

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

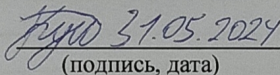
Профиль подготовки: «Вычислительные машины, комплексы, системы и сети»

БАКАЛАВРСКАЯ РАБОТА

на тему

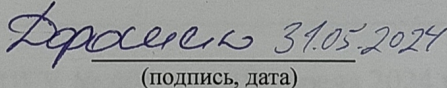
СИСТЕМА УПРАВЛЕНИЯ ПИТАНИЕМ
МОДУЛЯ КВАНТОВОЙ СЕТИ

Студент


(подпись, дата)

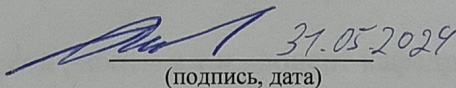
Культиясов Ярослав Кириллович
(ФИО полностью)

Руководитель


(подпись, дата)

Дорошенко И. Н.
(фамилия, инициалы)

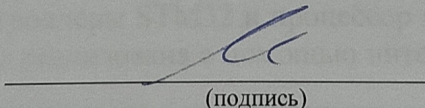
Нормоконтролёр


(подпись, дата)

Бычков А. С.
(фамилия, инициалы)

Работа допущена к защите (протокол заседания. кафедры от 27.05.2024г. № 11)

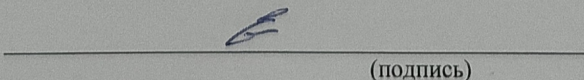
Заведующий кафедрой


(подпись)

Митрохин М.А.
(фамилия, инициалы)

Работа защищена с отметкой отлично (протокол заседания ГЭК от
10.06.2024 № 18)

Секретарь ГЭК


(подпись)

Гурин Е.И.
(фамилия, инициалы)

Пенза 2024

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

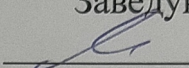
вычислительной техники

Кафедра

«Вычислительная техника»

«Утверждаю»

Заведующий кафедрой

 М.А. Митрохин

19 апреля 2024 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Культиясов Ярослав Кириллович гр. 20ВВП2 факультета вычислительной техники

2. Тема работы «Система управления питанием модуля квантовой сети»

Тема утверждена приказом ПГУ № 405/о от 19 апреля 2024 г.

3. Руководитель бакалаврской работы к.т.н., доцент кафедры ВТ ПГУ Дорошенко И. Н.

4. Задание на работу. Разработать программное обеспечение системы управления питанием модуля квантовой сети. Программное обеспечение системы управления питанием должно быть реализовано в соответствии с заложенной в модуль архитектурой. Система включает в себя микроконтроллеры STM32 и процессор на базе архитектуры ARM. Связь между элементами системы реализована с помощью интерфейса последовательного порта RS-232.

5. Перечень подлежащих разработке вопросов. Требуется произвести анализ предметной области: изучить техническую документацию к разрабатываемому модулю, изучить способы проектирования и разработки внутреннего программного обеспечения микроконтроллеров STM32. Также необходимо изучить средства разработки программного обеспечения.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Изучение предметной области	50	19.04.2024-30.04.2024	Дорошенко И. Н.
2.	Анализ требований	10	01.05.2024-02.05.2024	Дорошенко И. Н.
3.	Выбор технологий, применяемых при проектировании и разработке	50	03.05.2024-07.05.2024	Дорошенко И. Н.
4.	Описание алгоритмов	40	08.05.2024-11.05.2024	Дорошенко И. Н.
5.	Разработка системы управления питанием модуля квантовой сети	150	12.05.2024-29.05.2024	Дорошенко И. Н.
6.	Оформление пояснительной записки	24	30.05.2024-31.05.2024	Дорошенко И. Н.

Дата выдачи задания 19 апреля 2024 г.

Руководитель бакалаврской работы Дорошенко И. Н. 19.04.24
(подпись, дата)

Задание к исполнению принял студент Культиясов Я. К. 19 апреля 2024
(подпись, дата)

Работу к защите допустить
Декан факультета Фионова Л.Р. 06.06.2024
(подпись, дата)

Реферат

Бакалаврская работа содержит 102 л., 23 рис., 27 табл., 5 источн. и 3 прил.

ПРОГРАММИРОВАНИЕ, УПРАВЛЕНИЕ, ЭЛЕКТРОПИТАНИЕ,
МИКРОКОНТРОЛЛЕР, КВАНТОВАЯ СЕТЬ, ЖУРНАЛИРОВАНИЕ

Цель работы – разработать систему управления питанием модуля квантовой сети.

Разрабатываемая система должна обеспечивать непрерывный мониторинг напряжений на линиях питания и в случае возникновения аварийной ситуации, такой как выход напряжения за предельные значения, выключить питание платы для защиты оборудования. Также система должна обеспечивать журналирование аварийных ситуаций. Взаимодействие с пользователем должно сводиться к кнопке питания.

Результатом проделанной работы является система управления питанием модуля квантовой сети, отвечающая всем требованиям, которые предъявляются к системе питания. Конечная версия системы совмещает в себе простоту для пользователя и гибкость для расширения и взаимодействия с другими программными модулями. Пользователь полностью изолирован от системы управления питанием, благодаря чему исключается вероятность испортить изделие, при неразумном использовании. Это достигается за счёт автономности системы и сведения взаимодействия с пользователем к кнопке питания.

					ПГУ 1.090301.31.001 ПЗ							
Изм	Лист	№ докум.	Подпись	Дата	Система управления питанием модуля квантовой сети Пояснительная записка			Литера	Лист	Листов		
Разраб.		Культиясов Я. К.	<i>Кули</i>	31.05.24							2	102
Пров.		Дорошенко И. Н.	<i>Дорошенко</i>	31.05.24								
Н. Контр.		Бычков А. С.	<i>Бычков</i>	31.05.24				ФВТ, каф. ВТ гр. 20ВВП2				
Утв.		Митрохин М.А.	<i>Митрохин</i>	31.05.24								

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

АЦП – аналого-цифровой преобразователь

ВПО – внутреннее программное обеспечение

Кубит – наименьшая единица информации в квантовом компьютере

МК – микроконтроллер

ПЗУ – постоянное запоминающее устройство

ПУ – процессор управления

Сторожевой таймер – таймер, который необходимо сбрасывать в заданные промежутки времени, использующийся в схемах контроля над зависанием системы

УП – управление питанием

ШИМ-контроллер – контроллер для широтно-импульсной модуляции

ЯУП – ячейка управления питанием

HAL – слой аппаратных абстракций

UART – универсальный асинхронный приемопередатчик

					ПГУ 1.090301.31.001 ПЗ	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

Содержание

Реферат	2
Перечень сокращений и обозначений.....	3
Содержание	4
Введение.....	7
1 Обзор предметной области.....	9
1.1 Обоснование необходимости использования системы.....	9
1.2 Постановка задачи	10
1.2.1 Техническая база разрабатываемой системы.....	10
1.2.2 Требования к разрабатываемой системе	16
1.3 Обзор микроконтроллера STM32.....	17
1.4 Выбор средств разработки	19
1.4.1 Выбор языка программирования.....	19
1.4.2 Ввод о выборе языка программирования	21
1.4.3 Выбор среды разработки.....	22
1.4.4 Вывод о выборе среды разработки	24
2 Разработка системы управления питанием	25
2.1 Назначение функций GPIO портам микроконтроллеров.....	25
2.2 Разработка протокола взаимодействия.....	31
2.2.1 Взаимодействие элементов системы	31
2.2.2 Формат пакета информационного взаимодействия	34
2.2.3 Общие команды взаимодействия	39
2.2.4 Команды взаимодействия процессора управления и микроконтроллера ячейки управления питанием	40
2.2.5 Команды взаимодействия процессора управления и ведущего микроконтроллера	42
2.2.6 Команды взаимодействия ведущего и ведомого микроконтроллеров	48
2.3 Разработка внутреннего программного обеспечения микроконтроллеров	52
2.3.1 Общие положения о процессе работы микроконтроллеров.....	52

2.3.2	Разработка общих процессов для всех микроконтроллеров	53
2.3.3	Разработка процессов микроконтроллера ячейки управления питанием	55
2.3.4	Разработка процессов ведущего микроконтроллера управления питанием на плате управления	58
2.3.5	Разработка процессов ведомого микроконтроллера управления питанием на плате управления	63
2.4	Разработка драйвера обмена сообщениями с микроконтроллерами	64
3	Описание разработанной системы управления питанием	68
3.1	Руководство пользователя	68
3.2	Руководство программиста	68
3.2.1	Общие положения	68
3.2.2	Отправка сообщения микроконтроллеру	69
3.2.3	Подписка на уведомление от микроконтроллера	70
3.3	Описание внутреннего программного обеспечения микроконтроллера ячейки управления питанием	71
3.3.1	Модуль TxBuffer	71
3.3.2	Модуль RxBuffer	72
3.3.3	Модуль Main	72
3.3.4	Модуль System	72
3.3.5	Модуль Channel	73
3.4	Описание внутреннего программного обеспечения ведущего микроконтроллера управления питанием на плате управления	74
3.4.1	Модуль TxBuffer	74
3.4.2	Модуль RxBuffer	75
3.4.3	Модуль Main	75
3.4.4	Модуль System	75
3.4.5	Модуль PowerCheck	76
3.4.6	Модуль Channel	77
3.4.7	Модуль Log	78

3.5 Описание внутреннего программного обеспечения ведомого	
микроконтроллера управления питанием на плате управления	78
3.5.1 Модуль TxBuffer	78
3.5.2 Модуль RxBuffer	79
3.5.3 Модуль Main.....	79
3.5.4 Модуль System	79
3.5.5 Модуль PowerCheck.....	80
3.5.6 Модуль Channel.....	81
3.6 Описание драйвера обмена сообщениями с микроконтроллерами	82
3.6.1 Модуль MCUTransportCommunication	82
3.6.2 Модуль MCUTransportCommunicationSerialWrite	82
3.6.3 Модуль MCUTransportCommunicationSerialRead	82
4 Безопасность жизнедеятельности.....	84
4.1 Меры безопасности.....	84
4.2 Требования к рабочему месту.....	84
Заключение	85
Список использованных источников	86
Приложение А – UML-диаграммы.....	87
Приложение А.1 – Диаграмма последовательностей 1	848
Приложение А.2 – Диаграмма последовательностей 2.....	849
Приложение Б – Схемы принципиальные	90
Приложение Б.1 – Схема принципиальная 1	91
Приложение Б.2 – Схема принципиальная 2	92
Приложение В – Презентация.....	93

Введение

В настоящее время остро стоит вопрос защиты информации, передаваемой на большие расстояния, где невозможно изолировать канал передачи данных от предполагаемого злоумышленника. Поэтому информация шифруется ключом, который неизвестен сторонним лицам и не может быть получен в течение времени его использования.

В последнее время широко стали развиваться квантовые технологии передачи и генерации ключей. Квантовые сети позволяют защищать передаваемую информацию с помощью физических свойств кубитов, а не алгоритмов, требующих высокой вычислительной мощности для решения со стороны злоумышленника. Благодаря свойствам кубита стало возможным однозначно определить прослушивается ли канал, чтобы предотвратить утечку данных.

Как и любая другая коммуникационная сеть, квантовая сеть состоит из каналов передачи данных и коммутационных узлов. Каждый узел включает в себя ряд модулей с особыми требованиями. К таким требованиям относится невозможность повлиять на работу оборудования модулей из внешней среды, а также обеспечение отказоустойчивости.

Одним из внешних факторов воздействия на оборудование является его питание. Даже использование серверных отказоустойчивых блоков питания не может гарантировать отсутствие скачков и отклонений по напряжению. А для исключения воздействия со стороны источника питания используется система аппаратного мониторинга и управления, которая определяет отклонения по напряжению на всех линиях питания, ведёт журнал возникших ошибок и при необходимости предпринимает действия для защиты оборудования от повреждений.

В данной работе представлена система управления питанием модуля квантовой сети, которая включает в себя функционал системы аппаратного мониторинга и управления, ведёт подробный журнал событий, предоставляет

					ПГУ 1.090301.31.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

возможность включать и выключать питание как отдельного оборудования, так и всего модуля в целом. Отличительной особенностью данной системы, по сравнению с готовыми решениями представленными на рынке, является гибкость, достигаемая путём использования собственного протокола обмена между контроллерами и процессором.

					ПГУ 1.090301.31.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

1 Обзор предметной области

1.1 Обоснование необходимости использования системы

В современном мире существует множество устройств с требованием стабильного питания с отклонением по напряжению не более 5% и модуль квантовой сети не является исключением. Современные блоки питания предоставляют плате питание по основным линиям ± 12 , $+5$, $+3.3$ В, но компоненты платы зачастую требуют куда меньшего напряжения. Поэтому в модуле квантовой сети используются две цепи:

- первичная цепь, включающая в себя источник питания и каналы по которым ток идёт до плат;
- вторичная цепь, включающая в себя ШИМ-контроллер и линии питания, распределяющие ток по плате.

Сейчас системы питания редко выходят из строя или начинают подавать неверное напряжение, однако на подавляющем большинстве современных плат установлены микросхемы аппаратного мониторинга, которые собирают информацию о линиях питания вторичной цепи. Зачастую на специализированном оборудовании система мониторинга включена в систему управления питанием.

В случае с квантовой сетью, в каждом модуле система управления питанием просто необходима. Это обуславливается, тем что такие модули включают в себя чувствительное к питанию оборудование, например плата генератора шума, и требуют постоянного контроля за питанием, чтобы исключить возможность влияния на работу оборудования со стороны источника питания, а также обеспечить сохранность оборудования в случаях неисправности системы питания. Система управления питанием позволяет быстро реагировать на сбои в системе питания.

Кроме того система управления питанием может включать в себя дополнительный функционал, такой как:

- ведение журнала и предоставление его по требованию;

					ПГУ 1.090301.31.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

- включение и выключение определённого оборудования по требованию от процессора или другого оборудования;

- обработка внешних сигналов нажатия кнопок питания и датчиков вскрытия корпуса.

Важным фактором в разработке системы управления питанием является стабильность и автономность. Система должна обеспечивать постоянный контроль на протяжении всего времени питания от источника, а также быть самодостаточной, чтобы не требовалось вмешательство оператора. Для достижения этих целей используется схема из контроллеров напряжения и центрального процессора, который является ведущим в системе управления питанием.

1.2 Постановка задачи

1.2.1 Техническая база разрабатываемой системы

Требования к системе управления питанием были сформулированы с учётом технических требований и на основании задачи, которая предъявлена к разрабатываемому изделию.

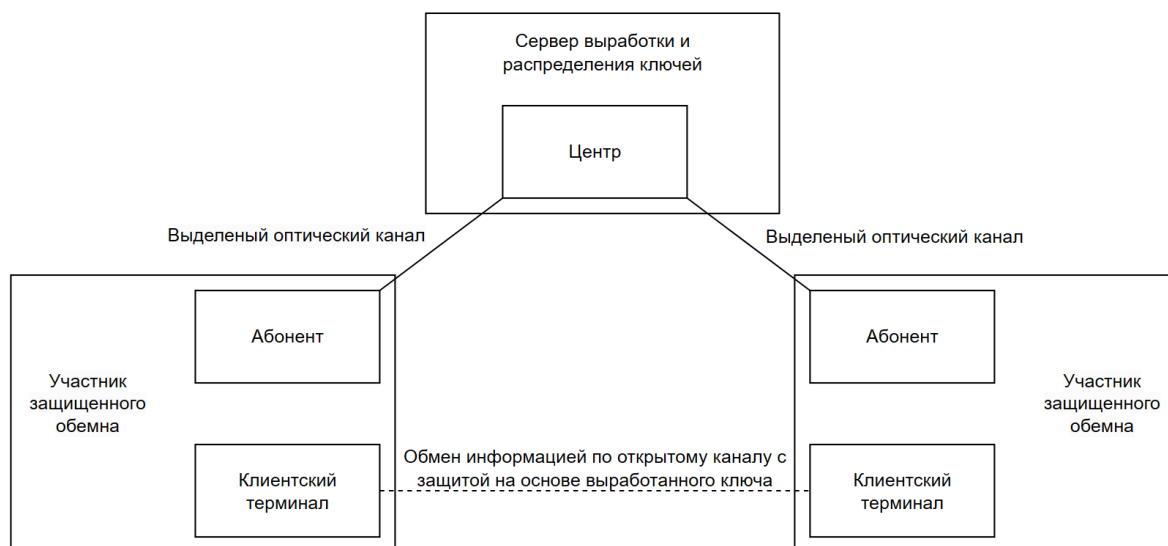


Рисунок 1 – Общая схема квантовой сети

На рисунке выше изображена общая схема квантовой сети, которая включает в себя двух участников защищённого обмена и сервер выработки и распределения ключей.

Каждый участник включает в себя клиентский терминал, выполненный на базе персонального компьютера, который отвечает за непосредственную связь между клиентами по открытому каналу с защитой на выработанном ключе, и абонент, выполненный в виде отдельного модуля, который ответственен за взаимодействие с сервером выработки и распределения ключей.

Сервер выработки и распределения ключей состоит из нескольких модулей, именуемых центр. Каждый модуль является отдельным устройством в корпусе.

Требуется разработать систему управления питанием для модуля квантовой сети, именуемого центр. Центр состоит из следующих функциональных плат:

- плата шифрования;
- плата шумогенератора;
- плата управления.

Для реализации системы управления питанием в модуль была заложена схема изображённая на рисунках 2 и 3.

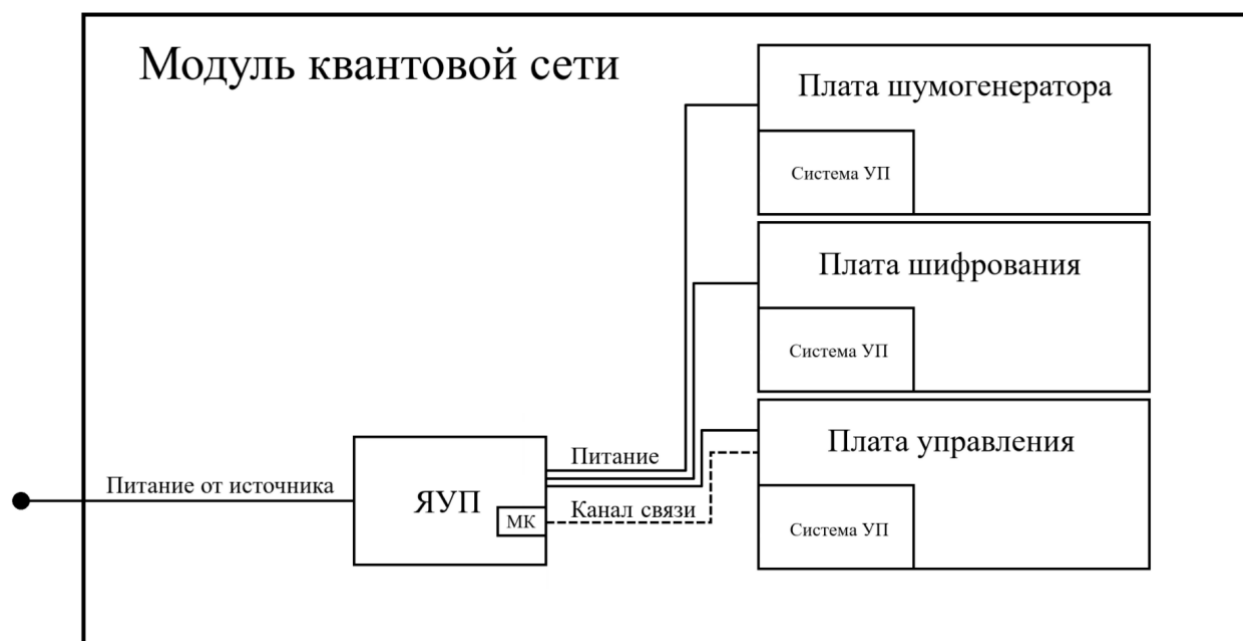


Рисунок 2 – Схема расположения основных блоков

Связь между микроконтроллером ячейки управления питанием (ЯУП) и процессором управления на плате управления реализована через UART.

Ячейка управления питанием, по своей задумке, является узлом подающим питание от источника ко всем платам, а также обеспечивающим связь платы управления с кнопкой питания, а также с органами управления, которые не имеют отношения к системе управления питанием, такими как дисплей и клавиатура. На ячейке управления питанием располагается один микроконтроллер, на который возложена задача обработки внешних сигналов от органов управления. В данном случае рассматривается включение питания по нажатию кнопки включения и уведомление операционной системы при повторном нажатии, с последующим выключением питания после завершения работы модуля.

На каждой плате, которая питается от ячейки управления питанием, установлена система управления питанием. Схема организации системы управления питанием на плате управления изображена на рисунке 3.

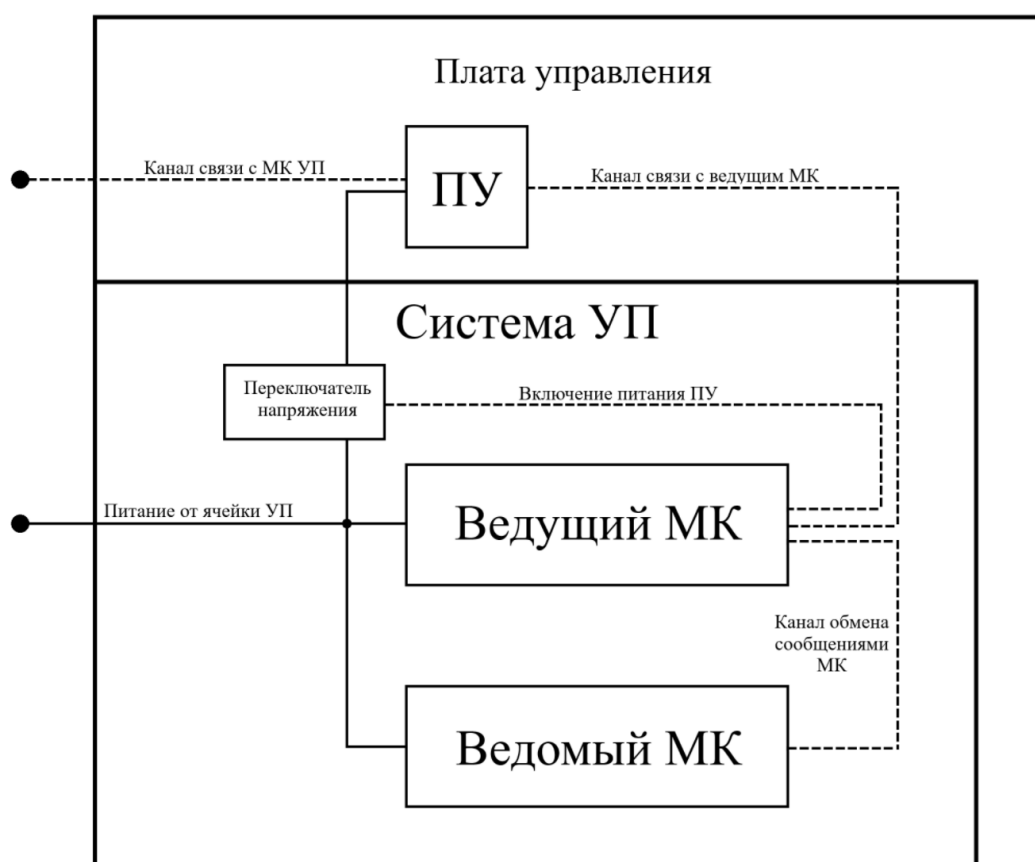


Рисунок 3 – Организация системы управления питанием

Система управления питанием (УП) на плате управления состоит из процессора управления (ПУ) и блока микроконтроллеров.

Блок микроконтроллеров может состоять из одного, двух и более микроконтроллеров. Это зависит от количества линий питания на плате, которые необходимо измерять, так как АЦП микроконтроллера может одновременно измерять только ограниченное число каналов. В этом блоке один микроконтроллер является ведущим. Ведущий отвечает на запросы от процессора управления, отправляет уведомления и организует работу подчинённых микроконтроллеров. При подаче питания первыми включаются микроконтроллеры. После измерения напряжений на линиях. Ведущий подаёт питание на процессор управления и остальное оборудование.

Процессор управления является главным в принятии решений всего модуля квантовой сети, в том числе он управляет системой управления питанием. На процессоре управления располагается драйвер обмена сообщениями с микроконтроллерами. Драйвер принимает и отправляет пакеты команд с микроконтроллерами. Для реализации системы управления питанием процессор управления ведёт обмен с контроллером ячейки управления питанием и блоком микроконтроллеров управления питанием.

Процессор управления реализован на базе программируемой логической интегральной схемы XC7Z030-2FBG484I. Микроконтроллеры используемые в данной системе это STM32L151VET6.

Обмен данными процессора управления с программным обеспечением микроконтроллеров осуществляется через интерфейс UART со следующими параметрами:

- скорость 115200 бит/сек.;
- биты данных 8;
- контроля чётности нет;
- стоповых бит 1;
- управления потоком нет.

Система управления питанием должна вести контроль напряжений на линиях питания. Контроль напряжений на каждой плате ведётся по своему, заданному набору линий, однако далее в работе, при рассмотрении отдельных плат в составе модуля квантовой сети, фокус смещён на плату управления, так как реализация системы управления питанием на разных платах отличается незначительно. Перечень линий питания с их предельными значениями перечислен в таблице 1.

Таблица 1 – Предельные значения по линиям питания на плате управления модуля квантовой сети

Номер линии	Номинальное значение, В	Минимальное значение, мВ	Максимальное значение, мВ
1	3.3	2970	3630
2	1	900	1100
3	1	900	1100
4	1.2	1080	1320
5	1.8	1620	1980
6	1.8	1620	1980
7	1.8	1620	1980
8	1.5	1350	1650
9	3.3	2970	3630
10	1	900	1100
11	1	900	1100
12	1.2	1080	1320
13	1.8	1620	1980
14	1.8	1620	1980
15	1.8	1620	1980

Номер линии	Номинальное значение, В	Минимальное значение, мВ	Максимальное значение, мВ
16	1.5	1350	1650
17	1.25	1125	1375
18	0.977	879	1075
19	2.5	2250	2750
20	3	2700	3300
21	5	4500	5500
22	-5	-4500	-5500
23	5	4500	5500
24	-5	-4500	-5500
25	5	4500	5500
26	15	13500	16500
27	-15	-13500	-16500
28	12	10800	13200
29	-12	-10800	-13200
30	9	8100	9900
31	-9	-8100	-9900
32	1,9	1710	2090
33	1,9	1710	2090
34	3.3	2970	3630
35	3.3	2970	3630
36	3.3	2970	3630
37	-1.7	1530	1870
38	-1.7	1530	1870

Номер линии	Номинальное значение, В	Минимальное значение, мВ	Максимальное значение, мВ
39	3.3	2970	3630
40	-13.75	-12375	-15125
41	-13.75	-12375	-15125
42	-13.75	-12375	-15125
43	-13.75	-12375	-15125
44	-13.75	-12375	-15125

1.2.2 Требования к разрабатываемой системе

Требуется разработать внутреннее программное обеспечение микроконтроллеров и драйвер для взаимодействия с ними. Программное обеспечение для реализации системы управления питанием должно быть реализовано в соответствии с заложенной в модуль архитектурой, описанной в пункте 1.2.1. Список устройств, составляющих систему управления питанием, состоит из следующих элементов:

- процессор управления;
- блок микроконтроллеров управления питанием на плате;
- микроконтроллер ячейки управления питанием.

Система управления питанием, в первую очередь, должна выполнять непрерывный мониторинг напряжений на линиях питания и в случае возникновения аварийной ситуации, такой как выход напряжения за предельные значения, заранее рассчитанные на этапе разработки платы, выключить питание платы для защиты оборудования. Также в системе управления питанием необходимо реализовать алгоритмы реакции системы на нажатия кнопки питания, так как это единственный способ взаимодействия пользователя с системой. Также необходимо разработать

гибкий протокол взаимодействия для возможности расширения системы в будущем, а также для возможности внедрения системы в другие модули.

Помимо основных задач система должна реализовывать следующий функционал:

- ведение журнала;
- программное включение и выключение питания навесного оборудования по команде от процессора;
- изменение пороговых значений на заданных линиях питания по команде от процессора;

1.3 Обзор микроконтроллера STM32

Для реализации системы управления питанием в модуль заложена схема на основе микроконтроллеров STM32L151VET6. Данный микроконтроллер изображён на рисунке 4.



Рисунок 4 – Микроконтроллер STM32L151VET6

Микроконтроллер относится к серии STM32L151xE, которая производится компанией STMicroelectronics.

Компания STMicroelectronics — это один из крупнейших производителей полупроводниковых электронных и микроэлектронных компонентов. Компания основана в 1987 году и активно развивается по сей день. Производственные мощности насчитывают 14 заводов, а основными покупателями продукции компании являются Apple, Bosch, Continental AG, HP, Huawei, Mobileye, Samsung, SpaceX и Tesla. Данные сведения позволяют ожидать стабильности от компании, что немаловажно при разработке любого электронного продукта, так как это позволяет быть уверенным, в том что на протяжении жизни проекта не будет дефицита компонентов.

Микроконтроллер STM32L151VET6 имеет 12-битный АЦП на 25 каналов, что позволяет одновременно измерять большее число линий питания, чем могут предложить аналоги в данной ценовой категории, а также микроконтроллер имеет встроенный модуль RTC, что позволяет вести отсчёт в реальном времени. Микроконтроллер обладает возможностью отладки внутреннего программного обеспечения с помощью последовательного порта стандарта RS-232 или JTAG. Кроме того компания STMicroelectronics предоставляет удобные средства разработки, а именно средство конфигурации микроконтроллеров и микропроцессоров CubeMX и более доступный, на фоне конкурентов, программатор ST-LINK. Также STM32 поддерживается в ведущих на данный момент средах разработки Keil и IAR, а также в SW4STM32 разработанной специально для работы с продуктами данной компании. [1]

Основные характеристики данного микроконтроллера перечислены в таблице 2. [2]

					ПГУ 1.090301.31.001 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 2 – Характеристики микроконтроллера

Наименование характеристики	Характеристика
Тип ядра	ARM Cortex M3
Разрядность ядра, Бит	32
Частота, МГц	32
DMIPS/МГц	1.25
Flash память, КБ	512
RAM, КБ	80
EEPROM, КБ	16
Разрешение АЦП, Бит	12
Каналы АЦП	25
Поддерживаемые интерфейсы	I2C, I2S, SPI, USART, USB
Количество I/O	83
Тип посадки	SMD/SMT
Тип корпуса	LQFP-100
Диапазон температур, °C	-40...+105
Напряжение питания, В	1.65-3.6

1.4 Выбор средств разработки

1.4.1 Выбор языка программирования

Для разработки внутреннего программного обеспечения микроконтроллеров STM32 используются языки программирования Си и C++. Оба языка подходят как для реализации программного обеспечения микроконтроллеров, так и для реализации драйвера для процессора управления.

Си – это универсальный, императивный язык программирования поддерживающий структурное программирование и сочетающий в себе возможности языков высокого и низкого уровня. Язык был разработан для переноса семейства UNIX операционных систем с ассемблера на более удобный в использовании язык программирования. Синтаксис языка близок к типичным машинным командам, благодаря чему может использоваться в ситуациях, где необходимо прямое взаимодействие с памятью. [3]

Основные достоинства языка Си:

- простой синтаксис, позволяющий собирать конструкции любой сложности;
- возможность подключения готовых модулей для расширения возможностей языка и встроенных команд;
- непосредственный доступ к памяти и возможность использования указателей;
- явные области видимости имён;
- возможность использования структур и объединений;
- широкий набор инструментов для работы с битовыми операциями.

Основные недостатки языка Си:

- отсутствие средств автоматического определения ошибок в использовании и освобождении памяти;
- компилятор обнаруживает только самые грубые ошибки в синтаксисе.

C++ является высокоуровневым языком программирования общего назначения, в котором используется статическая типизация, и который может применяться для создания самых разнообразных приложений. В настоящее время язык C++ является одним из самых популярных и распространённых языков программирования. C++ является прямым наследником языка C. При разработке языка C++ ставили целью сделать такой же быстрый и низкоуровневый язык как C, но с поддержкой объектно-ориентированной парадигмы программирования. Синтаксис C++ очень похож на синтаксис C –

					ПГУ 1.090301.31.001 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

значительная часть программ обратно совместимы и могут запускаться с компиляторов обоих языков.

Основные достоинства языка C++:

- в C++ используются глобальные данные и глобальные функции, которые не используются во многих других языках высокого уровня в области компьютерных наук;
- программы на C++ могут использовать мультипарадигменное программирование;
- возможность подключения готовых модулей для расширения возможностей языка и встроенных команд;
- непосредственный доступ к памяти и возможность использования указателей;
- C++ может использоваться в системном программировании, чего не могут позволить большинство других языков высокого уровня;
- возможность использования структур и объединений;
- широкий набор инструментов для работы с битовыми операциями.

Основные недостатки языка C++:

- Более сложные конструкции языка, которые могут привести к ошибкам;
- компилятор не обнаруживает ошибки при работе с памятью.

1.4.2 Ввод о выборе языка программирования

Оба языка подходят для использования в условиях поставленной задачи, однако, в результате сравнительного анализа, для разработки был выбран язык Си, как более удобный для применения в поставленной задаче. Причина выбора в том, что Си обладает более простым и явным синтаксисом, что делает код более прозрачным. В тоже время конструкции языка C++ являются избыточными для реализации поставленной задачи, так как задача не требует применения ООП и его сложных конструкций.

1.4.3 Выбор среды разработки

В качестве инструментов разработки программ STM32 были рассмотрены:

- средство конфигурации микроконтроллеров и микропроцессоров CubeMX, позволяющее сократить время разработки за счёт генерации кода инициализации;

- среда разработки IAR IDE;

- среда разработки KEIL IDE.

Также в качестве инструмента разработки драйвера обмена пакетами были рассмотрены:

- среда разработки Visual Studio;

- среда разработки CLion .

CubeMX – это графический инструмент, который позволяет легко конфигурировать микроконтроллеры и микропроцессоры STM32, а также генерировать соответствующий код инициализации на языке Си. Данный инструмент поддерживает все актуальные серии STM32 и сильно упрощает создание проекта для будущей программы за счёт установки HAL и необходимых библиотек, удобного графического редактора, позволяющего инициализировать периферийные устройства, и генератора кода.

Средство генерации кода инициализации CubeMX было выбрано, вследствие того что на данный момент не имеет аналогов. Это достигается за счёт того что CubeMX производится компанией STMicroelectronics как и сами микроконтроллеры STM32. Оно является бесплатным и моментально получает обновления при появлении новых микроконтроллеров в линейке.

IAR IDE – это полноценная среда разработки программ на ARM архитектуре включающая в себя компилятор языка Си и C++ разработанный компанией IAR Systems и предлагающий расширенную глобальную и целевую оптимизацию. Инструмент позволяет писать, компилировать и конфигурировать бинарный файл внутреннего программного обеспечения (ВПО) микроконтроллера, а также отлаживать программы STM32 на

устройстве при подключении программатора ST-LINK. IAR IDE поддерживает все 32-разрядные ядра ARM от основных поставщиков и некоторые 64-разрядные, а благодаря интегрированному дополнению к статическому анализу C-STAT и надстройке для анализа среды выполнения C-RUN набор инструментов помогает обеспечить качество кода на протяжении всего процесса разработки и тестирования.

KEIL IDE – это мощный инструмент для создания и редактирования проектов любой степени сложности, который позволяет как работать с исходными текстами внутреннего программного обеспечения, так и проводить программирование ПЗУ микроконтроллера с помощью встроенных инструментов, а также данная среда разработки позволяет проводить внутрисхемную отладку кода. В IDE используется язык программирования C и C++. Так же KEIL имеет в своём распоряжении симулятор, который позволяет эмулировать некоторое аппаратное оборудование, например такое как интерфейс UART.

В ходе сравнения сред разработки IAR IDE и KEIL IDE не выявлен какой-либо фаворит. Обе среды предоставляют широкий функционал достаточный для реализации поставленной задачи, однако необходимо выбрать что-то одно. Выбор пал на IAR IDE. Данная среда разработки предоставляет достаточный функционал для того чтобы полностью покрыть весь спектр задач.

Visual Studio – это одна из самых популярных в мире IDE, которую можно использовать для редактирования, отладки и сборки кода. Данная IDE поддерживает практически все возможные архитектуры, позволяет создавать широкий спектр разнообразных приложений и может расширять свой функционал, путём установки модулей расширения.

CLion - это популярная среда, предназначенная для кросс-платформенной разработки на языках C и C++, которая может быть использована для редактирования, отладки и сборки кода. В совокупности с этим данная IDE включает в себя мощный отладчик и средства

динамического анализа кода, позволяющие быстро находить и устранять возникающие проблемы.

Обе среды предоставляют широкий функционал достаточный для реализации поставленной задачи, однако выбор пал на Visual Studio, так как данная среда разработки является более простой в освоении. Помимо этого данная среда является более универсальной за счёт расширений, которые можно установить прямо из установщика.

1.4.4 Вывод о выборе среды разработки

В результате изучения и сравнения доступных сред разработки, на основании их достоинств и недостатков, был выбран следующий комплект инструментов и сред разработки:

- инструмент для конфигурации микроконтроллеров CubeMX;
- среда разработки IAR IDE;
- среда разработки Visual Studio.

Данный набор инструментов позволяет вести эффективную разработку программ микроконтроллеров и драйвера процессора на всех этапах разработки.

2 Разработка системы управления питанием

2.1 Назначение функций GPIO портам микроконтроллеров

Подключение микроконтроллеров на печатные платы произведено согласно технической документации. В данной работе рассматриваются следующие микроконтроллеры:

- микроконтроллер ячейки управления питанием;
- ведущий микроконтроллер системы управления питанием, расположенный на плате управления;
- ведомый микроконтроллер системы управления питанием, расположенный на плате управления.

Важно подчеркнуть тот факт, что на данных схемах отмечены только те контакты микроконтроллеров, которые относятся к системе управления питанием модуля квантовой сети.

Схема подключения микроконтроллера ячейки управления питанием изображена на рисунке 5.

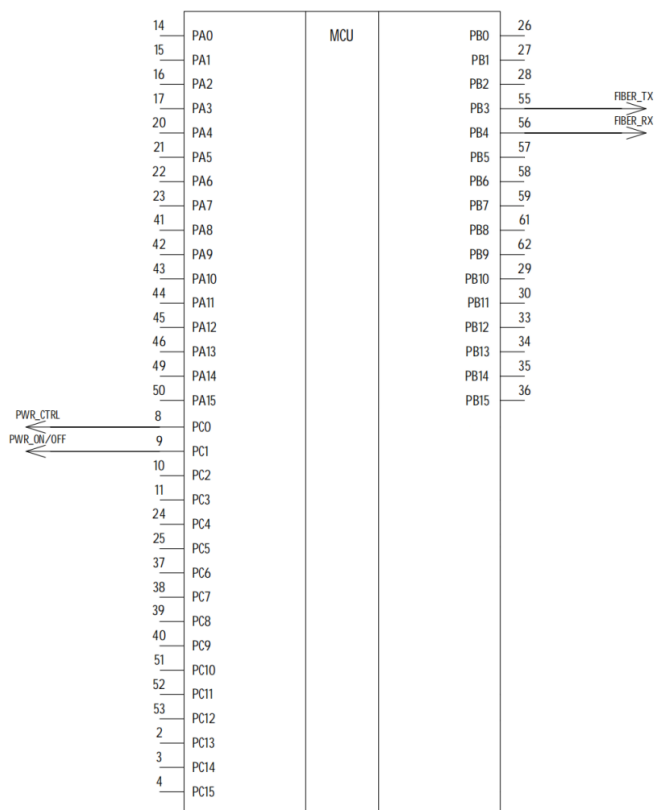


Рисунок 5 – Микроконтроллер ячейки управления питанием

Функция микроконтроллера ячейки управления питанием в системе управления питанием заключается во включении и выключении питания всего модуля в целом. За это отвечает контакт PWR_CTRL. Условно можно сказать, что питание модуля есть, если PWR_CTRL находится в состоянии логической единицы, иначе питание не подаётся.

Включение питания происходит при нажатии кнопки питания. Микроконтроллер определяет статус нажатия кнопки питания по входному контакту PWR_ON/OFF. Логический ноль обозначает, что кнопка нажата.

Выключение питания происходит по двум сценариям:

- кнопка питания была нажата, когда питание уже подаётся. Это запустит процесс завершения работы модуля квантовой сети;
- кнопка питания зажата в течение 5 секунд, что приводит к принудительному выключению питания, вне зависимости от обстоятельств. При таком выключении модуль успеет завершить лишь наиболее важные процессы, до того как питание будет отключено.

Процесс завершения работы модуля квантовой сети запускается процессором управления после получения уведомления о нажатии кнопки питания. Уведомление отправляется микроконтроллером сразу после нажатия кнопки питания, при условии, что питание модуля в данный момент подаётся. После завершения работы модуля процессор управления отправляет запрос на выключение питания. Уведомление и запрос передаются через UART, TX и RX которого располагаются на контактах PB3 и PB4.

Основная часть работы системы управления питанием возложена на блоки микроконтроллеров управления питанием. Схема подключения ведущего микроконтроллера управления питанием на плате управления изображена на рисунке 6.

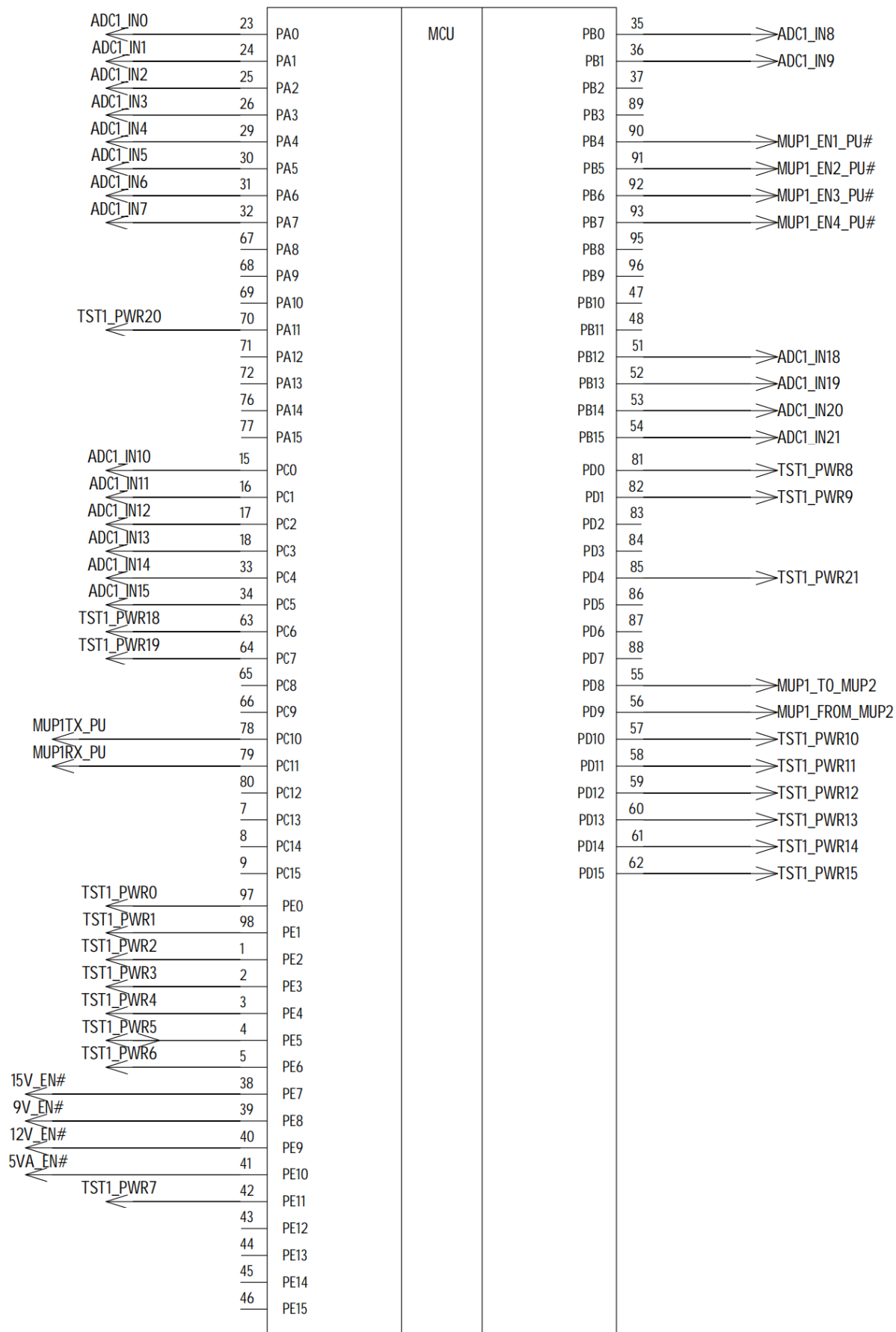


Рисунок 6 – Ведущий микроконтроллер управления питанием

Среди функций ведущего микроконтроллера управления питанием на плате управления можно выделить следующие:

- включение основных линий питания ± 5 , ± 9 , ± 12 , ± 15 В;
- измерение напряжений на всех линиях питания;
- включение питания процессора управления;
- организация связи с процессором управления и ведомым микроконтроллером;
- выключение питания всей платы при возникновении аварийной ситуации;
- ведение журнала аварийных ситуаций.

За включение основных линий питания отвечают контакты 15V_EN, 9V_EN, 12V_EN, 5V_EN. Питание включается когда на контакт подаётся логическая единица, что происходит во время инициализации микроконтроллера. Для выключения питания по этим линиям, соответственно, необходимо подать логический ноль.

За измерение линий питания отвечает встроенный 12-битный АЦП. Задействованы каналы 0-15 и 18-21, так как каналы 16 и 17 являются внутренними каналами для измерения температуры и опорного напряжения соответственно. Входы АЦП на схеме обозначаются как ADC1_IN. На ведущем микроконтроллере ведётся контроль за 20 линиями питания, а оставшиеся 24 линии контролируются ведомым микроконтроллером. Микроконтроллер в течении работы модуля квантовой сети постоянно проверяет напряжения по заданным линиям. Если на какой-нибудь линии питание выйдет за предельное значение, то микроконтроллер зарегистрирует аварийную ситуацию и выключит питание платы, а также добавит в журнал данное событие.

Включение питания процессора управления происходит после измерения напряжений на линиях. Включение происходит путём подачи логической единицы на контакты MUP1_EN_PU, после чего на процессор подаётся питание по линиям:

					ПГУ 1.090301.31.001 ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подпись	Дата		

- 1 В и 4 В – контакт MUP1_EN1_PU;
- 1.8 В – контакт MUP1_EN2_PU;
- 1.5 В и 3.3 В – контакт MUP1_EN3_PU;
- 2.5 В и 5 В – контакт MUP1_EN4_PU.

Связь с процессором управления организована через UART, TX и RX которого расположены на контактах MUP1TX_PU и MUP1RX_PU. Также организована связь с ведомым микроконтроллером, но через контакты MUP1_TO_MUP2 и MUP1_FROM_MUP2.

Контакты TST1_PWR включены совместно с контактами ADC1_IN и предназначены для тестирования работы микроконтроллера относительно измерения напряжения. Подробнее процесс тестирования будет рассмотрен в пункте 2.3.3.

Среди функций ведомого микроконтроллера управления питанием на плате управления можно выделить следующие:

- измерение напряжений на всех линиях питания;
- включение и выключение элементов платы по команде процессора управления;
- выключение питания всей платы при возникновении аварийной ситуации.

Измерение и контроль напряжений на линиях питания происходит идентично ведущему микроконтроллеру, но контроль идёт по 24 линиям питания.

Схема подключения ведомого микроконтроллера управления питанием на плате управления изображена на рисунке 7.

Связь с ведущим микроконтроллером организована через UART, TX и RX которого расположены на контактах MUP1_FROM_MUP2 и MUP1_TO_MUP2. Также есть связь с процессором управления, но она предназначена только для обмена базовыми командами, которые будут рассмотрены далее. Связь с процессором управления идёт через UART контакты MUP2TX_PU и MUP2RX_PU.

					ПГУ 1.090301.31.001 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

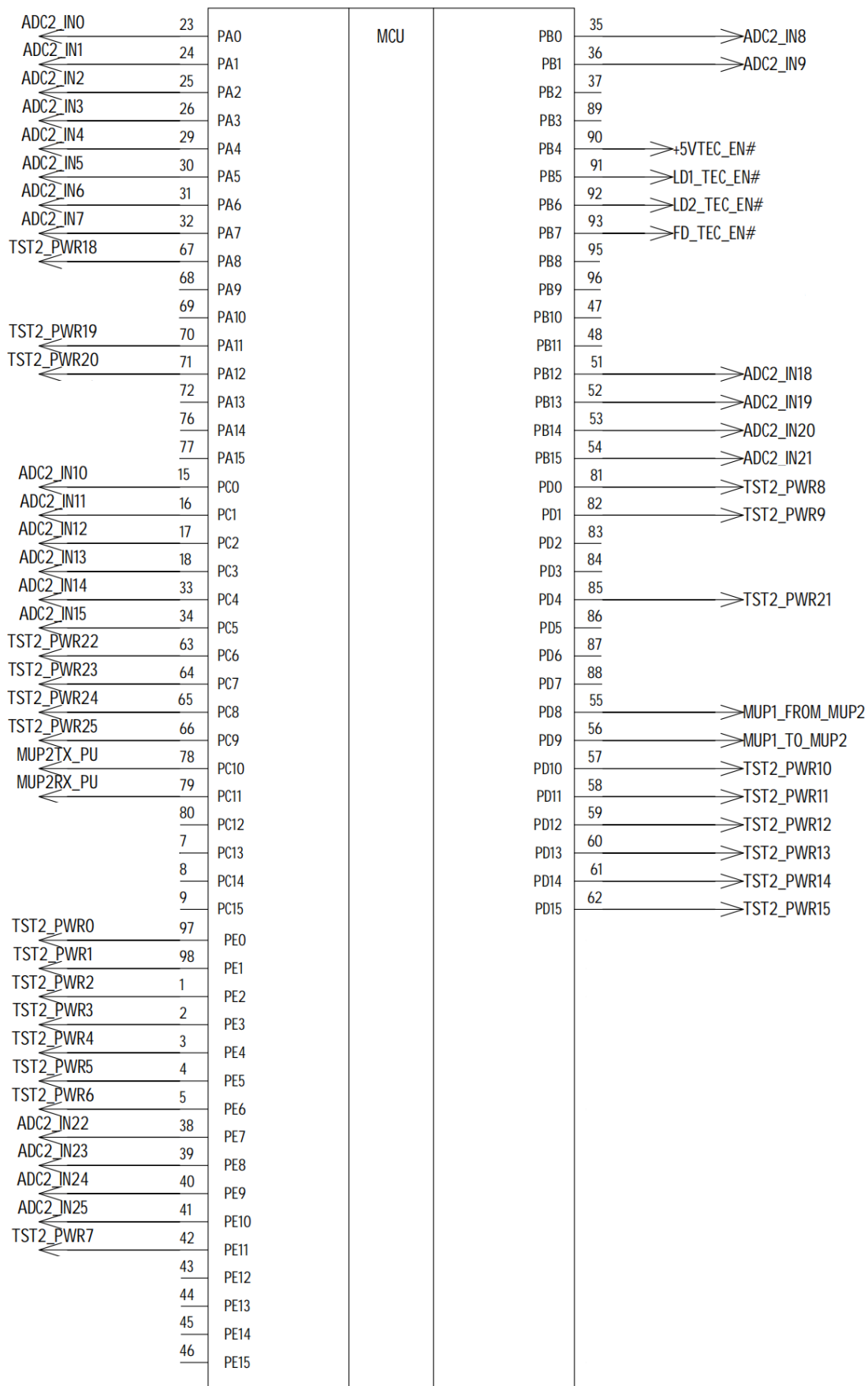


Рисунок 7 – Ведомый микроконтроллер управления питанием

Включение элементов происходит по команде ведущего микроконтроллера, который, в свою очередь, получает команду на включение или выключение от процессора управления. За это отвечают следующие контакты:

- контакт +5VTEC_EN включает общее питание лазеров;
- контакт LD1_TEC_EN включает лазер 1;
- контакт LD2_TEC_EN включает лазер 2;
- контакт FD_TEC_EN включает фотодиод.

2.2 Разработка протокола взаимодействия

2.2.1 Взаимодействие элементов системы

Взаимодействие элементов системы управления питанием происходит путём передачи пакетов запросов и уведомлений. Формат пакетов будет рассмотрен в пункте 2.1.2. UML диаграмма взаимодействия процессора и микроконтроллеров изображена на рисунке 8.

На UML диаграмме рассмотрены две операции:

- включение лазера по инициативе процессора;
- выключение модуля по нажатию кнопки питания.

Для включения лазера процессор управления составляет и отправляет запрос ведущему микроконтроллеру. Ведущий, в свою очередь, формирует и отправляет запрос уже ведомому микроконтроллеру, который включает лазер. После включения лазера, ведомый микроконтроллер возвращает результат операции ведущему, который, в свою очередь, возвращает ответ процессору управления со статусом операции.

При обнаружении нажатия кнопки питания микроконтроллер ячейки управления питанием отправляет уведомление процессору управления. Получив уведомление, процессор управления запускает процесс завершения работы, заключительным этапом которого является отправка запроса на выключение питания.

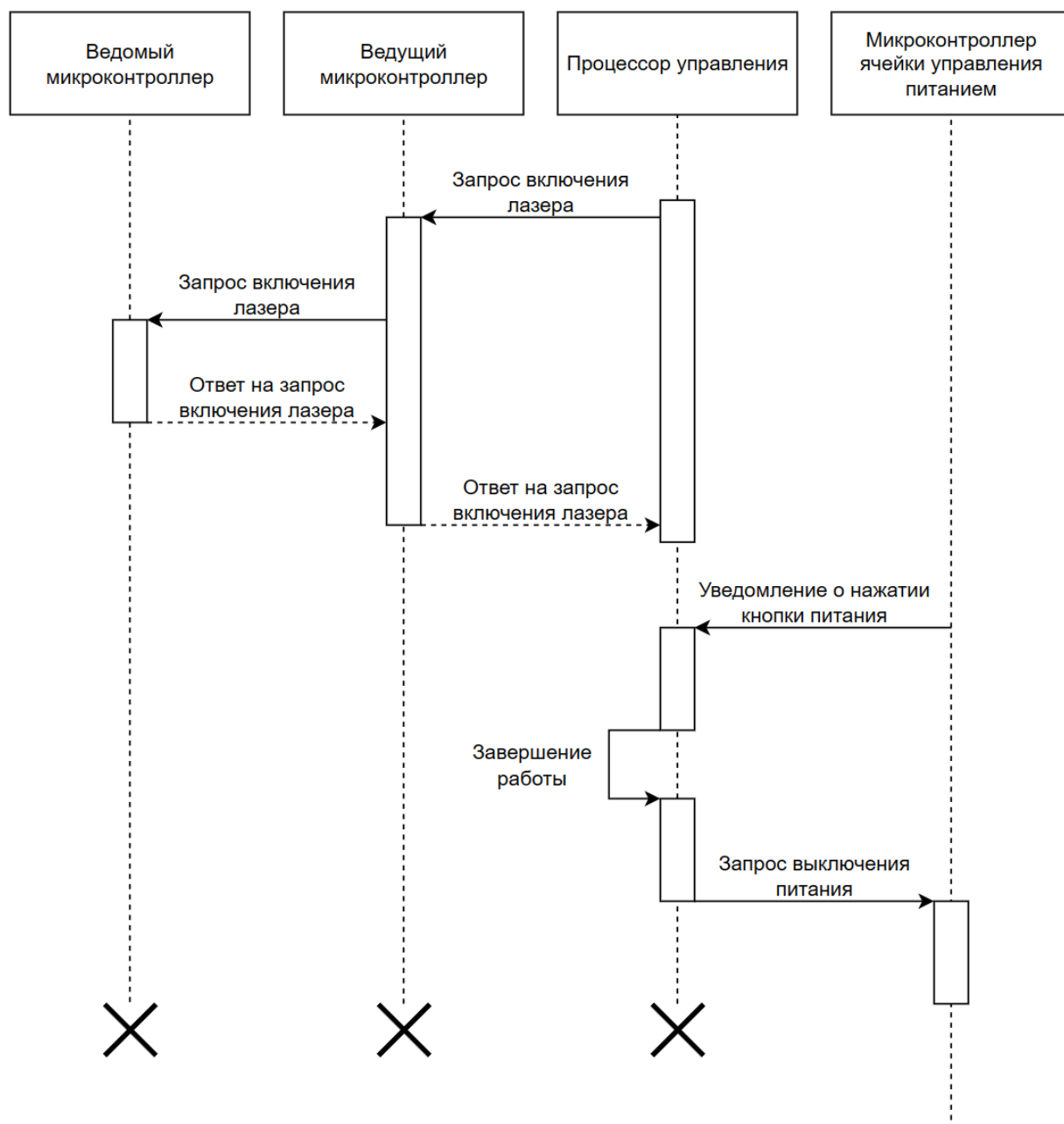


Рисунок 8 – Временная UML диаграмма взаимодействия

На рисунке 9 изображена временная диаграмма включения и выключения элементов системы управления питанием.

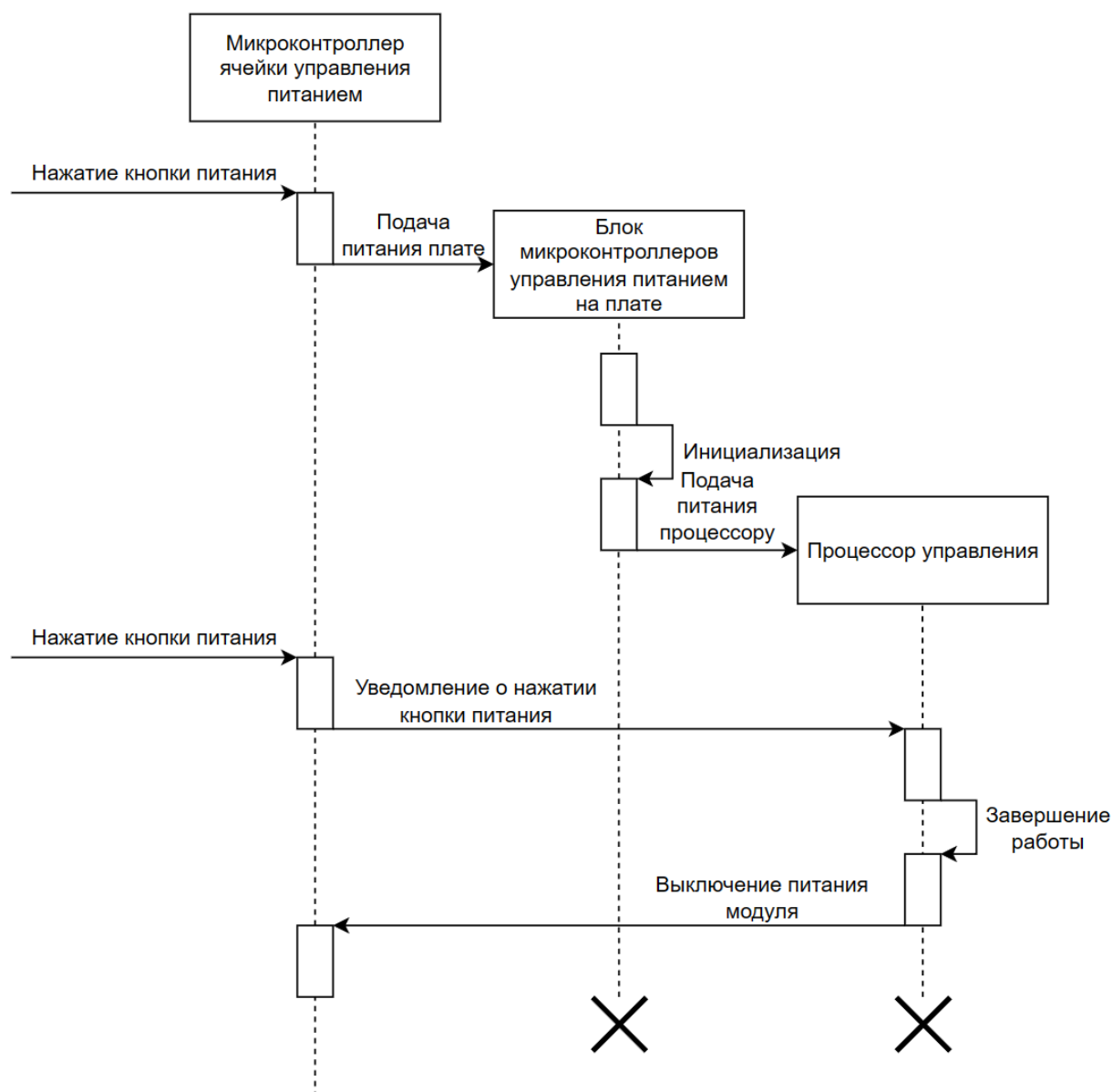


Рисунок 9 – Временная UML диаграмма включения и выключения элементов системы управления питанием

После нажатия кнопки питания микроконтроллер ячейки управления подаёт напряжение на платы модуля. Каждая плата подключена к питанию таким образом, чтобы сначала включились микроконтроллеры, проверяющие напряжение. После проверки напряжений на всех линиях ведущий в блоке микроконтроллер включает напряжение на всей плате.

После повторного нажатия кнопки питания микроконтроллер ячейки управления питанием отправляет уведомление процессору управления о данном событии. Получив уведомление, процессор завершает работу и

отправляет команду на выключение питания. В конце микроконтроллер ячейки управления питанием выключает питание всех плат в составе модуля.

2.2.2 Формат пакета информационного взаимодействия

Обмен данными процессора управления с программным обеспечением микроконтроллера осуществляется через интерфейс UART со следующими параметрами:

- скорость 115200 бит/сек.;
- биты данных 8;
- контроля чётности нет;
- стоповых бит 1;
- управления потоком нет.

Данные по интерфейсу UART передаются пакетами формата, представленного в таблице 3.

Таблица 3 – Формат пакета информационного взаимодействия процессора управления с микроконтроллерами

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
END	0	2	Управляющий код начала пакета со значением C00Ch
DEVICE	2	2	Код устройства (каждому микроконтроллеру присваивается свой уникальный код)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
SEQUENCE_NUMBER	4	4	Номер последовательности. При работе по схеме запрос- ответ номер последовательности в пакете ответа должен совпадать с номером последовательности в пакете запроса
SIZE	8	4	Длина поля DATA
CRC1	12	2	Контрольная сумма CRC- 16 CCITT в сетевом порядке байт. Вычисляется для потока байт полей DEVICE, SEQUENCE_NUMBER и SIZE
CRC2	14	2	Контрольная сумма CRC- 16 CCITT в сетевом порядке байт. Вычисляется для потока байт поля DATA
DATA	16	SIZE	Данные пакета

Взаимодействие осуществляется по двум схемам:

– по принципу запрос от процессора управления, ответ от микроконтроллера;

					ПГУ 1.090301.31.001 ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

– передача оповещения от микроконтроллера в процессор управления.

Стоит учесть, что передача уведомления от микроконтроллера не прерывает передачу данных от другого устройства, в то же время само уведомление передаётся в процессор управления единым блоком данных. Ответ от процессора управления к микроконтроллеру на уведомления не отсылаться.

В таблице 4 представлено перечисление возможных значений поля DEVICE для микроконтроллеров системы управления питания. Неуказанные значения принадлежат микроконтроллерам других систем.

Таблица 4 – Перечисление кодов устройств (поле DEVICE)

Имя устройства	Значение	Микроконтроллер	Схема взаимодействия	Описание
MKUP_MCU	05h	МК ЯУП	Запрос-ответ	МК ЯУП (внутренний функционал)
MKUP_POWER_BUTTON	06h	МК ЯУП	Уведомления	Кнопка включения/выключения устройства
MKUP_PU_MASTER	0Fh	МК УП	Запрос-ответ	МК УП основной
MKUP_PU_SLAVE	10h	МК УП	Запрос-ответ	МК УП вспомогательный

Формат поля DATA пакета информационного взаимодействия зависит от функциональной роли пакета. При работе по схеме запрос-ответ поле

DATA запроса содержит заголовок команды, за которым следуют параметры команды. Формат поля DATA запроса представлен в таблице 5.

Таблица 5 – Формат поля DATA запроса

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
COMMAND	0	4	Код команды
PARAMS	4	SIZE – sizeof(COMMAND)	Параметры команды

Формат поля PARAMS определяется значением поля COMMAND.

Поле DATA ответа на запрос содержит статус и, если это необходимо, данные. Поле заполняется только в случае успешного завершения запроса. Формат поля DATA ответа на запрос представлен в таблице 6.

Таблица 6 – Формат поля DATA ответа на запрос

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
COMMAND	0	4	Код команды, на которую отправляется ответ. Должен совпадать с кодом команды в запросе
STATUS	4	4	Статус завершения выполнения операции.
DATA	8	SIZE – sizeof(COMMAND) – sizeof(STATUS)	Данные ответа

Поле DATA уведомления содержит код уведомления, за которым следуют параметры уведомления. Формат поля DATA уведомления представлен в таблице 7.

Таблица 7 – Формат поля DATA уведомления

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
NOTIFY_CODE	0	4	Код уведомления
PARAMS	4	SIZE – sizeof(NOTIFY_CODE)	Параметры команды

Формат поля PARAMS определяется значением поля NOTIFY_CODE.

В таблице 8 перечислены пулы кодов команд и уведомлений, закреплённых за каждым конкретным устройством. Незанятые коды принадлежат микроконтроллерам других систем.

Таблица 8 – Перечисление кодов команд и уведомлений, закреплённых за каждым конкретным устройством

Имя устройства	Закрепленный интервал кодов команд/уведомлений
Общие команды для устройств, работающих по схеме запрос-ответ	01h – 80h
MKUP_POWER_BUTTON	281h – 300h
MKUP_MCU	301h – 380h
MKUP_PU_MASTER	781h – 800h
MKUP_PU_SLAVE	801h – 880h

2.2.3 Общие команды взаимодействия

По схеме запрос-ответ можно выделить ряд общих команд, которые поддерживаются каждым микроконтроллером, вне зависимости от его предназначения. Общие команды приведены в таблице 9.

Таблица 9 – Перечисление общих команд взаимодействия (COMMAND)

Имя команды	Значение	Описание
PING	01h	Проверка канала связи
DO_INTEGRITY_CONTROL	02h	Провести контроль целостности ВПО
GET_SOFTWARE_SIZE	03h	Получение размера образа ВПО
GET_SOFTWARE_BLOCK	04h	Получение данных блока образа ВПО
GET_VERSION	05h	Получение версии ВПО

Команда PING отправляет пакет с полем PARAMS заполненным произвольными данными от процессора управления до микроконтроллера. Успешным результатом является совпадение полей вернувшегося ответа с отправленным запросом.

Команда DO_INTEGRITY_CONTROL отправляет пакет только с кодом команды. Ответный пакет содержит контрольную сумму ВПО рассчитанную на микроконтроллере. На процессоре управления заранее известны контрольные суммы всех устройств. Благодаря чему процессор управления автоматически проверяет соответствие и целостность ВПО на микроконтроллерах.

Команда GET_SOFTWARE_SIZE содержит только код команды. Возвращаемое значение равно размеру ВПО микроконтроллера в байтах.

Команда GET_SOFTWARE_BLOCK запрашивает блок образа ВПО. Формат параметров запроса на получение данных блока образа ВПО представлен в таблице 10.

Таблица 10 – Формат параметров запроса на получение данных блока образа ВПО (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
OFFSET	0	4	Смещение запрашиваемого блока данных относительно начала образа ВПО
COUNT	4	4	Размер запрашиваемого блока данных ВПО

В таблице 11 представлен формат ответа на запрос получения данных блока образа ВПО.

Таблица 11 – Формат параметров ответа на запрос получения данных блока образа ВПО (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
DATA	0	N	Данные запрашиваемого блока образа ВПО

Команда GET_VERSION получает версию внутреннего программного обеспечения. Формат параметров запроса на получение версий ВПО состоит лишь из кода запроса, формат ответа представлен в таблице 12.

2.2.4 Команды взаимодействия процессора управления и микроконтроллера ячейки управления питанием

Внутреннее программное обеспечение микроконтроллера ячейки управления питанием поддерживает обработку общих команд, описанных в пункте 2.1.3, а также специализированных команд, которые перечислены в таблице 13.

Таблица 12 – Формат ответа на запрос версий ВПО (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Type	0	4	Значение типа принимает значения: 0 – Плата управления 1 – Плата шумогенератора 2 – Плата шифрования
MajorVersion	4	2	Старшая часть версии
MinorVersion	6	2	Младшая часть версии

Таблица 13 – Перечисление команд к панели управления питанием (COMMAND)

Имя команды	Значение	Описание
MKPI_MCU_POWER_OFF	301h	Выключить питание модуля
MKPI_MCU_REBOOT	302h	Перезагрузка модуля

По получению команды MKPI_MCU_POWER_OFF микроконтроллер выключает питание модуля квантовой сети. Формат запроса выключения питания содержит лишь код команды. Формат ответа на запрос выключения питания не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению команды MKPI_MCU_REBOOT микроконтроллер выключает и снова включает питание модуля. Формат запроса перезагрузки содержит лишь код команды.

Микроконтроллер ячейки управления питанием включает в себя механизм отправки уведомлений. Микроконтроллер отправляет уведомление с кодом POWER_KEY_DOWN (281h), когда кнопка питания нажата.

2.2.5 Команды взаимодействия процессора управления и ведущего микроконтроллера

Внутреннее программное обеспечение ведущего микроконтроллера на плате управления поддерживает обработку общих команд, описанных в пункте 2.1.3, а также специализированных команд, которые перечислены в таблице 14.

Таблица 14 – Перечисление команд к ведущему микроконтроллеру (COMMAND)

Имя команды	Значение	Описание
MKUP_PU_MASTER_SET_POWER_STATE	781h	Установить состояние питания элементов платы
MKUP_PU_MASTER_GET_POWER_STATE	782h	Получить состояние питания элементов платы
MKUP_PU_MASTER_SET_CONTROL_VALUE	783h	Установить контрольное значение канала
MKUP_PU_MASTER_GET_ERROR_STATE	784h	Получить состояние ошибок питания

Имя команды	Значение	Описание
MKUP_PU_MASTER_RESET_ERROR_STATE	785h	Сбросить ошибки питания
MKUP_PU_MASTER_SET_TIME	786h	Установить календарь
MKUP_PU_MASTER_TESTING	787h	Провести тестирование
MKUP_PU_MASTER_GET_TESTING_RESULT	788h	Получить результат тестирования
MKUP_PU_MASTER_JOURNAL_RECORDS_SIZE	789h	Получить размер журнала
MKUP_PU_MASTER_JOURNAL_RECORD_DATA	78Ah	Получить запись журнала
MKUP_PU_MASTER_JOURNAL_CLEAR	78Bh	Очистить запись журнала

По получению команды MKUP_PU_MASTER_SET_POWER_STATE микроконтроллер включает или выключение питания указанных элементов платы управления. Формат запроса содержит код команды, новое состояние и маску модулей, описание представлено в таблице 15. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению команды MKUP_PU_MASTER_GET_POWER_STATE микроконтроллер передаёт состояние включено или выключено питания указанного элемента платы управления. Формат запроса получения состояния питания элементов платы содержит только код команды.

Таблица 15 – Формат параметров запроса на установку состояния питания элементов платы управления (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Mask	0	4	Битовая маска элементов: 0 бит – общее питание лазеров 1 бит – питание лазера 1 2 бит – питание лазера 2 3 бит – питание фотодиода
State	4	4	Новое состояние: 0 – выключено 1 – включено

Ответ на запрос получения состояния питания элементов платы содержит битовую маску, где бит установленный в единицу означает, что элемент включён, а нуль значит, что элемент выключен. Формат ответа на запрос представлен в таблице 16.

Таблица 16 – Формат параметров ответа на запрос получения состояния питания элементов платы управления (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
State_Mask	0	4	Битовая маска элементов: 0 бит – общее питание лазеров 1 бит – питание лазера 1 2 бит – питание лазера 2 3 бит – питание фотодиода

По получению управляющей команды
MKUP_PU_MASTER_SET_CONTROL_VALUE микроконтроллер

устанавливает контрольное значение на заданную линию питания. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS. Формат запроса установки контрольного значения питания представлен в таблице 17.

Таблица 17 – Формат параметров запроса установки контрольного значения (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Chanel	0	4	Номер канала
ControlValue	4	8	Контрольное значение (мВ)

По получению команды MKUP_PU_MASTER_GET_ERROR_STATE микроконтроллер передаёт состояние ошибок питания платы управления. Формат запроса получения состояния ошибок питания содержит только код команды. Ответ на запрос получения состояния ошибок питания представлен в таблице 18.

Таблица 18 – Формат параметров ответа на запрос получения состояния ошибок питания платы управления (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Error_Mask	0	4	Битовая маска модулей платы управления: 0 бит – общее питание лазеров 1 бит – питание фотодиода Состояние: 0 – ошибка не зафиксирована 1 – ошибка зафиксирована

По получению управляющей команды MKUP_PU_MASTER_RESET_ERROR_STATE микроконтроллер выполняет сброс ошибок питания платы управления. Запрос сброса ошибок питания платы управления не содержит дополнительных данных. Ответ на запрос сброса ошибок питания платы управления не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению команды MKUP_PU_MASTER_SET_TIME микроконтроллер выполняет установку даты и времени. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS. Формат запроса установки календаря содержит код команды, дату и время, описание представлено в таблице 19.

Таблица 19 – Формат параметров запроса установки календаря (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Year	0	2	Год
Month	2	2	Месяц
DayOfWeek	4	2	День недели
Day	6	2	Число
Hour	8	2	Час
Minute	10	2	Минута
Second	12	2	Секунда
Milliseconds	14	2	Миллисекунда

По получению команды MKUP_PU_MASTER_TESTING микроконтроллер выполняет тест встроенного АЦП, а также АЦП ведомого. Запрос на проведение тестирования не содержит дополнительных данных. Ответ на запрос тестирования не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению управляющей команды MKUP_PU_MASTER_GET_TESTING_RESULT микроконтроллер передаёт результат тестирования. Запрос на получение результата тестирования не содержит дополнительных данных. Формат ответа на запрос получения результата тестирования представлен в таблице 20.

Таблица 20 – Формат параметров ответа на запрос получения результата тестирования (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
TestComplete	0	1	Значение: 0 – тестирование не окончено 1 – ошибка не зафиксирована 2 – зафиксирована ошибка
Channel	1	1	Номер канала
Voltage	2	2	Напряжение в мВ

По получению управляющей команды MKUP_PU_MASTER_JOURNAL_RECORDS_SIZE микроконтроллер передаёт размер журнала. Формат запроса на получение размера журнала содержит только код команды. Ответ на запрос содержит размер журнала.

По получению управляющей команды MKUP_PU_MASTER_JOURNAL_RECORD_DATA микроконтроллер передаёт указанный блок памяти. Формат запроса на получение записи журнала содержит код команды, смещение и размер записи, описание представлено в таблице 21.

Таблица 21 – Формат параметров запроса получения записи журнала (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Offset	0	4	Смещение начала возвращаемого участка
Size	4	4	Размер участка

Формат ответа на запрос получения записи журнала представлен в таблице 22.

Таблица 22 – Формат параметров ответа на запрос получения записи журнала (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Offset	0	4	Смещение начала возвращаемого участка
Size	4	4	Размер участка
Data	8	Size	Участок записи

По получению команды MKUP_PU_MASTER_JOURNAL_CLEAR микроконтроллер очищает журнал. Формат запроса на очистку содержит только код команды. Формат ответа на запрос очистки не содержит поля FDATA, успешность завершения отображает поле STATUS.

2.2.6 Команды взаимодействия ведущего и ведомого микроконтроллеров

Внутреннее программное обеспечение ведомого микроконтроллера на плате управления поддерживает обработку общих команд, описанных в

пункте 2.1.3, а также специализированных команд, которые перечислены в таблице 23.

Таблица 23 – Перечисление команд к ведущему микроконтроллеру (COMMAND)

Имя команды	Значение	Описание
MKUP_PU_SLAVE_VOLTAGE_ON	801h	Включение линий питания
MKUP_PU_SLAVE_VOLTAGE_OFF	802h	Выключение линий питания
MKUP_PU_SLAVE_RESET_WATCHDOG_TIMER	803h	Сброс следающего таймера
MKUP_PU_SLAVE_WATCHDOG_TIMER	804h	Сработал сторожевой таймер
MKUP_PU_SLAVE_TESTING	805h	Начать тестирование
MKUP_PU_SLAVE_GET_TESTING_RESULT	806h	Получить результат тестирования
MKUP_PU_SLAVE_POWER_FAULT	807h	Произошла ошибка питания
MKUP_PU_SLAVE_SET_CONTROL_VALUE	808h	Установить контрольное значение канала

По получению команды MKUP_PU_SLAVE_VOLTAGE_ON микроконтроллер включает питание указанных линий. Формат запроса содержит код команды и маску линий питания. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению команды MKUP_PU_SLAVE_VOLTAGE_OFF микроконтроллер выключает питание указанных линий. Формат запроса содержит код команды и маску линий питания. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению управляющей команды MKUP_PU_SLAVE_RESET_WATCHDOG_TIMER микроконтроллер сбрасывает сторожевой таймер. Формат запроса содержит только код команды. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению команды MKUP_PU_SLAVE_TESTING микроконтроллер выполняет тест встроенного АЦП. Запрос на проведение тестирования не содержит дополнительных данных. Ответ на запрос тестирования не содержит поля DATA, успешность завершения отображает поле STATUS.

По получению управляющей команды MKUP_PU_SLAVE_GET_TESTING_RESULT микроконтроллер передаёт результат тестирования. Запрос на получение результата тестирования не содержит дополнительных данных. Формат ответа на запрос получения результата тестирования представлен в таблице 24.

Микроконтроллер отправляет мастеру уведомление MKUP_PU_SLAVE_WATCHDOG_TIMER, когда не получает команды на сброс следящего таймера более 2 секунд. Уведомление о срабатывании следящего таймера не содержит дополнительных данных.

Таблица 24 – Формат параметров ответа на запрос получения результата тестирования (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
TestComplete	0	1	Значение: 0 – тестирование не окончено 1 – ошибка не зафиксирована 2 – зафиксирована ошибка
Channel	1	1	Номер канала
Voltage	2	2	Напряжение в мВ

Микроконтроллер отправляет мастеру уведомление MKUP_PU_SLAVE_POWER_FAULT, когда напряжение на линии питания выходит за допустимый предел. Формат уведомления об ошибке питания представлен в таблице 25.

Таблица 25 – Формат параметров ответа на запрос получения результата тестирования (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Channel	0	2	Номер канала
Voltage	2	2	Напряжение в мВ

По получению управляющей команды MKUP_PU_SLAVE_SET_CONTROL_VALUE микроконтроллер устанавливает контрольное значение на заданную линию питания. Формат ответа на запрос не содержит поля DATA, успешность завершения отображает поле STATUS. Формат запроса установки контрольного значения питания представлен в таблице 26.

Таблица 26 – Формат параметров запроса установки контрольного значения (PARAMS)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Описание
Chanel	0	4	Номер канала
ControlValue	4	8	Контрольное значение (мВ)

2.3 Разработка внутреннего программного обеспечения микроконтроллеров

2.3.1 Общие положения о процессе работы микроконтроллеров

Вся работа микроконтроллеров организована с помощью бесконечного цикла, поочерёдно вызывающего основные модули внутреннего программного обеспечения.

Каждый модуль внутреннего программного обеспечения устроен, так чтобы при вызове его главной функции, в заданной последовательности, исполнялись малые блоки кода. Таким образом, достигается эмуляция многозадачности. [4] Так, например, при проведении контроля целостности подсчёт контрольной суммы происходит в 4 этапа:

- инициализация таблицы перестановки для прямого подсчёта;
- проведение прямого подсчёта контрольной суммы;
- инициализация таблицы перестановки для обратного подсчёта;
- проведение обратного подсчёта контрольной суммы.

Между этими этапами микроконтроллер совершает другие действия. Такой механизм позволяет не задерживать реакцию микроконтроллера на внешние события, такие как исполнение полученного запроса или обнаружение нажатия кнопки питания, в тоже время не прерывая других исполняющихся процессов.

Внутреннее программное обеспечение микроконтроллеров можно разделить на 2 группы процессов:

– фоновые процессы, которые работают с момента инициализации до выключения питания;

– активные процессы, которые начинают работу при возникновении события, например прерывания.

Все фоновые процессы, кроме процесса приёма, уникальны для каждого микроконтроллера и зависят от функции, выполняемой микроконтроллером. Активные процессы также зависят от функции микроконтроллера, однако все микроконтроллеры включают в себя общие процессы ответа на базовые запросы.

2.3.2 Разработка общих процессов для всех микроконтроллеров

В группу общих процессов входят:

– фоновый процесс получения запросов;

– активные процессы обработки общих запросов к микроконтроллерам.

Процесс приёма происходит путём сбора пакета побайтово. Стартом сборки пакета и работы алгоритма является получение сигнатуры начала пакета. Сигнатура начала пакета — это последовательность из двух байт 0x0C и 0xC0.

Алгоритм работы фонового процесса приёма пакетов запросов изображён на рисунке 10.

Обработка общих запросов включает в себя обязательную поддержку следующих команд:

– команда проверки связи PING;

– команда получения размера внутреннего программного обеспечения GET_SOFTWARE_SIZE;

– команда получения блока данных внутреннего программного обеспечения GET_SOFTWARE_BLOCK;

– команда проверки целостности внутреннего программного обеспечения DO_INTEGRITY_CONTROL;

– команда получения версии GET_VERSION.

При получении команды PING микроконтроллер копирует поле DATA запроса в одноимённое поле ответа и отправляет его.

Команды GET_SOFTWARE_SIZE и GET_VERSION являются самыми простыми из всех. Данные команды возвращают константы прописанные во внутреннее программное обеспечение.

При получении команды GET_SOFTWARE_BLOCK микроконтроллер проверяет параметры запроса, если параметры верны, то копирует блок программной части микроконтроллера. Схема алгоритма получения блока внутреннего программного обеспечения микроконтроллера представлена на рисунке 11.

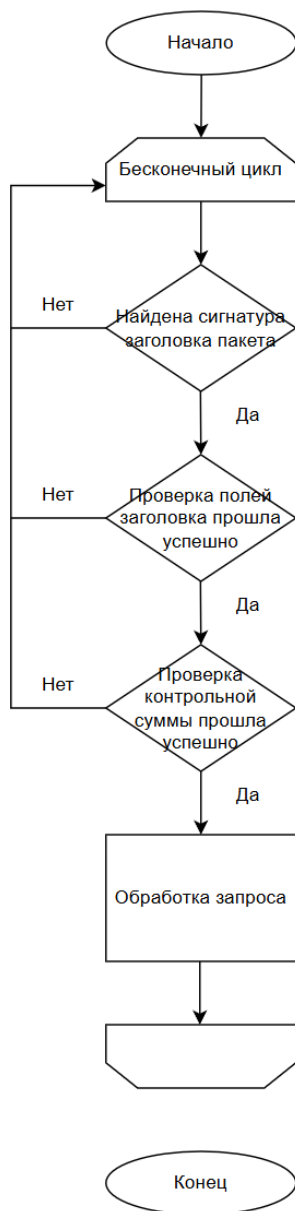


Рисунок 10 – Схема алгоритма приёма пакетов запросов

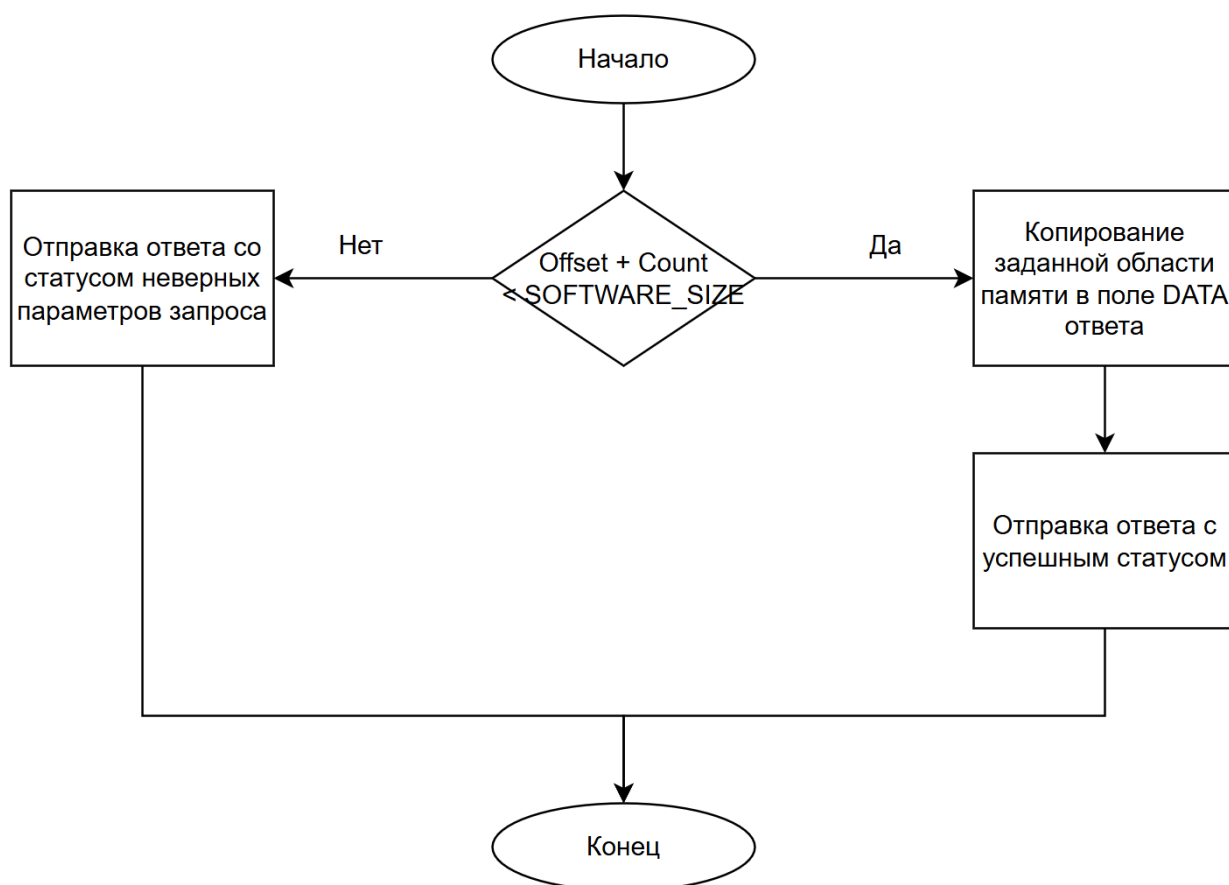


Рисунок 11 – Схема алгоритма получения блока данных внутреннего программного обеспечения

При получении команды DO_INTEGRITY_CONTROL микроконтроллер вычисляет контрольную сумму внутреннего программного обеспечения, используя алгоритм вычисления имитовставки описанный в ГОСТ Р 34.13 – 2015 в двух направлениях. Схема алгоритма вычисления контрольной суммы представлена на рисунке 12.

2.3.3 Разработка процессов микроконтроллера ячейки управления питанием

Микроконтроллер ячейки управления питанием включает в себя множество функций, однако в данной работе рассматривается только его часть в системе управления питанием.

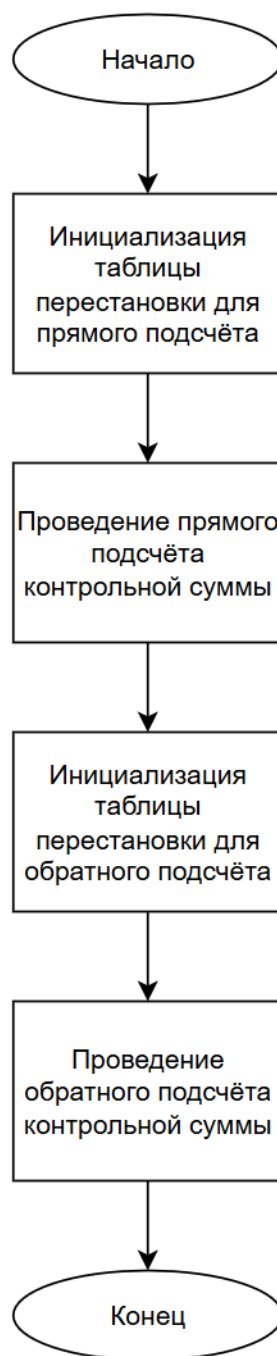


Рисунок 12 – Схема алгоритма вычисления контрольной суммы

Одним из фоновых процессов данного микроконтроллера является процесс получения статуса кнопки питания, с помощью опроса соответствующего контакта см. пункт 1.4.2. Для избежания проблемыдребезга контактов используется программная защита. Для однозначного определения статуса кнопки, она должна оставаться в неизменном положении в течение 50 миллисекунд.

Схема алгоритма получения статуса нажатия кнопки питания изображена на рисунке 13.

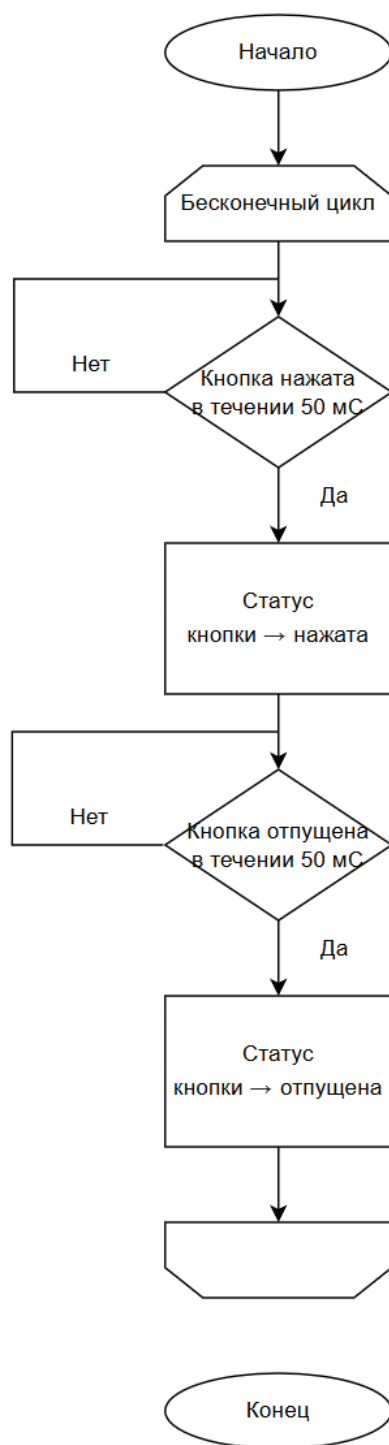


Рисунок 13 – Схема алгоритма получения статуса кнопки

Если статус кнопки питания переходит в нажатое состояние, то вызывается обработчик кнопки питания, совершающий определённые действия в зависимости от контекста. Схема алгоритма обработчика кнопки питания изображена на рисунке 14.

При получении запроса на выключение питания модуля микроконтроллер выключает питание. В тоже время при получении запроса на перезагрузку, микроконтроллер сначала выключает питание модуля, а потом, через 2 секунды, его включает.

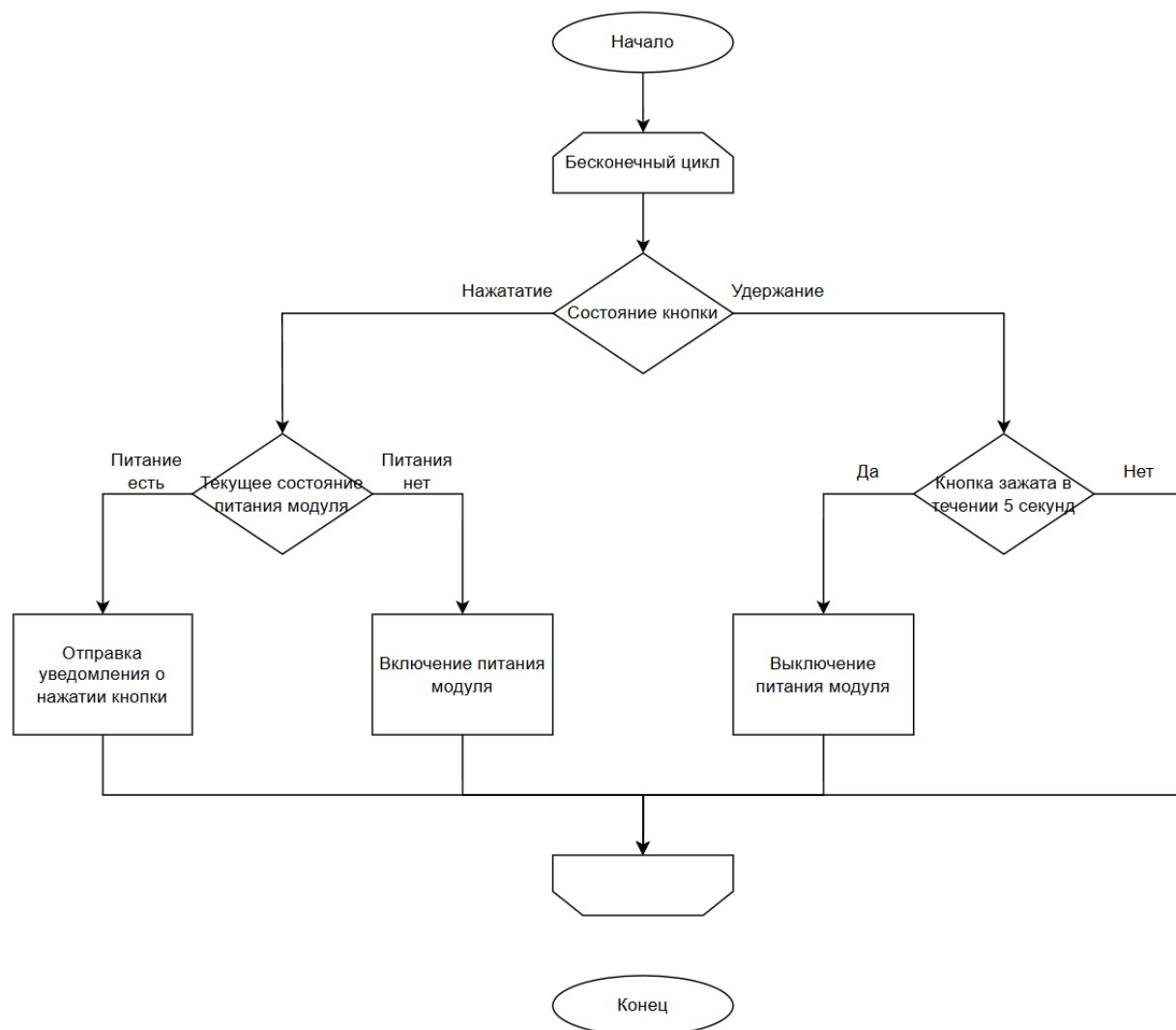


Рисунок 14 – Схема алгоритма обработчика кнопки питания

2.3.4 Разработка процессов ведущего микроконтроллера управления питанием на плате управления

Ведущий микроконтроллер управления питанием отвечает за измерение напряжения на заданных линиях питания, а также за организацию работы ведомого микроконтроллера.

Главная задача микроконтроллера это измерение напряжения. Она оформлена в виде фонового процесса, текущего на протяжении всего

времени работы модуля. Схема алгоритма измерения напряжения на линиях питания изображена на рисунке 15.

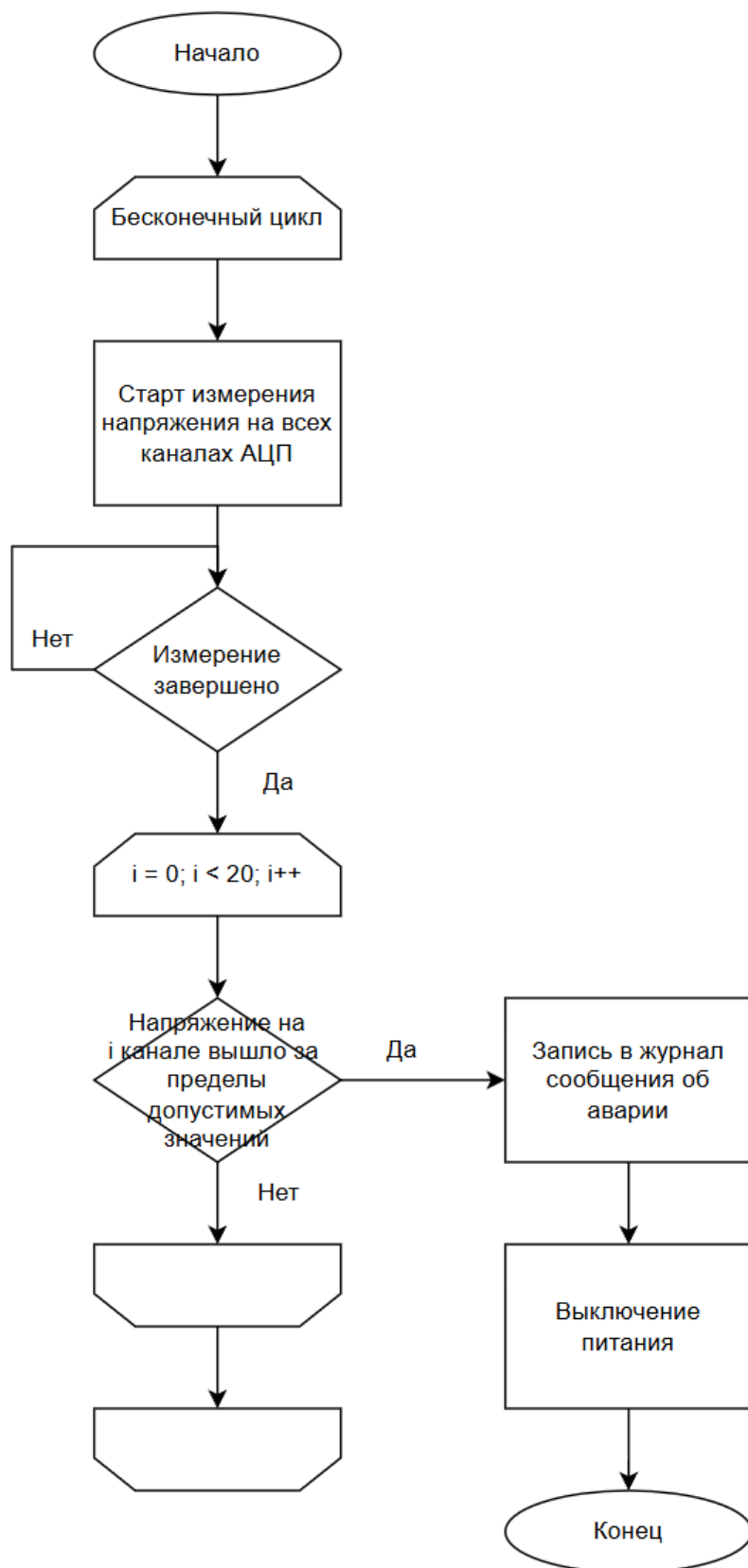


Рисунок 15 – Схема алгоритма измерения напряжения на линиях питания

Также ведущий микроконтроллер раз в секунду отправляет ведомому команду на сброс следящего таймера. Если от ведомого пришёл ответ, то работа продолжается, иначе происходит авария по причине потери связи.

Активно ведущий микроконтроллер выполняет работу по получению команд, согласно пункту 2.1.5.

Команда MKUP_PU_MASTER_SET_POWER_STATE заставляет микроконтроллер преобразовать пришедшую с командой маску устройств в маску соответствующую команде MKUP_PU_SLAVE_VOLTAGE_ON или, в случае если команда в режиме выключения, в маску соответствующую команде MKUP_PU_SLAVE_VOLTAGE_OFF.

Команда MKUP_PU_MASTER_GET_POWER_STATE заставляет микроконтроллер вернуть текущую маску питания устройств, путём копирования в поле DATA ответа на запрос.

Команда MKUP_PU_MASTER_SET_CONTROL_VALUE заставляет микроконтроллер высчитать 10% отклонения от заданного в команде напряжения и записать их в таблицу предельных значений.

По получению команды MKUP_PU_MASTER_GET_ERROR_STATE микроконтроллер передаёт состояние ошибок питания платы управления, путём копирования в поле DATA ответа на запрос.

По получению управляющей команды MKUP_PU_MASTER_RESET_ERROR_STATE микроконтроллер сбрасывает в ноль маску состояния ошибок питания.

При получении команды MKUP_PU_MASTER_SET_TIME микроконтроллер устанавливает для модуля RTC время и дату. Схема алгоритма установки даты и времени модуля RTC изображена на рисунке 16.

По получению команды MKUP_PU_MASTER_TESTING микроконтроллер начинает тестирование. В ходе тестирования выясняется может ли микроконтроллер определить выход напряжения за допустимые пределы. Каждая линия питания, измеряемая микроконтроллером,

					ПГУ 1.090301.31.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

сопровождается переключателем для смещения напряжения на линии в нужный момент (контакт TST_PWR). Тестирование происходит в 2 этапа:

- проверка повышенного напряжения на всех каналах;
- проверка пониженного напряжения на всех каналах.

Схема алгоритма тестирования микроконтроллера изображена на рисунке 17.

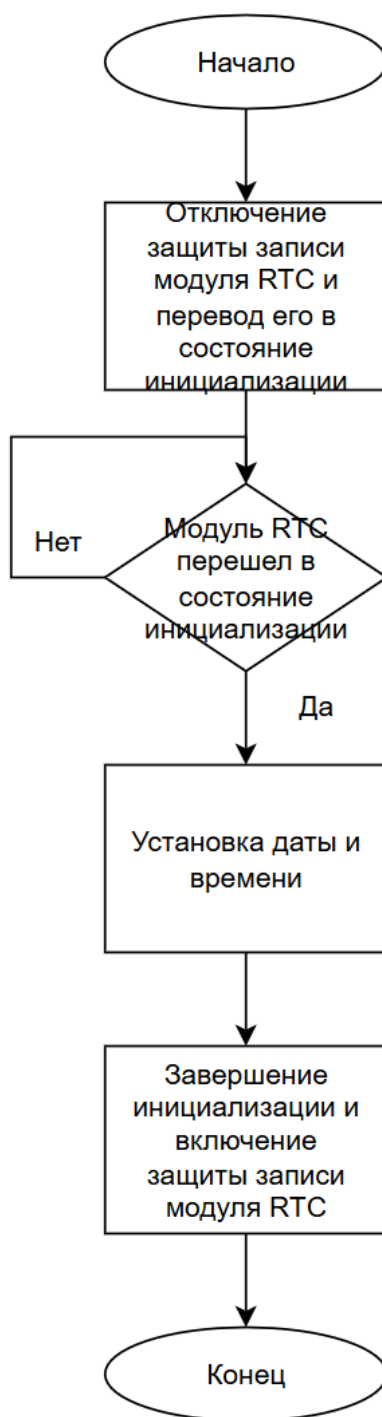


Рисунок 16 – Схема алгоритма установки даты и времени

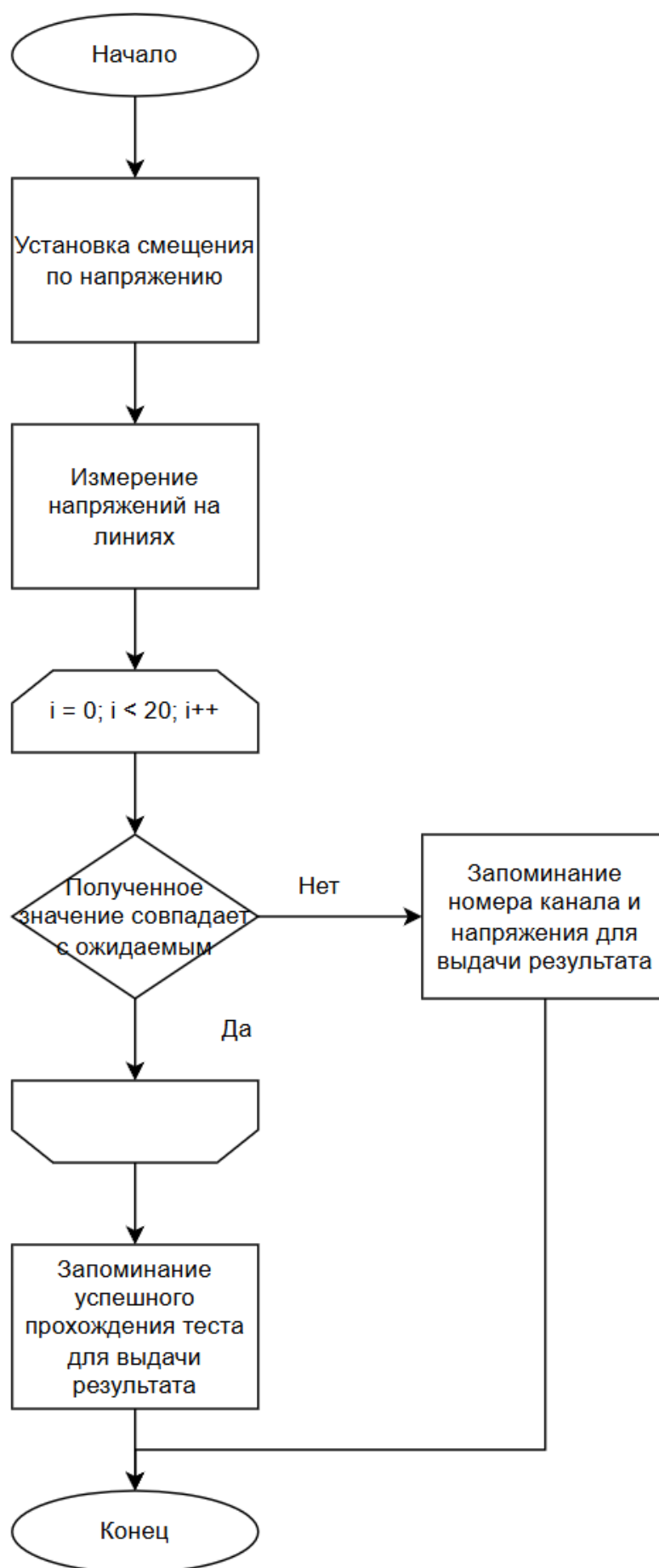


Рисунок 17 – Схема алгоритма тестирования микроконтроллера

По получению управляющей команды MKUP_PU_MASTER_GET_TESTING_RESULT микроконтроллер передаёт результат тестирования, записанный во время последнего теста.

По получению управляющей команды MKUP_PU_MASTER_JOURNAL_RECORDS_SIZE микроконтроллер передаёт размер журнала, вычисляемый как адрес конца минус адрес начала.

По получению управляющей команды MKUP_PU_MASTER_JOURNAL_RECORD_DATA микроконтроллер передаёт указанный блок памяти, если он удовлетворяет требованиям. Алгоритм выполнения данной команды аналогичен алгоритму получения блока данных описанного в пункте 2.3.2, но применим только в области журнала.

По получению команды MKUP_PU_MASTER_JOURNAL_CLEAR микроконтроллер очищает журнал.

2.3.5 Разработка процессов ведомого микроконтроллера управления питанием на плате управления

Ведомый микроконтроллер управления питанием отвечает за измерение напряжения на заданных линиях питания. Он подчиняется ведущему и, таким образом, расширяет функционал блока микроконтроллеров, как если бы это был один микроконтроллер.

Команды MKUP_PU_SLAVE_TESTING, MKUP_PU_SLAVE_GET_TESTING_RESULT и MKUP_PU_SLAVE_SET_CONTROL_VALUE исполняются аналогично ведущему. Также микроконтроллер постоянно ведёт контроль напряжений на заданных линиях аналогично ведущему.

Команды MKUP_PU_SLAVE_VOLTAGE_ON и MKUP_PU_SLAVE_VOLTAGE_OFF приходят от мастера и содержат маску включения или выключения питания на линиях.

Команда MKUP_PU_SLAVE_RESET_WATCHDOG_TIMER сбрасывает сторожевой таймер микроконтроллера.

					ПГУ 1.090301.31.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

Ведомый микроконтроллер отправляет уведомления ведущему в случае появления ошибок. Если сработал сторожевой таймер, то отправляется уведомление с командой MKUP_PU_SLAVE_WATCHDOG_TIMER, после чего микроконтроллер записывает ошибку в журнал и выключает питание. Если зарегистрирована ошибка в напряжении, то отправляется уведомление с командой MKUP_PU_SLAVE_POWER_FAULT, после чего микроконтроллер записывает ошибку в журнал и выключает питание.

2.4 Разработка драйвера обмена сообщениями с микроконтроллерами

В операционной системе уже присутствует драйвер контроллера UART канала, поэтому за фактическую отправку и приём данных отвечает он. Драйвер обмена сообщениями выполняет задачу проверки и обработки принятых пакетов, а также их отправку.

Выполнение запроса состоит из двух частей:

- отправка пакета запроса микроконтроллеру;
- получение ответа на запрос от микроконтроллера.

Для отправки драйвер получает на вход только поле DATA запроса, так как заголовок перед отправкой заполняется в драйвере. Из-за ограничения скорости передачи данных через UART канал драйвер не может сразу отправить пакет к микроконтроллеру. Запросы ставятся в очередь, которая со временем исполняется, однако запрос может находиться в ожидании только 5 секунд, иначе запрос отменяется по причине истечения времени выполнения. Схема алгоритма постановки запроса в очередь изображена на рисунке 18.

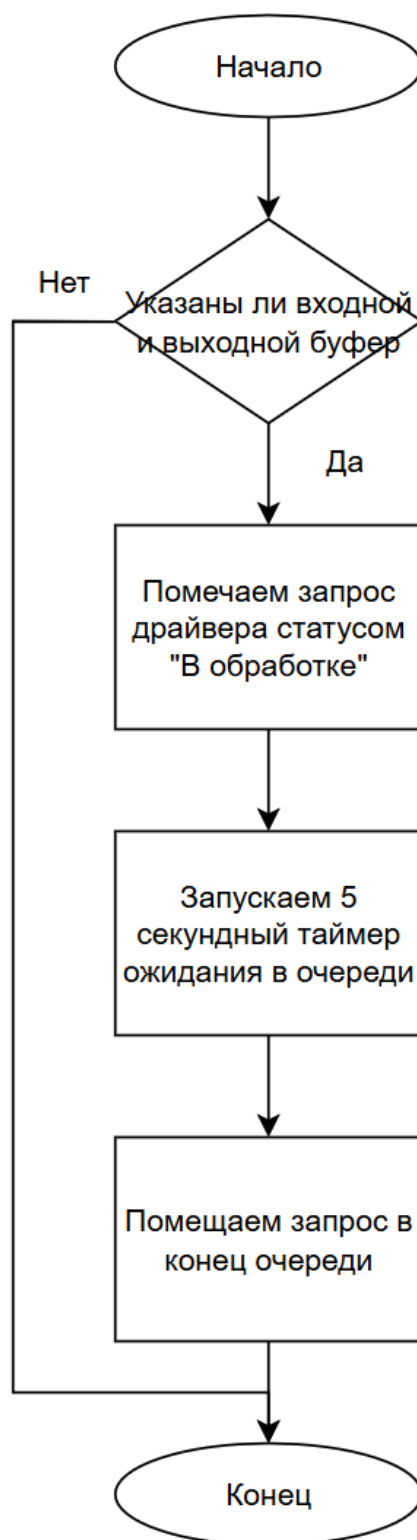


Рисунок 18 – Схема алгоритма постановки запроса в очередь

После того как запрос поступил на исполнение он заполняется данными заголовка и отправляется к соответствующему микроконтроллеру. Схема алгоритма отправки пакета изображена на рисунке 19.

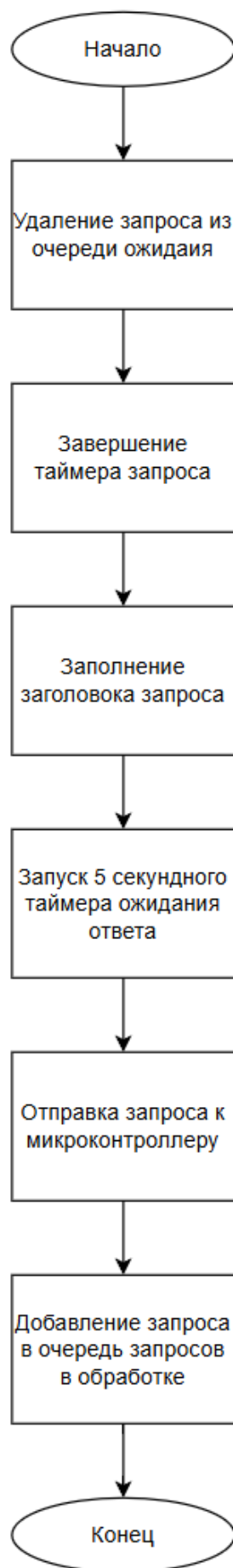


Рисунок 19 – Схема алгоритма отправки пакета

На протяжении всего времени существования драйвер поддерживает активной нить чтения информации поступающей от микроконтроллеров. Схема алгоритма получения входящих пакетов изображена на рисунке 20.



Рисунок 20 – Схема алгоритма получения входящих пакетов

3 Описание разработанной системы управления питанием

3.1 Руководство пользователя

Вследствие того, что система управления питанием скрыта от конечного пользователя, то работу системы, для рядового пользователя, можно свести к функционалу кнопки питания.

С помощью кнопки питания пользователь может выполнить следующие действия:

- включение модуля, при нажатии кнопки;
- выключение модуля, при повторном нажатии;
- экстренное выключение модуля, посредством отключения питания, при удержании кнопки в течении 5 секунд.

3.2 Руководство программиста

3.2.1 Общие положения

Главным в системе управления питанием модуля квантовой сети является процессор управления, расположенный на плате управления. Именно он является инициатором большинства действий в системе. За взаимодействие процессора управления с микроконтроллерами отвечает драйвер обмена сообщениями. Драйвер установлен в операционной системе и не требует установки или запуска, он автоматически стартует при запуске ОС. Для управления драйвером обмена сообщениями используются службы.

Взаимодействие с драйвером происходит по модели запрос-ответ. Служба открывает нужное устройство, ответственное за обмен с конкретным микроконтроллером, и отправляет запрос записи, входным буфером которого является поле DATA пакета согласно структуре описанной в пункте 2.1.2. После обработки драйвер завершает запрос со статусом завершения.

Каждый запрос к микроконтроллеру происходит по типовой модели, поэтому в данном руководстве рассмотрены только 2 уникальных способа взаимодействия:

- отправка пакета запроса микроконтроллеру;

– подписка на уведомление от микроконтроллера.

Оба вида взаимодействия осуществляются через команду DeviceIoControl, встроенную в подключаемый файл Windows.h.

3.2.2 Отправка сообщения микроконтроллеру

В качестве примера использована команда, которая должна включить фотодиод, однако это может быть любая другая команда, так как механизм взаимодействия не зависит от команды.

Первым этапом необходимо получить описатель устройства. В данном случае это можно сделать через символическую ссылку на устройство драйвера MKUP_PU_MASTER с помощью команды CreateFile. [5]

```
hDevice = CreateFile(L"\\\\.\\MKUP_PU_MASTER",  
GENERIC_WRITE|GENERIC_READ, 0, NULL, OPEN_EXISTING,  
FILE_ATTRIBUTE_NORMAL, NULL);
```

Далее следует подготовить буфер для приёма ответа и пакет для отправки. Заполняется только поле DATA пакета, так как заголовок пакета создаётся в драйвере. Пример пакета взаимодействия с командой MKUP_PU_MASTER_SET_POWER_STATE, предназначенного устройству MKUP_PU_MASTER представлен в таблице 27.

Таблица 27 – Пример пакета взаимодействия
MKUP_PU_MASTER_SET_POWER_STATE (DATA)

Название поля	Смещение поля, в байтах	Длина поля, в байтах	Значение
COMMAND	0	4	0x781
Mask	4	4	0x8
State	8	4	0x1

После того как все подготовлено можно отправить запрос с помощью команды DeviceIoControl. Драйвер поддерживает работу в асинхронном режиме, но для примера используется синхронный режим.

```
DeviceIoControl(hDevice, IOCTL_DEVICE_REQUEST, lpInBuffer,
sizeof(lpInBuffer), lpOutBuffer, sizeof(lpOutBuffer),
lpBytesReturned, NULL);
```

Успешность завершения операции определяется возвращаемым значением команды DeviceIoControl. Если DeviceIoControl возвращает TRUE, то содержимое ответа от микроконтроллера можно получить в выходном буфере.

Во время выполнения команды, с помощью инструмента логирования и внутреннего отладчика операционной системы, были получены лог драйвера об операциях в течении времени исполнения команды и дампы памяти отправленного и принятого пакетов. Лог и дампы памяти изображены на рисунках 21,22.

```
[INFO](MCUTransportInsertIntoWriteQueue:447) Успешная постановка запроса в очередь на отправку пакета в порт 0 от устройства
MKUP_MSU_MASTER.
[INFO](MCUTransportWriteSerialThread:130) Отправляется запрос с номером последовательности 7 устройству 15
[INFO](ReadComplete:383) Сработал ReadComplete при чтении данных из порта 0, статус 0x00000000.
[INFO](ProcessingHeader:90) Приняли заголовок ответа от устройства с кодом 15, с номером последовательности 7
[INFO](ReadComplete:383) Сработал ReadComplete при чтении данных из порта 0, статус 0x00000000.
[INFO](ProcessingData:298) Приняли данные ответа от устройства с кодом 15, с номером последовательности 7
```

Рисунок 21 – Лог драйвера обмена пакетами за период выполнения **команды**

B755CB88	000FC00C	00000007	Заголовок
B755CBC0	0000000C	FD823FA2	
B755CBC8	00000781	00000002	Поле DATA запроса
B755CBD0	00000001	CCCCCCCC	
B017B4F8	00000781	00000000	Поле DATA ответа

Рисунок 22 – Данные пакетов

3.2.3 Подписка на уведомление от микроконтроллера

В качестве примера будет использовано уведомление микроконтроллера ячейки управления питанием о нажатии кнопки питания.

Первым этапом пользователь драйвера должен получить описатель устройства. Это можно сделать через символическую ссылку на устройство драйвера MKUP_POWER_BUTTON с помощью команды CreateFile.

```
hDevice = CreateFile(L"\\\\.\\MKUP_POWER_BUTTON",
GENERIC_WRITE|GENERIC_READ, 0, NULL, OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL, NULL);
```

Далее следует подготовить буфер для приёма уведомления. В случае с уведомлением от микроконтроллера ячейки управления питанием о нажатии кнопки питания можно ограничиться буфером в 4 байта, так как уведомление содержит только код команды.

После того как все подготовлено можно подписаться на уведомление с помощью команды DeviceIoControl. Подписаться можно как в синхронном, так и в асинхронном режиме.

```
DeviceIoControl(hDevice, IOCTL_NOTIFY, NULL, NULL, lpOutBuffer,
sizeof(lpOutBuffer), lpBytesReturned, lpOverlapped);
```

Драйвер завершит запрос когда от микроконтроллера ячейки управления питанием придёт уведомление о нажатии кнопки питания. Это и будет означать что уведомление пришло, а в выходном буфере можно посмотреть поле DATA.

С помощью внутреннего отладчика операционной системы, был получен дамп памяти пакета уведомления. Пакет уведомления изображён на рисунке 23.

C12C1F48	0006C00C	00000000	Заголовок
C12C1F50	00000004	412C5877	
C12C1F58	00000281	00000000	Поле DATA уведомления

Рисунок 23 – Данные пакета уведомления

3.3 Описание внутреннего программного обеспечения микроконтроллера ячейки управления питанием

Внутреннее программное обеспечение микроконтроллера разбито на отдельные модули, каждый из которых отвечает за свои функции. Модули разделены по разным файлам.

3.3.1 Модуль TxBuffer

Данный модуль представляет собой абстракцию для работы с UART Tx. Модуль включает базовые функции для работы Tx канала.

Основные функции модуля:

					ПГУ 1.090301.31.001 ПЗ	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

- функция TxBufferInitialize инициализирует и настраивает DMA контроллер UART Tx для работы;
- функция TxBufferFree деинициализирует DMA контроллер UART;
- функция TxBufferFlush очищает FIFO очередь буфера;
- функция TxHandler производит отправку FIFO очереди буфера;
- функция TxPutByte добавляет в FIFO очередь буфера один байт;
- функция TxPutBuffer добавляет в FIFO очередь буфера содержимое другого буфера.

3.3.2 Модуль RxBuffer

Данный модуль представляет собой абстракцию для работы с UART Rx. Модуль включает базовые функции для работы Rx канала.

Основные функции модуля:

- функция RxBufferInitialize инициализирует и настраивает DMA контроллер UART Rx для работы;
- функция RxBufferFree деинициализирует DMA контроллер UART;
- функция RxBufferFlush очищает FIFO очередь буфера;
- функция RxGetByte забирает из FIFO очереди буфера один байт;
- функция RxGetBuffer забирает из FIFO очереди буфера определённое количество байт, если это возможно.

3.3.3 Модуль Main

Данный модуль является точкой входа внутреннего программного обеспечения микроконтроллера. Здесь объявляются прерывания и проходит инициализация. Модуль располагает единственной функцией main.

Функция main запускает бесконечный цикл, в котором работают остальные модули, воссоздавая многопоточность.

3.3.4 Модуль System

Данный модуль ответствен за внутренние процессы микроконтроллера и контроль GPIO портов. Именно тут происходит обработка всех действий, которые совершает микроконтроллер.

Основные функции модуля:

					ПГУ 1.090301.31.001 ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		

- функция `SystemInitialize` производит настройку GPIO портов и прерываний, заводит внутренний таймер, а также инициализирует периферию микроконтроллера;
- функция `TimerHandler` обрабатывает прерывание таймера, которое возникает один раз в секунду;
- функция `Timer` возвращает количество тактов таймера с предыдущего вызова, что позволяет измерять время между циклами;
- функция `Power` включает или выключает питание модуля;
- функция `Reboot` запускает процесс перезапуска питания;
- функция `RebootBackground` выполняет процесс перезапуска питания;
- функция `KbBackground` опрашивает состояние кнопки питания;
- функции `Crc16` и `Crc32` подсчитывают кумулятивную сумму;
- функция `SystemBackground` вызывает функции `KbBackground` и `RebootBackground`.

3.3.5 Модуль Channel

Данный модуль обеспечивает связь микроконтроллера ячейки управления питанием с процессором управления, расположенным на плате управления, используя UART.

Основные функции модуля:

- функция `ChannelInitialize` инициализирует интерфейс, производит настройку GPIO контактов и прерываний связанных с UART, а также инициализирует `TxBuffer` и `RxBuffer`;
- функция `CheckPacket` проверяет кумулятивную сумму пакета, чтобы удостовериться в корректности пришедших данных;
- функция `CreatePacket` создаёт пакет для отправки по UART каналу;
- функция `RecvByte` собирает поток байт с UART канала в буфер, если в потоке найдена сигнатура пакета;
- функция `RecvPacket` проверяет данные пакета;

- функция ProcessWorkPacket представляет собой набор обработчиков, из которых выбирается соответствующий по коду команды;
- функция ProcessKey выполняет действия в зависимости от статуса кнопки питания;
- функция CtrlBackground подсчитывает контрольную сумму внутреннего программного обеспечения;
- функция RecvBackground организует процесс приёма информации с UART канала;
- функция ChannelBackground вызывает функции RecvBackground и ProcessKey.

3.4 Описание внутреннего программного обеспечения ведущего микроконтроллера управления питанием на плате управления

Внутреннее программное обеспечение микроконтроллера разбито на отдельные модули, каждый из которых отвечает за свои функции. Модули разделены по разным файлам.

3.4.1 Модуль TxBuffer

Данный модуль представляет собой абстракцию для работы с UART Tx. Модуль включает базовые функции для работы Tx канала.

Основные функции модуля:

- функция TxBufferInitialize инициализирует и настраивает DMA контроллер UART Tx для работы;
- функция TxBufferFree деинициализирует DMA контроллер UART;
- функция TxBufferFlush очищает FIFO очередь буфера;
- функция TxHandler производит отправку FIFO очереди буфера;
- функция TxPutByte добавляет в FIFO очередь буфера один байт;
- функция TxPutBuffer добавляет в FIFO очередь буфера содержимое другого буфера.

3.4.2 Модуль RxBuffer

Данный модуль представляет собой абстракцию для работы с UART Rx. Модуль включает базовые функции для работы Rx канала.

Основные функции модуля:

- функция RxBufferInitialize инициализирует и настраивает DMA контроллер UART Rx для работы;
- функция RxBufferFree деинициализирует DMA контроллер UART;
- функция RxBufferFlush очищает FIFO очередь буфера;
- функция RxGetByte забирает из FIFO очереди буфера один байт;
- функция RxGetBuffer забирает из FIFO очереди буфера определённое количество байт, если это возможно.

3.4.3 Модуль Main

Данный модуль является точкой входа внутреннего программного обеспечения микроконтроллера. Здесь объявляются прерывания и проходит инициализация. Модуль располагает единственной функцией main.

Функция main запускает бесконечный цикл, в котором работают остальные модули, воссоздавая многопоточность.

3.4.4 Модуль System

Данный модуль ответствен за внутренние процессы микроконтроллера и контроль GPIO портов. Именно тут происходит обработка всех действий, которые совершает микроконтроллер.

Основные функции модуля:

- функция SystemInitialize инициализирует таблицу контроля напряжения, производит настройку GPIO портов и прерываний, заводит внутренний таймер, а также инициализирует периферию микроконтроллера;
- функция TimerHandler обрабатывает прерывание таймера, которое возникает один раз в секунду;
- функция Timer возвращает количество тактов таймера с предыдущего вызова, что позволяет измерять время между циклами;

- функция SetCalendar преобразует структуру календаря во время с 1 января 1970 года и устанавливает это значение в RTC модуль для дальнейшего использования;
- функция GetCalendar получает время с RTC модуля, преобразует его в структуру календаря и возвращает это значение;
- функция PowerOn включает питание заданных элементов платы;
- функция PowerOff выключает питание заданных элементов платы;
- функции Crc16 и Crc32 подсчитывают кумулятивную сумму.

3.4.5 Модуль PowerCheck

Данный модуль выполняет проверку напряжения на линиях питания в течение всего времени работы модуля квантовой сети.

Основные функции модуля:

- функция PowerCheckInitialize настраивает АЦП и его каналы для измерения напряжения на линиях питания;
- функция VoltageCalc подсчитывает реальное значение напряжения на канале АЦП;
- функция SetLimits устанавливает предельные значения для заданной линии питания;
- функция TestMode запускает тестирование АЦП;
- функция GetTestResult возвращает результаты тестирования АЦП;
- функция PowerADCBackground ведёт подсчёт напряжений на каналах АЦП и записывает их для дальнейшего использования;
- функция PowerCheckBackground сравнивает значения, полученные с помощью АЦП, со значениями из таблицы контроля напряжений, а также проводит тестирование АЦП, если оно запущено;
- функция PowerCheckBackground вызывает функции PowerADCBackground и PowerCheckBackground.

3.4.6 Модуль Channel

Данный модуль обеспечивает связь ведущего микроконтроллера с ведомым, а также с процессором управления, используя для этого UART каналы.

Основные функции модуля:

- функция ChannelInitialize инициализирует интерфейсы, производит настройку GPIO контактов и прерываний связанных с UART, а также инициализирует TxBuffer и RxBuffer для каждого канала;
- функция CheckPacket проверяет кумулятивную сумму пакета, чтобы удостовериться в корректности пришедших данных;
- функция CreatePacket создаёт пакет для отправки по UART каналу;
- функция RecvByte_PU собирает поток байт с UART канала процессора управления в буфер, если в потоке найдена сигнатура пакета;
- функция RecvByte_MCUS собирает поток байт с UART канала ведомого микроконтроллера в буфер, если в потоке найдена сигнатура пакета;
- функция RecvPacket проверяет данные пакета;
- функция ProcessWorkPacket_PU представляет собой набор обработчиков для команд процессора управления, из которых выбирается соответствующий по коду команды;
- функция ProcessWorkPacket_MCUS представляет собой набор обработчиков для команд ведомого микроконтроллера, из которых выбирается соответствующий по коду команды;
- функция CtrlBackground подсчитывает контрольную сумму внутреннего программного обеспечения;
- функция RecvBackground организует процесс приёма информации с UART канала;
- функция PowerCheckBackground_MCUM проверяет наличие ошибок питания на ведущем микроконтроллере и запускает аварию, если ошибки есть;

					ПГУ 1.090301.31.001 ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

– функция PowerCheckBackground_MCUS проверяет наличие ошибок питания на ведомом микроконтроллере и запускает аварию, если ошибки есть;

– функция PowerCheckBackground организует работу модуля контроля напряжений;

– функция PowerCheckSetAlarm создает запись об аварии в журнале;

– функция ChannelBackground последовательно вызывает функции CtrlBackground, RecvBackground, PowerCheckBackground_MCUM, PowerCheckBackground_MCUS и PowerCheckBackground.

3.4.7 Модуль Log

Данный модуль отвечает за ведение журнала. Журнал представляет из себя кольцевой буфер расположенный в EEPROM памяти.

Основные функции модуля:

– функция LogInitialize подготавливает кольцевой буфер в памяти, если его нет или проверяет целостность буфера, если он есть;

– функция Set добавляет запись в журнал;

– функция Get вычитывает запись из журнала;

– функция Clear очищает запись журнала;

– функция LogBackground организует работу журнала и синхронизирует к нему доступ.

3.5 Описание внутреннего программного обеспечения ведомого микроконтроллера управления питанием на плате управления

Внутреннее программное обеспечение микроконтроллера разбито на отдельные модули, каждый из которых отвечает за свои функции. Модули разделены по разным файлам.

3.5.1 Модуль TxBuffer

Данный модуль представляет собой абстракцию для работы с UART Tx. Модуль включает базовые функции для работы Tx канала.

Основные функции модуля:

					ПГУ 1.090301.31.001 ПЗ	Лист
						78
Изм.	Лист	№ докум.	Подпись	Дата		

- функция TxBufferInitialize инициализирует и настраивает DMA контроллер UART Tx для работы;
- функция TxBufferFree деинициализирует DMA контроллер UART;
- функция TxBufferFlush очищает FIFO очередь буфера;
- функция TxHandler производит отправку FIFO очереди буфера;
- функция TxPutByte добавляет в FIFO очередь буфера один байт;
- функция TxPutBuffer добавляет в FIFO очередь буфера содержимое другого буфера.

3.5.2 Модуль RxBuffer

Данный модуль представляет собой абстракцию для работы с UART Rx. Модуль включает базовые функции для работы Rx канала.

Основные функции модуля:

- функция RxBufferInitialize инициализирует и настраивает DMA контроллер UART Rx для работы;
- функция RxBufferFree деинициализирует DMA контроллер UART;
- функция RxBufferFlush очищает FIFO очередь буфера;
- функция RxGetByte забирает из FIFO очереди буфера один байт;
- функция RxGetBuffer забирает из FIFO очереди буфера определённое количество байт, если это возможно.

3.5.3 Модуль Main

Данный модуль является точкой входа внутреннего программного обеспечения микроконтроллера. Здесь объявляются прерывания и проходит инициализация. Модуль располагает единственной функцией main.

Функция main запускает бесконечный цикл, в котором работают остальные модули, воссоздавая многопоточность.

3.5.4 Модуль System

Данный модуль ответствен за внутренние процессы микроконтроллера и контроль GPIO портов. Именно тут происходит обработка всех действий, которые совершает микроконтроллер.

Основные функции модуля:

					ПГУ 1.090301.31.001 ПЗ	Лист
						79
Изм.	Лист	№ докум.	Подпись	Дата		

- функция `SystemInitialize` инициализирует таблицу контроля напряжения, производит настройку GPIO портов и прерываний, заводит внутренний таймер, а также инициализирует периферию микроконтроллера;
- функция `TimerHandler` обрабатывает прерывание таймера, которое возникает один раз в секунду;
- функция `Timer` возвращает количество тактов таймера с предыдущего вызова, что позволяет измерять время между циклами;
- функция `PowerOn` включает питание заданных элементов платы;
- функция `PowerOff` выключает питание заданных элементов платы;
- функции `Crc16` и `Crc32` подсчитывают кумулятивную сумму.

3.5.5 Модуль `PowerCheck`

Данный модуль выполняет проверку напряжения на линиях питания в течение всего времени работы модуля квантовой сети.

Основные функции модуля:

- функция `PowerCheckInitialize` настраивает АЦП и его каналы для измерения напряжения на линиях питания;
- функция `VoltageCalc` подсчитывает реальное значение напряжения на канале АЦП;
- функция `SetLimits` устанавливает предельные значения для заданной линии питания;
- функция `TestMode` запускает тестирование АЦП;
- функция `GetTestResult` возвращает результаты тестирования АЦП;
- функция `PowerADCBackground` ведёт подсчёт напряжений на каналах АЦП и записывает их для дальнейшего использования;
- функция `PowerCheckBackground` сравнивает значения, полученные с помощью АЦП, со значениями из таблицы контроля напряжений, а также проводит тестирование АЦП, если оно запущено;
- функция `PowerCheckBackground` вызывает функции `PowerADCBackground` и `PowerCheckBackground`.

3.5.6 Модуль Channel

Данный модуль обеспечивает связь ведомого микроконтроллера с ведущим, а также с процессором управления, используя для этого UART каналы.

Основные функции модуля:

- функция ChannelInitialize инициализирует интерфейсы, производит настройку GPIO контактов и прерываний связанных с UART, а также инициализирует TxBuffer и RxBuffer для каждого канала;
- функция CheckPacket проверяет кумулятивную сумму пакета, чтобы удостовериться в корректности пришедших данных;
- функция CreatePacket создаёт пакет для отправки по UART каналу;
- функция RecvByte_PU собирает поток байт с UART канала процессора управления в буфер, если в потоке найдена сигнатура пакета;
- функция RecvByte_MCUM собирает поток байт с UART канала ведущего микроконтроллера в буфер, если в потоке найдена сигнатура пакета;
- функция RecvPacket проверяет данные пакета;
- функция ProcessWorkPacket_PU представляет собой набор обработчиков для команд процессора управления, из которых выбирается соответствующий по коду команды;
- функция ProcessWorkPacket_MCUM представляет собой набор обработчиков для команд ведущего микроконтроллера, из которых выбирается соответствующий по коду команды;
- функция CtrlBackground подсчитывает контрольную сумму внутреннего программного обеспечения;
- функция RecvBackground организует процесс приёма информации с UART канала;
- функция PowerCheckBackground организует работу модуля контроля напряжений;
- функция PowerCheckSetAlarm создает запись об аварии в журнале;

					ПГУ 1.090301.31.001 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

– функция ChannelBackground последовательно вызывает функции CtrlBackground, RecvBackground и PowerCheckBackground.

3.6 Описание драйвера обмена сообщениями с микроконтроллерами

3.6.1 Модуль MCUTransportCommunication

Данный модуль реализует базовый функционал драйвера.

Основные функции модуля:

– функция MCUTransportCommunicationUnload осуществляет завершение всех исполняющихся процессов и очистку памяти при выгрузке драйвера;

– функция MCUTransportAddDevice осуществляет добавление нового устройства драйвера;

– функция DriverEntry является точкой входа драйвера;

3.6.2 Модуль MCUTransportCommunicationSerialWrite

Данный модуль отвечает за отправку пакетов к микроконтроллерам.

Основные функции модуля:

– функция MCUTransportWriteSerialThread осуществляет отправку пакетов к микроконтроллерам;

– функция CompleteWriteRequestAlreadySended завершает все запросы, находящиеся на обработке;

– функция MCUTransportHeaderFill заполняет заголовок пакета.

3.6.3 Модуль MCUTransportCommunicationSerialRead

Данный модуль отвечает за получение пакетов от микроконтроллеров.

Основные функции модуля:

– функция ProcessingStartSign осуществляет поиск сигнатуры пакета;

– функция ProcessingHeader осуществляет проверку заголовка по контрольной сумме;

– функция ProcessingRequestPacket завершает обработку пакета для запроса;

- функция ProcessingNotifyPacket завершает обработку пакета для уведомления;
- функция ProcessingData проверяет поле DATA пакета;
- функция ReadComplete отложенный вызов при получении данных;
- функция MCUTransportReadSerialThread осуществляет получение пакетов от микроконтроллеров.
- Модуль MCUTransportRequestDevices
- Данный модуль организует очередь отправки.
- Основные функции модуля:
 - функция MCUTransportNotSendedyetRequestTimeout завершает запрос по истечении времени ожидания в очереди;
 - функция MCUTransportAlreadySendedRequestTimeout завершает запрос по истечении времени ожидания ответа;
 - функция MCUTransportInsertIntoWriteQueue помещает запрос в очередь ожидания.

4 Безопасность жизнедеятельности

4.1 Меры безопасности

Для безопасного программирования элементов модуля и введения системы управления питанием в работу необходимо соблюдать следующие требования:

– при работе с изделием должны быть соблюдены требования безопасности, указанные в приказе «Об утверждении Правил по охране труда при эксплуатации электроустановок», зарегистрированного в Минюсте России 30.12.2020 N 61957.

– при проведении работ необходимо соблюдать требования безопасности по обслуживанию применённых средств измерений и контроля, а также испытательного оборудования.

– к работе с изделием допускаются лица, прошедшие инструктаж по технике безопасности.

4.2 Требования к рабочему месту

Для работы с оборудованием необходимо соблюдать следующие требования:

– все виды работ с модулем и его составляющими должны проводиться в помещении с нормальными климатическими условиями, согласно ГОСТ 15150-69.

– рабочее место должно быть оборудовано с учётом требований по защите от статического электричества согласно ГОСТ 11 073.062-2001.

– рабочее место должно позволять подключать корпус модуля и антистатический браслет работника к контуру защитного заземления.

– к рабочему месту должно быть подведено напряжение 220 В частотой 50 Гц для питания оборудования.

					ПГУ 1.090301.31.001 ПЗ	Лист
						84
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения поставленной задачи по разработке системы управления питанием модуля квантовой сети была разработана система, отвечающая всем требованиям, которые предъявляются к системе управления питанием. Конечная версия системы совмещает в себе простоту для пользователя и гибкость для расширения и взаимодействия с другими программными модулями. Пользователь полностью изолирован от системы управления питанием, благодаря чему исключается вероятность испортить изделие, при неразумном использовании. Это достигается за счёт автономности системы и сведения взаимодействия с пользователем к кнопке питания.

В ходе разработки были освоены навыки разработки программного обеспечения в условия ограниченных ресурсов, так как микроконтроллеры обладают куда меньшими объёмами как постоянной, так и оперативной памяти, а также программа выполняется на одном ядре без возможности задействовать стандартные механизмы многопоточности.

Однако, разработанная система может быть расширена, добавлением нового функционала. Так, например, микроконтроллеры управления питанием на платах могут дополнительно измерять температуру на критических узлах системы питания, за счёт установки датчиков температуры или терморезисторов в соответствующих узлах.

За счёт модульности системы и гибкого протокола взаимодействия, данную систему можно расширять и переносить на другие модули квантовой сети, так как она не привязана к функционалу конкретного модуля. Что позволяет в перспективе использовать данную систему управления питанием на всех модулях квантовой сети.

					ПГУ 1.090301.31.001 ПЗ	Лист
						85
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

1. Джеффри Браун, Discovering the STM32 Microcontroller \ Англ., изд. 14с8а1е (2016-06-05), Индианский университет, 2016. - 244 с.
2. Документация по STM32. [Электронный ресурс] \ Режим доступа: https://www.st.com/content/st_com/en/support/learning/stm32-education.html, свободный. (дата обращения 25.04.2024).
3. Керниган Б., Ритчи Д., Язык программирования Си \ Пер. с англ., 3-е изд., испр. — СПб.: «Невский Диалект», 2001. - 352 с.
4. Кармине Новьелло, Mastering STM32 \ Англ., изд. 0.21 (2017-02-06), Leanpub, 2017. - 792 с.
5. Джеффри Рихтер, Создание эффективных Win32-приложений с учетом специфики 64-разрядной версии Windows: Для профессионалов \ Пер. с англ.: Ю.Е. Купцевич, А.Р. Врублевский. 4. изд., Рус. ред. — СПб.: Питер, 2001. - 722 с.

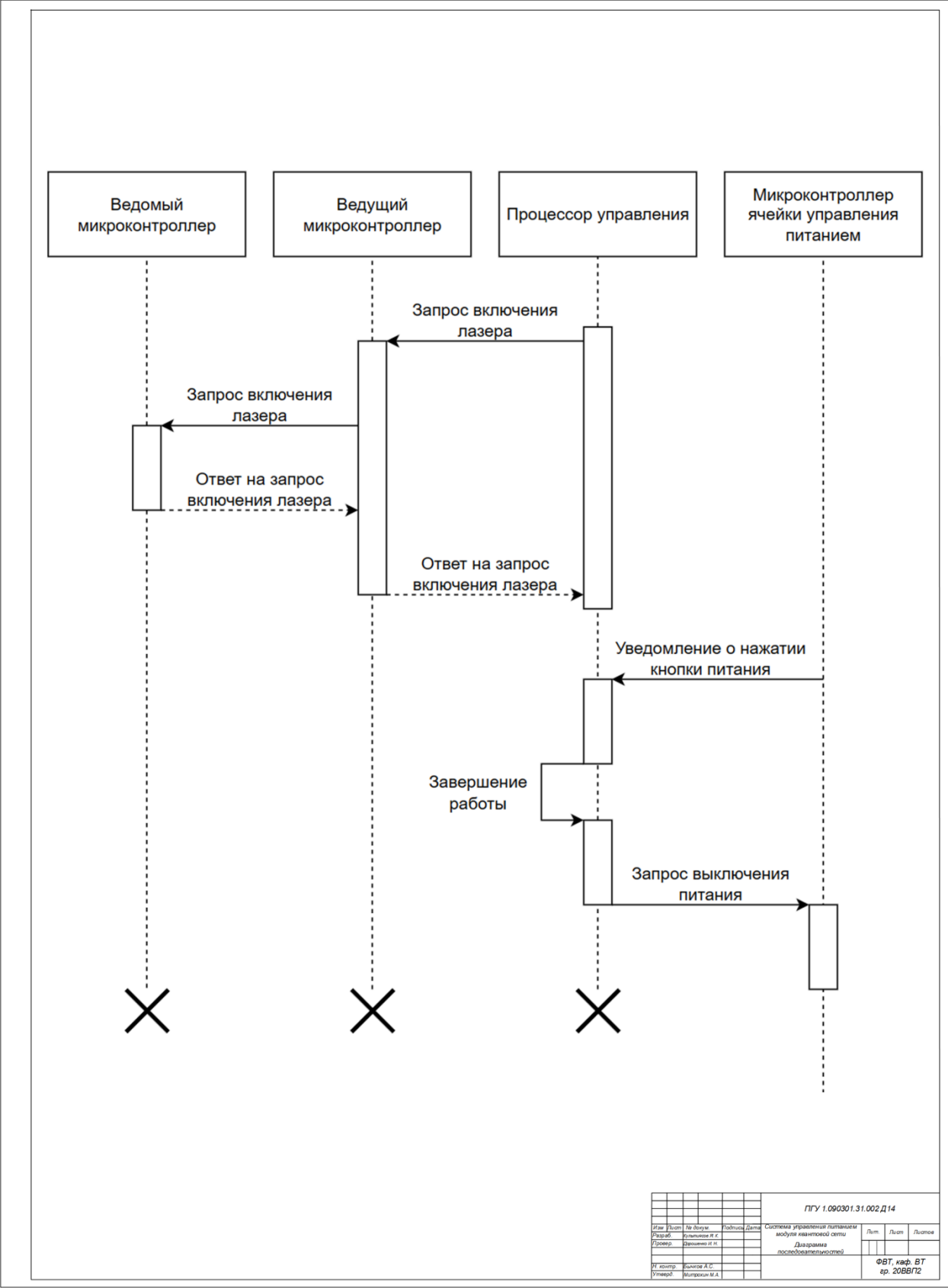
Приложение А

(Обязательное)

Система управления питанием модуля квантовой сети

UML-диаграммы

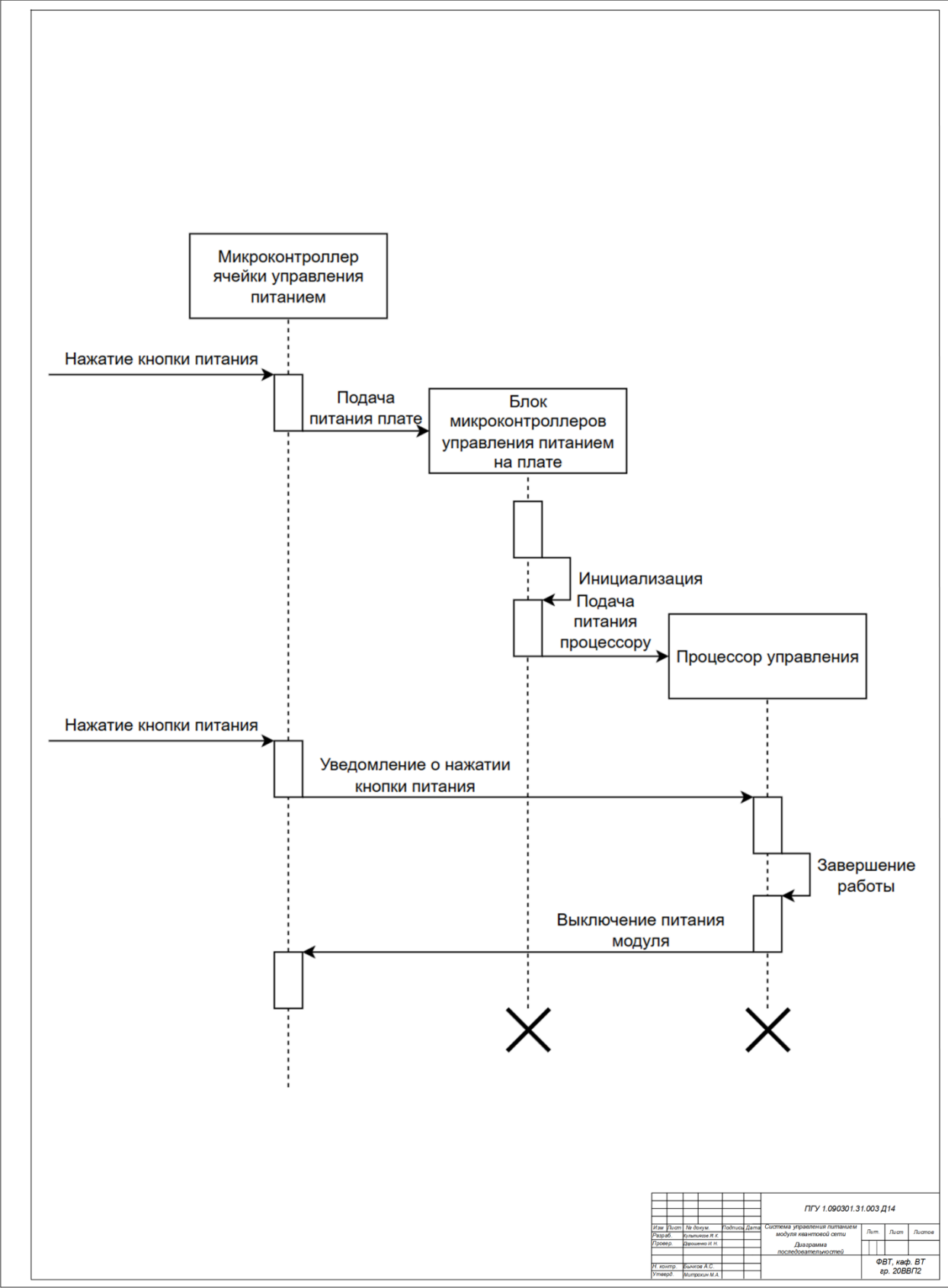
Приложение А.1 – Диаграмма последовательностей 1



									ПГУ 1.090301.31.002 Д14				
Изм.	Введ.	№ докум.	Подпись	Дата	Система управления питанием модуля коаксиальной связи (Диаграмма последовательностей)				Лист	Лист	Листов	ФВТ, каф. ВТ ер. 20ВВГ2	
Разработ.	Утвержден	И.с.	Должность	И.п.									
Провер.													
И. контрол.	Выполнен	А.С.											
Утвержден		Игорьков	М.А.										

Изм.	Лист	№ докум.	Подпись	Дата

Приложение А.2 – Диаграмма последовательностей 2



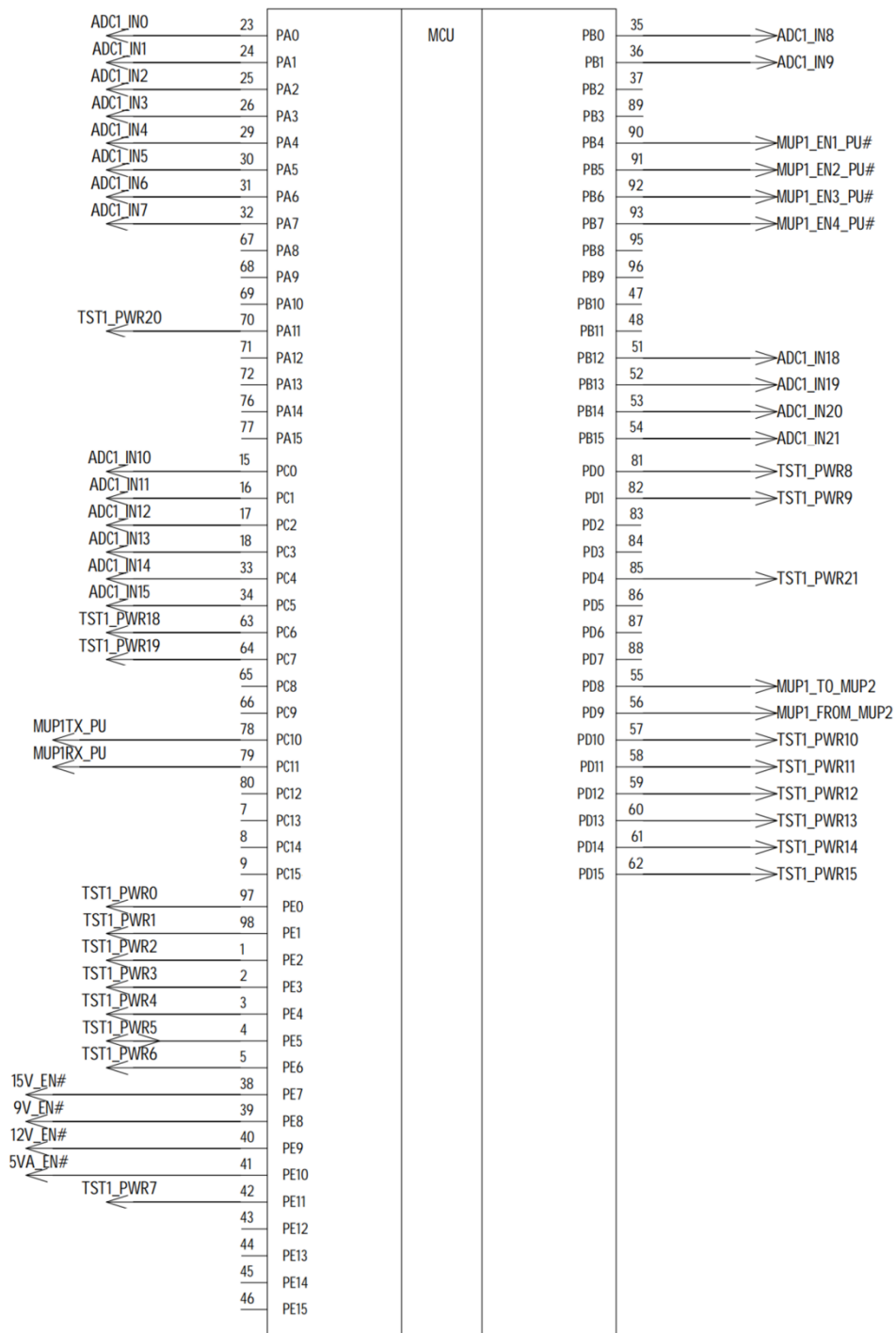
Приложение Б

(Обязательное)

Система управления питанием модуля квантовой сети

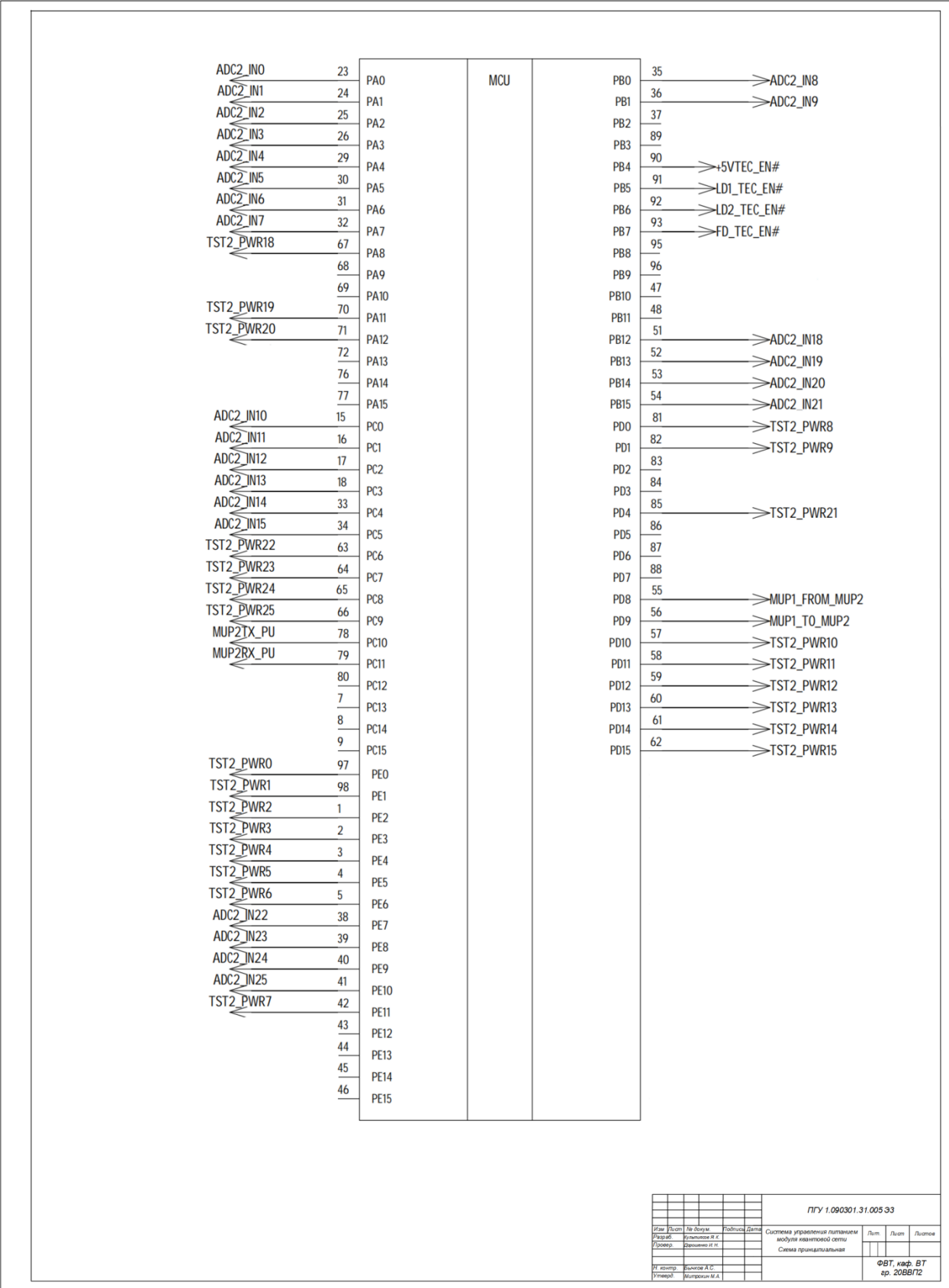
Схемы принципиальные

Приложение Б.1 – Схема принципиальная 1



				ПГУ 1.090301.31.004 33		
Иван	Иван	Не догум	Подпись	Дата	Система управления питанием модуля квантовой сети	Лит.
Разраб.	Кульков И. К.				Схема принципиальная	Лит.
Провер.	Давыденко И. Н.					Лит.
						Лит.
И. инж.	Былков А. В.					ФБТ, каф. ВТ гр. 20ВВП2
Утверд.	Митрофанов М. А.					

Приложение Б.2 – Схема принципиальная 2



Приложение В

(Обязательное)

Система управления питанием модуля квантовой сети

Презентация

Выпускная квалификационная работа на тему «Система управления питанием модуля квантовой сети»

Выполнил:

студент группы 20ВВП2

Культиясов Я.К.

					ПГУ 1.090301.31.001 ПЗ	Лист
						94
Изм.	Лист	№ докум.	Подпись	Дата		

Цели работы

Основной целью данной выпускной квалификационной работы является разработка программного обеспечения, предназначенного для управления питанием модуля квантовой сети.

Задача управления питанием сформирована на основе требований к разрабатываемым модулям сервера выработки и распределения ключей.

Необходимость системы управления питанием обусловлена следующими причинами:

- исключение внешнего влияния на модуль квантовой сети;
- обеспечение сохранности оборудования.

Решаемые задачи

Список основных задач решаемых системой управления питанием:

- непрерывный мониторинг напряжений на линиях питания;
- защита оборудования при выходе напряжения за допустимые пределы;
- реакция модуля на нажатия кнопки включения питания.

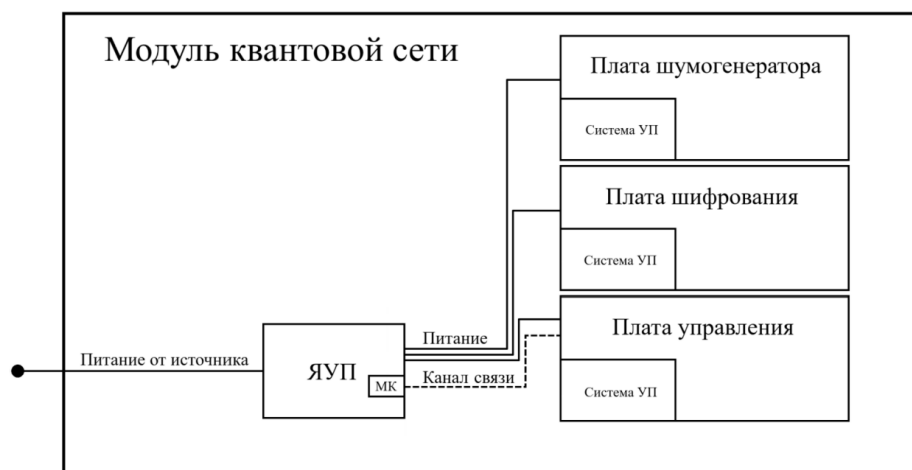
Список дополнительных задач решаемых системой управления питанием:

- ведение журнала аварийных ситуаций;
- возможность корректировки пороговых значений напряжения.

Общая схема квантовой сети



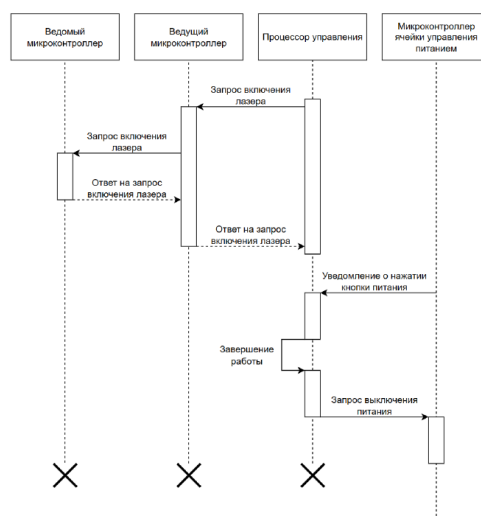
Схема связи основных блоков модуля



Организация системы управления питанием на плате управления



Временная UML диаграмма взаимодействия



Структура заголовка пакета

Название поля	Смещение поля, байт	Длина поля, байт	Описание
END	0	2	Сигнатура пакета
DEVICE	2	2	Код устройства
SEQUENCE_NUMBER	4	4	Номер последовательности.
SIZE	8	4	Длина поля DATA
CRC1	12	2	Контрольная сумма заголовка пакета
CRC2	14	2	Контрольная сумма поля DATA
DATA	16	SIZE	Данные пакета

Структура записи журнала

Название поля	Смещение поля, байт	Длина поля, байт	Описание
AlarmCode	0	1	Код события
Device	1	2	Код устройства
Channel	3	1	Номер линии.
Voltage	4	2	Значение напряжения
<u>Hour</u>	6	1	Час
<u>Minute</u>	7	1	Минута
<u>Second</u>	8	1	Секунда
<u>Day</u>	9	1	День
<u>Month</u>	10	1	Месяц
<u>Year</u>	11	1	Год

Заключение

В результате выполнения выпускной квалификационной работы было разработано программное обеспечение для микроконтроллеров и драйвер взаимодействия с ними.

Разработанное ПО решает все поставленные задачи по контролю питания.

Достоинства разработанной системы:

- автономность;
- расширяемость.

					ПГУ 1.090301.31.001 ПЗ	Лист
						99
Изм.	Лист	№ докум.	Подпись	Дата		

Перспективы применения системы

На данный момент времени разработанная система питания используется в реализации модуля «Центр», однако в перспективе перенос системы и на модуль «Абонент».

Возможные варианты развития системы:

- реализация механизма измерения температуры;
- реализация перевода модуля в режим малого потребления.

					ПГУ 1.090301.31.001 ПЗ	Лист
						100
Изм.	Лист	№ докум.	Подпись	Дата		

Спасибо за внимание