

Министерство образования и науки Российской Федерации
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
«Высшая школа электроники и компьютерных наук»
Кафедра «Инфокоммуникационных технологий»

ПРОЕКТ ПРОВЕРЕН
Рецензент

В. М. Гриб

“_25_” ____ декабря ____ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ
Заведующий кафедрой ИКТ

Даровских С.Н.

“_27_” ____ декабря ____ 2017 г.

Индуктивный цифро-аналоговый преобразователь перемещения

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОМУ КВАЛИФИКАЦИОННОМУ ПРОЕКТУ (ВКП)
ЮУрГУ – Д.11.05.01.2017.320.00 ПЗ ВКП

Консультанты:

По экономике

Н.С. Дзензелюк

“_01_” ____ декабря ____ 2017 г.

Руководитель проекта:

И.В. Фокин

“_25_” ____ декабря ____ 2017 г.

По охране труда

И.С. Окраинская

“_29_” ____ роября ____ 2017 г.

Автор проекта:

студент группы КЭ-609

Г. А. Твердохлебов

“_29_” ____ декабря ____ 2017 г.

Нормоконтролер:

В.Д. Спицина

“_08_” ____ января ____ 2018 г.

Челябинск
2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1 Постановка задачи.....	7
1.1 Описание выполняемых функций	7
1.2 Анализ технического задания	8
2 Технический раздел.....	9
2.1 Построение структурной схемы	9
2.2 Разработка принципиальной схемы	9
2.2.1 Выбор микроконтроллера в качестве блока обработки информации.....	10
2.2.2 Схема электрическая принципиальная измерительного блока	13
2.2.3 Схема электрическая принципиальная блока генератора с излучателем	14
2.2.4 Схема электрическая принципиальная блока цифровой обработки.....	16
2.2.5 Схема электрическая принципиальная блока формирования выходного сигнала	17
2.2.6 Схема электрическая принципиальная блока преобразования питания	19
3 Конструкторский раздел.....	20
3.1 Разработка блоков «Электронного помощника оператора».....	20
3.2. Разработка узлов печатных	20
4 Разработка программного обеспечения	23
4.1 Принципы и методы программирования ИЦАПП. Структура ПО ИЦАПП	23
4.2 Методика отладки ПО ИЦАПП.	25
4.3 Описание и обоснование методов отладки ПО ИЦАПП	26
5 Организационно-экономический раздел.....	28
5.1 Сетевое планирование	28
5.2 Построение сетевого графика	28

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

5.3 Расчёт параметров событий сетевого графика.....	32
5.4 Расчёт параметров работ сетевого графика	33
5.5 Расчёт параметров СГ в целом.....	36
5.6 Расчет производственных затрат на проведение ОКР	37
5.7 Расчёт заработной платы	41
5.8 Накладные расходы.....	41
5.9 Расчет итоговых затрат	42
5.10 Расчет экономического эффекта от производства изделия	42
6 Охрана труда.....	43
6.1 Анализ опасных и вредных производственных факторов, источником которых является спроектированное устройство.....	43
6.2 Анализ условий эксплуатации проектируемого устройства	43
6.3 Обеспечение безопасности устройства.....	44
6.4 Требования к эргономичности.....	45
ЗАКЛЮЧЕНИЕ	46
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	47
ПРИЛОЖЕНИЕ А	49

ВВЕДЕНИЕ

Опыт эксплуатации технических систем, таких как производственные станки, находящихся под управлением оператора показал, что человек не всегда в состоянии самостоятельно уследить за правильной работой станка.

Воздействие внешних факторов, таких как высокие показатели температуры, влажности; запыленность воздуха, солнечная радиация усиливают усталость оператора, снижают внимание, контроль за приборами и аварийной сигнализацией. Аварийные режимы работы системы - выявляются несвоевременно, с опозданием.

Ошибочные действия оператора выводят из строя дорогостоящие агрегаты и узлы, а так же к браку производимой продукции.

Возникла необходимость автоматизировать управление системой аварийной остановки оборудования. На оператора при этом возлагается контроль за исправной работой системы управления.

В процессе эксплуатации оборудования выявляется необходимость мониторинга параметров работы агрегатов и узлов оборудования объекта.

Это привело к задаче создания устройства «Индуктивный цифро-аналоговый преобразователь перемещения», отвечающий за преобразование перемещения края металлического листа в электрический сигнал для обеспечения работы системы автоматического управления.

Данному устройству присвоено наименование «Индуктивный цифро-аналоговый преобразователь перемещения» (сокращенно ИЦАПП).

ИЦАПП предназначен для того, чтобы:

- а) повысить безопасность и упростить эксплуатацию оборудования;
- б) своевременно обнаружить неисправность и предотвратить выход из строя агрегатов и узлов, а так же предотвратить брак продукции;
- в) автоматически, независимо от действий оператора, предотвратить аварийные ситуации и поломку оборудования, совместно с автоматической системой управления оборудованием.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

1 Постановка задачи

1.1 Описание выполняемых функций

Целью разрабатываемого дипломного проекта является создание устройства индуктивный цифро-аналоговый преобразователь перемещения (ИЦАПП).

ИЦАПП предназначен для бесконтактного преобразования перемещения края кромки листа в электрический сигнал для управления исполнительным устройством.

Управление исполнительным устройством системы автоматического управления осуществляется с помощью выходных аналоговых сигналов:

А) при сопротивлении нагрузки не меньше 4,7 кОм:

- 1) двухполярный сигнал по напряжению ± 6 В;
- 2) однополярный сигнал по напряжению 0...10 В.

Б) при сопротивлении нагрузки не меньше 250 Ом: уровень ограничения сигнала по току 4...20 мА.

Выходные сигналы одинаковы по информационной составляющей, это сделано для того, чтобы можно было управлять несколькими устройствами, а так же для подключения измерительной техники.

При обнаружении отклонений в работе агрегатов и узлов объекта, ИЦАПП должен обеспечить регулировку отклонений в работе исполняющего устройства.

ИЦАПП должен обеспечивать выполнение перечисленных требований:

- при двухполярном напряжении питания ± 15 В. Максимальный ток потребления не должен превышать 190 мА.

- при однополярном напряжении питания 24 В. Максимальный ток потребления не должен превышать 230 мА.

При разработке следует обеспечить устойчивость системы к внешним воздействиям температуры от 0 °С до плюс 65 °С при влажности воздуха 85%.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

1.2 Анализ технического задания

Современные технические системы разрабатываются с применением новейших материалов, технологий, систем автоматического контроля и управления, с применением передовых программных продуктов. Благодаря появлению на рынке различных интеллектуальных цифровых микросхем, микроконтроллеров появилась возможность реализовать большое число функций разрабатываемых устройств, чем самым повысить автоматизацию систем, повысить покупательскую способность. Это привело к тому, что ядром многих устройств стала схемотехника, содержащая в основном цифровые компоненты.

С использованием методов и средств цифрового управления необходимо разработать такое устройство, в котором управление всеми основными процессами и функциями происходит именно на уровне цифровых микросхем или элементов. При этом требуется преобразование входного сигнала до уровня, который может обработать АЦП. Рассмотрим все выполняемые задачи подробнее.

Для преобразования входного сигнала до уровня который сможет преобразовать АЦП потребуется схема усилителя сигнала.

Для осуществления основных функций, таких как фильтрация входных линий, анализ входных сигналов, управление выходными каскадами целесообразно применить микропроцессорную технику. Рациональны однокристалльные микроЭВМ или микроконтроллеры, на основе которых возможна реализация достаточно сложных алгоритмов управления без существенного схемотехнического усложнения. Фильтрация входного сигнала и управление выходными каскадами может быть возложено на микроконтроллере.

Так же немаловажным фактором, который следует учесть при выборе компонентов ИЦАП - это сложные условия эксплуатации, которые обуславливают необходимость применения радиоэлектронных элементов, имеющих повышенную прочность конструкции, высокую надежность при эксплуатации, поэтому при выборе элементов требуется учесть устойчивость к внешним воздействиям.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

2 Технический раздел

2.1 Построение структурной схемы

Для построения структурной схемы ИЦАПП требуется определить основные структурные блоки, связи между ними, а также их взаимодействие с блоками внешних устройств.

Из технического задания на ИЦАПП ясно, что в разрабатываемом устройстве должны присутствовать такие блоки:

- блок преобразования напряжения питания;
- блок генератора;
- блок измерителя;
- блок цифровой обработки;

Создание программатора не является целью данного дипломного проектирования, так как эти решения уже давно разработаны и унифицированы. По этому поводу можно лишь отметить, что все функции по контролю за программированием блока обработки информации можно возложить на ЭВМ, разработав лишь надлежащий кабельный комплект.

Научно-технический прогресс позволяет на аппаратном уровне объединить некоторые функциональные блоки в одном электронном компоненте - микроконтроллере.

Схемотехнические решения каждого блока ИЦАПП приведены в следующем разделе [1].

Структурная схема устройства ИЦАПП представлена на чертеже ЮурГУ-Д.11.05.01.2017.320.00 Э2.

2.2 Разработка принципиальной схемы

Научно-технический прогресс в приборостроении характеризуется применением новейших материалов, технологий, систем автоматического контроля и управления, разработкой передовых программных продуктов.

					ЮурГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

В техническом задании указаны требования на применение перспективной элементной базы, в том числе и импортной, не имеющей отечественных аналогов, поэтому в разработке постараемся применить перспективные радиокомпоненты и элементы [2,3].

2.2.1 Выбор микроконтроллера в качестве блока обработки информации

Контроллер предназначен для осуществления функций:

- чтения дискретных сигналов внешних датчиков;
- фильтрация полученной информации;
- управления выходными каскадами.

Реализация схемы контроллера возможна двумя вариантами:

- с использованием устройств фиксированной логики;
- с использованием микроконтроллеров, микропроцессоров.

Устройства фиксированной логики при создании схемы использовать не выгодно, так как это резко увеличивает габариты, массу, энергопотребление изделия, снижает надежность. Кроме того, невозможно быстро и удобно изменять параметры настроек. Применение микроконтроллеров позволяет избавиться от перечисленных недостатков и построить удобное в работе, функционально насыщенное изделие.

Для обработки входных аналоговых и дискретных данных применим в качестве арифметико-логического устройства микроконтроллер семейства ARM Cortex-M0. Выберем из семейства контроллер LPC822M101JDH20 из-за его приемлемых электрических и функциональных характеристик, изложенных ниже [4].

Прибор LPC822M101JDH20 является 32-разрядным микроконтроллером на основе ARM Cortex-M0 архитектуры.

Особенности микроконтроллера LPC822M101JDH20:

- ARM Cortex-M0 + работает на частотах до 30 МГц с одноклапным множителем и быстрым одноклапным портом ввода-вывода;

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

- последовательный отладчик (SWD) с четырьмя точками прерывания и двумя точками наблюдения. Поддерживается ограниченное сканирование JTAG (BSDL);

- емкость внутрисистемно программируемой Flash памяти 16 кБайт, 1000 циклов стирания/записи;

- встроенная RAM емкостью 4 кБайт;

- API-интерфейсы встроенного ПЗУ для АЦП, SPI, I2C, USART, конфигурация питания (профили мощности) и целочисленное разделение;

- высокоскоростной интерфейс GPIO, подключенный к шине ARM Cortex-M0 +IO с программируемыми линиями I/O до 16 контактов, программируемым режимом открытого стока, входным инвертором и цифровым фильтром. Управление направлением GPIO поддерживает независимую установку/очистку/переключение отдельных бит;

- высоковольтный драйвер вывода источника (20 мА) на четырех контактах;

- возможность генерации прерываний GPIO с функцией логического соответствия шаблону на восьми входах GPIO;

- конфигурируемый пользователем таймер (SCTimer/ШИМ) с функциями ввода и вывода для тайминга и ШИМ-приложений. Каждый вход SCTimer/ШИМ мультиплексируется, чтобы можно было выбирать из нескольких источников ввода, таких как контакты, прерывание АЦП или выход компаратора;

- четырехканальный многорежимный таймер (MRT) для генерации повторяющихся прерываний с четырьмя программируемыми фиксированными скоростями;

- таймер самостоятельного запуска (WKT), синхронизированный от IRC, низкоэнергетического, низкочастотного внутреннего генератора или внешнего входного тактового сигнала в постоянно действующей области мощности;

- один 12-разрядный АЦП с несколькими внутренними и внешними триггерными входами, и частотой дискретизации до 1,2 мега выборок в секунду. АЦП поддерживает две независимые последовательности преобразования;

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

- компаратор с четырьмя входными контактами и внешним или внутренним опорным напряжением;
 - три интерфейса UART с функциями выводов, назначенными через матрицу переключателей и один общий генератор дробных бодовых скоростей;
 - два контроллера SPI с функциями выводов, назначенными через матрицу переключателей;
 - четыре интерфейса I2C-шины. Один I2C поддерживает Fast-Plus Plus со скоростью передачи данных 1 Мбит/с на двух реальных контактах с открытым сливом и режиме прослушивания. Три I2C поддерживают скорость передачи данных до 400 кбит/с на стандартных цифровых выводах;
 - внутренний RC-генератор 12 МГц с точностью до полутора процентов, который может быть использован в качестве системных часов;
 - кварцевый генератор с рабочим диапазоном от 1 МГц до 25 МГц;
 - программируемый тактирующий генератор с частотным диапазоном от 9,4 кГц до 2,3 МГц;
 - PLL позволяет процессору работать с максимальной частотой процессора без необходимости в высокочастотном кристалле. Может запускаться от системного генератора, внешнего тактового входа или внутреннего RC-генератора;
 - потребляемая мощность в активном режиме до 90 мкА/МГц в маломощном режиме с использованием RC в качестве источника синхронизации;
 - уменьшенные режимы питания: спящий режим, режим глубокого сна, режим пониженного энергопотребления и режим глубокого отключения;
 - диапазон напряжения питания от 1,8 до 3,6 В;
 - диапазон рабочих температур от минус 40 °С до плюс 105 °С;
- Микроконтроллер LPC822M101JDH20 поддерживается большим количеством средств разработки программ и систем, включающих: Си-компиляторы, микроассемблеры, отладчики/симуляторы программ, внутрисхемные эмуляторы и отладочные устройства.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

16 Кбайт Flash-памяти программного кода микроконтроллера вполне хватит для описания всего алгоритма работы системы ИЦАПП.

2.2.2 Схема электрическая принципиальная измерительного блока

Схема электрическая принципиальная измерительного блока (В1 Измерительная часть) представлена на чертеже ЮурГУ-Д.11.05.01.2017.320.00 ЭЗ в графической части дипломного проекта.

Для преобразования входного напряжения с низким уровнем сигнала в напряжение, необходимое для работы с микроконтроллером, применим схему инвертирующего усилителя и схему детектирования.

Усилительный каскад собран по схеме с инвертирующими усилителями, в основе которых лежит операционный усилитель TL084IDT. Было принято решение сделать усилитель на 3 каскадах для более стабильного усиления. [5]

Амплитудное детектирование возможно при помощи нелинейных цепей или синхронных детекторов. Детекторы с нелинейными элементами получили большее распространение благодаря своей простоте. Типовая принципиальная схема АД на диоде представлена на рисунке 1.

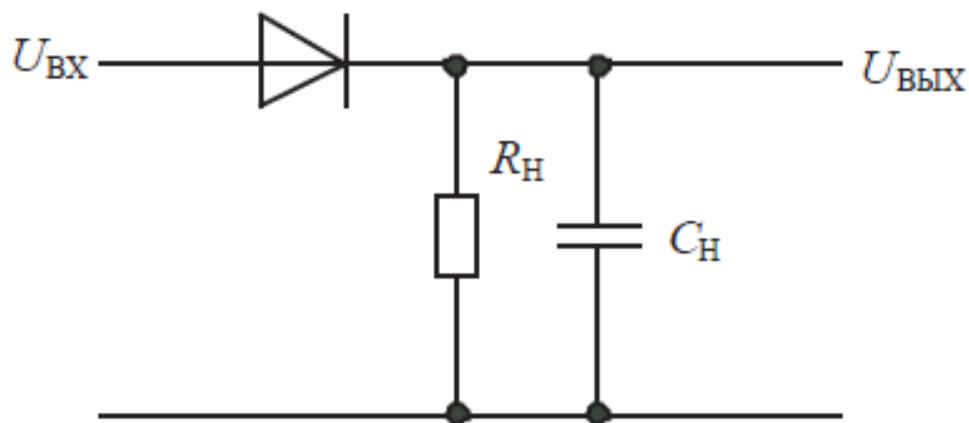


Рисунок 1 – Принципиальная схема диодного амплитудного детектора

2.2.3 Схема электрическая принципиальная блока генератора

Схема электрическая принципиальная блока генератора (В2 Генератор) представлена на чертеже ЮурГУ-Д.11.05.01.2017.320.00 Э3 в графической части дипломного проекта.

Блок состоит из нескольких каскадов:

- генератор;
- предусилитель;
- усилитель мощности класса D;
- излучатель.

Для генератора сигнала используем схему моста Вина, на основе операционного усилителя TL972ID. Типовая принципиальная схема генератора на мосте Вина представлена на рисунке 2.

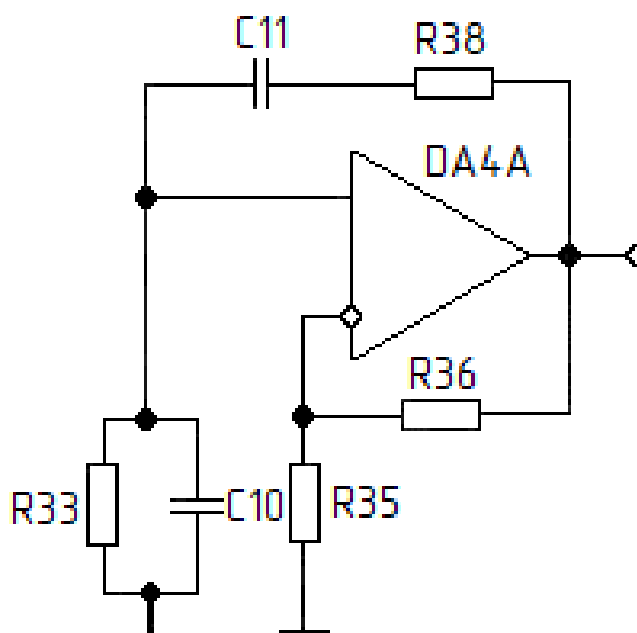


Рисунок 2 – Генератор на мосте Вина

Если величины сопротивлений R_{33} и R_{38} , а также ёмкости C_{10} и C_{11} не слишком сильно разнятся, то такая цепь имеет сглаженный квазирезонанс, то есть коэффициент передачи напряжения от правого по схеме вывода R_{38} (входной сигнал) в точку соединения R_{33} , C_{10} , C_{11} (выходной сигнал) имеет максимум на некоторой частоте.

Наиболее простая формула для квазирезонансной частоты имеет место при равенствах:

$$R_{33} = R_{38} = R \quad \text{и} \quad C_{10} = C_{11} = C,$$

при этом частота квазирезонанса равна:

$$f = \frac{1}{2\pi RC},$$

при ёмкости конденсатора C 100 пФ и частоте колебания сигнала 40 кГц, сопротивление R будет приблизительно 39 кОм.

На частоте квазирезонанса фазовый сдвиг выходного сигнала моста Вина относительно входного сигнала равен нулю, а модуль коэффициента передачи равен $1/3$ ($R_{36} = 2 \cdot R_{35}$). Если включить в петле обратной связи, охватывающей вход и выход моста Вина активный не инвертирующий усилительный элемент, в идеале не имеющий фазового сдвига, с коэффициентом передачи более трёх, то в контуре возникнут нарастающие до бесконечности по амплитуде автоколебания, так как в этом контуре не выполняется критерий устойчивости для линейных систем.

Решение проблемы обеспечения требуемого коэффициента усиления (K_u) давно известно: применить в качестве R_{35} или R_{36} какой-нибудь нелинейный или управляемый элемент, который будет подстраиваться таким образом, чтобы обеспечить заданный K_u при некоем определённом размахе выходного сигнала. Обычно ставят терморезисторы, миниатюрные лампочки, оптроны, либо полевые транзисторы (наш случай). Для достижения низкого ТНД необходимо обеспечить, чтобы нелинейность данного управляющего элемента не проявлялась на частотах генерации. Для лампочек и терморезисторов на частотах генерации от десятков Герц и выше это условие легко выполнимо за счёт тепловой инерционности оных. Полевым же транзистором необходимо управлять используя детектор с соотношением большой постоянной времени.

Для предварительного усиления сигнала используем схему инвертирующего усилителя, на основе того же операционного усилителя, что и для генератора.

Для усилительного каскада используем микросхему усилителя мощности класса D (TDA8920CJ). Схема усилителя мощности с небольшими коррекциями и

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

дополнениями построена на основе документации микросхемы. Были добавлены схемы плавного включения питания и его фильтрации.

В качестве излучателя используется стандартный LC колебательный контур, с электротехническими параметрами аналогичного устройства.

2.2.4 Схема электрическая принципиальная блока цифровой обработки

Схема электрическая принципиальная блока цифровой обработки (В3 Цифровая часть) представлена на чертеже ЮурГУ-Д.11.05.01.2017.320.00 Э3 в графической части дипломного проекта.

Блок цифровой обработки состоит из микроконтроллера DD2, двух периферийных устройств DD1 и DA7 и схем формирования выходных сигналов, которыми управляет микроконтроллер.

Микроконтроллер DD2 работает с внутренним тактовым генератором. Выводы «9» (I2C_SDA) и «10» (I2C_SCL) главного (master) устройства интерфейса передачи данных I2C для обмена данными с периферийными устройствами DD1 и DA7.

Микросхема DD1 (STML75M2F) является цифровым температурным сенсором с датчиком перегрева. В котором выход «3» (\overline{OS}/INT) отвечает за оповещение контроллера о превышении температурного порога, так же по интерфейсу I2C датчик сообщает о текущей температуре устройства, в результате чего микроконтроллер делает поправку в управлении исполнительным устройством.

Для увеличения точности преобразования входного сигнала в цифровой, необходим внешний АЦП. Это даёт, в отличии от встроенного АЦП, повышение точности преобразования за счёт встроенного опорного напряжения. Этим параметрам соответствует микросхема DA7 (ADS1114IDGST) представляющая собой 16-ти разрядный АЦП с встроенным опорным напряжением.

Для точного управления схемой формирования выходных сигналов используем встроенные в микроконтроллер таймеры. Контроллер формирует ШИМ сигналы управления (PWM_1 и PWM_2).

					ЮурГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

PWM_1 управляет схемой формирования двух выходных сигналов по напряжению, как указано в ТЗ.

PWM_2 управляет схемой формирования выходного сигнала по току.

2.2.5 Схема электрическая принципиальная блока преобразования питания

Схема электрическая принципиальная блока преобразования питания (В5 Питание) представлена на чертеже ЮУрГУ-Д.11.05.01.2017.320.00 ЭЗ в графической части дипломного проекта.

Блок формирования выходных сигналов состоит из трёх схем преобразования сигнала:

- по двухполярному напряжению с уровнем ограничения сигнала ± 6 В;
- по однополярному напряжению с уровнем ограничения сигнала $0 \dots 10$ В;
- по току с уровнем ограничения сигнала $4 \dots 20$ мА.

Преобразователь сигнала по двухполярному выходному напряжению составлен из двух схем: инвертирующем сумматоре на операционном усилителе; и двухтактном эмиттерном повторителе.

Для достижения заданного уровня напряжения рассчитаем инвертирующий сумматор на операционном усилителе по формуле:

$$U_{out} = -R_{23} \cdot \left(\frac{U_{PWM}}{R_{21}} + \frac{U_{оп}}{R_{100}} \right),$$

где U_{out} – уровень выходного сигнала (± 6 В),

R_{23} – сопротивление обратной связи усилителя,

R_{100} – ограничивающее сопротивление (680 кОм),

U_{PWM} – уровень управляющего сигнала ($0 \dots 3,3$ В),

$U_{оп}$ – подтягивающее напряжение (минус 5 В).

Схема на выходе устройства должна обеспечивать ± 6 В. Поэтому составим систему уравнений двух состояний выходного сигнала:

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

$$\begin{cases} U_{PWM1} = -R_{oc} \cdot \left(\frac{U_{PWM1}}{R_{21}} + \frac{U_{оп}}{R_{100}} \right); \\ U_{PWM2} = -R_{oc} \cdot \left(\frac{U_{PWM2}}{R_{21}} + \frac{U_{оп}}{R_{100}} \right); \end{cases} \quad \begin{cases} 6 = -R_{23} \cdot \left(-\frac{5}{R_{100}} \right); \\ -6 = -R_{23} \cdot \left(\frac{3,3}{R_{21}} + \frac{5}{R_{100}} \right). \end{cases}$$

упростим эту систему уравнений:

$$\begin{cases} R_{100} = \frac{5}{6} \cdot R_{23}; \\ R_{21} = \frac{3,3 \cdot R_{23}^2}{6 \cdot (R_{23} + 1)}. \end{cases}$$

из первого уравнения выразим R_{oc} и подставив во второе и решим систему:

$$\begin{cases} R_{23} = \frac{6}{5} \cdot R_{100}; \\ R_{21} = \frac{3,3 \cdot \left(\frac{6 \cdot R_{100}}{5} \right)^2}{6 \cdot \left(\frac{6 \cdot R_{100}}{5} + 1 \right)}; \end{cases} \quad \begin{cases} R_{23} = \frac{6}{5} \cdot R_{100}; \\ R_{21} = \frac{19,8 \cdot 462400 \cdot 10^6}{20400 \cdot 10^3 + 25}; \end{cases} \quad \begin{cases} R_{23} \approx 510 \text{ кОм}; \\ R_{21} \approx 430 \text{ кОм}. \end{cases}$$

Преобразователь сигнала по однополярному выходному напряжению построен на выходном измерительном усилителе с регулируемым коэффициентом усиления (AD8226ARMZ).

В документации на микросхему приведена схема подключения и формула расчёта коэффициента усиления:

$$V_{out} = G(V_{in+} - V_{in-}) + V_{ref},$$

где V_{in+} и V_{ref} равны нулю, так как подтянуты на землю, отсюда найдём $G = 3$.

Далее, найдём сопротивление усилителя (R_{61}) по формуле:

$$G = 1 + \frac{49,4 \text{ кОм}}{R_{61}},$$

отсюда:

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

$$R_{61} = \frac{49,4 \text{ кОм}}{G - 1} \approx 24 \text{ кОм.}$$

Преобразователь сигнала по току построен на той же схеме, что и преобразователь сигнала по однополярному выходному напряжению без управления коэффициентом усиления и добавлением передатчика тока на микросхеме XTR115U. Согласно технической документации максимальный входной ток на входе «2» микросхемы DA11 составляет 200 мкА, а управляющее напряжение с выхода «7» микросхемы DA10 - 3,3 В, отсюда по закону Ома рассчитаем сопротивление резистора R₆₄:

$$R_{64} = \frac{U_{DA10-7}}{I_{DA11-2}} \approx 16 \text{ кОм.}$$

2.2.6 Схема электрическая принципиальная блока преобразования питания

Схема электрическая принципиальная блока преобразования питания (В5 Питание) представлена на чертеже ЮурГУ-Д.11.05.01.2017.320.00 ЭЗ в графической части дипломного проекта.

Для стабильной работы всего устройства микросхемам необходимо стабильное питание с разными напряжениями питания в том числе и двухполярного. Так же необходима защита устройства от ошибочного подключения питания.

Для этих целей было использовано несколько микросхем понижения и стабилизации напряжения:

- схема на полевых транзисторах, для защиты устройства от переплюсовки.
- микросхема URA2415LD-20W-H, в качестве понижающего двухполюсного DC – DC преобразователя.
- микросхема L78L33ABD-TR, для питания микроконтроллера и других микросхем с использованием схемы защиты от короткого замыкания с высоким выходным током, предложенной разработчиками.
- микросхемы регуляторы положительного и отрицательного напряжения L78L05ABD13-TR и L79L05ABD13-TR, для питания остальной периферии.

					ЮурГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

3 Конструкторский раздел

3.1 Разработка блоков ИЦАПП

Разрабатываемые блоки относятся к классу наземной РЭА, устанавливаемой на неподвижном объекте. Общими для данного класса аппаратуры являются повышенные требования к вибро- и ударопрочности, вибро- и удароустойчивости, пылебрызгозащищённости, высокой влажности [6].

Промышленностью разработаны и выпускаются типовые ряды блоков для размещения на различных носителях. В данном случае, учитывая специфику использования устройства как оригинальную для каждого объекта установки, целесообразно разработать блоки специальной конструкции.

Разрабатываемое изделие относится к классу необслуживаемых.

Для удобства ремонта, блоки выполнены в виде разборной конструкции, состоящей из основания, для крепления соединителей и элементов крепления к корпусу объекта, и кожуха. Для уплотнения неподвижного соединения «основание-кожух» и для требования пылебрызгозащищённости используется резиновая прокладка. Длина внутриблочного монтажа обеспечивает простоту и лёгкость извлечения печатных узлов для их тестирования и наладки в составе блока. Для размещения силовых тепловыделяющих элементов внутри блока, применяется электроизоляционная теплопроводная прокладка для отвода тепла на корпус устройства.

3.2 Разработка узлов печатных

Все электронные схемы ИЦАПП расположены в печатных узлах. Основой печатного узла является печатная плата (ПП). ПП объединяет группу радиоэлементов в функционально законченный узел и должна обеспечить требуемые механические и электрические параметры в заданных условиях эксплуатации при минимальных затратах. Для этого необходимо:

- выбрать тип ПП;
- определить класс точности выполнения печатного рисунка;

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

- выбрать способ изготовления ПП;
- установить габаритные размеры и конфигурацию платы;
- разместить навесные элементы;
- определить размеры элементов рисунка;
- определить требования к рисунку ПП в отдельных местах;
- разместить их на плате и осуществить трассировку;
- разработать конструкторскую документацию.

Материал основания ПП одновременно выполняет функции конструктивной опоры для элементов схемы и теплоотводящей среды, обеспечивает электрическую изоляцию в тяжелых условиях и должен обеспечивать стабильность физико-механических и электрических параметров ПП после воздействия механических нагрузок, климатических факторов и агрессивных химических сред.

Толщина ПП выбирается из рекомендованного ряда толщин и должна быть увязана с диаметрами применяемых металлизированных отверстий (для качественной металлизации отношение диаметра металлизированного отверстия к толщине платы должно быть более 0,33). С учетом вышеперечисленных требований и технологических возможностей сторонних компаний выбираем в качестве материала печатной платы стеклотекстолит СФ-2-50Г-1,5 В.С. ГОСТ 10316-78.

Существует пять классов точности ПП, отличающиеся конфигурацией печатного рисунка. Класс точности выбирается исходя из плотности связей, технологических возможностей производства и допустимых рабочих напряжений для проводников. Согласно этим данным выбираем третий класс точности.

Поскольку проектируемый печатный узел содержит большое количество электрических связей между элементами, используем двухстороннюю ПП. Исходя из производственных возможностей, выбираем комбинированный позитивный метод изготовления печатных плат с классом точности 2 по ОСТ4ГО.010.026-85, который имеет следующие достоинства: низкая трудоёмкость, высокая разрешающая способность, возможность полной автоматизации процесса изготовления. В каче-

стве технологических отверстий используются крепежные отверстия в углах платы.

В соответствии с технологическими возможностями выбираем размеры плат (согласно ГОСТ 10317-79 «Платы печатные. Основные размеры» размеры каждой стороны ПП должны быть кратными 5 мм). Т.к. площадь платы определена, пытаемся разместить навесные элементы и выполнить трассировку печатных проводников с помощью ЭВМ автоматизированным методом.

Размещение элементов производим с двух сторон. Такое размещение обеспечит проектирование более простого и надежного корпуса. Для уменьшения накладных расходов целесообразно убрать проводящий рисунок из-под крупных элементов печатной платы, что позволит избежать использования прокладок при установке элементов.

Для трассировки печатных проводников используем программный продукт Altium Designer версии 2015 года. Данное программное обеспечение является наиболее эффективным из ряда других аналогичных программ (Microsim Design, OrCAD 7.0, Route 3.0), и самое главное - позволяет использовать часть уже разработанных библиотек элементов.

После выполнения размещения элементов и трассировки печатных проводников, выполняем доработку отдельных не разведенных соединений вручную и оформляем конструкторскую документацию

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

4 Разработка программного обеспечения

4.1 Принципы и методы программирования ИЦАПП. Структура ПО ИЦАПП

Структура программного обеспечения ИЦАПП была разработана с учетом следующих исходных данных:

- структурная схема системы;
- алгоритмы работы системы;
- выбранный тип микроконтроллера.

В общем виде ПО системы состоит из функциональных программ, системных программ и обработчиков прерываний.

Функциональной будем называть программу, обеспечивающую выполнение одной из штатных функций системы.

Системной будем называть программу, обеспечивающую обмен данными либо обработку данных, используемых в своей работе функциональными программами.

Необходимость обеспечения многозадачного режима функционирования ПО ИЦАПП объясняется необходимостью параллельного выполнения блоками системы разнотипных задач (периодического опроса датчиков, периодической проверки цепей датчиков и исполнительных элементов, функциональных задач и т.п.).

Кроме того, реализация многозадачного режима функционирования ПО конструктивных единиц ИЦАПП даёт возможность его разделения на отдельные функционально завершённые модули. Это позволяет назначать разные уровни приоритета для разных программ.

Разбиение функциональных программ на отдельные фрагменты, каждый из которых завершается командой возврата в диспетчер задач, даёт значительные преимущества по сравнению с написанием программ, непрерывно выполняющихся до завершения реализуемых ими алгоритмов:

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23

- отказ от циклических проверок условий внутри программ обеспечивает высвобождение вычислительных ресурсов микроконтроллера, поскольку допустимое время выполнения операций агрегатами и узлами объекта много больше времени проверки условия выполнения и выдачи очередного воздействия микроконтроллером;

- обеспечение возможности расчёта загрузки микроконтроллера;

- значительное упрощение процесса отладки ввиду минимизации действий внутри каждого фрагмента и отсутствия зависимости между условиями и действиями, описанными в разных фрагментах.

Упрощение процесса отладки также достигается благодаря облегчению поиска ошибочно написанного фрагмента.

Система управления считается работающей в режиме реального времени по отношению к управляемому объекту, если время реакции системы на изменение входных воздействий меньше длительность самого короткого процесса в объекте, который выполняется под действием ИЭ.

Согласно «Циклограмме работы агрегатов и узлов объекта», длительность самой короткой операции составляет 0,05 с.

Время реакции системы на изменение входных воздействий определяется суммой длительностей двух временных интервалов:

- временем фильтрации входных сигналов (в нашем случае – 0,003 с);

- периодом вызова функциональных программ, формирующих выходные сигналы по изменению входных воздействий (0,01 с).

Таким образом, время реакции на изменение входных воздействий для разрабатываемой системы составляет 0,013 с, что почти в 4 раза меньше длительности самой короткой операции механизма зарядания. Следовательно, разрабатываемая система является системой реального времени.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

Аппаратные прерывания вызываются аperiодическими событиями (например, завершение записи байта в ЭПЗУ микроконтроллера, завершение приёма или передачи байта по последовательной линии связи, завершение аналого-цифрового преобразования, изменение сигнала на входе одного из внешних прерываний и т. п.). События, вызывающие аппаратные прерывания, как правило, требуют немедленной (в пределах нескольких десятков микросекунд) реакции со стороны программного обеспечения, поэтому подпрограммы обработки аппаратных прерываний имеют в системе наивысший приоритет и выполняются за минимально возможное время.

Выбранная структура построения позволяет обеспечивать перенос программного обеспечения на другой тип микроконтроллера.

Изменения, необходимые для такого переноса, составят не более пяти процентов от общего объёма исходного кода.

Необходимость переноса части ПО ИЦАПП на другой тип микроконтроллера может быть обусловлена в основном возможностью использовать наработки по созданию ПО ИЦАПП при разработке программного обеспечения аналогичных систем слежения на других типах микроконтроллеров.

После написания кода программы для ИЦАПП и корректировки ошибок в отладчиках пакетов программ, её можно скомпилировать в Hex-файл «прошивки» контроллера. Листинг программного кода приведён в Приложении А.

4.2 Методика отладки ПО ИЦАПП

Как и отладку любой другой системы управления отладку ИЦАПП можно условно разделить на два этапа:

- до установки на объект;
- в составе объекта.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

До начала отладки в составе объекта должны быть удачно завершены следующие испытания системы:

- проверка функционирования аппаратной части системы;
- проверка работоспособности при воздействии предельных значений температуры - 0°C и плюс 65°C;
- проверка на механические воздействия;
- проверка соответствия программного обеспечения требованиям к системе по выполнению задач, в части алгоритмов работы ИЦАПП.

В составе объекта проверка производится на соответствие требованиям по выполнению задач аппаратной и программной частей системы одновременно.

Таким образом, процесс отладки и настройки системы до установки на объект можно разделить на:

- отладку программного обеспечения системы;
- отладку аппаратной части системы.

4.3 Описание и обоснование методов отладки ПО ИЦАПП

Используемые для разработки ПО ИЦАПП программный пакет Keil uVision включает в себя отладочные средства. Отладка отдельных программ в Keil uVision в основном производилась в режиме автоматической трассировки, с одной точкой останова на каждый программный фрагмент.

Поскольку используемые программные отладчики, не позволяют создавать автоматические модели управляемых объектов, пригодные для полномасштабной отладки функциональных программ, окончательная отладка ПО будет произведена после установки системы на объект.

Выбранные методы и принципы программирования обеспечивают следующие удобства при отладке и корректировке ПО:

- возможность производить отладку алгоритмов непосредственно в составе объекта без существенного увеличения трудозатрат, благодаря простоте и

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

«прозрачности» функциональных программ, полученной при построении ПО по принципу модульности и фрагментарности;

- возможность 100%-ой проверки системного ПО и обработчиков прерываний без использования аппаратной части ИЦАПП благодаря почти полной независимости системного ПО друг от друга и от функциональных программ, а также наличия соответствующих пакетов разработки и отладки;

- гибкость программного обеспечения благодаря независимости функциональных модулей друг от друга;

- возможность расширения количества функционального ПО путём простого «присоединения» новых функциональных модулей.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

5 Организационно-экономический раздел

5.1 Сетевое планирование

Метод сетевого планирования применяется для оптимизации планирования и управления сложными разветвлёнными комплексами работ. При применении данного метода достигается:

- координация работ, выполняемых различными исполнителями;
- установление между исполнителями необходимой связи;
- выявление решающих направлений и работ;
- целесообразное и планомерное сокращение сроков.

5.2 Построение сетевого графика

Сетевой график лежит в основе сетевого планирования. Он является графическим изображением комплекса взаимосвязанных работ, выполняемых в определенной последовательности. Основными элементами графика являются работа, событие, путь.

Работа - это действия и процессы, связанные с затратами ресурсов и времени. Событие - это результат одной или нескольких работ (кроме исходного события). На графике работы изображаются стрелками, события – кружками, пример приведен на рисунке 3.

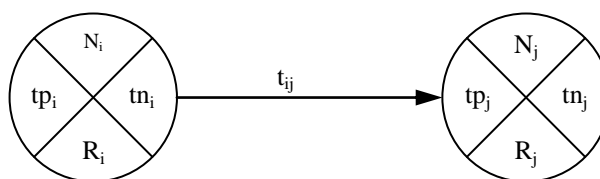


Рисунок 3 – Фрагмент сетевого графика

где: N_i - номер события;

i - номер предшествующего события;

j - номер данного события;

Изм.	Лист	№ докум.	Подпись	Дата

t_{p_i} - ранний срок свершения события, срок, необходимый для выполнения всех предшествующих работ без растягивания их во времени;

t_{n_i} - поздний срок свершения i -го события, срыв которого приводит к срыву срока завершаемого события;

R - резерв времени;

t_{ij} - продолжительность работ.

Ожидаемая продолжительность работы t_{ij} в СГ рассчитывается по принятой двух оценочной методике, исходя из минимальной $t_{ij \text{ мин}}$ и максимальной $t_{ij \text{ макс}}$ оценок продолжительности. При этом предполагается, что минимальная оценка соответствует наиболее благоприятным условиям работы, а максимальная - наиболее неблагоприятным.

Ожидаемая продолжительность каждой работы определяется по формуле

$$t_{ij \text{ ож}} = 0,6 t_{ij \text{ мин}} + 0,4 t_{ij \text{ макс}}.$$

Среднеквадратическое отклонение δ_{ij} продолжительности в двухоценочной методике рассчитывается по формуле

$$\delta_{ij} = 0,2 \cdot (t_{ij \text{ макс}} - t_{ij \text{ мин}}).$$

Дисперсия определяется по формуле

$$D_{ij} = 0,04 \cdot (t_{ij \text{ макс}} - t_{ij \text{ мин}})^2.$$

Рассчитанные значения заносятся в таблицу 1 по возрастанию кодов работ. В зависимости от содержания и сложности каждой работы численность исполнителей принимать от одного до четырёх человек (1-руководитель, 2-инженер, 1-лаборант). Код работы составляется из номеров начального и конечного событий этой работы, отделяемых запятой (i, j).

Таблица 1 - Перечень, параметры и вероятностные характеристики работ СГ

Код	Наименование работы	Продолж.			Исполн.			СКО	Дисп дн2
		мин.	макс.	ожид.	рук.	инж.	лаб.		
0-1	1	8	14	12	1	0	0	1,2	1,44
1-2	2	21	37	31	1	1	0	3,2	10,24
1-3	6	7	12	10	0	1	0	1,0	1,00
1-4	18	13	19	17	0	0	1	1,2	1,44
2-5	3	13	21	18	1	1	0	1,6	2,56
3-6	7	17	35	28	0	1	0	3,6	12,96
4-8	5	4	8	6	0	0	1	0,8	0,64
5-6	4	15	24	20	1	1	0	1,8	3,24
6-7	9	19	33	27	1	1	0	2,8	7,84
6-8	11	6	18	13	0	1	0	2,4	5,76
7-8	10	24	44	36	1	1	0	4,0	16,00
8-9	8	16	28	23	1	2	0	2,4	5,76
8-10	12	5	6	6	0	0	1	0,2	0,04
9-11	14	20	39	31	1	2	0	3,8	14,44
10-11	13	18	34	28	0	0	1	3,2	10,24
11-12	22	12	21	17	0	1	0	1,8	3,24
11-13	15	32	42	38	1	1	0	2,0	4,00
11-14	16	19	29	25	0	0	1	2,0	4,00
12-15	24	13	14	14	0	1	0	0,2	0,04
13-15	19	20	45	35	1	1	0	5,0	25,00
14-15	17	12	17	15	0	0	1	1,0	1,00
15-16	20	18	34	28	1	2	1	3,2	10,24
16-17	23	5	11	9	0	1	1	1,2	1,44
16-18	21	8	21	16	1	1	0	2,6	6,76
17-18	25	14	16	15	0	1	1	0,4	0,16
18-19	26	7	9	8	1	0	0	0,4	0,16
19-20	29	22	37	31	1	1	0	3,0	9,00
19-21	27	6	7	7	0	1	0	0,2	0,04
19-22	31	20	25	23	0	0	1	1,0	1,00
20-22	30	26	46	38	1	1	0	4,0	16,00
21-22	28	25	35	31	0	1	0	2,0	4,00
22-23	32	5	5	5	1	0	0	0,0	0,00

В таблице 1 указаны следующие наименования работ:

- 1 Получение и анализ технического задания на разработку (модернизацию) прибора.
- 2 Подбор технической документации (разработок родственных фирм, включая зарубежные).
- 3 Анализ технической документации (выявление аналогов и эффек-

тивных технических решений).

4 Составление рекомендаций по результатам анализа технической документации.

5 Анализ затрат на разработку, изготовление и эксплуатацию прототипа.

6 Выявление аналогов, изготовленных в натуре (в металле).

7 Анализ аналогов и выбор прототипа

8 Разработка технических предложений по модернизации прототипа и разработке нового прибора. Согласование с заказчиком.

9 Анализ функциональной недостаточности и избыточности прототипа.

10 Выдвижение и проработка предложений по модернизации прототипа.

11 Изучение условий и режимов эксплуатации разрабатываемого (модернизируемого) прибора.

12 Анализ литературы по БЖД.

13 Разработка мероприятий по БЖД.

14 Доработка модели прототипа в соответствии с предложениями по его модернизации.

15 Моделирование работы прототипа с учетом предложений по его модернизации в требуемых условиях и режимах.

16 Ранжирование реализуемых прототипом функций по критериям качества.

17 Ранжирование реализуемых прототипом функций по затратам на их разработку, изготовление и эксплуатацию.

18 Разработка СГ.

19 Выявление эффективных технических решений требуемых функций (работ) в аналогах, изготовленных в натуре (металле).

20 Разработка модели нового прибора.

21 Моделирование работы нового прибора в требуемых режимах и условиях.

22 Разработка функциональной схемы модернизируемого прототипа.

23 Разработка функциональной схемы нового прибора.

- 24 Разработка структурной схемы модернизируемого прототипа.
- 25 Разработка структурной схемы нового прибора.
- 26 Выбор оптимального варианта и согласование с заказчиком.
- 27 Разработка основных принципиальных конструкторских решений.
- 28 Разработка конструкторских решений сборочных единиц и деталей.
- 29 Разработка основных принципиальных технологических решений.
- 30 Разработка технологических решений сборочных единиц и деталей.
- 31 Расчет показателей эффективности.
- 32 Сдача заказчику

5.3 Расчёт параметров событий сетевого графика

Нумерация событий ведётся по возрастанию номера слева направо и сверху вниз. Ранний срок свершения исходного (нулевого) события СГ принимается равным нулю. Ранний срок свершения данного промежуточного события рассчитывается путём сравнения сумм, состоящих из раннего срока свершения события, непосредственно предшествующего данному и длительности работы. Так как данное событие не может свершиться, пока не закончится последняя из непосредственно предшествующих ему работ, очевидно, что в качестве раннего срока свершения события принимается максимальная из сравниваемых сумм. Рассчитанный таким способом ранний срок свершения завершающего события всего СГ принимается в качестве его же позднего срока свершения. Это означает, что завершающее событие СГ никаким резервом времени не располагает.

Поздний срок свершения данного промежуточного события определяется при просмотре СГ в обратном направлении. Для этого сопоставляются разности между поздним сроком свершения события, непосредственно следующего за данным, и продолжительности работы, соединяющей соответствующее событие с данным. Так как ни одна из непосредственно следующих за данным событием работ не может начаться, пока не свершится само данное событие, очевидно, его поздний срок свершения равен минимуму из подсчитанных разностей.

Правильность расчёта поздних сроков свершения событий СГ подтверждается

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

получением нулевого позднего срока свершения исходного события.

Резерв времени образуется у тех событий, для которых поздний срок свершения больше раннего, и он равен их разности. Если же эти сроки равны, событие резервом времени не располагает и, следовательно, лежит на критическом пути. Результаты расчёта сводятся в таблицу 2.

Таблица 2 - Параметры событий сетевого графика

Номер события	Сроки свершения		Резерв
	Ранний	Поздний	
0	0	0	0
1	12	12	0
2	43	43	0
3	22	53	31
4	29	138	109
5	61	61	0
6	81	81	0
7	108	108	0
8	144	144	0
9	167	167	0
10	150	170	20
11	198	198	0
12	215	257	42
13	236	236	0
14	223	254	31
15	271	271	0
16	299	299	0
17	308	308	0
18	323	323	0
19	331	331	0
20	362	362	0
21	338	369	31
22	400	400	0
23	405	405	0

5.4 Расчёт параметров работ сетевого графика

Ранний срок начала работы $T_{рнij}$ совпадает с ранним сроком свершения её начального события. Поздний срок начала работы $T_{пнij}$ можно получить, если из позднего срока свершения её конечного события вычесть её ожидаемую продолжительность. Ранний срок окончания работы $T_{роij}$ образуется прибавлением её продолжительности к раннему сроку свершения её начального события. Поздний срок окон-

чания работы $T_{поij}$ совпадает с поздним сроком свершения её конечного события.

Для всех работ критического пути, как не имеющих резервов времени, ранний срок начала совпадает с поздним сроком начала, а ранний срок окончания - с поздним сроком окончания. Работы, не лежащие на критическом пути, обладают резервами времени.

Полный резерв времени работы $R_{пij}$ образуется вычитанием из позднего срока свершения её конечного события раннего срока свершения её начального события и её ожидаемой продолжительности. Частный резерв времени первого рода $R^1_{пij}$ равен разности поздних сроков свершения её конечного и начального событий за вычетом её ожидаемой продолжительности. Частный резерв времени второго рода $R^2_{пij}$ равен разности ранних сроков свершения её конечного и начального событий за вычетом её ожидаемой продолжительности. Свободный (независимый) резерв времени работы $R_{сij}$ образуется вычитанием из раннего срока свершения её конечного события позднего срока свершения её начального события и её ожидаемой продолжительности. Свободный резерв времени может быть отрицательным.

Правильность расчётов резервов времени работы можно проверить по следующим соотношениям:

- а) сумма полного и свободного резерва работы равна сумме двух частных её резервов;
- б) поздний и ранний сроки начала работы, а также поздний и ранний сроки её окончания всегда отличаются на величину её полного резерва.

Для работ, лежащих на критическом пути никаких резервов времени нет и, следовательно, коэффициент напряжённости $K_{нij}$ таких работ равен единице. Если работа не лежит на критическом пути, она располагает резервами времени и её коэффициент напряжённости меньше единицы. Его величина подсчитывается как отношение суммы продолжительностей отрезков максимального пути, проходящего через данную работу, не совпадающих с критическим путём $t_{максij}$ к сумме продолжительностей отрезков критического пути, не совпадающих с максимальным путём, проходящим через эту работу $t_{крij}$.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

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

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

Результаты сведены в таблицу 3.

Таблица 3- Параметры работ сетевого графика

Код	Т ожид.	Ранний ср.св.	Поздний ср.св.	Ранний ср.св.	Поздний ср.св.	Резервы времени				
						полный	Част. 1 рода	Част. 2 рода	Своб.	Коэф. напр.
0-1	12	0	0	12	12	0	0	0	0	1,0
1-2	31	12	12	43	43	0	0	0	0	1,0
1-3	10	12	43	22	53	31	31	0	-31	0,6
1-4	17	12	121	29	138	109	109	0	-109	0,2
2-5	18	43	43	61	61	0	0	0	0	1,0
3-6	28	22	53	50	81	31	0	31	-31	0,6
4-8	6	29	138	35	144	109	0	109	-109	0,2
5-6	20	61	61	81	81	0	0	0	0	1,0
6-7	27	81	81	108	108	0	0	0	0	1,0
6-8	13	81	131	94	144	50	50	50	-50	0,2
7-8	36	108	108	144	144	0	0	0	0	1,0
8-9	23	144	144	167	167	0	0	0	0	1,0
8-10	6	144	164	150	170	20	20	0	-20	0,6
9-11	31	167	167	198	198	0	0	0	0	1,0
10-11	28	150	170	178	198	20	0	20	-20	0,6
11-12	17	198	240	215	257	42	42	0	-42	0,4
11-13	38	198	198	236	236	0	0	0	0	1,0
11-14	25	198	229	223	254	31	31	0	-31	0,6
12-15	14	215	257	229	271	42	0	42	0	0,4
13-15	35	236	236	271	271	0	0	0	0	1,0
14-15	17	223	254	240	271	31	0	31	-31	0,6
15-16	28	271	271	299	299	0	0	0	0	1,0
16-17	9	299	299	308	308	0	0	0	0	1,0
16-18	16	299	307	315	323	8	8	8	-8	0,7
17-18	15	308	308	323	323	0	0	0	0	1,0
18-19	8	323	323	331	331	0	0	0	0	1,0
19-20	31	331	331	362	362	0	0	0	0	1,0
19-21	7	331	362	338	369	31	31	0	-31	0,6
19-22	23	331	377	354	400	46	46	46	-46	0,3
20-22	38	362	362	400	400	0	0	0	0	1,0
21-22	31	338	369	369	400	31	0	31	-31	0,6
22-23	5	400	400	405	405	0	0	0	0	1,0

Учитывая все рассчитанные данные, строится сетевой график представленный в дополнении ЮУрГУ-Д.11.05.01.2017.320.00 Д5.

5.5 Расчёт параметров СГ в целом

В этом разделе рассчитываются следующие параметры СГ:

- количество событий $n_c=23$ в СГ, включая исходное;
- количество работ $n_p= 32$ в СГ;
- коэффициент сложности $k_c= 1,391$ СГ, равный отношению количества работ к количеству событий в СГ.

Критический путь $L_{кр}$ в СГ проходит через события и работы, не обладающие резервами времени, и имеет, следовательно, максимальную продолжительность $t_{кр}=405$, равную сроку свершения завершающего события.

Дисперсия срока наступления завершающего события определяется в соответствии с центральной предельной теоремой теории вероятностей как сумма дисперсии работ критического пути:

$$D= 179,68.$$

Вероятность свершения завершающего события в срок, равный продолжительности критического пути, равна $p(t_{св} \leq t_{кр}) = 0,5$. Если директивный срок T_d установлен меньше продолжительности критического пути, вероятность свершения события к директивному сроку меньше 0,5 и может быть рассчитана с помощью функции распределения нормального отклонения (нормированная функция Лапласа) $\Phi(i)+0,5$. Нормальное отклонение "и" равно разности между директивным сроком и продолжительностью критического пути (с учётом знака), отнесённой к среднеквадратическому отклонению продолжительности критического пути.

В задаче требуется определить вероятности свершения завершающего события в сроки, указанные в таблице 4.

Среднеквадратическое отклонение продолжительности критического пути определяется по формуле

$$\delta_{кр} = \sqrt{D_{ткр}} = 13,405,$$

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

где $D_{\text{кр}}$ - дисперсия срока наступления завершающего события, равная сумме дисперсий работ критического пути.

Таблица 4 - Вероятности свершения завершающего события в директивный срок от года до полутора лет

Наименование показателя	Отношение директивного срока к критическому
Критический срок	405,000
Среднеквадратическое отклонение	13,405
Директивный срок	400,000
Нормальное отклонение	-0,373
Вероятность свершения	0,355

Вероятность свершения больше 0,35, что удовлетворяет условия, поэтому перерасчёт СГ не нужен.

5.6 Расчет производственных затрат на проведение ОКР

Целью опытно-конструкторских работ была разработка, создание и отладка работы системы управления двухплоскостной платформы. В смету затрат на ОКР входят следующие статьи:

- материалы и компоненты;
- услуги сторонних организаций;
- покупные изделия;
- заработная плата;
- накладные расходы.

Ведомости статей затрат приведены в таблицах 5-7.

Таблица 5 - Перечень материалов и компонентов

Наименование	Количество, шт	Цена, руб	Стоимость, руб.
Излучающая катушка			
1 Катушка индуктивности	1	10,000	10,000
2 Перемычка МГШВ 0,35 L=40 мм Белый	2	0,500	1,000
3 Печатная плата	1	1,835	1,835
4 Конденсатор К73-17с-вариант 2- 630В-1000 пФ ±10%	1	3,388	3,388
5 Прокладка	1	0,500	0,500
6 Шайба 3 65Г 06 ГОСТ 6402-70	7	0,800	5,600
7 Шайба 3.01.08кп.016 ГОСТ 10450-78	4	0,800	3,200
8 Корпус катушки	1	1,001	1,001

Продолжение таблицы 5

Наименование	Количество, шт	Цена, руб	Стоимость, руб.
9 Планка	1	1,001	1,001
10 Винт самонарезающий 3x8 ГОСТ 11650-80	7	2,500	17,500
11 Лепесток 1-2-3/2x20-06 ГОСТ 22375-77	2	1,000	2,000
12 Лепесток 1-1-3,2x18-06 ГОСТ 22376-77	1	1,000	1,000
13 Катушка индуктивности	1	10,000	10,000
14 Перемычка МГШВ 0,35 L=40 мм Белый	2	0,500	1,000
15 Печатная плата	1	1,835	1,835
16 Прокладка	1	0,500	0,500
17 Шайба 3 65Г 06 ГОСТ 6402-70	7	0,800	5,600
18 Шайба 3.01.08кп.016 ГОСТ 10450-78	4	0,800	3,200
19 Корпус катушки	1	1,001	1,001
20 Планка	1	1,001	1,001
21 Винт самонарезающий 3x8 ГОСТ 11650-80	7	2,500	17,500
22 Лепесток 1-2-3/2x20-06 ГОСТ 22375-77	2	1,000	2,000
23 Лепесток 1-1-3,2x18-06 ГОСТ 22376-77	1	1,000	1,000
Корпус			
24 Перемычка МГШВ 0,35 L=140 мм Белый	4	0,500	4,000
25 Жгут СБ	1	2,000	1,000
26 Светодиод АЛ307КМ	1	4,969	1,000
27 Крышка	1	1,001	1,000
28 Гайка М3-6Н.5.А.016 ГОСТ 5927-70	4	2,000	4,000
29 Гайка М4-6Н.5.А.016 ГОСТ 5927-70	6	2,500	6,000
30 Гайка М5-6Н.5.А.016 ГОСТ 5927-70	8	3,000	8,000
31 Винт М4-6gx10.36.016 ГОСТ 17475-80	8	2,500	8,000
32 Винт М3x10.08X18Н10 ГОСТ 1491-80	4	0,899	4,000
33 Винт М3-6gx22.36.016 ГОСТ 1491-80	2	2,000	2,000
34 Винт М5-6gx16.36.016 ГОСТ 1491-80	6	3,000	6,000
35 Винт М5-6gx20.36.016 ГОСТ 17475-80	8	3,000	8,000
36 Прокладка	1	1,000	1,000
37 Шайба 3 65Г 06 ГОСТ 6402-70	6	0,600	6,000
38 Шайба 4 65Г 06 ГОСТ 6402-70	6	0,800	6,000
39 Шайба 5 65Г 06 ГОСТ 6402-70	14	1,000	14,000
40 Шайба 3.01.08кп.016 ГОСТ 10450-78	5	0,600	5,000
41 Шайба 4.01.08кп.016 ГОСТ 10450-78	6	0,800	6,000
42 Шайба 5.01.08кп.016 ГОСТ 10450-78	14	1,000	14,000
43 Планка	1	1,001	1,000
44 Вилка 2PM22Б10Ш1А	1	410,000	1,000
45 Стойка	6	1,001	6,000
46 Корпус СБ	1	100,000	1,000
47 Радиатор	1	1,001	1,000
48 Лепесток 1-1-3,2x18-06 ГОСТ 22376-77	1	1,000	1,000
49 Лента ПВХ 15x 0,2 синий первый сорт ГОСТ 16214-86	1	10,000	1,000
50 Шнур-чулок ШЧС-1,0 ОСТ17-184-2003	1	20,000	20,000

Продолжение таблицы 5

Наименование	Количество, шт	Цена, руб	Стоимость, руб.
Печатная плата			
51 Диод LL4148	4	0,384	1,538
52 Диод 1N5819	2	0,862	1,724
53 Светодиод АЛ307КМ	1	4,969	4,969
54 Чип конденсатор СС 0805 25В-2,2 мкФ Х7R ±10%	2	1,435	2,871
55 Чип конденсатор СС 0805 25В-2,2 мкФ Х7R ±10%	2	1,435	2,871
56 Чип конденсатор СС 0805 50В-0,01 мкФ Х7R ±10%	1	0,177	0,177
57 Чип конденсатор СС 0805 50В-0,1 мкФ Х7R ±10%	11	0,164	1,807
58 Чип конденсатор СС 0805 50В-0,33 мкФ Х7R ±10%	1	0,428	0,428
59 Чип конденсатор СС 0805 50В-0,47 мкФ Х7R ±10%	9	0,725	6,529
60 Чип конденсатор СС 0805 50В-1 мкФ Х7R ±10%	1	0,667	0,667
61 Чип конденсатор СС 0805 50В-1 мкФ Х7R ±10%	1	0,667	0,667
62 Чип конденсатор СС 0805 50В-4,7 мкФ Х7R ±10%	1	2,864	2,864
63 Чип конденсатор СС 0805 50В-100 пФ NPO(COG) ±5%	8	0,192	1,538
64 Чип конденсатор СС 0805 50В-220 пФ NPO(COG) ±5%	5	0,233	1,163
65 Чип конденсатор СС 0805 50В-1500 пФ NPO(COG) ±5%	1	0,713	0,713
66 Чип конденсатор СС 0805 50В-6800 пФ NPO(COG) ±5%	4	0,384	1,538
67 Чип конденсатор СС 1206 50В-0,015 мкФ Х7R ±10%	2	0,202	0,403
68 Чип конденсатор СС 1206 50В-1 мкФ Х7R ±10%	1	1,389	1,389
69 Чип конденсатор СС 1206 50В-2,2 мкФ Х7R ±10%	1	2,120	2,120
70 Чип резистор CR 0805-0,125-0 Ом ±5%	2	0,074	0,149
71 Чип резистор CR 0805-0,125-1 кОм ±5%	2	0,071	0,143
72 Чип резистор CR 0805-0,125-1 кОм ±5%	2	0,071	0,143
73 Чип резистор CR 0805-0,125-1 МОм ±5%	1	0,084	0,084
74 Чип резистор CR 0805-0,125-2 кОм ±5%	1	0,099	0,099
75 Чип резистор CR 0805-0,125-2 кОм ±5%	2	0,099	0,198
76 Чип резистор CR 0805-0,125-2,4 кОм ±5%	1	0,093	0,093
77 Чип резистор CR 0805-0,125-3 кОм ±5%	1	0,074	0,074
78 Чип резистор CR 0805-0,125-3,6 кОм ±5%	2	0,050	0,099
79 Чип резистор CR 0805-0,125-4,7 кОм ±5%	2	0,105	0,211
80 Чип резистор CR 0805-0,125-4,7 кОм ±5%	2	0,105	0,211
81 Чип резистор CR 0805-0,125-5,1 кОм ±5%	4	0,071	0,285
82 Чип резистор CR 0805-0,125-5,6 кОм ±5%	1	0,192	0,192
83 Чип резистор CR 0805-0,125-5,6 кОм ±5%	1	0,192	0,192
84 Чип резистор CR 0805-0,125-10 Ом ±5%	2	0,093	0,186
85 Чип резистор CR 0805-0,125-10 кОм ±5%	9	0,071	0,642
86 Чип резистор CR 0805-0,125-10 кОм ±5%	5	0,071	0,357
87 Чип резистор CR 0805-0,125-10 кОм ±5%	1	0,071	0,071
88 Чип резистор CR 0805-0,125-27 кОм ±5%	1	0,093	0,093
89 Чип резистор CR 0805-0,125-30 кОм ±5%	1	0,087	0,087
90 Чип резистор CR 0805-0,125-39 кОм ±5%	2	0,078	0,155
91 Чип резистор CR 0805-0,125-39 кОм ±5%	3	0,078	0,233

Изм.	Лист	№ докум.	Подпись	Дата

ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ

Лист

39

Продолжение таблицы 5

Наименование	Количество, шт	Цена, руб	Стоимость, руб.
92 Чип резистор CR 0805-0,125-51 кОм ±5%	1	0,071	0,071
93 Чип резистор CR 0805-0,125-100 кОм ±5%	8	0,071	0,570
94 Чип резистор CR 0805-0,125-200 кОм ±5%	2	0,090	0,180
95 Чип резистор CR 0805-0,125-470 Ом ±5%	1	0,074	0,074
96 Чип резистор CR 1206-0,25-4,7 кОм ±5%	1	0,105	0,105
97 Чип резистор CR 1206-0,25-5,1 кОм ±5%	2	0,140	0,279
98 Чип резистор CR 1206-0,25-10 Ом ±5%	2	0,071	0,143
99 Чип резистор CR 2512-1-0,1 Ом ±5%	2	0,100	0,199
100 Чип резистор CR 0805-0,125-10 МОм ±5%	2	0,199	0,397
101 Чип резистор CR 0805-0,125-160 Ом ±5%	1	0,186	0,186
102 Чип резистор CR 0805-0,125-180 Ом ±5%	1	0,157	0,157
103 Электролит 25В-100 мкФ (6x11)	1	0,849	0,849
104 Электролит 63В-470 мкФ ±20% SR (12x20)	2	1,001	2,003
105 Электролит ELZET 50В-10 мкФ (5x7) +105С	1	0,583	0,583
106 Электролит ELZET 63В-2,2 мкФ ±20% SS (4x7)	1	1,026	1,026
107 Электролит SAMWHA 63В-100 мкФ ±20% (8x11,5) +105С	5	3,252	16,260
108 Электролит JAMICON 63В-47 мкФ (6,3x11) +105С	1	1,358	1,358
109 Электролит JAMICON 63В-47 мкФ (6,3x11) +105С	2	1,358	2,716
110 Супрессор SMBJ36CA	1	4,501	4,501
111 Микросхема L79L05ABD13-TR	1	8,410	8,410
112 Микросхема URA2415LD-20W-H	1	1305,401	1305,401
113 Микросхема LP2951ACMX-3,3	1	52,268	52,268
114 Микросхема STLM75M2F	1	50,490	50,490
115 Микросхема XTR115U/2K5	1	182,860	182,860
116 Микросхема L78L05ABD13-TR	1	5,840	5,840
117 Микросхема TDA8920CJ/N1.112	1	195,600	195,600
118 Микросхема ADG419BRZ	1	111,600	111,600
119 Микросхема ADS1114IDGST	1	292,361	292,361
120 Микросхема TL084CPWR	1	9,700	9,700
121 Микросхема TL972ID	1	23,151	23,151
122 Микросхема AD8226ARMZ	2	122,599	245,198
123 Микросхема LPC822M101JDH20	1	85,176	85,176
124 Микросхема LM393D	1	4,994	4,994
125 Печатная плата	1	262,908	262,908
126 Чип-транзистор IRF7341	1	25,678	25,678
127 Чип-транзистор IRF7342	1	18,278	18,278
128 Чип-транзистор BDP949	1	15,773	15,773
129 Чип-транзистор BDP953	1	10,335	10,335
130 Чип-транзистор BDP954	1	16,855	16,855
131 Стабилитрон BZV55C9V1	2	0,462	0,924
132 Стабилитрон BZV85C15V	1	1,017	1,017
133 Стабилитрон BZV85C39	1	1,634	1,634

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ

Продолжение таблицы 5

Наименование	Количество, шт	Цена, руб	Стоимость, руб.
134 Стабилитрон BZV85C5V6	1	1,429	1,429
135 Корпусный постоянный RSF5SS-2,2 Ом ±5%	4	9,750	38,998
136 Дроссель CW68 120К	2	3,478	6,956
137 Колодка штыревая PLD-10	1	10,128	10,128
138 Выводной транзистор КП103И1	1	19,620	19,620
Итого:			3888,44

Таблица 6 – Ведомость об услугах сторонних организаций.

Наименование	Кол-во, шт	Цена, руб	Стоимость, руб.
1 Основной печатный узел	1	5560	5560
Итого:			5560

Таблица 7 – Перечень покупных изделий.

Наименование	Кол-во, шт	Цена, руб	Стоимость, руб.
1 Программный продукт Keil uVision5	1	25800	25800
2 Программатор	1	5000	5000
Итого:			30800

5.7 Расчёт заработной платы

Среднедневная заработная плата (з/п) одного инженера рассчитывается делением среднемесячной заработной платы одного инженера (основной и дополнительной) на среднее число рабочих дней в месяце (22), установленное в законодательном порядке. Данные и расчёты приведены в таблице 8.

Таблица 8 - Трудоёмкость и сметная стоимость работ сетевого графика.

Сотрудники	Занятость, дн.	Оклад Руб./мес.	Средняя з/п, руб.	Выплата, руб.
Руководитель - 1	397	30000	1500	595500
Инженер - 2	598	25000	1250	747500
Лаборант - 1	174	10000	500	87000
Начисление:				429000
Итого:				1859000

5.8 Накладные расходы

На накладные расходы берём 10 % от суммы итоговых стоимостей всех про-

шлых расходов.

Итого: 189925 рублей.

5.9 Расчет итоговых затрат

Затраты на ОКР определяем по смете затрат согласно таблицы 9.

Таблица 9 – Смета затрат на ОКР

Наименование статьи	Сумма, руб.
Материалы и компоненты	3888
Слуги сторонних организаций	5560
Покупные изделия	30800
Заработная плата	1859000
Накладные расходы	189925
Итого:	2089173

5.10 Расчет экономического эффекта от производства изделия

В ходе проектирования было разработано устройство, стоимость разработки модели в сторонней организации 1000 рублей за один час работы. Для опытных специалистов на разработку данной модели при восьми часовом рабочем дне потребовалось бы около 13,5 месяцев работы (405 рабочих дня). Таким образом получаем 3240000 рублей. Так же данную модель нужно тестировать и проверять. Тестирование и устранение неполадок заняло бы в сторонней организации около месяца (22 рабочих дней). Инженеры по тестированию получают 750 рублей за час своего рабочего времени. Рассчитав стоимость получим 132000 рублей. Сложив все затраты для разработки данного проекта в сторонних организациях получаем минимум 3372000 рублей, что значительно превышает наши затраты на разработку данного проекта (2089173 рублей).

6 Охрана труда

Разрабатываемый индуктивный цифро-аналоговый преобразователь перемещения относится к устройствам автоматического регулирования.

6.1 Анализ опасных и вредных производственных факторов, источником которых является спроектированное устройство

Из группы физически опасных и вредных факторов определяем, что разработанное устройство не относится к разряду движущих машин и механизмов, у него отсутствуют подвижные элементы. Изделие не является источником поступления пыли в воздух рабочей зоны, источником повышения температуры воздуха эксплуатируемой зоны, не создает шум и не является источником вибрации, излучает электромагнитные колебания, но частота 40кГц и мощность 10Вт работает на излучение в окружающую среду. Электромагнитное поле создается между элементами конструкции (чувствительными элементами), и не излучается в окружающее пространство.

Так как разработанное устройство относится к электронным средствам, то действия группы химически и биологически опасных и вредных факторов не рассматриваем.

Из всего выше сказанного можно определить, что устройство не является источником опасных и вредных факторов.

Разработанное устройство не является источником ионизирующих излучений и источником распространения вредных веществ в воздухе рабочей зоны. В то же время конструкция устройства обеспечивает проведение мероприятий по дезактивации, дезинфекции, дегазации изделия.

6.2 Анализ условий эксплуатации проектируемого устройства

Разрабатываемое устройство предназначено для эксплуатации в различных технологических установках.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

Исходя из данных технического задания, конструкция устройства разработана таким образом, чтобы обеспечивать нормальное функционирование устройства при температуре от 0 °С до плюс 65 °С и влажности не более 85%.

6.3 Обеспечение безопасности устройства

Конструкция аппаратуры исключает возможность попадания электрического напряжения на наружные металлические части.

Выполнение всех рабочих функций происходит при: однополярном питающем напряжении плюс 24В источником напряжения, двухполярном питающем напряжении плюс-минус 15В. Согласно ГОСТ 12.0.007.0-75 по способу защиты человека от поражения электрическим током, изделие относится к классу 3, то есть применение дополнительных мер по обеспечению электробезопасности к данному устройству не предъявляется.

Пожарная безопасность изделия и его элементов обеспечивается исключением использования в конструкции изделия легковоспламеняющихся материалов в соответствии с ГОСТ 12.1.004-91 ССБТ.

Корпус устройства разработан таким образом, что кромки углов закруглены и не могут быть источником физических повреждений.

Крепление аппаратуры на объекте исключает ее перемещение во всех направлениях от вибраций, возникающих в условиях эксплуатации.

Конструкция устройства обеспечивает нормальное функционирование устройства при температуре от 0 °С до плюс 65 °С и влажности не более 85%.

Требования соблюдаются исключением возможности неправильной установки и сочленения разъемных блоков. Все подключения к датчику производить при отключенном напряжении питания. Ввод проводов в корпус устройства осуществляется через изоляционные детали. При этом исключается возможность повреждения проводов и их изоляции в процессе монтажа и эксплуатации изделия.

Корпус изделия соединён с основными частями изделия в единую конструкцию, закрывают опасную зону и снимаются только при помощи инструмента. Ввод

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

проводников в корпус устройства осуществляется через изоляционные детали. Конструкция и материал устройства исключает возможность случайного прикосновения к токоведущим частям, электрических перекрытий, а также замыкания проводников на корпус и накоротко.

Электромагнитное поле, создаваемое между двумя чувствительными элементами расположены на расстоянии 50 мм друг от друга, что допускает только локальное воздействие на организм человека. Согласно СанПиН 2.2.4.3359-16 пункту 8, предельно допустимый уровень (ПДУ) электрического и магнитного полей рассчитывается по формулам:

$$E = \sqrt{\frac{\text{ЭЭ}_E}{T}} = \sqrt{\frac{20000}{12}} = 40,8 \frac{\text{В}}{\text{м}}; \quad H = \sqrt{\frac{\text{ЭЭ}_H}{T}} = \sqrt{\frac{200}{12}} = 4,08 \frac{\text{А}}{\text{м}},$$

где, E-напряжённость электрического поля,

H-напряжённость магнитного поля,

T-время воздействия за смену,

ЭЭ-энергетическая экспозиция.

6.4 Требования к эргономичности

При разработке аппаратуры обеспечивается удобство выполнения функциональных обязанностей, достаточные физические (в том числе зрительные и слуховые) связи между оператором и аппаратурой.

Устройство оборудовано светодиодом, сигнализирующим о подключении к питанию, так же надписью с названием и серийным номером. Органы управления на данном устройстве отсутствуют.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

ЗАКЛЮЧЕНИЕ

Разработанное устройство индуктивный цифро-аналоговый преобразователь перемещения позволяет осуществлять управление исполнительным устройством.

С использованием в качестве блока управления микроконтроллера архитектуры ARM Cortex-M0 LPC822M101JDH20 и программного продукта для этого процессора обеспечивается высокая надёжность проводимых проверок, т.е. получения достоверных результатов.

Конструктивные и схемотехнические решения выполнены с учётом современных направлений в разработках радиоэлектронной аппаратуры. Компактная конструкция блока (размеры) позволяет легко разместить устройство используя приспособления для крепежа в удобном месте.

Разработан комплект конструкторской документации на изготовление устройства.

С помощью разработанного устройства управления исполнительным устройством человек полностью исключается из процесса регулировки работающего оборудования. Оператору необходимо только следить за критическими ситуациями.

Разработаны мероприятия по охране труда при эксплуатации изделия.

Рассчитаны экономические показатели. Так, себестоимость одного комплекта изделия при мелкосерийном производстве составляет 50300 руб., а срок окупаемости разработки около 0,7 года с начала разработки системы.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		46

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Александров К.К., Кузьмина Е.Г. Электротехнические чертежи и схемы, 1990.-288с.
2. Хоровец П., Хилл У. Искусство схемотехники: В 3-х томах: Т.1. Пер. с англ. М.: Мир,1993.-413с.
3. Хоровец П., Хилл У. Искусство схемотехники: В 3-х томах: Т.2. Пер. с англ. М.: Мир,1993.-452с.
4. Справочная информация и технические данные расположенные в сети «Internet» по адресу www.microchip.com
5. Нефедов А.В. Интегральные микросхемы и их зарубежные аналоги: Справочник. Т.10.-М.: ИП РадиоСофт,2000.-544с.
6. Разработка и оформление конструкторской документации радиоэлектронной аппаратуры: Справочник. Под ред. Романычевой Э.Т.-М.: Радио и связь, 1989.-448 с.
7. Гелль П.П., Иванов-Есипович Н.К. Конструирование и микроминиатюризация радиоэлектронной аппаратуры.: Учебник для вузов.-Л.:Энергоиздат.1984.-536с.
8. Зайцев А. А. и др. Полупроводниковые приборы. Транзисторы малой мощности: Справочник - 2-е изд. - М: Радио и связь,1995.-384с.
9. Справочная книга радиолюбителя конструктора: В2-х книгах. Кн.2. Под ред. чистякова. М.: Радио и связь, 1993.-336с.
10. Сетевые методы планирования и управления: Методические указания к курсовому проекту для студентов приборостроительного факультета / Составители: В.С. Зинкевич, Л.А. Баев, Н.П. Мешковой. - Челябинск Изд. ЮУрГУ, 1998. - 22с.
11. Организационно-экономический раздел дипломного проекта конструкторского направления: Учеб. пособие для студентов приборостроительного фа-

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		47

культета / Мешковой Н.П., Закиров Р.Ш., Зинкевич В.С., Попов Г.С.; Под ред. Н.П. Мешкового. - Челябинск: ЧГТУ, 1990. - 53 с.

12. Методические указания по дипломному проектированию для студентов приборостроительного факультета. (Раздел «Безопасность жизнедеятельности») / Составители: Н.В. Глотова, И. С. Окраинская; Под ред. А.И. Сидорова. - Челябинск: ЮУрГУ, 1999. - 16 с.

13. Безопасность жизнедеятельности: Учебник для вузов / С. В. Белов, А. В. Ильницкая, А. Ф. Козьяков и др.; Под общ. Ред. С. В. Белова. - М.: Высшая школа, 1999.

14. Усатенко С.Т., Каченюк Т.К., Терехова М.В. Выполнение электрических схем по ЕСКД. Справочник. - 2-е изд., перераб. и доп. - М.: Издательство стандартов, 1992. - 316 с.

15. Стандарт предприятия: Курсовые и дипломные проекты. Общие требования к оформлению учебной документации. СТП ЧГТУ 04 - 96. - Челябинск, ЧПИ, 1987. - 2ч.

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
						48
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ А

Листинг программного кода для дипломного проекта «индуктивный цифро-аналоговый преобразователь перемещения»

Файл VTIYU1820.c

```
#include "LPC82x.h"
#include <stdlib.h>
#include "MyDefs.h"
#include "GPIO.h"
#ifdef RS232
#include "Serial.h"
#endif
#include "TLE4997.h"
#include "Hex.h"
#include "ADC.h"
#include "ACT.h"
#include "I2C.h"
#include "VTIYU1820.h"
//-----
#define SECTOR_SIZE 1024
volatile uint32_t ms_ticks;
uint32_t max_adc, min_adc, phase_tle4997;
uint32_t timer_max_min;
bool bl_tuned, bl_good_conf;
uint32_t timer_test_tx;
uint32_t timer_skip_adc;
bool bl_skip_adc;
uint32_t timer_DisLevelToogle;
//-----
uint32_t cr;
#ifdef RS232
uint32_t phase_rx_from_pc_ascii;
#endif
#endif
```

					ИОУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

```

uint16_t tle4997_adr;
uint16_t tle4997_data;
uint32_t timer_repower_tle4997;
//-----
void InitAll(void);
void InitVars(void);
uint32_t GetContCnt1ms(void);
int8_t ShiftDig2ByteTLE4997(uint8_t a_dig, uint16_t *bt);
uint16_t ReadTLE4997(uint8_t adr);
#ifdef RS232
void RxFromPcAscii_proc(void);
#endif
void WorkWithADCdata_proc(void);
void TLE4997(void);
int32_t GetLockedStep(uint32_t val1, uint32_t val2);
void Restart100msAdcPeriod(void);
void EnAllInts(void);
void RePowerTLE4997(void);
//-----
void SysTick_Handler(void)
{
    ms_ticks++;
}
//-----
int main(void)
{
    #define CRP1 0x12345678
    #define CRP2 0x87654321
    #define CRP3 0x43218765
    uint32_t tmp_32;
    InitAll();
    __enable_irq();
    tmp_32=(uint32_t>(&main);
    if(tmp_32>0x1000)
    {
        uint8_t* buf_sec0=malloc(SECTOR_SIZE);

```



```

const uint8_t *ptr_sec0=0;
if(buf_sec0!=NULL)
{
#ifdef RS232
    SendHexDword((uint32_t)buf_sec0);
    SendCr();
#endif
    for(uint32_t i=0; i<SECTOR_SIZE; i++)
        buf_sec0[i]=ptr_sec0[i];
    for(int i=3; i>=0; i--)
    {
        tmp_32<<=8;
        tmp_32|=buf_sec0[0x2fc+i];
    }
#ifdef RS232
    SendHexDword(tmp_32);
    SendCr();
#endif
    if(tmp_32==CRP2)
    {
        tmp_32=CRP3;
        for(int i=0; i<4; i++)
        {
            buf_sec0[0x2fc+i]=tmp_32;
            tmp_32>>=8;
        }
        __disable_irq();
        tmp_32=1;
        for (; tmp_32<5; tmp_32++)
        {
            if (u32IAP_PrepareSectors(0, 0)!=IAP_STA_CMD_SUCCESS) continue;
            if (u32IAP_EraseSectors(0, 0)!=IAP_STA_CMD_SUCCESS) continue;
            if (u32IAP_PrepareSectors(0, 0)!=IAP_STA_CMD_SUCCESS) continue;
            if (u32IAP_CopyRAMToFlash(0*SECTOR_SIZE, (uint32_t)buf_sec0, SECTOR_SIZE)!=IAP_STA_CMD_SUCCESS) continue;

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

```

        if (u32IAP_Compare(0*SECTOR_SIZE, (uint32_t)buf_sec0, SECTOR_SIZE)!=IAP_STA_CMD_SUCCESS) continue;
        break;
    }
    __enable_irq();
#ifdef RS232
    SendHexByte(tmp_32);
    SendCr();
#endif
    if(tmp_32>=4)
        u32IAP_EraseSectors(0, (16384/SECTOR_SIZE)-1);
    }
    free(buf_sec0);
}
}
#ifdef EN_WDT
    InitWDT();
#endif
#ifdef RS232
    #ifndef DIV_4
        SendString("Ã×Â (ÂÒÈÐ 7072) V-1.00. ©Òâêî.\r\n");
    #else
        SendString("Ã×Â (ÂÒÈÐ 7072.4) V-1.00. ©Òâêî.\r\n");
    #endif
#endif
do
{
    tmp_32=ReadADC();
}while (tmp_32==UM1);
max_adc=0;
min_adc=0xffff;

while(true)
{
#ifdef EN_WDT
    ResetWDT();

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		52

```

#endif
#ifdef RS232
    RxFromPcAscii_proc();
#endif
    WorkWithADCdata_proc();
    I2C_proc();
    TLE4997Timers_proc();
}
}
//-----
void InitAll(void)
{
    SystemCoreClockUpdate();

    InitVars();
    InitGpio();
    SysTick_Config(SystemCoreClock/1000);
    NVIC_SetPriority(SysTick_IRQn, 1);
    RePowerTLE4997();
#ifdef RS232
    InitUART0(9600);
#else
    LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 7);
    LPC_SWM->PINASSIGN0 |= 0xffff;
    LPC_SYSCON->SYSAHBCLKCTRL &= ~(1 << 7);
#endif
    InitADC();
    InitACMP();
    InitSCT();
    InitMRT();
    InitI2C();
    InitCRC();
    bl_good_conf=ReadConf();
    Restart100msAdcPeriod();
}
//-----

```

```

void InitVars(void)
{
    ms_ticks=0;
    cr=false;
    timer_test_tx=0;
}
//-----
uint32_t GetContCnt1ms(void)
{
    return ms_ticks;
}
//-----
#ifdef RS232
void RxFromPcAscii_proc(void)
{
    uint16_t w_tmp;
    int16_t tmp=GetByteFromRxPC();
    if (tmp==-1) return;
    switch(tmp)
    {
        case 'R':
        case 'r':
            cr=false;
            tle4997_adr=0;
            phase_rx_from_pc_ascii=1;
            return;
        case 'W':
        case 'w':
            cr=false;
            tle4997_adr=tle4997_data=0;
            phase_rx_from_pc_ascii=4;
            return;
    }
    switch(phase_rx_from_pc_ascii)
    {
        case 1:

```

```

if (ShiftDig2ByteTLE4997(tmp, &tle4997_adr))
{
if(tle4997_adr!=0xff)
{
SendHexByte(tle4997_adr);
SendByteToPc(':');
w_tmp=ReadTLE4997(tle4997_adr);
SendHexWord(w_tmp);
SendCr();
}
else
{
SendFrame(PrepCmd(0, 0), false);
CloseDoutTLE4997();
ClrClkTLE4997();
SendByteToPc('Q');
SendByteToPc('.');
SendCr();
}
}
break;
case 2:
case 3:
phase_rx_from_pc_ascii=0;
break;
case 4:
ShiftDig2ByteTLE4997(tmp, &tle4997_adr);
break;
case 5:
WriteAndShowTLE4997();
break;
case 6:
if (ShiftDig2ByteTLE4997(tmp, &tle4997_data))
WriteAndShowTLE4997();
break;
}

```

```

}
#endif
//-----
void WorkWithADCdata_proc(void)
{
    uint32_t tmp_32, dwPeri;
    int16_t data;
    if(bl_skip_adc)
    {
        if((GetContCnt1ms()-timer_skip_adc)<5)
            return;
        bl_skip_adc=false;
    }
    data=ReadADC();
    if(data== -1) return;

    if(data>max_adc)
    {
        max_adc=data;
        Restart100msAdcPeriod();
        return;
    }
    if(data<min_adc)
    {
        min_adc=data;
        Restart100msAdcPeriod();
        return;
    }

    tmp_32=GetContCnt1ms()-timer_max_min;
    if(phase_tle4997<4)
        dwPeri=10;
    else
        dwPeri=100;
    if(tmp_32>=dwPeri)
    {

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

```

TLE4997();
Restart100msAdcPeriod();
max_adc=0;
min_adc=0xffff;
}
}
//-----
void TLE4997(void)
{
int32_t nTmp32;
uint32_t tmp_32;
static int32_t StepOffset;
static uint32_t LowLevelTLE4997rangeMin, HighLevelTLE4997rangeMax, LowLevelT-
LE4997Middlerange, HighLevelTLE4997Middlerange, LowLevelTLE4997rangeGood, HighLevelT-
LE4997rangeGood;
switch (phase_tle4997)
{
case 0:
PowerOnTLE4997();
phase_tle4997++;
break;
case 1:
if(!TestTLE4997Present(0x8f))
break;
StopTLE4997();
SetGainOffset(GAIN_ADR, 0, false);
SetGainOffset(OFFSET_ADR, -400, false);
WriteTLE4997(0x16, 0x7d51);
ReStartTLE4997();
phase_tle4997++;
break;
case 2:
LowLevelTLE4997rangeMin=min_adc;
SetGainOffset(OFFSET_ADR, 399, true);
phase_tle4997++;
break;
}
}

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		57

```

case 3:
    HighLevelTLE4997rangeMax=max_adc;
    tmp_32=(HighLevelTLE4997rangeMax-LowLevelTLE4997rangeMin)/20;
    HighLevelTLE4997rangeMax-=(tmp_32<<2);
    LowLevelTLE4997rangeMin+=(tmp_32<<2);
    HighLevelTLE4997rangeGood=HighLevelTLE4997rangeMax-(tmp_32<<1);
    LowLevelTLE4997rangeGood=LowLevelTLE4997rangeMin+(tmp_32<<1);
    LowLevelTLE4997Middlerange=1361-tmp_32;
    HighLevelTLE4997Middlerange=1361+tmp_32;
    StepOffset=-200;
    if(bl_good_conf)
    {
        StopTLE4997();
        SetGainOffset(GAIN_ADR, GetGain(), false);
        SetGainOffset(OFFSET_ADR, GetOffset(), false);
        ReStartTLE4997();
        EnAllInts();
    }
    else
        SetGainOffset(GAIN_ADR, 399, true);
    bl_tuned=false;
    DisLevelToggle();
    phase_tle4997++;
    break;
case 4:
    if(GetblEnLevelToggle())
    {
        if((max_adc-min_adc)<100 || (GetContCnt1ms()-timer_DisLevelToggle)>300)
            DisLevelToggle();
    }
    nTmp32=(max_adc+min_adc)/2;
    if(nTmp32>HighLevelTLE4997Middlerange)
    {
        StepOffset=-GetLockedStep(nTmp32, HighLevelTLE4997Middlerange);
        if(StepOffset>0)
        {

```



```

    }
}
break;
}
else
{
    StepOffset=GetLockedStep(LowLevelTLE4997Middlerange, nTmp32);
    if(StepOffset<0)
    {
        StepOffset=-StepOffset;
        if(StepOffset>=2)
            StepOffset/=2;
    }
}
}
SetGainOffset(OFFSET_ADR, GetCurOffset()+StepOffset, true);
break;
}
}
//-----
int32_t GetLockedStep(uint32_t val1, uint32_t val2)
{
    uint32_t Tmp32_2;
    uint32_t tmp_32=(val1-val2);
    Tmp32_2=tmp_32>>4;
    if (Tmp32_2==0)
        Tmp32_2=1;
    return Tmp32_2;
}
//-----
void Restart100msAdcPeriod(void)
{
    timer_max_min=GetContCnt1ms();
}
//-----
void EnAllInts(void)

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

```

{
  if(!bl_tuned)
  {
    EnableSCTint();
    EnableMRTint();
    bl_tuned=true;
  }
}
//-----
void RePowerTLE4997(void)
{
  PowerOffTLE4997();
  timer_repower_tle4997=GetContCnt1ms();
  phase_tle4997=0;
  InitTLE4997();
}
//-----
uint16_t GetTLE4997adr(void)
{
  return tle4997_adr;
}
//-----
uint16_t GetTLE4997data(void)
{
  return tle4997_adr;
}
//-----
void SetCr(uint32_t val)
{
  cr=val;
}
//-----
uint32_t GetCr(void)
{
  return cr;
}

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

```

//-----
#ifdef RS232
void SetPhase_RX_from_PC_ASCII(uint32_t val)
{
    phase_rx_from_pc_ascii=val;
}
//-----
void AddToPhase_RX_from_PC_ASCII(uint32_t val)
{
    phase_rx_from_pc_ascii+=val;
}
//-----
#endif

void InitSkipADC(void)
{
    bl_skip_adc=true;
    timer_skip_adc=GetContCnt1ms();
}
//-----

bool TestTLE4997Present(uint32_t adr)
{
    uint32_t tmp_32=ReadTLE4997(adr) & (~0x200);
    if (tmp_32==0xF93D)
        return true;
    PowerOffTLE4997();
    phase_tle4997=0;
    return false;
}
//-----

bool TestEn2EEProm(void)
{
    return GetblEnLevelToggle() && bl_tuned;
}
//-----

void ResTimeOutDisLevelToogle(void)
{

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		62

```

timer_DisLevelToggle=GetContCnt1ms();
}
//-----

gpio.c

#include "LPC82x.h"
#include "MyDefs.h"
#include "GPIO.h"
//-----

#define DIN_TLE4997_MASK (1<<1)
#define OUT_INV_MASK (1<<4)
#define NMOSI_MASK (1<<8)
#define MOSI_MASK (1<<9)
#define SH_DN_MASK (1<<13)
#define OUT_MASK (1<<14)
#define CLK_MASK (1<<15)

bool bl_en_level_toggle, bl_high;
//-----

void InitGpio (void)
{
    LPC_SYSCON->SYSAHBCLKCTRL |= ((1 << 6) | (1 << 7) | (1 << 18));
// enable GPIO and SWM clocks and IOCON
    LPC_SWM->PINENABLE0 |= ((3<<6) | (1<<1) | (1<<9) | (1<<16) | (1<<23) | (1<<2) |
(1<<15));
    LPC_SWM->PINENABLE0 &= ~((1<<3) | (1<<22) | (1<<11) | (1<<12));
    LPC_IOCON->PIO0_0=1<<7;
    LPC_IOCON->PIO0_17=1<<7;
    LPC_IOCON->PIO0_23=1<<7;
    LPC_IOCON->PIO0_11=1<<7;
    LPC_IOCON->PIO0_10=1<<7;
    LPC_IOCON->PIO0_14=1<<7;
#ifdef RS232
    LPC_IOCON->PIO0_4=1<<7;
#endif
    LPC_SYSCON->SYSAHBCLKCTRL &= ~((1 << 7) | (1 << 18));
// disable SWM clock and IOCON

```

```

#ifdef RS232
    LPC_GPIO_PORT->DIR0 = (OUT_MASK | NMOSI_MASK | MOSI_MASK | CLK_MASK |
SH_DN_MASK);          // configure GPIO as output
    LPC_GPIO_PORT->CLR0 = (OUT_MASK | NMOSI_MASK | MOSI_MASK |
CLK_MASK);           // configure GPIO as low
#else
    LPC_GPIO_PORT->DIR0 = (OUT_MASK | OUT_INV_MASK | NMOSI_MASK |
MOSI_MASK | CLK_MASK | SH_DN_MASK);    // configure GPIO as output
    LPC_GPIO_PORT->CLR0 = (OUT_MASK | OUT_INV_MASK | NMOSI_MASK |
MOSI_MASK | CLK_MASK);           // configure GPIO as low
#endif

CloseDoutTLE4997();
ClrCikTLE4997();
bl_high=false;
EnLevelToogle();
OutPutFreqHigh();
DisLevelToogle();
}
//-----
void SetDoutTLE4997(void)
{
    __disable_irq();
    LPC_GPIO_PORT->CLR0=MOSI_MASK | NMOSI_MASK;
    DelayGPIO();
    LPC_GPIO_PORT->SET0=MOSI_MASK;
    __enable_irq();
}
//-----
void ClrDoutTLE4997(void)
{
    __disable_irq();
    LPC_GPIO_PORT->CLR0=MOSI_MASK | NMOSI_MASK;
    DelayGPIO();
    LPC_GPIO_PORT->SET0=NMOSI_MASK;
    __enable_irq();
}
//-----

```

Изм.	Лист	№ докум.	Подпись	Дата

```

void CloseDoutTLE4997(void)
{
    LPC_GPIO_PORT->CLR0=MOSI_MASK | NMOSI_MASK;
}
//-----

void SetClkTLE4997(void)
{
    LPC_GPIO_PORT->SET0=CLK_MASK;
}
//-----

void ClrClkTLE4997(void)
{
    LPC_GPIO_PORT->CLR0=CLK_MASK;
}
//-----

int32_t GetMiso(void)
{
    if (LPC_GPIO_PORT->PIN0 & DIN_TLE4997_MASK)
        return false;
    else
        return true;
}
//-----

void DelayGPIO(void)
{
    __NOP();
    __NOP();
}
//-----

void EnLevelToogle(void)
{
    bl_en_level_toggle=true;
}
//-----

void DisLevelToogle(void)
{

```

```

    bl_en_level_toggle=false;
}
//-----
bool GetblEnLevelToggle(void)
{
    return bl_en_level_toggle;
}
//-----
void PowerOnTLE4997(void)
{
    LPC_GPIO_PORT->CLR0=SH_DN_MASK;
}
//-----
void PowerOffTLE4997(void)
{
    LPC_GPIO_PORT->SET0=SH_DN_MASK;
}
//-----
void OutPutFreqLow(void)
{
    bl_high=false;
    if(bl_en_level_toggle)
    {
        LPC_GPIO_PORT->SET0=OUT_MASK;
#ifdef RS232
        LPC_GPIO_PORT->CLR0=OUT_INV_MASK;
#endif
    }
}
//-----
void OutPutFreqHigh(void)
{
    bl_high=true;
    LPC_GPIO_PORT->CLR0=OUT_MASK;
#ifdef RS232
    LPC_GPIO_PORT->SET0=OUT_INV_MASK;
#endif
}

```



```

#endif
}
//-----
void ToggleOutPutFreq(void)
{
    if (bl_high)
        OutPutFreqLow();
    else
        OutPutFreqHigh();
}
//-----

```

Serial.c

```

#ifdef RS232
#include "LPC82x.h"
#include "MyDefs.h"
#include "Serial.h"
#include "WDT.h"
//-----
#define SIZEOF_TX_BUF 64
#define SIZEOF_RX_BUF 64
volatile uint32_t tail_tx_buf, head_tx_buf, tail_rx_buf, head_rx_buf;
volatile bool bl_sending;
uint8_t tx_buf[SIZEOF_TX_BUF];
uint8_t rx_buf[SIZEOF_RX_BUF];
//-----
void InitUART0(uint32_t baud_rate)
{
    uint32_t tmp;
    uint32_t clk;
    bl_sending=false;
    head_tx_buf=tail_tx_buf=tail_rx_buf=head_rx_buf=0;
    LPC_SYSCON->SYSAHBCLKCTRL |= ((1 << 7) | (1<<14)); // enable
SWM and UART0 clocks
    tmp=LPC_SWM->PINASSIGN0 & ~0xffff;

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		67

```

LPC_SWM->PINASSIGN0=tmp | (4<<0) | (0<<8);
LPC_SYSCON->SYSAHBCLKCTRL &= ~(1 << 7); // disable
SWM clock
clk = SystemCoreClock;

LPC_SYSCON->UARTCLKDIV = 1;
LPC_SYSCON->PRESETCTRL &= ~(1<<3); // reset uart0
LPC_SYSCON->PRESETCTRL |= (1<<3); // release uart0
LPC_SYSCON->UARTFRGDIV = 0xFF;
LPC_SYSCON->UARTFRGMULT = 4;
LPC_USART0->CFG = (1<<2);
LPC_USART0->BRG = clk/16/ baud_rate-1;
LPC_USART0->INTENSET=(1<<0);

NVIC_SetPriority(UART0_IRQn, 2);
NVIC_EnableIRQ(UART0_IRQn);

LPC_USART0->CFG |= 1<<0; // enable uart
}
//-----
void SendByteToPc(uint8_t bt)
{
    volatile uint32_t tmp;
    __disable_irq();
    if(!bl_sending)
    {
        bl_sending=true;
        LPC_USART0->TXDAT=bt;
        LPC_USART0->INTENSET=(1<<3);
        __enable_irq();
    }
    else
    {
#ifdef EN_WDT
        ResetWDT();
#endif
    }
}

```

```

while(((head_tx_buf-tail_tx_buf) & (SIZEOF_TX_BUF-1))>SIZEOF_TX_BUF*3/4)
{
    __enable_irq();
    __NOP();
    __disable_irq();
};
tmp=head_tx_buf;
tx_buf[tmp++]=bt;
head_tx_buf=tmp & (SIZEOF_TX_BUF-1);
__enable_irq();
}
}
//-----
void UART0_IRQHandler(void)
{
    if (LPC_USART0->INTSTAT & (1<<0))
    {
        rx_buf[head_rx_buf++]=LPC_USART0->RXDAT;
        head_rx_buf &= (SIZEOF_RX_BUF-1);
    }
    if ((LPC_USART0->INTSTAT & (1<<3)) && bl_sending)
    {
        if(head_tx_buf!=tail_tx_buf)
        {
            LPC_USART0->TXDAT=tx_buf[tail_tx_buf++];
            tail_tx_buf &= (SIZEOF_TX_BUF-1);
        }
        else
        {
            bl_sending=false;
            LPC_USART0->INTENCLR=(1<<3);
        }
    }
}
//-----
int16_t GetByteFromRxPC(void)

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		69

```

{
    uint8_t uTmp8;
    if(head_rx_buf==tail_rx_buf)
        return -1;
    uTmp8=rx_buf[tail_rx_buf++];
    tail_rx_buf &= (SIZEOF_RX_BUF-1);
    return uTmp8;
}
//-----
#endif
//-----

TLE4997.c

#include "GPIO.h"
#include "TLE4997.h"
#include "Hex.h"
#ifdef RS232
#include "Serial.h"
#endif
#include "ACMP.h"
#include "SCT.h"
#include "ADC.h"
#include "I2C.h"
#include "VTIYU1820.h"
//-----
#define READ_DATA_FROM_GIVEN_ADDRESS_CMD4997 1
#define WRITE_DATA_TO_GIVEN_ADDRESS_CMD4997 9
//-----
typedef struct
{
    int32_t val;
    uint32_t timer;
    uint32_t timeout;
    bool bl_stored;
} aParament;

```

```

//-----
aParament GainPar, OffsPar;
//-----

void DelayClkTLE4997(void);
#ifdef RS232
int8_t ShiftDig2ByteTLE4997(uint8_t a_dig, uint16_t *bt);
#endif
uint8_t CountParity(uint8_t bt);
uint32_t GetFrame(void);
uint32_t PrepData(uint16_t w_data);
void SetParAndResTimer(aParament *str, int32_t par, uint32_t adr, bool bl_stop_start);
void TestTimerAndStore(aParament *str);
//-----

void InitTLE4997(void)
{
    SetCr(false);
#ifdef RS232
    SetPhase_RX_from_PC_ASCII(0);
#endif
    CloseDoutTLE4997();
    OffsPar.bl_stored=true;
    GainPar.bl_stored=true;
    GainPar.val=-1000;
    OffsPar.val=-1000;
    GainPar.timeout=57500;
    OffsPar.timeout=62500;
}
//-----

void SendFrame(uint32_t dw_frame, bool bl_first)
{
    if(!bl_first)
    {
        SetClkTLE4997();
        DelayClkTLE4997();
        ClrClkTLE4997();
    }
}

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

```

for(uint32_t i=0; i<21; i++)
{
    if(dw_frame & 1)
        SetDoutTLE4997();
    else
        ClrDoutTLE4997();
    DelayClkTLE4997();
    SetClkTLE4997();
    DelayClkTLE4997();
    ClrClkTLE4997();
    dw_frame>>=1;
}
InitSkipADC();
}
//-----
void DelayClkTLE4997(void)
{
    for(uint32_t i=0; i<24; i++)
        DelayGPIO();
}
//-----
uint32_t PrepCmd(uint8_t adr, uint8_t Cmd)
{
    uint8_t bt0=Cmd<<1 | 1;
    uint8_t bt1=(adr & 0x3f) <<1 | 1;
    uint8_t bt2=3<<3;
    bt2^=CountParity(bt2);
    bt2^=CountParity(bt1);
    bt2^=CountParity(bt0);
    return (uint32_t)bt2<<16 | bt1<<8 | bt0;;
}
//-----
uint16_t ReadTLE4997(uint8_t adr)
{
    SendFrame(PrepCmd(adr, READ_DATA_FROM_GIVEN_ADDRESS_CMD4997), (adr &
0x80)>0);
}

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		72

```

CloseDoutTLE4997();
adr&=7;
uint32_t dw_result=GetFrame();
if(!(dw_result & 1)) return (uint16_t)-1;
if(!(dw_result & 1<<20)) return (uint16_t)-1;
if(!((dw_result>>17 & 7) == adr)) return (uint16_t)-1;
return dw_result>>1 & 0xffff;
}
//-----
#ifdef RS232
int8_t ShiftDig2ByteTLE4997(uint8_t a_dig, uint16_t *bt)
{
switch(a_dig)
{
case ',':
SetCr(false);
AddToPhase_RX_from_PC_ASCII(2);
break;
case 10:
case 13:
case ';':
if(!GetCr())
{
SetCr(true);
AddToPhase_RX_from_PC_ASCII(1);
return true;
}
break;
default:
{
SetCr(false);
if(a_dig>='a')
a_dig&=0xdf;
a_dig-=0x30;
if(a_dig>=0x80) break;
if(a_dig>9)

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		73

```

    a_dig-=7;
    if(a_dig<=0xf)
    {
        (*bt)<<=4;
        (*bt)|=a_dig;
    }
}
}
return false;
}
#endif

//-----
uint8_t CountParity(uint8_t bt)
{
    uint8_t parity=0;
    uint8_t bt_mask=1;
    for(uint8_t i=0; i<8; i++)
    {
        if (bt & bt_mask)
        {
            if(i & 1)
                parity ^= 2;
            else
                parity ^= 4;
        }
        bt_mask <<= 1;
    }
    return parity;
}

//-----
uint32_t GetFrame(void)
{
    uint32_t dw_result=0;
    for (uint32_t i=0; i<21; i++)
    {
        SetCikTLE4997();
    }
}

```

Изм.	Лист	№ докум.	Подпись	Дата


```

DelayClkTLE4997();
ClrClkTLE4997();
if(GetMiso())
    dw_result |= 1<<21;
dw_result>>=1;
DelayClkTLE4997();
}
SetClkTLE4997();
DelayClkTLE4997();
ClrClkTLE4997();
DelayClkTLE4997();
InitSkipADC();
return dw_result;
}
//-----
#ifdef RS232
void WriteAndShowTLE4997(void)
{
    uint16_t adr, data;
    adr=GetTLE4997adr();
    data=GetTLE4997data();
    WriteTLE4997(adr, data);
    SetPhase_RX_from_PC_ASCII(0);
    SendHexByte(adr);
    SendByteToPc('=');
    SendHexWord(data);
    SendCr();
}
#endif
//-----
void WriteTLE4997(uint8_t adr, uint16_t data)
{
    SendFrame(PrepCmd(adr, WRITE_DATA_TO_GIVEN_ADDRESS_CMD4997), false);
    SendFrame(PrepData(data), false);
    CloseDoutTLE4997();
}

```

```

//-----
uint32_t PrepData(uint16_t w_data)
{
    uint32_t dw_result=w_data<<1 | 1;
    uint8_t bt2=dw_result>>16 | 1<<4;
    bt2^=CountParity(bt2);
    bt2^=CountParity(dw_result>>8);
    bt2^=CountParity(dw_result);
    dw_result |= (uint32_t)bt2<<16;
    return dw_result;
}
//-----
void StopTLE4997(void)
{
    WriteTLE4997(0x21, 0x640);
}
//-----
void ReStartTLE4997(void)
{
    WriteTLE4997(0x21, 0x240);
}
//-----
void SetGainOffset(uint32_t adr, int32_t gain, bool bl_stop_start)
{
    if(gain<-400)
        gain=-400;
    else
    {
        if(gain>399)
            gain=399;
    }
    if(adr==GAIN_ADR)
        SetParAndResTimer(&GainPar, gain, adr, bl_stop_start);
    if(adr==OFFSET_ADR)
        SetParAndResTimer(&OffsPar, gain, adr, bl_stop_start);
}

```

Изм.	Лист	№ докум.	Подпись	Дата

```

//-----
int32_t GetCurGain(void)
{
    return GainPar.val;
}
//-----
int32_t GetCurOffset(void)
{
    return OffsPar.val;
}
//-----
void SetParAndResTimer(aParamant *str, int32_t par, uint32_t adr, bool bl_stop_start)
{
    static int32_t prev;
    static uint32_t cnt=0;
    str->timer=GetContCnt1ms();
    str->bl_stored=false;
    DisSCTint();
    if(str->val==par)
    {
        if(prev!=par)
        {
            prev=par;
            cnt=0;
            return;
        }
        else
        {
            if(++cnt>=3)
            {
                TestTLE4997Present(0xf);
                par=-par;
                par/=2;
                prev=par;
                cnt=0;
            }
        }
    }
}

```

```

    else
        return;
    }
}
cnt=0;
str->val=par;
if(bl_stop_start)
    StopTLE4997();
WriteTLE4997(adr, par*4096/100+16384);
if(bl_stop_start)
    ReStartTLE4997();
EnSCTint();
}
//-----
void TLE4997Timers_proc(void)
{
    TestTimerAndStore(&OffsPar);
    TestTimerAndStore(&GainPar);
}
//-----
void TestTimerAndStore(aParament *str)
{
    if(!str->bl_stored && ((GetContCnt1ms()-str->timer)>=str->timeout))
    {
        str->bl_stored=true;
        if(str==&OffsPar)
            SetOffset(str->val);
        else
            SetGain(str->val);
    }
}
//-----

HEX.c

#ifdef RS232

```

```

#include "Hex.h"
#include "Serial.h"
//-----
void SendHexByte(uint8_t bt)
{
    SendTetr(bt>>4);
    SendTetr(bt);
}
//-----
void SendHexWord(uint16_t wrd)
{
    SendHexByte(wrd>>8);
    SendHexByte(wrd);
}
//-----
void SendTetr(uint8_t tetr)
{
    tetr=(tetr & 0xf)+0x30;
    if(tetr>0x39)
        tetr+=7;
    SendByteToPc(tetr);
}
//-----
void SendString(const int8_t *ptr)
{
    while(*ptr!=0) SendByteToPc(*(ptr++));
}
//-----
void SendCr(void)
{
    SendByteToPc('\r');
    SendByteToPc('\n');
}
//-----
void SendHexDword(uint32_t dwrд)
{

```

Изм.	Лист	№ докум.	Подпись	Дата

```

SendHexWord(dwrд>>16);
SendHexWord(dwrд);
}
//-----
#endif

ADC.c

#include "LPC82x.h"
#include "MyDefs.h"
//-----
void InitADC(void)
{
    LPC_SYSCON->PDRUNCFG &= ~(1<<4);
    LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 24);
    LPC_ADC->TRM=0;
    LPC_ADC->CTRL=(SystemCoreClock/500000-1) | (1<<30);
    while(LPC_ADC->CTRL & (1<<30));
    LPC_ADC->CTRL=0;
    LPC_ADC->SEQA_CTRL=(1<<9) | (1UL<<18) | (1UL<<26) | (1UL<<31);
}
//-----
int32_t ReadADC(void)
{
    uint32_t tmp_32;
    uint32_t adc_dat=LPC_ADC->DAT9;
    if(adc_dat & (1UL<<31))
    {
        tmp_32=(adc_dat>>4) & 0xfff;
        LPC_ADC->SEQA_CTRL=(1<<9) | (1UL<<18) | (1UL<<26) | (1UL<<31);
        return tmp_32;
    }
    else
        return -1;
}
//-----

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		80

ACT.c

```
#include "LPC82x.h"
#include "MyDefs.h"
#include "SCT.h"
#include "GPIO.h"
#include "MRT.h"
#include "VTIYU1820.h"

//-----
//-----

volatile bool bl_first;
volatile uint32_t disChangePeriodCnt;
//-----

void InitSCT(void)
{
    bl_first=true;
    LPC_SYSCON->SYSAHBCLKCTRL |= (1 << 8);
    LPC_SYSCON->PRESETCTRL &= ~(1<<8); // Reset SCT.
    LPC_SYSCON->PRESETCTRL |= (1<<8); // Release SCT.
    LPC_SCT->CTRL = (0<<5) | // SCTimer clock = 24 MHz,
                    (1<<2); // and halt.
    LPC_SCT->CONFIG = (1<<0); // 32 bits.
    LPC_SCT->REGMODE = (1 << 0); // Match/capture register 0 is capture.
    LPC_SCT->EV0_STATE = 0x00000001; // Event 0 only happens in state 0.
    LPC_SCT->EV0_CTRL = (0 << 6) | // IOSEL [9:6] = SCT_IN0,
                        (2 << 10) | // IOCOND [11:10] = falling edge,
                        (2 << 12) | // COMBMODE[13:12] = uses IO condition only,
                        (1 << 14) | // STATELD [14] = STATEV is loaded into state,
                        (0 << 15); // STATEV [15] = new state is 0.
    LPC_SCT->CAPCTRL0 = (1 << 0); // Event 0 causes capture 0 to be loaded.
    LPC_SCT->LIMIT = 0x0001; // Events 0 is used as counter limit.
    LPC_SCT->EVEN = 0x00000001; // Events 0 generate interrupts.
    LPC_INPUTMUX->SCT0_INMUX0=5; // Connect the comparator output to the SCT.
    NVIC_SetPriority(SCT_IRQn, 0);
    LPC_SCT->CTRL &= ~(1 << 2); // Unhalt it.
```

```

}
//-----
void SCT_IRQHandler(void)
{
    uint32_t tmp_32;
#ifdef DIV_4
    static uint32_t phase_div4=0;
#endif
    if (LPC_SCT->EVFLAG & (1 << 0))
    {
        LPC_SCT->EVFLAG = (1 << 0);    // Clear event 0 flag.
        tmp_32=LPC_SCT->CAP0;
        if(disChangePeriodCnt>3) return;
        ResTimeOutDisLevelToogle();
        if(disChangePeriodCnt>1)
        {
            disChangePeriodCnt--;
            return;
        }
        if(bl_first)
        {
            bl_first=false;
            tmp_32=1600;
        }
#ifdef DIV_4
        if(++phase_div4 & 3)==0)
            SetIntervalMRT((tmp_32<<1)-1);
        #else
            SetIntervalMRT((tmp_32>>1)-1);
        #endif
    }
}
//-----
void EnableSCTint(void)
{
    NVIC_EnableIRQ(SCT_IRQn);    // Enable SCTimer/PWM interrupt.
}

```



```

}
//-----
void DisSCTint(void)
{
    disChangePeriodCnt=4;
}
//-----
void EnSCTint(void)
{
    disChangePeriodCnt=3;
}
//-----

```

I2C.c

```

#include "LPC82x.h"
#include "MyDefs.h"
#include "I2C.h"
#include "CRC.h"
#include "VTIYU1820.h"
#include "WDT.h"
//-----
#define ADR_CFG1 0x3
#define ADR_CFG2 0x23
//-----
typedef struct
{
    int16_t gain;
    int16_t offset;
    uint16_t number;
    uint16_t crc;
}sTLE4997Cfg;
//-----
typedef union
{
    sTLE4997Cfg CfgS;
}

```

```

uint8_t CfgU[sizeof (sTLE4997Cfg)];
}uTLE4997Cfg;
//-----
uTLE4997Cfg cfg1, cfg2;
uint32_t i2c_adr, i2c_cnt, phase_i2c, bl_need_save_conf1, wrt_timer, i2c_proc_timer;
uint8_t *i2c_ptr;
bool bl_en_timeout_i2c_proc;
//-----
void SetData2I2C(uint32_t data); //Загрузка данных для пе-
редачи.
bool GetMasterPendingFromI2C(uint32_t dw_val);
void SetDataAndCont(uint32_t data);
bool ReadOneConf(uTLE4997Cfg *uni, uint32_t adr);
void CopyCfg2ToCfg1(void);
void CopyCfg1ToCfg2(void);
void WaitPhaseI2C(void);
bool CountCrcArray(uTLE4997Cfg *uni, bool bl_test);
void CopyCfg(uTLE4997Cfg *s, uTLE4997Cfg *d);
void InitWriteArray2I2C(uint32_t adr, uint32_t cnt, uint8_t *ptr);
void InitReadArray2I2C(uint32_t adr, uint32_t cnt, uint8_t *ptr);
void FixConf(void);
void SetStart(void);
void SetStop(void);
void SetCont(void);
void InitArray2I2C(uint32_t adr, uint32_t cnt, uint8_t *ptr, uint32_t phase);
void ResetI2c(void);
void StartWrDelay(void);
uint32_t ReadTwoConfs(void);
void RestartI2C_proc_timer(void);
bool TestTimeOutI2C_proc(void);
//-----
void InitI2C(void)
{
    bl_need_save_conf1=false;
    LPC_SYSCON->SYSAHBCLKCTRL |= (1<<5); //Разрешаем
I2C0 clock.

```

```

ResetI2c();
bl_en_timeout_i2c_proc=false;
}
//-----
void SetData2I2C(uint32_t data)
{ //Загрузка данных для передачи.
LPC_I2C0->MSTDAT=data;
}
//-----
bool GetMasterPendingFromI2C(uint32_t dw_val)
{
uint32_t tmp=LPC_I2C0->STAT;
if (tmp & 1)
return ((tmp>>1) & 7)==dw_val;
return false;
}
//-----
void InitWriteArray2I2C(uint32_t adr, uint32_t cnt, uint8_t *ptr)
{
InitArray2I2C(adr, cnt, ptr, 1);
}
//-----
void InitReadArray2I2C(uint32_t adr, uint32_t cnt, uint8_t *ptr)
{
InitArray2I2C(adr, cnt, ptr, 5);
}
//-----
void I2C_proc(void)
{
if(TestTimeOutI2C_proc())
{
ResetI2c();
phase_i2c=0;
}
switch(phase_i2c)
{

```

Изм.	Лист	№ докум.	Подпись	Дата

```

case 0:
    if(bl_need_save_conf1)
    {
        bl_need_save_conf1=false;
        InitWriteArray2I2C(ADR_CFG1, sizeof(uTLE4997Cfg), cfg1.CfgU);
    }
    break;
case 1:
case 5:
    if(GetMasterPendingFromI2C(2))
    {
        SetDataAndCont(i2c_adr>>8);
        phase_i2c++;
    }
    break;
case 2:
case 6:
    if(GetMasterPendingFromI2C(2))
    {
        SetDataAndCont(i2c_adr);
        phase_i2c++;
    }
    break;
case 3:
    if(GetMasterPendingFromI2C(2))
    {
        if(i2c_cnt)
        {
            SetDataAndCont(*(i2c_ptr++));
            i2c_cnt--;
        }
        else
            StartWrDelay();
    }
    break;
case 4:

```

```

    if((GetContCnt1ms()-wrt_timer)>=3)
        phase_i2c=0;
    break;
case 7:
    if(GetMasterPendingFromI2C(2))
    {
        SetData2I2C(0xa1); //Загрузка данных для пере-
дачи (slave-адрес для чтения).
        SetStart();
        phase_i2c++;
    }
    break;
case 8:
    if(GetMasterPendingFromI2C(1))
    {
        *(i2c_ptr++)=LPC_I2C0->MSTDAT;
        if(--i2c_cnt)
            SetCont();
        else
            StartWrDelay();
    }
    break;
}
}
//-----
void SetDataAndCont(uint32_t data)
{
    SetData2I2C(data); //Загрузка данных для передачи
(передача очередного байта).
    SetCont();
}
//-----
bool ReadOneConf(uTLE4997Cfg *uni, uint32_t adr)
{
    InitReadArray2I2C(adr, sizeof(uTLE4997Cfg), uni->CfgU);
    WaitPhaseI2C();
}

```

```

return CountCrcArray(uni, true);
}
//-----
bool ReadConf(void)
{
for(uint32_t i=3; i>0; i--)
{
switch(ReadTwoConfs())
{
case 0: //Обе конфигурации бракованные.
cfg1.CfgS.gain=399;
cfg1.CfgS.offset=399;
cfg1.CfgS.number=0;
CountCrcArray(&cfg1, false);
CopyCfg1ToCfg2();
WaitPhaseI2C();
case 2: //Только первая конфигурация бракованная.
CopyCfg2ToCfg1();
WaitPhaseI2C();
break;
case 1: //Только вторая конфигурация бракованная.
CopyCfg1ToCfg2();
WaitPhaseI2C();
break;
case 3: //Обе конфигурации исправные.
if(cfg1.CfgS.number==cfg2.CfgS.number) break;
if(cfg1.CfgS.number<cfg2.CfgS.number)
CopyCfg2ToCfg1();
else
CopyCfg1ToCfg2();
WaitPhaseI2C();
}
if (ReadTwoConfs()==3) return true;
}
return false;
}

```

```

//-----
void CopyCfg2ToCfg1(void)
{
    CopyCfg(&cfg2, &cfg1);
    InitWriteArray2I2C(ADR_CFG1, sizeof(uTLE4997Cfg), cfg1.CfgU);
}
//-----

void CopyCfg1ToCfg2(void)
{
    CopyCfg(&cfg1, &cfg2);
    InitWriteArray2I2C(ADR_CFG2, sizeof(uTLE4997Cfg), cfg2.CfgU);
}
//-----

void WaitPhaseI2C(void)
{
    do
    {
#ifdef EN_WDT
        ResetWDT();
#endif
        I2C_proc();
    } while(phase_i2c);
}
//-----

bool CountCrcArray(uTLE4997Cfg *uni, bool bl_test)
{
    uint32_t dw_tmp=sizeof(uTLE4997Cfg)-2;
    ResetCRC();
    for(uint32_t i=0; i< dw_tmp; i++)
        CountCrc(uni->CfgU[i]);
    dw_tmp=GetCRCResult();
    if(!bl_test)
    {
        uni->CfgS.crc=dw_tmp;
        return true;
    }
}

```

```

else
    return uni->CfgS.crc==dw_tmp;
}
//-----
void CopyCfg(uTLE4997Cfg *s, uTLE4997Cfg *d)
{
    for(uint32_t i=0; i<sizeof(uTLE4997Cfg); i++)
        d->CfgU[i]=s->CfgU[i];
}
//-----
uint32_t GetOffset(void)
{
    return cfg1.CfgS.offset;
}
//-----
uint32_t GetGain(void)
{
    return cfg1.CfgS.gain;
}
//-----
void SetOffset(uint32_t offs)
{
    int16_t tmp=offs-cfg1.CfgS.offset;
    if (tmp>-5 && tmp<5) return;
    if(!TestEn2EEProm()) return;
    cfg1.CfgS.offset=offs;
    FixConf();
}
//-----
void SetGain(uint32_t gain)
{
    int16_t tmp=gain-cfg1.CfgS.gain;
    if (tmp>-5 && tmp<5) return;
    if(!TestEn2EEProm()) return;
    cfg1.CfgS.gain=gain;
    FixConf();
}

```



```

}
//-----
void FixConf(void)
{
    cfg1.CfgS.number++;
    CountCrcArray(&cfg1, false);
    CopyCfg1ToCfg2();
    bl_need_save_conf1=true;
}
//-----
void SetStart(void)
{
    LPC_I2C0->MSTCTL = (1<<1);           //Set MSTSTART.
    RestartI2C_proc_timer();
}
//-----
void SetStop(void)
{
    LPC_I2C0->MSTCTL = (1<<2);           //Set MSTSTOP.
    RestartI2C_proc_timer();
}
//-----
void SetCont(void)
{
    LPC_I2C0->MSTCTL = (1<<0);           //Set MSTCONTINUE.
    RestartI2C_proc_timer();
}
//-----
void InitArray2I2C(uint32_t adr, uint32_t cnt, uint8_t *ptr, uint32_t phase)
{
    if(cnt==0) return;
    ResetI2c();
    i2c_adr=adr;
    i2c_cnt=cnt;
    i2c_ptr=ptr;
    RestartI2C_proc_timer();
}

```

```

do
{
#ifdef EN_WDT
    ResetWDT();
#endif

    if(TestTimeOutI2C_proc())
    {
        ResetI2c();
        return;
    }
}

while(!GetMasterPendingFromI2C(0));
SetData2I2C(0xa0); //Загрузка данных для передачи
(slave-адрес для записи).
SetStart();
phase_i2c=phase;
}
//-----
void ResetI2c(void)
{
    LPC_SYSCON->PRESETCTRL &= ~(1<<6); //Сброс
    LPC_SYSCON->PRESETCTRL |= (1<<6); //I2C0.
    LPC_I2C0->MSTTIME=0; //По два такта на уровень.
    LPC_I2C0->CLKDIV=59; //SCL 100KHz.
    LPC_I2C0->CFG=(1<<0); //Master enabled.
    phase_i2c=0;
}
//-----
void StartWrDelay(void)
{
    SetStop();
    wrt_timer=GetContCnt1ms();
    phase_i2c=4;
}
//-----
uint32_t ReadTwoConfs(void)

```

					ЮУрГУ-Д.11.05.01.2017.320.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		92

```

{
uint32_t tmp=0;
if(ReadOneConf(&cfg1, ADR_CFG1))
    tmp++;
if(ReadOneConf(&cfg2, ADR_CFG2))
    tmp+=2;
return tmp;
}
//-----
void RestartI2C_proc_timer(void)
{
i2c_proc_timer=GetContCnt1ms();
bl_en_timeout_i2c_proc=true;
}
//-----
bool TestTimeOutI2C_proc(void)
{
if(!bl_en_timeout_i2c_proc)
    return false;
bool blTmp=(GetContCnt1ms()-i2c_proc_timer) >= 5;
if(blTmp)
    bl_en_timeout_i2c_proc=false;
return blTmp;
}
//-----

```