,00
56	Контроллер-ST7565S	1	760	760,00
Итого				3841,34

При выполнении НИОКР учитываются затраты на расходные материалы, необходимые для создания модели готового устройства, так как создание конкурентоспособной единицы невозможно без их использования.

Расходные материалы и их стоимость приведены в таблице 6.

Таблица 6 – Смета затрат на расходные материалы

№ расх. матер.	Наименование	Количество, шт.	Цена (за ед. товара), руб	Стоимость, руб
1	Канифоль жидкая 100мл.	2	68,19	136,38
2	Припой SN60PB40 1.0мм 500гр.без флюса	2	1107,01	2214,00
3	Флюс Kester 959T безотмывочный (30мл.)	1	87,4	87,40
4	Изопропанол, 500 мл.	1	150	150,00
5	Паяльная паста с припоем ПОС-70	2	549,14	1098,30
6	Термоскотч толщина 0.28 мм ширина 12 мм длина 33 м t=800С	1	333,8	333,80
7	Набор инструментов для пайки ZD-920A	1	447	447,00
Итого				4466,90

Сложив приведенные выше расчеты, получаем полную стоимость материальных затрат на используемые электронные компоненты и расходные материалы: $3841,34 + 4466,88 = 8308,22$ рублей.

Следующим элементом суммы является амортизация. Применимо к цифровой части стенда она выражается в износе оборудования для производства.

Срок полезного использования персонального компьютера, на котором осуществлялось программирование составляет 5 лет. Стоимость компьютера составляет 15000 рублей. Эксплуатация компьютера длилась 2,5 месяца. Согласно приведенным данным получаем:

$$P_{использ} = \frac{T_{использ} \cdot P}{T_{общ}} = \frac{2,5 \cdot 15000}{60} = 625 \text{ рублей.} \quad (8.15)$$

Ежемесячная амортизация равна:

$$P_{ежемес} = \frac{15000}{60} = 250 \text{ рублей.} \quad (8.16)$$

При расчете амортизации учитывается, что проектирование и разработка стенда осуществляются на территории самого предприятия, поэтому в амортизационные расходы нет необходимости включать стоимость арендной платы помещения. Затраты на оплату труда включают в себя прямую заработную плату рабочим, премии, а также надбавку заработной платы в виде уральского регионального коэффициента. Итоговая заработная плата персонала рассчитывается по формуле:

$$ЗП_{полн.} = (ОК + ЗП_{прем.} + ЗП_{коэф.}) \cdot t, \quad (8.17)$$

где: $ЗП_{полн.}$ – полная заработная плата, руб.;

$ОК$ – оклад, руб./день;

$ЗП_{прем.}$ – премия, руб./день;

$ЗП_{коэф.}$ – региональный уральский коэффициент, руб./день;

t – трудоемкость работ, дней.

Премияльная ставка равна 15% от прямой заработной платы работника. Уральский региональный коэффициент работ также составляет 15%. Исходя из этих данных приводится расчет полной заработной платы в таблице 7.

Таблица 7 – Трудоемкость и заработная плата исполнителей работ

Исполнители	Оклад, руб./мес.	Премия, руб.	Трудоемкость, дни.	Уральский региональный коэффициент(15%),руб.	Полная заработная плата, руб
Руководитель	35000	5250	12	5250	22752
Инженер	25000	3750	13	3750	17589
Лаборант	12000	1800	20	1800	13000
Итого					53341

К прочим затратам относятся незначительные в рамках разработки проекта расходы, возникающие на определенных этапах реализации продукции. К ним относятся: доставка заказанных микросхем, расходы на документацию и отчетность.

Расходы на печатные материалы и отчетность: исходя из стоимости печати одного чертежа формата А1 20 руб. за лист, и одного листа формата А4 2 руб. за лист, получаем:

$$P_{расх. мат} = 6 \cdot 24 + 100 \cdot 2 = 340 \text{ рублей.} \quad (8.18)$$

Расходы на услуги доставки: доставка осуществляется наложенным платежом, поэтому учитывается помимо ее основной стоимости входит оплата услуг фирмы доставки. В сумме данные расходы составляют 120 рублей.

К прочим затратам также относится энергопотребление ПК. В процессе программирования необходимо обеспечение питания компьютера от сети 220 Вольт. При расчете использовались данные по городу Челябинску, стоимость электроэнергии составляет 1,6 руб/кВт*час. Системный блок ПК потребляет 0,4 киловатта. Соответственно, затраты на электроэнергию:

$$P_{эл.} = T \cdot 1,6 \cdot P_{потреб} = 400 \cdot 1,6 \cdot 0,4 = 256 \text{ рублей,} \quad (8.19)$$

где: $P_{эл}$ - затраты на электроэнергию,

T -время работы ПК в часах,

$P_{потреб}$ - потребление ПК в кВт.

Кроме использования в стенде цифровой проектируемой части стенда, в нем также был создан корпус. Расчеты затрат на разработку корпуса и внешнего вида устройства согласно данным предприятия составляют примерно 2 тыс.руб.

Итого, полная себестоимость конкурентоспособной разработки составляет:

$$C=8308,22+625+53341+300+250+120+256+2000=65200,22 \text{ рубля.}$$

8.5 Экономический эффект прибора

Разработанное устройство предназначено для использования внутри компании, а не для продажи. Использование данного прибора позволит уменьшить общее время проверки продукции, выпускаемой ООО «ПЛАНАР».

Также это позволит увеличить количество выпускаемых приборов и сократить количество людей на этапе проверки оборудования.

Срок окупаемости эмулятора составит один месяц, так как по данным компании, количество проданных устройств за год составляет более двух тысяч. Если учесть, что за один месяц компания продает хотя бы два измерителя ИТ-100, то доход в один месяц составит 341200 рублей, что в пять раз больше стоимости эмулятора.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной дипломной работы проанализированы результаты и подведены итоги исследований в области определения характеристик зарядных устройств в радиоизмерительных системах, выпускаемых компанией ООО «ПЛАНАР».

В настоящее время разработанное устройство имеет возможность по модернизации своего функционала путем добавления характеристик других видов аккумуляторов, при этом не требуется вносить изменений в схемотехническую часть стенда.

Планируется дальнейшее развитие разработки, усовершенствование и оптимизация алгоритмов программной части управляющего контроллера.

Рассчитанные экономические показатели помогут в дальнейшем определить стоимостную оценку всех работ для создания конкурентоспособной продукции, а описанные в разделе безопасности жизнедеятельности правила пользования прибором оградить человека от опасных и вредных производственных факторов, возникающих в процессе эксплуатации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1) Электронный ресурс зарядка литий-ионных аккумуляторов:
<http://www.gaw.ru/html.cgi/txt/app/micros/avr/AVR146.htm> (Internet)
- 2) П.Хоровиц, У. Хилл. - Искусство схемотехники: в 2-х т. т.1 – М.:Изд-во: «Мир» 1986. – 598 с.: ил.
- 3) Клеменс Б. – Язык С в XXI веке (теория и принципы построения) - М.:ДМК Пресс 2015. – 376 с.: ил.
- 4) Прата С. - Язык программирования С. Лекции и упражнения – М.:Изд-во «Вильямс», 2013. – 960 с.: ил.
- 5) Макконнелл С. – Совершенный код. Мастер класс - М.:Изд-во: «Русская редакция», 2010. – 896 с.: ил.
- 6) Сабунин А. - Altium Designer. Новые решения в проектировании электронных устройств – М.:Изд-во «Солон-пресс», 2009. – 432 с.: ил.
- 7) Мактас М.Я. - Проектирование печатных плат в САПР Altium Designer – М.Изд-во «УлГТУ», 2014. – 139 с.: ил.
- 8) Абчук В.А. - Менеджмент. Учебник – М.: Изд-во «Союз», 2002. – 463 с.
- 9) Мешковой Н.П., Закиров Р.Ш., Зинкевич В.С., Попов С.Г. - Учебное пособие для студентов приборостроительного факультета – Челябинск: ЧГТУ, 1990. – 53 с.
- 10) Скворцов Ю.В. - Организационно-экономические вопросы в дипломном проектировании –М.: Изд-во «Студент», 2012. – 374 с.: ил.
- 11) Зотов Б.И - Безопасность жизнедеятельности на производстве: учебник. - 2-е изд., перераб. и доп. – Изд-во: «КолосС», 2004. – 432 с.: ил.
- 12) Бекасова В.Н., Боровик С.И., Глотова Н.В., Давлятшин В.Г., В.Г. Зеленкин, Л.М. Киселева, Окраинская И.С. – Безопасность жизнедеятельности в дипломных проектах – Изд-во «ЮУрГУ», 2007. – 166 с.: ил.
- 13) СТО ЮУрГУ 04 – 2008: Курсовое и дипломное проектирование: Общие требования к оформлению – Издательство «ЮУрГУ», 2008. – 56 с.

- 14) Романычева Э.Т., Иванова А.К., Куликов А.С., Миронова Н.Г. - Разработка и оформление конструкторской документации РЭА – Изд-во «Радио и связь», 1989. – 448 с.: ил.
- 15) Техническая документация стенда ТСВ-01
- 16) ГОСТ 2.001-93 ЕСКД. «Общие положения» Москва: ИПК Изд-во стандартов
- 17) ГОСТ 2.301-68 ЕСКД. «Форматы» Москва: ИПК Изд-во стандартов
- 18) ГОСТ 2.302-68 ЕСКД. Масштабы. Москва: ИПК Изд-во стандартов
- 19) ГОСТ 2.303-68 ЕСКД. Линии. Москва: ИПК Изд-во стандартов
- 20) ГОСТ 2.304-81 ЕСКД. Шрифты чертёжные. Москва: ИПК Изд-во стандартов
- 21) ГОСТ 2.701-2008 ЕСКД. Схемы. Виды и типы. Общие требования к выполнению. Москва: ИПК Изд-во стандартов
- 22) ГОСТ 2.702-2011 ЕСКД. Правила выполнения электрических схем. Москва: ИПК Изд-во стандартов
- 23) ГОСТ 12.2.007.0-75 ССБТ. Изделия электротехнические. Общие требования безопасности. Москва: ИПК Изд-во стандартов

ПРИЛОЖЕНИЕ А

Исходный код программы

Файл «main.c»

```
#include "./board/board.h"
#include "./fw/fw.h"
#include "./fw/timeservice.h"
#include "./stend.h"
#include "./lib/glcd/glcd.h"
#include "./drv/acc_meas.h"
#include "./drv/term_emul.h"
#include "./drv/charge_emul.h"
#include <stdio.h>
#define MEMPOOL_ITEMS_COUNT 64
#define MEMPOOL_ITEM_SIZE_DWORDS 4
__task int main( void )
{
    printf("debug\n");
    /* BSP */
    Board_Init();
    /* initialize memory pools... */
    static struct{
        uint32_t data[MEMPOOL_ITEM_SIZE_DWORDS];
    } mempool_storage[MEMPOOL_ITEMS_COUNT];
    fw_mempool_init( mempool_storage, sizeof( mempool_storage ), sizeof( mempool_storage[0] ) );
    /* initialize the framework */
    fw_init();
    /* initialize time service... */
    timeservice_init();
    /* init publish-subscribe... */
    fw_ps_init( NULL, 0 );
    /* explicitly invoke the active objects' ctors... */
    stend_ao_ctor( &accEmulStendAO, NULL );
    /* start the active objects... */
    static const tEvent* ao_ev_queue_storage[EVENTS_QUEUE_LENGTH +
(sizeof(RINGBUF_T)+3)/4];
```

```

fw_ao_start( (void*)&accEmulStendAO, /* AO descriptor */
            1u, /* AO priority(also used in FW as AO unique ID) */
            ao_ev_queue_storage, EVENTS_QUEUE_LENGTH, /* evt queue */
            (void*)0, 0u, /* no per-thread stack */
            (tEvent*)0 ); /* no initialization event */

/* run the QF application */
fw_run();
while(1);
}

```

Файл «stend.c»

```

#include "stend.h"
#include <assert.h>
#include "../macro.h"
#include "../signals.h"
#include "../lib/glcd/glcd.h"
#include "../drv/charge_emul.h"
#include "../drv/term_emul.h"
#define DEBUGID "[stend.c]: "
#define DBGOUT( level, ... ) //TSANDBG_PRINT( level, DEBUGID __VA_ARGS__ )

static void stend_fsm_l_ctor(tStendFsmL* mem);
static void stend_fsm_l_init(hcState this, hSm sm, hcState* next);
static void stend_fsm_l_entry(hcState this, hSm sm);
static hcState stend_fsm_l_dispatch(hcState this, hSm sm, const struct tEvent* ev, hcState* next);
static const tEvent* tmr_get_event(void* object, struct tTimer* timer);
static void stend_test_handle_event(tStend* stend ,const struct tEvent* ev);
tStendAO accEmulStendAO;
static const tTimerEventGetter tmr_ev_getter = { .object = NULL, .get_event = tmr_get_event };
static const tStendEv temp_test_done = {
    .super = { .sig = STEND_UPDATED },
    .id = stendEVENT_TEMP_TEST_DONE
};
static const tStendEv charging_test_done = {
    .super = { .sig = STEND_UPDATED },
    .id = stendEVENT_CHARGING_TEST_DONE
}

```

```

};
DEFINE_FSML_VTBL( stend_fsm_l_vtbl,
    .init = stend_fsm_l_init,
    .entry = stend_fsm_l_entry,
    .dispatch = stend_fsm_l_dispatch );
void stend_ao_ctor(tStendAO* this, void* mainmenu) {
    assert(this);
    (void)mainmenu;
    stend_fsm_l_ctor( &this->sm );
    //Construct Active Object
    fw_ao_ctor( (tAO*)this, (hSm)&this->sm );
    //Subscribe to all signals
    fw_ps_subscribe_all( (tAO*)this, MAX_SIG );
    term_emul_init();
    charge_emul_init();
}
void kb_handler_publish_event( const tKbEventData* data ){
    static tKbEvent kb_event = { .super = { .sig = KB_EVENT_SIG } };
    kb_event.data = *data;
    fw_ps_publish_sig((const tEvent*)&kb_event);
}
void stend_test_start(tStend* stend) {
    stend->flag = 1;
    charging_test_start(&stend->charge);
}
static void stend_fsm_l_ctor(tStendFsmL* handler) {
    handler->super.vtbl = &stend_fsm_l_vtbl;
    //View
    view_ctor(&handler->view, &handler->stend);
    //Timer
    timer_ctor(&handler->timer,
        &(const tTimerCfg){.period
TIMESERVICE_MS_TO_TICKS(KB_HANDLER_CALL_PERIOD_MS),
        .ev_getter = &tmr_ev_getter});
    charging_test_ctor(&handler->stend.charge);
}

```

```

        temp_test_ctor(&handler->stend.temp_test);
    }
static void stend_fsm_l_entry( hcState this, hSm sm ) {
    (void)sm;
    tStendFsmL* fsm_l = (tStendFsmL*)this;
    sm_init((hSm)&fsm_l->view);
    timer_start( &fsm_l->timer, NULL );
}
static void stend_fsm_l_init( hcState this, hSm sm, hcState* next ) {
    (void)sm;
    (void)next;
    tStendFsmL* stend = (tStendFsmL*)this;
    stend->super.successor = NULL;
}
static hcState stend_fsm_l_dispatch( hcState this, hSm sm, const struct tEvent* ev, hcState* next ) {
    (void)sm;
    tStendFsmL* fsm_l = (tStendFsmL*)this;
    tStend* stend = &fsm_l->stend;
    switch ( ev->sig ) {
        case VIEW_TIMER_SIG:
            sm_dispatch((hSm)&fsm_l->view, ev);
            break;
        case KB_EVENT_SIG:
            sm_dispatch((hSm)&fsm_l->view, ev);
            break;
        case KB_TIMER_SIG:
            kb_handler();
            break;
        case VIEW_MODEL_UPDATED_SIG:
            sm_dispatch((hSm)&fsm_l->view, ev);
            break;
        case CHARGING_TEST_SIG:
        case TEMP_TEST_SIG:
            stend_test_handle_event(stend, ev);
            break;
    }
}

```

```

    }
    return NULL;
}
/* Timers event getters */
static const tEvent* tmr_get_event(void* object, struct tTimer* timer) {
    (void)object;
    (void)timer;
    static const tEvent event = {.sig = KB_TIMER_SIG};
    return &event;
}
static void stend_test_handle_event(tStend* stend ,const struct tEvent* ev) {
    if (ev->sig == CHARGING_TEST_SIG) {
        const tChargingTestEvent* ct_event = (const tChargingTestEvent*) ev;
        if (ct_event->id == charging_testCHARGED_COMPLETE) {
            fw_ps_publish_sig((tEvent*)&charging_test_done);
            temp_test_start(&stend->temp_test);
        } else {
            charging_test_handle_event(&stend->charge, ev);
        }
    } else if (ev->sig == TEMP_TEST_SIG) {
        const tChargingTempEvent* temp_event = (const tChargingTempEvent*) ev;
        if (temp_event->id == charging_tempCHARGED_COMPLETE)
            fw_ps_publish_sig((tEvent*)&temp_test_done);
        else
            temp_test_handle_event(&stend->temp_test, ev);
    }
}

```

Файл «acc_meas.c»

```

#include "charging_test.h"
#include "temp_test.h"
#include "acc_meas.h"
#include "board.h"
#include <stdbool.h>
#include <math.h>
#include "./signals.h"

```

```

#include "../fw/fw.h"
#define LAST_MEAS 8
typedef struct tVaccMeas {
    int v_ref_mv;
} tVaccMeas;
typedef struct tData {
    uint16_t i_acc_code;
    uint16_t v_acc_code;
    uint16_t v_acc_ref_code;
    uint16_t i_leak_acc_code;
    int v_acc_ref_mv;
    Status i_acc_status;
    Status v_acc_status;
    Status v_acc_ref_status;
    Status i_leak_acc_status;
} tData;
typedef struct tAccMeas {
    tVaccMeas vaccMeas;
    tData acc_meas_data;
    tAccMeasParam param_meas;
    const tEvent* insignal;
    bool async_mode;
    int result_ready;
} tAccMeas;
static inline int convert_adc_code_to_v(int code);
static inline int convert_adc_code_to_i(int code);
static inline int convert_adc_code_to_i_leak(int code);
static inline int calc_adc_v_ref_from_precise_v_ref(int code);
static inline void on_result_done();
static tAccMeas meas;
void ADC_IRQHandler(void) {
    static int counter = 0;
    static int done_flag = 0;
    uint32_t dr_chan0 = LPC_ADC->DR[ADC_CH0];
    uint32_t dr_chan1 = LPC_ADC->DR[ADC_CH1];

```

```

uint32_t dr_chan2 = LPC_ADC->DR[ADC_CH2];
uint32_t dr_chan3 = LPC_ADC->DR[ADC_CH3];
if (ADC_DR_DONE(dr_chan0) == 1) {
    meas.acc_meas_data.i_acc_code += ADC_DR_RESULT(dr_chan0);
done_flag |= 1 << ADC_CH0;
}
if (ADC_DR_DONE(dr_chan1) == 1) {
    meas.acc_meas_data.v_acc_code += ADC_DR_RESULT(dr_chan1);
    done_flag |= 1 << ADC_CH1;
}
if (ADC_DR_DONE(dr_chan2) == 1) {
    meas.acc_meas_data.i_leak_acc_code += ADC_DR_RESULT(dr_chan2);
    done_flag |= 1 << ADC_CH2;
}
if (ADC_DR_DONE(dr_chan3) == 1) {
    meas.acc_meas_data.v_acc_ref_code += ADC_DR_RESULT(dr_chan3);
    done_flag |= 1 << ADC_CH3;
}
if (done_flag == ((1 << ADC_CH0) | (1 << ADC_CH1) | (1 << ADC_CH2) | (1 << ADC_CH3)))
{
    done_flag = 0;
    counter++;
}
if (counter == LAST_MEAS) {
    LPC_ADC->CR &= ~ADC_CR_BURST;
    LPC_ADC->INTEN &= ~((1UL << ADC_CH0) |
        (1UL << ADC_CH1) |
        (1UL << ADC_CH2) |
        (1UL << ADC_CH3));
    meas.acc_meas_data.i_acc_code /= 8;
    meas.acc_meas_data.v_acc_code /= 8;
    meas.acc_meas_data.i_leak_acc_code /= 8;
    meas.acc_meas_data.v_acc_ref_code /= 8;
    counter = 0;
    if (meas.async_mode == true) {

```

```

        meas.acc_meas_data.v_acc_ref_mv =
calc_adc_v_ref_from_precise_v_ref(meas.acc_meas_data.v_acc_ref_code);
        meas.param_meas.v_acc =
convert_adc_code_to_v(meas.acc_meas_data.v_acc_code);
        meas.param_meas.i_acc =
convert_adc_code_to_i(meas.acc_meas_data.i_acc_code);
        meas.param_meas.i_leak_acc =
convert_adc_code_to_i_leak(meas.acc_meas_data.i_leak_acc_code);
        on_result_done();
    } else {meas.result_ready = 1;}
}
}

void acc_meas_return(tAccMeasParam* param) {
    param->i_acc = meas.param_meas.i_acc;
    param->v_acc = meas.param_meas.v_acc;
    param->i_leak_acc = meas.param_meas.i_leak_acc;
}

void acc_meas_async(const void* event) {
    meas.async_mode = true;
    meas.insignal = event;
    LPC_ADC->INTEN |= ((1UL << ADC_CH0) | (1UL << ADC_CH1) | (1UL << ADC_CH2) |
(1UL << ADC_CH3));
    Chip_ADC_SetBurstCmd(LPC_ADC, ENABLE);
}

void acc_meas_sync(tAccMeasParam* param) {
    meas.async_mode = false;
    LPC_ADC->INTEN |= ((1UL << ADC_CH0) | (1UL << ADC_CH1) | (1UL << ADC_CH2) |
(1UL << ADC_CH3));
    Chip_ADC_SetBurstCmd(LPC_ADC, ENABLE);
    // Waiting
    while (!meas.result_ready) continue;
    // Converting result
    meas.acc_meas_data.v_acc_ref_mv =
calc_adc_v_ref_from_precise_v_ref(meas.acc_meas_data.v_acc_ref_code);
    param->v_acc = convert_adc_code_to_v(meas.acc_meas_data.v_acc_code);

```



```

    param->i_acc = convert_adc_code_to_i(meas.acc_meas_data.i_acc_code);
    param->i_leak_acc = convert_adc_code_to_i_leak(meas.acc_meas_data.i_leak_acc_code);
}
#define ADC_FULLSCALE_CODE 4096
#define RDIV_COEFF_DIVIDER 1000
static inline int calc_adc_v_ref_from_precise_v_ref(int code) {
    meas.vaccMeas.v_ref_mv = (2048 * ADC_FULLSCALE_CODE / code);
    return meas.vaccMeas.v_ref_mv;
}
static inline int convert_adc_code_to_v(int code) {
    return (meas.vaccMeas.v_ref_mv * code / ADC_FULLSCALE_CODE);
}
static inline int convert_adc_code_to_i(int code) {
    return convert_adc_code_to_v(code)*10 / 11;
}
static inline int convert_adc_code_to_i_leak(int code) {
    #define GAIN 50
    #define R_SHUNT 5.1f
    int V_OUT = convert_adc_code_to_v(code);
    return (V_OUT - meas.vaccMeas.v_ref_mv) / (GAIN * (int)R_SHUNT);
}
static inline void on_result_done() {
    fw_ps_publish_sig(meas.insignal);
}

```

Файл «charging_test»

```

#include "charging_test.h"
#define STEP_RESIST 1
static inline const tTimerEventGetter* get_ev_getter(tTimer* timer);
static const tEvent* get_event(void* object, struct tTimer* timer);
static inline void charging_meas_to_resist(tChargingTest* charge);
static inline void charging_test_init(tChargingTest* charge);
static const tChargingTestEvent charg_time_out = {
    .super = { .sig = CHARGING_TEST_SIG },
    .id = charging_testEVENT_MODE_TIME_OUT
};

```

```

static const tChargingTestEvent charge_done = {
    .super = { .sig = CHARGING_TEST_SIG },
    .id = charging_testCHARGED_COMPLETE
};

static const tChargingTestEvent result_charge_done = {
    .super = { .sig = CHARGING_TEST_SIG },
    .id = acc_measRESULT_CHARGE_COMPLETE
};

void charging_test_handle_event(tChargingTest* charge, const tEvent* event) {
    const tChargingTestEvent* ct_event = (const tChargingTestEvent*) event;
    switch(ct_event->id) {
        case charging_testEVENT_MODE_START:
            charging_test_init(charge);
            charge_emul_set_resist(charge->engine_posit);
            timer_start(&charge->timer, NULL);
            break;
        case charging_testEVENT_MODE_TIME_OUT:
            acc_meas_async((tEvent*)&result_charge_done);
            break;
        case acc_measRESULT_CHARGE_COMPLETE:
            charging_meas_to_resist(charge);
            break;
    }
}

void charging_test_ctor(tChargingTest* charge) {
    timer_ctor(&charge->timer,          &(const          tTimerCfg){.period          =
TIMESERVICE_MS_TO_TICKS(1000),
            .ev_getter = get_ev_getter(&charge->timer),
            .f_oneShot = 1 });
}

void charging_test_dtor(tChargingTest* charge) {
    memset(charge, 0, sizeof(*charge));
    timer_reset(&charge->timer);
}

```

```

void charging_test_return(tChargingTest* charge) {
    charge->i_charge_back = charge->i_charge[SIZE_CHARGE-1];
    charge->i_leak_charge_back = charge->i_leak_charge[SIZE_CHARGE-1];
    charge->v_charge_back = charge->v_charge[SIZE_CHARGE-1];
}

static inline void charging_test_init(tChargingTest* charge) {
    charge->i = 0;
    memset(charge->i_charge, 0, sizeof(charge->i_charge));
    memset(charge->v_charge, 0, sizeof(charge->v_charge));
    memset(charge->i_leak_charge, 0, sizeof(charge->i_leak_charge));
    charge->engine_posit = 127;
}

static inline void charging_meas_to_resist(tChargingTest* charge) {
    tAccMeasParam result;
    acc_meas_return(&result);
    charge->i_charge[charge->i] = result.i_acc;
    charge->v_charge[charge->i] = result.v_acc;
    charge->i_leak_charge[charge->i] = result.i_leak_acc;
    charge->engine_posit -= STEP_RESIST;
    charge->i++;
    if(charge->i == SIZE_CHARGE) {
        fw_ps_publish_sig((tEvent*)&charge_done);
    } else {
        charge_emul_set_resist(charge->engine_posit);
        timer_start(&charge->timer, NULL);
    }
}

static const tEvent* get_event(void* object, struct tTimer* timer) {
    (void)timer;
    (void)object;
    static const tChargingTestEvent event = {.super.sig = CHARGING_TEST_SIG,.id =
charging_testEVENT_MODE_TIME_OUT};
    return (const tEvent*)&event;
}

```

```

static inline const tTimerEventGetter* get_ev_getter(tTimer* timer) {
    static tTimerEventGetter ev_getter;
    ev_getter.object = timer;
    ev_getter.get_event = get_event;
    return &ev_getter;
}

```

Файл «temp_test»

```

#include "temp_test.h"
#include "charging_test.h"
#define STEP_RESIST      1
#define MAX_TEMP1 50
#define MAX_TEMP2 55
#define MIN_TEMP1 35
#define MIN_TEMP2 40
static inline const tTimerEventGetter* get_ev_getter(tTimer* timer);
static const tEvent* get_event(void* object, struct tTimer* timer);
static inline void temp_test_to_resist(tChargingTemp* temp_test);
static inline void temp_test_init(tChargingTemp* temp_test);
static inline void temp_test_done (tChargingTemp* temp_test);
static const tChargingTempEvent temp_time_out = {
    .super = { .sig = TEMP_TEST_SIG },
    .id = charging_tempEVENT_MODE_TIME_OUT
};
static const tChargingTempEvent charge_done = {
    .super = { .sig = TEMP_TEST_SIG },
    .id = charging_tempCHARGED_COMPLETE
};
static const tChargingTempEvent result_temp_done = {
    .super = { .sig = TEMP_TEST_SIG },
    .id = acc_measRESULT_TEMP_COMPLETE
};
void temp_test_handle_event(tChargingTemp* temp_test, const tEvent* event) {
    const tChargingTempEvent* temp_event = (const tChargingTempEvent*) event;
    switch(temp_event->id) {
        case charging_tempEVENT_MODE_START:

```

```

        temp_test_init(temp_test);
        term_emul_set_resist_by_temp(temp_test->temp);
        timer_start(&temp_test->timer, NULL);
        break;
    case charging_tempEVENT_MODE_TIME_OUT:
        acc_meas_async((tEvent*)&result_temp_done);
        break;
    case acc_measRESULT_TEMP_COMPLETE:
        temp_test_to_resist(temp_test);
        break;
    }
}

void temp_test_ctor(tChargingTemp* temp_test) {
    timer_ctor(&temp_test->timer,          &(const          tTimerCfg){.period          =
TIMESERVICE_MS_TO_TICKS(1000),
                .ev_getter = get_ev_getter(&temp_test->timer),
                .f_one-shot = 1 });
}

void temp_test_dtor(tChargingTemp* temp_test) {
    memset(temp_test, 0, sizeof(*temp_test));
    timer_reset(&temp_test->timer);
}

void temp_test_return(tChargingTemp* temp_test) {
    temp_test->max_temp;
    temp_test->min_temp;
    temp_test->max_temp_ok;
    temp_test->min_temp_ok;
}

static inline void temp_test_init(tChargingTemp* temp_test) {
    temp_test->max_temp = 0;
    temp_test->min_temp = 0;
    temp_test->temp = 22;
    temp_test->direction = true;
}

```

```

static inline void temp_test_to_resist(tChargingTemp* temp_test) {
    tAccMeasParam result;
    acc_meas_return(&result);
    if (temp_test->direction == true) {
        if (result.v_acc > 0) {
            temp_test_done(temp_test);
            temp_test->temp++;
            term_emul_set_resist_by_temp(temp_test->temp);
            timer_start(&temp_test->timer, NULL);
            if (temp_test->temp == 95) {
                fw_ps_publish_sig((tEvent*)&charge_done);
            }
        } else if (result.v_acc == 0) {
            temp_test->direction = false;
            temp_test_done(temp_test);
            term_emul_set_resist_by_temp(temp_test->temp);
            timer_start(&temp_test->timer, NULL);
        }
    } else if (temp_test->direction == false) {
        if (result.v_acc == 0) {
            temp_test->temp--;
            if (temp_test->temp == 22){
                fw_ps_publish_sig((tEvent*)&charge_done);
            } else {temp_test_done(temp_test);
                term_emul_set_resist_by_temp(temp_test->temp);
                timer_start(&temp_test->timer, NULL);
            }
        } else {temp_test_done(temp_test);
            fw_ps_publish_sig((tEvent*)&charge_done);
        }
    }
}

static inline void temp_test_done(tChargingTemp* temp_test) {
    if (temp_test->max_temp > MAX_TEMP2) {
        temp_test->max_temp_ok = false;
    }
}

```

```

    } else if (temp_test->max_temp >= MAX_TEMP1 && temp_test->max_temp <= MAX_TEMP2)
    {
        temp_test->max_temp = temp_test->temp;
        temp_test->max_temp_ok = true;
    }
    if(temp_test->min_temp < MIN_TEMP1) {
        temp_test->min_temp_ok = false;
    } else if (temp_test->min_temp >= MIN_TEMP1 && temp_test->min_temp <= MIN_TEMP2) {
        temp_test->min_temp = temp_test->temp;
        temp_test->min_temp_ok = true;
    }
}

static const tEvent* get_event(void* object, struct tTimer* timer) {
    (void)timer;
    (void)object;
    static const tChargingTempEvent event = {.super.sig = TEMP_TEST_SIG,.id =
charging_tempEVENT_MODE_TIME_OUT};
    return (const tEvent*)&event;
}

static inline const tTimerEventGetter* get_ev_getter(tTimer* timer) {
    static tTimerEventGetter ev_getter;
    ev_getter.object = timer;
    ev_getter.get_event = get_event;
    return &ev_getter;
}

```

Файл «board.c»

```

#include "board.h"
#include "string.h"
#include "retarget.h"
#include "kb.h"

static inline void Board_Kb_Init(void);
static inline void Board_SPI_Init(void);
static inline void Board_Glcd_Init(void);
static inline void Board_I2C_Init(void);
static inline void Board_UART_Init( void );

```

```

static inline void Board_ADC_Init( void );
/* System oscillator rate and RTC oscillator rate */
const uint32_t OscRateIn = 12000000;
const uint32_t RTCOscRateIn = 32768;
/* Initialize debug output via UART for board */
void Board_Debug_Init(void) {
#if defined(DEBUG_ENABLE)
    Chip_UART_Init(DEBUG_UART);
    Chip_UART_SetBaud(DEBUG_UART, 115200);
    Chip_UART_ConfigData(DEBUG_UART, UART_LCR_WLEN8 | UART_LCR_SBS_1BIT |
UART_LCR_PARITY_DIS);
    /* Enable UART Transmit */
    Chip_UART_TXEnable(DEBUG_UART);
#endif
}
/* Sends a character on the UART */
void Board_UARTPutChar(char ch) {
#if defined(DEBUG_ENABLE)
    while ((Chip_UART_ReadLineStatus(DEBUG_UART) & UART_LSR_THRE) == 0) {}
    Chip_UART_SendByte(DEBUG_UART, (uint8_t) ch);
#endif
}
/* Gets a character from the UART, returns EOF if no character is ready */
int Board_UARTGetChar(void) {
#if defined(DEBUG_ENABLE)
    if (Chip_UART_ReadLineStatus(DEBUG_UART) & UART_LSR_RDR) {
        return (int) Chip_UART_ReadByte(DEBUG_UART);
    }
#endif
    return EOF;
}
/* Outputs a string on the debug UART */
void Board_UARTPutSTR(char *str) {
#if defined(DEBUG_ENABLE)
    while (*str != '\0') {

```



```

        Board_UARTPutChar(*str++);
    }
#endif
}

void Board_Init(void) {
    /* Sets up DEBUG UART */
    DEBUGINIT();
    /* Initializes GPIO */
    Chip_GPIO_Init(LPC_GPIO);
    Chip_IOCON_Init(LPC_IOCON);
    /* Peripheral setup */
    Board_Kb_Init();
    Board_SPI_Init();
    Board_Glcd_Init();
    Board_I2C_Init();
    Board_UART_Init();
    Board_ADC_Init();
    /* Debug */
    Board_Debug_Init();
}

/* Initialize KB control lines */
static inline void Board_Kb_Init(void) {
    Chip_GPIO_SetPinDIRInput(LPC_GPIO, KB_GPIO_PORT, KB_SW0_GPIO_PIN);
    Chip_GPIO_SetPinDIRInput(LPC_GPIO, KB_GPIO_PORT, KB_SW1_GPIO_PIN);
    Chip_GPIO_SetPinDIRInput(LPC_GPIO, KB_GPIO_PORT, KB_SW2_GPIO_PIN);
}

/* Initialize SPI */
static inline void Board_SPI_Init(void) {
    Chip_GPIO_SetPinDIROutput(LPC_GPIO, SPI_GPIO_PORT, SPI_SSEL_GPIO_PIN);
    Board_SPI_DeassertSSEL();
    /* Set up SSP0 interface */
    Chip_Clock_EnablePeriphClock(SYSCTL_CLOCK_SSP0);
    Chip_SSP_Set_Mode(LPC_SSP0, SSP_MODE_MASTER);
    Chip_SSP_SetFormat(LPC_SSP0,          SSP_BITS_8,          SSP_FRAMEFORMAT_SPI,
SSP_CLOCK_CPHA0_CPOL0);
}

```

```

    Chip_SSP_SetBitRate(LPC_SSP0, 4000000);
    Chip_SSP_Enable(LPC_SSP0);
}
/* Initialize GLCD control lines */
static inline void Board_Glcd_Init(void) {
    Chip_GPIO_SetPinDIROutput(LPC_GPIO, GLCD_GPIO_PORT, GLCD_RS_GPIO_PIN);
    Chip_GPIO_SetPinDIROutput(LPC_GPIO, GLCD_GPIO_PORT, GLCD_A0_GPIO_PIN);
}
/* Initialize I2C */
static inline void Board_I2C_Init(void) {
    Chip_IOCON_SetI2CPad(LPC_IOCON, I2CPADCFG_STD_MODE);
    Chip_I2C_Init( I2C0 );
    Chip_I2C_SetClockRate( I2C0, 400000 );
    NVIC_EnableIRQ(I2C_IRQn);
}
/* Initialize UART */
static inline void Board_UART_Init( void ) {
}

/* Initialize ADC */
static inline void Board_ADC_Init( void ) {
    ADC_CLOCK_SETUP_T adc_clock_setup = { .adcRate = 100000,
                                           .bitsAccuracy = 12,
                                           .burstMode = false };
    Chip_ADC_Init( LPC_ADC, &adc_clock_setup );
    NVIC_EnableIRQ(ADC_IRQn);
    LPC_ADC->CR |= ((1UL << ADC_CH0) | (1UL << ADC_CH1) | (1UL << ADC_CH2) | (1UL
<< ADC_CH3));
}

```

Файл «glcd.c»

```

#include "glcd.h"
#include <stdio.h>
#include <string.h>
#include <stdint.h>
#include "board.h"

```

```

#include "fonts\font4x5.h"
#include "fonts\font5x8.h"
#include "fonts\font12x16.h"

// absolute address of LCD Controller #0 CTRL and DATA registers
#define CMD_ADDR0
#define DATA_ADDR      1
#define GLCD_SET_PAGE          0xB0 // 1011XXXX: set lcd GLCD_PAGES_COUNT (Y)
address 0...8
#define GLCD_SET_X_ADDR_H      0x10 // 0001XXXX: set lcd X address-старшие биты
#define GLCD_SET_X_ADDR_L      0x00 // 0000XXXX: set lcd X address-младшие биты
// ST7565 command set
#define GLCD_ON_DISPLAY        0xAE // 1010111X: 1 - turn display on
#define GLCD_BIAS_SET          0xA2 // 1010001X: 0:1/9bias,1:1/7bias
#define GLCD_ADC_SELECT        0xA0 // 1010000X: 0:normal,1:revers
#define GLCD_COM_SELECT        0xC0 // 1100X000: select output scan direction
0:normal,1:revers
#define GLCD_RES_RATIO         0x20 // 00100XXX: internal resistor ratio Rb/Ra
#define GLCD_SET_V5            0x81 // 10000001: Set V5 output voltage, volume -
2 byte
#define GLCD_POWER_SET         0x28 // 00101XXX: Select internal power supply
#define GLCD_BOOST_SET         0xF8 // 11111000: Select booster ratio, volume - 2
byte (000000XX)
#define GLCD_SET_DISPLAY       0xA6 // 1010011X: Set LCD 0:normal,1:revers
#define GLCD_SET_POINTS        0xA4 // 1010010X: Set points 0:normal,1:all points ON
#define GLCD_INT_RESET         0xE2 // 11100010: Internal reset
#define GLCD_SET_MODE_INVERS   0xA7 // 10100111: Set LCD display mode to revers
#define GLCD_SET_MODE_NORMAL   0xA6 // 10100110: Set LCD display mode to normal
/* флаги функции MenuShowScrolling */
#define MENU_SCROLL_UP         (1ul << 0)
#define MENU_SCROLL_DOWN      (1ul << 1)
#define MENU_SCROLL_RET        (1ul << 2)
#define GLCD_PAGES_COUNT      8
#define GLCD_COLUMNS_COUNT    128
static inline void reset_delay(void);

```

```

static inline void write( uint8_t byte );
static inline void write_cmd( uint8_t cmd );
static inline void write_data( uint8_t data );
static inline void set_address( uint8_t column, uint8_t page );
static inline void buffer_block_set(uint8_t x,uint8_t y,uint8_t data, uint8_t n);
static inline void buff_draw_vline(uint8_t x,uint8_t y,uint8_t h, tLcdColor color);
static inline void buff_draw_hline(uint8_t x,uint8_t y, uint8_t w, tLcdColor color);
static inline void send_region(uint8_t x,uint8_t y,uint8_t w,uint8_t h);
static inline void buff_draw_vline(uint8_t x,uint8_t y,uint8_t h, tLcdColor color);
static inline unsigned char symbol_decode(unsigned char c);
static inline void draw_text(uint8_t x, uint8_t y, const char* s, tLcdColor color);
static inline void draw_symb(uint8_t x, uint8_t y, unsigned char c, tLcdColor color);
static inline void draw_rectangle(uint8_t x, uint8_t y, uint8_t w, uint8_t h, tLcdColor color);
static inline void draw_frame(uint8_t x, uint8_t y, uint8_t w, uint8_t h, tLcdColor color);
static inline uint8_t align(uint8_t field_l, uint8_t object_l, tLcdAlign align);
uint8_t bufferGLCD[GLCD_PAGES_COUNT][GLCD_COLUMNS_COUNT];
void glcd_init(void) {
    Board_Glcd_Rst_Clr();
    Board_Glcd_Rst_Set();
    reset_delay();
    write_cmd(GLCD_BIAS_SET | 0x00); // 1/9 bias
    write_cmd(GLCD_ADC_SELECT | 0x00); // normal (column adr inc: 0 - 83h)
    write_cmd(GLCD_COM_SELECT | 0x08); // revers (row adr inc: 63 -> 0)
    write_cmd(GLCD_RES_RATIO | 0x03); //
    //contrast
    write_cmd(GLCD_SET_V5);
    write_cmd(50);
    //mode
    write_cmd(GLCD_SET_MODE_NORMAL);
    write_cmd(GLCD_BOOST_SET);
    write_cmd(0);
    write_cmd(GLCD_POWER_SET | 0x07);

    write_cmd(GLCD_ON_DISPLAY | 0x01); // display on
    glcd_clear_screen( cBLACK );
}

```

```

}
void glcd_clear_screen(tLcdColor color) {
    for (uint8_t page=0; page<GLCD_PAGES_COUNT ;page++)
    {
        for (uint8_t column=0; column<GLCD_COLUMNS_COUNT; column++)
        {
            bufferGLCD[page][column]= (color==cWHITE ? 0xff : 0x00) ;
        }
    }
    send_region(0,0, GLCD_XPIXELS, GLCD_YPIXELS);
}
void glcd_draw_rectangle(uint8_t x, uint8_t y, uint8_t w, uint8_t h, tLcdColor color) {
    uint8_t i;
    for(i = x; i < x+w; i++)
    {
        buff_draw_vline(i,y,h,color);
    }
    send_region(x, y, w, h);
}
void glcd_draw_frame(uint8_t x, uint8_t y, uint8_t w, uint8_t h, tLcdColor color)
{
    buff_draw_hline(x, y, w, color);
    buff_draw_hline(x, y+h-1, w, color);
    buff_draw_vline(x, y, h,color);
    buff_draw_vline(x+w-1, y, h,color);
    send_region(x, y,w ,h );
}
void glcd_draw_symb(uint8_t x, uint8_t y, unsigned char c, tLcdColor color)
{
    uint8_t data, i;
    c = symbol_decode(c);
    for (i = 0; i <=4 ; i++)
    {
        if (color == cWHITE)
        {

```

```

        data = font5x8[c][i];
    }
    else { data = ~( font5x8[c][i]);}
    buffer_block_set(x+i,y,data,8);
}
send_region(x, y, 5, 7);
}
void glcd_draw_vline(uint8_t x,uint8_t y,uint8_t h, tLcdColor color)
{
    uint8_t data, n, h_left, y_curr;
    data = (color == cWHITE ? 0xff : 0x00);
    n = h % 8;
    buffer_block_set(x,y,data,n);
    h_left = h-n;
    y_curr = y+n;
    for (; h_left > 0; h_left -= 8, y_curr += 8)
    {
        buffer_block_set(x,y_curr,data,8);
    }
    send_region(x,y,1,h);
}
void glcd_draw_str(const char *data, const tDrawStrCfg* cfg)
{
    uint8_t x1, y1;
    draw_rectangle(cfg->x, cfg->y, cfg->w, cfg->h, !cfg->font_color);
    x1=(cfg->x)+align((cfg->w), 6*strlen(data)-1 , cfg->valign);
    y1=(cfg->y)+align(cfg->h, 7, cfg->halign);
    draw_text(x1, y1, data, cfg->font_color);
    send_region(cfg->x, cfg->y, cfg->w, cfg->h);
}
#pragma optimize=none
static inline void reset_delay(void){
    for (int cnt = 100; --cnt;) {
        asm("nop");
        asm("nop");
    }
}

```

```

    }
}
static inline void write( uint8_t byte ) {
    Board_SPI_AssertSSEL(); {
        while ( !(LPC_SSP0->SR & SSP_STAT_TNF) ) continue;
        LPC_SSP0->DR = byte;
        while ( (LPC_SSP0->SR & SSP_STAT_BSY) ) continue; //SSP0SR_bit.BSY == 1 )
        while ( !(LPC_SSP0->SR & SSP_STAT_RNE) ) continue;
        byte = (uint8_t)LPC_SSP0->DR;
        (void)byte;
    } Board_SPI_DeassertSSEL();
}
static inline void write_data( uint8_t data ) {
    write(data);
}
static inline void write_cmd( uint8_t data ) {
    Board_Glcd_A0_Clr(); {
        write(data);
    } Board_Glcd_A0_Set();
}
static inline void set_address( uint8_t column, uint8_t page )
{
    write_cmd(0xB0|(page&0x0F));
    write_cmd(0x10|(column>>4));
    write_cmd(0x00|(column&0x0f));
}
static inline void send_region(uint8_t x,uint8_t y,uint8_t w,uint8_t h)
{
    uint8_t page, column;
    uint8_t cnt= (y%8+h+7)/8;
    for(page=(y/8); 0<cnt; page++, cnt--)
    {
        for (column=x; column<x+w; column++)
        {
            set_address(column,page);
        }
    }
}

```

```

        write_data(bufferGLCD[page][column]);
    }
}
static inline void draw_symb(uint8_t x, uint8_t y, unsigned char c, tLcdColor color)
{
    uint8_t data, i;
    c=symbol_decode(c);
    for (i = 0; i <=4 ; i++)
    {
        if (color == cWHITE)
        {
            data= font5x8[c][i];
        }
        else {data= ~( font5x8[c][i]);}
        buffer_block_set(x+i,y,data,7);
    }
}
static inline void draw_text(uint8_t x, uint8_t y, const char* s, tLcdColor color)
{
    unsigned char z=0;
    while (*s)
    {
        draw_symb(x+z, y, *s, color);
        z+=6;
        ++s;
    }
}
static inline void draw_rectangle(uint8_t x, uint8_t y, uint8_t w, uint8_t h, tLcdColor color) {
    uint8_t i;
    for(i=x; i<x+w; i++)
    {
        buff_draw_vline(i,y,h,color);
    }
}

```



```

static inline void draw_frame(uint8_t x, uint8_t y, uint8_t w, uint8_t h, tLcdColor color)
{
    buff_draw_hline(x, y, w, color);
    buff_draw_hline(x, y+h-1, w, color);

    buff_draw_vline(x, y, h,color);
    buff_draw_vline(x+w-1, y, h,color);
}
static inline unsigned char symbol_decode(unsigned char c)
{
    if(32<=c && c<='~')
    {
        c=c-32;
    }
    else if(192<=c && c<=255)
    {
        c=c-96;
    }
    else
    {
        c=255;
    }
    return c;
}
static inline void buffer_block_set(uint8_t x,uint8_t y,uint8_t data, uint8_t n)
{
    uint8_t page, column;
    column=x, page=y/8;
    uint8_t data_mask=(uint8_t)((1<<n)-1);
    uint8_t block_mask=(uint8_t)(data_mask<<(y%8));
    bufferGLCD[page][column]&= ~block_mask;
    bufferGLCD[page][column]|= (data<<(y%8)) & block_mask;

    int8_t n_left=(int8_t)(n-(8-(y%8)));
    if (n_left>=0)

```

```

    {
        ++page;
        block_mask=(uint8_t)((1<<n_left)-1);
        bufferGLCD[page][column]&= ~block_mask;
        bufferGLCD[page][column]|= (data>>(n-n_left)) & block_mask;
    }
}
static inline void buff_draw_vline(uint8_t x,uint8_t y,uint8_t h, tLcdColor color)
{
    uint8_t data, n, h_left, y_curr;
    data= (color==cWHITE ? 0xff : 0x00);
    n=h%8;
    buffer_block_set(x,y,data,n);
    h_left=h-n;
    y_curr=y+n;
    for (; h_left>0; h_left-=8, y_curr+=8)
    {
        buffer_block_set(x,y_curr,data,8);
    }
}
static inline void buff_draw_hline(uint8_t x,uint8_t y, uint8_t w, tLcdColor color)
{
    uint8_t data, i;
    data= (color==cWHITE ? 0xff : 0x00);
    for ( i=x; i<=x+w; i++)
    {
        buffer_block_set(i,y,data,1);
    }
}
static inline uint8_t align(uint8_t field_l, uint8_t object_l, tLcdAlign align)
{
    uint8_t offset;
    if (align==algnhCENTER) // algnvCENTER=algnhCENTER
    {
        offset=(field_l-object_l)/2;
    }
}

```

```
}  
else if(align==algnhLEFT)// algnvTOP=algnhLEFT  
{  
    offset=0;  
}  
else if(align==algnhRIGHT)// algnvBOTTOM=algnhRIGHT  
{  
    offset=field_1-object_1;  
}  
return (offset);  
}
```