

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт машиностроения

(наименование института полностью)

Кафедра "Промышленная электроника"

(наименование кафедры/департамента/центра полностью)

11.03.04 Электроника и нанoeлектроника

(код и наименование направления подготовки, специальности)

Электроника и робототехника

(направленность (профиль) / специализация)

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Цифровая плата мониторинга состояния источника питания»

Обучающийся

С. А. Хуртин

(Инициалы Фамилия)

(личная подпись)

Руководитель

А. В. Прядилов

(Инициалы Фамилия)

Тольятти 2024

Аннотация

Цифровая плата мониторинга состояния источника питания.

Объем 59 с., 37 рис., 7 табл., 25 источников.

Объектом разработки является устройство мониторинга состояния источников питания.

Цель работы – получение готового устройства контроля БП с индикацией тока и напряжения с возможностью отключения нагрузки и управлением вентилятора.

Задачи работы заключались в отработке навыков прототипирования и получении практических навыков в интегрированных средах разработки.

Степень внедрения: устройство по разработанной документации является опытным образцом.

Областью применения данного устройства являются блоки питания любого типа, что реализуется за счет автономного исполнения. Применение устройства благоприятно сказывается на качестве питающего напряжения, так как в функционал включены защиты по превышению тока, напряжения и температуры.

Дипломный проект состоит из пояснительной записки и графической части. Пояснительная записка включает следующие разделы: введение, обзор и анализ существующих решений индикации напряжения, формулирование основных задач ВКР, выбор структурной схемы, выбор комплектующих, разработка принципиальной схемы, разработка программы для микроконтроллера, компоновка и конструирование платы управления, эскизная компоновка корпуса изделия, заключение и список используемой литературы.

Графическая часть выполнена на 6 листах формата А1 и включает структурную схему, схему электрическую принципиальную, сборочный чертеж, эскиз корпуса, схему алгоритма и схему подключения.

Annotation

Digital power supply status monitoring board.

Volume 58 pages, 37 figures, 7 tables, 25 sources.

The object of the development is a device for monitoring the status of power supplies.

The purpose of the work is to obtain a ready-made PSU monitoring device with current and voltage indication with the possibility of switching off the load and fan control.

The objectives of the work were to develop prototyping skills and gain practical skills in integrated development environments.

The degree of implementation: the device is a prototype according to the developed documentation.

The scope of application of this device is power supplies of any type, which is realized through autonomous execution. The use of the device has a positive effect on the quality of the supply voltage, since the functional includes protections for excess current, voltage and temperature.

The graduation project consists of an explanatory note and a graphic part. The explanatory note includes the following sections: introduction, review and analysis of existing voltage indication solutions, formulation of the main tasks of the final qualifying work, selection of a block diagram, selection of components, development of a schematic diagram, development of a program for a microcontroller, layout and design of a control board, sketch layout of the product body, conclusion and a list of used literature.

The graphic part is made on 6 sheets of A1 format and includes a block diagram, an electrical schematic diagram, an assembly drawing, a sketch of the housing, an algorithm diagram and a connection diagram.

Содержание

Введение.....	5
1 Обзор и анализ существующих решений. Формулирование задач ВКР.....	6
2 Выбор структурной схемы	10
3 Выбор комплектующих	12
3.1 Выбор микроконтроллера	12
3.2 Выбор датчика температуры.....	17
3.3 Выбор токового шунта	20
3.4 Выбор блока отключения нагрузки.....	21
3.5 Выбор регулятора напряжения.....	23
3.6 Выбор LCD дисплея.....	24
4 Разработка принципиальной схемы	27
4.1 Регулятор напряжения.....	27
4.2 Датчик температуры	29
4.3 Схема токового шунта и блока отключения нагрузки	30
4.4 Схемы LCD дисплея и кнопок.....	31
4.5 Схема вентилятора.....	32
4.6 Схема микроконтроллера.....	33
5 Компоновка и конструирование платы управления.....	34
6 Эскизная компоновка корпуса изделия	39
7 Разработка управляющей программы.....	41
7.1 Создание и настройка проекта.....	41
7.2 Описание исходного кода программы.....	47
Заключение	56
Список используемой литературы и используемых источников.....	57

Введение

Блок питания (БП) является важным компонентом любого электронного устройства. Он выполняет функцию преобразования первичного напряжения (например, из электрической сети или аккумулятора) в величину, необходимую для работы различного оборудования. Надежность и стабильность блока питания играет ключевую роль в общей работоспособности системы.

Практически любое электронное устройство имеет в своем составе либо полноценный блок питания, либо его отдельные исполнительные части, поэтому вне зависимости от типа блока питания необходимо следить за параметрами выдаваемого напряжения, особенно если речь идет о высокой стабильности. Для этих целей существуют специализированные устройства контроля параметров БП, которые выполняются в разных видах. Это могут быть как аппаратные блоки в ШИМ-контроллерах (широтно-импульсная модуляция), являясь частью схемотехники импульсных БП, так и внешние автономные устройства, частично перенимающие на себя некоторые функции контроля и защиты.

В данной работе будет выполнена разработка прототипа автономного устройства цифрового мониторинга состояния источника питания, который будет включать в себя функции отображения параметров, их хранение, контроль характеристик, а также управление исполнительными элементами.

1 Обзор и анализ существующих решений. Формулирование задач ВКР

Перед началом разработки устройства необходимо выполнить анализ существующих решений. Определить основные требования к функциональным возможностям, выбрать соответствующий диапазон измерения необходимых величин, предположить теоретическую конструкцию. Все это позволит более точно сформулировать задачи выпускной квалификационной работы.

На рынке существует довольно большой выбор различных устройств разного конструктивного исполнения. Большая часть из них представлена в виде конструктивных решений, которые будут описаны далее.

Встраиваемый в корпус блок, как на рисунке 1, применяется при монтаже на стенку устройства и служит для индикации различных величин. Имеет свой собственный корпус, за счет которого и выполняется крепление.



Рисунок 1 – Встраиваемый вольтамперметр

Индикаторы для монтажа на DIN-рейку, как на рисунке 2, применяются в электрощитах, где могут быть совмещены с исполнительными устройствами, например, с устройством защитного отключения или с реле принудительной вентиляции.



Рисунок 2 – Индикаторы на DIN рейку

Так же существуют индикаторы величин, выполненные на печатной плате без корпуса, пример на рисунке 3. Такая реализация более универсальна, но требует дополнительного проектирования для установки в какое-либо устройство. Применение такого вида индикаторов обоснована экономически, так как использование готовых блоков может сильно сократить время и бюджет разрабатываемого устройства без потери функциональности.

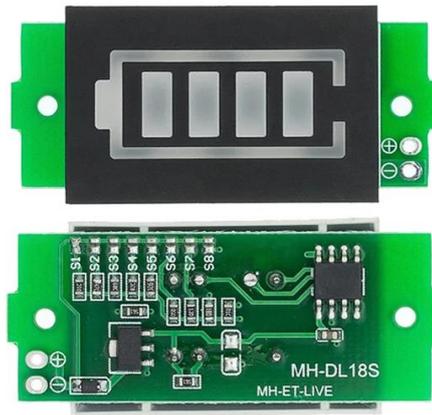


Рисунок 3 – Индикатор заряда батареи

Помимо блоков, которые являются компонентом итогового устройства, так же есть множество функционально законченных устройств. Большая их часть используется в быту, например, для подсчета потребляемой энергии или замера емкости аккумуляторов, пример на рисунке 4.



Рисунок 4 – USB ваттметр

Сформулируем основные задач ВКР.

В данной работе устройство выполнено как функционально законченное устройство.

Исходные данные устройства:

- внешнее автономное исполнение;
- питание платы 12В DC;
- диапазон измерения тока 0 – 10А;
- диапазон измерения напряжения 0 – 200В;
- цифровое отображение напряжения и тока;
- скользящий график на графическом дисплее;
- возможность отключения нагрузки при перегрузках (перегреве);
- измерение температуры блока питания;
- управление вентилятором охлаждения в автоматическом и непрерывном режимах;
- цифровая индикация температуры блока питания.

Печатная плата расположена в корпусе, элементы управления и дисплей расположены на лицевой панели. На торце корпуса расположены разъемы для подключения датчиков и необходимых цепей от блока питания.

В данном разделе был выполнен обзор и анализ существующих решений, а также были сформулированы задач ВКР.

2 Выбор структурной схемы

Основываясь на исходных данных для устройства, цифровая плата мониторинга состояния источника питания должна состоять из множества блоков.

Микроконтроллер – главный исполнительный блок в схеме. В его задачах заключена вся логика устройства: расчет необходимых значений, отработка защит, выполнение команд пользователя, индикация результатов на дисплее, отрисовка пользовательского интерфейса.

Регулятор напряжения – питание логической части устройства.

Токовый шунт с обвязкой – датчик тока с усилителем для измерения потребления подключенной нагрузки и защиты от перегрузки.

Температурный датчик – определение температуры блока питания для функционирования защиты от перегрева.

Вентилятор – встроенный вентилятор блока питания для поддержания допустимого температурного режима работы.

Блок отключения нагрузки – часть схемы для возможности ручного отключения или для отработки защит по превышению температуры или перегрузке.

ЖК дисплей – пользовательская индикация значений параметров блока питания в текстовом и графическом виде, а также определение состояния работы.

Кнопки – цепи ввода команд пользователя для изменения режима работы устройства.

Итоговая структурная схема устройства изображена на рисунке 5.

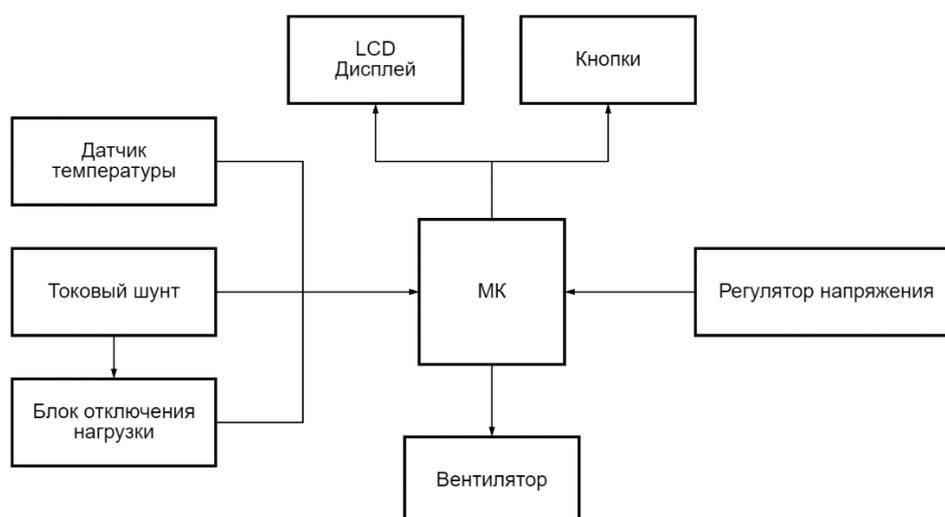


Рисунок 5 – Структурная схема устройства

В данном разделе был выполнен выбор структурной схемы устройства на основе исходных данных задачи.

3 Выбор комплектующих

Вариаций возможных версий данного устройства может быть великое множество. Моей целью заключается выбор таких компонентов, которые будут легко доступны к покупке, при этом имеющие адекватную стоимость.

3.1 Выбор микроконтроллера

В рамках задач устройства, выбор микроконтроллера ограничивается только его объемом ПЗУ, так как необходим достаточный объем для хранения элементов графики ЖК дисплея. Все остальные периферийные устройства микроконтроллера можно расширить. Недостаточное количество портов ввода-вывода общего назначения (GPIO) можно нарастить с помощью сдвиговых регистров или расширителей портов. Отсутствие интерфейсов передачи данных (UART, I2C, SPI) можно реализовать программным методом. Таким образом, предполагаю, что минимально необходимый объемом ПЗУ составит 8 кБ.

Далее необходимо определить производителя микроконтроллеров. Из самых популярных на рынке можно найти три варианта: Atmel, ST Semiconductor, и Espressif.

Микроконтроллеры Atmel известны своим семейством AVR, которое предлагает богатый выбор моделей под различные требования. Несмотря на то, что производство началось аж в 1997 году, функциональные возможности чипов до сих пор покрывают многие задачи автоматизации. В настоящее время Atmega328 является одним из самых популярных микроконтроллеров, так как применяется в платформе Arduino, внешний вид платы на рисунке 6. [11]. Под

требования устройства не подойдут лишь младшие модели серии Tiny, поэтому итоговый выбор довольно большой.



Рисунок 6 – Arduino UNO на базе Atmega328

В таблице 1 приведены некоторые возможные модели микроконтроллеров для применения в устройстве [11].

Таблица 1 – Модели AVR микроконтроллеров

Модель	Тактовая частота	ОЗУ	ПЗУ	Кол-во GPIO
ATmega2560	16МГц	8 Кбайт	256 КБайт	86
ATmega328	16МГц	2 Кбайт	32 КБайт	23
ATtiny88	16МГц	512 байт	8 КБайт	26

Микроконтроллеры ST семейства STM32 известны высокой производительностью и более богатым набором периферии. Компания ST запустила производство ассортимента в 2007 году. Первые серии начались с линейки STM32F1, которая развивается и выпускается до сих пор. Семейство 32-разрядных микроконтроллеров на процессоре ARM Cortex сочетает

великолепную производительность, низкое энергопотребление и простоту разработки. Большой ассортимент моделей, огромный набор программного обеспечения и инструментов, обеспечивают высокую долю рынка. В большинстве случаев, это идеальный выбор, как для небольших проектов, так и для крупных промышленных устройств.

ST выпускает собственные отладочные платы для прототипирования под линейками Discovery и Nucleo, на рисунке 7 внешний вид платы [12]. Удобство этих плат заключается в наличии распаянного программатора ST-LINK, который позволяет выполнять прошивку и отладку микроконтроллера.

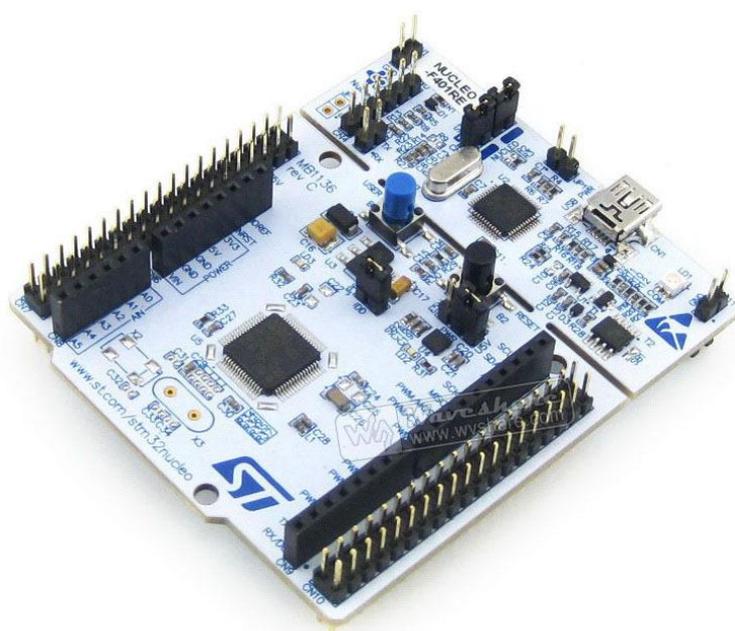


Рисунок 7 – STM32 Nucleo-401RE

Для задач устройства подойдут даже самые младшие модели, такие как STM32F0x0. Некоторые возможные модели микроконтроллеров STM32 для применения представлены в таблице 2 [12].

Таблица 2 – Модели STM32 микроконтроллеров

Модель	Тактовая частота	ОЗУ	ПЗУ	Кол-во GPIO
STM32F103C8T6	72МГц	20 Кбайт	64/128 КБайт	37
STM32L051C8T6	16МГц	8 Кбайт	64 КБайт	37
STM32F030F4P6	48МГц	4 Кбайт	16 КБайт	15

Еще один возможный вариант – микроконтроллеры ESP. Espressif Systems — международная компания-разработчик полупроводников, основанная в 2008 году. Специализируется на разработке микроконтроллеров с низким энергопотреблением, высокой производительностью и аппаратной поддержкой беспроводной передачи данных. В 2014 году компания выпускает продукт, ставший широко известным - ESP8266. Чип быстро завоевал популярность благодаря своей аппаратной поддержке интерфейса Wi-Fi и низкой цене. Espressif Systems свободно распространяет свой комплект разработчика. В этот комплект входит компилятор GCC, библиотеки Espressif и загрузочная утилита XTCOM. Библиотеки поставляются в виде скомпилированных библиотек, без исходных текстов. Для покупки доступно множество отладочных плат с различными версиями микроконтроллера. Одна из таких представлена на рисунке 8.



Рисунок 8 – Отладочная плата NodeMCU на базе ESP8266

Некоторые возможные модели микроконтроллеров ESP для применения представлены в таблице 3 [10].

Таблица 3 – Модели ESP микроконтроллеров

Модель	Тактовая частота	ОЗУ	ПЗУ	Кол-во GPIO
ESP8266	80/160 МГц	82 кБ	1 МБ	11
ESP32-PICO-D4	240МГц	520 кБ	4 МБ	42

В моем владении имеется отладочная плата STM32 Nucleo-Z767ZI на базе микроконтроллера STM32F767ZIT6 (ARM Cortex-M7), которая будет использована для прототипа в данной ВКР, её внешний вид на рисунке 9.

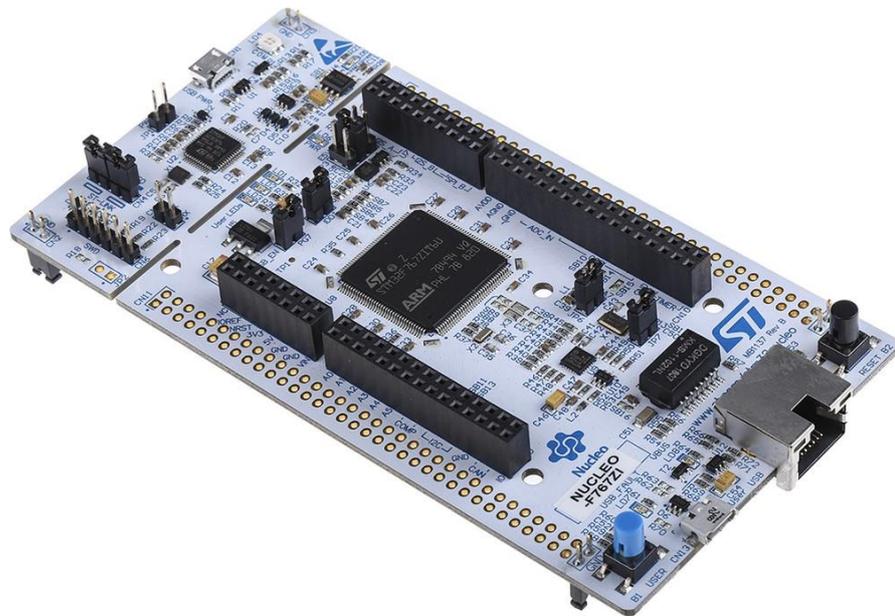


Рисунок 9 – Отладочная плата Nucleo-Z767ZI

В данном подразделе был выбран микроконтроллер для устройства, а также рассмотрены различные варианты.

3.2 Выбор датчика температуры

Выбор температурных датчиков довольно велик. В первую очередь, необходимо подобрать такой, который подходил бы под требуемый диапазон, который, в свою очередь, сильно зависит от исполнения самого блока питания.

Самые горячие части в импульсном блоке питания: высоковольтные силовые ключи, трансформатор и выходной выпрямитель. Чаще всего температурный датчик следует устанавливать на выходном выпрямителе, где и наблюдается самый сильный нагрев. В основе выпрямителя используются полупроводниковые элементы с рабочей температурой до $+175^{\circ}\text{C}$, но это

допустимая температура кристалла, тогда как температуру корпуса не следует допускать выше $+70^{\circ}\text{C}$. Нижнюю границу можно установить в 0°C , по причине сильной зависимости от исполнения корпуса блока питания, которые могут быть как открытого исполнения, так и целиком залитые компаундом.

Датчики температуры можно разделить на цифровые и аналоговые. В основе цифровых заключена микросхема, которая сама занимается оцифровкой аналогового датчика и выдает готовое значение по цифровым шинам данных.

Пример цифрового термометра - DS18B20, внешний вид на рисунке 10. Обеспечивает измерение температуры и имеет функцию сигнализации с энергонезависимыми программируемыми пользователем верхней и нижней точками срабатывания. DS18B20 взаимодействует по однопроводной шине 1-Wire, которая по определению требует только одной линии передачи данных. Он имеет диапазон рабочих температур от -55°C до $+125^{\circ}\text{C}$ и точность измерения $\pm 0,5^{\circ}\text{C}$. Кроме того, DS18B20 может получать питание непосредственно от линии передачи данных (паразитное питание), устраняя необходимость во внешнем источнике питания [13].

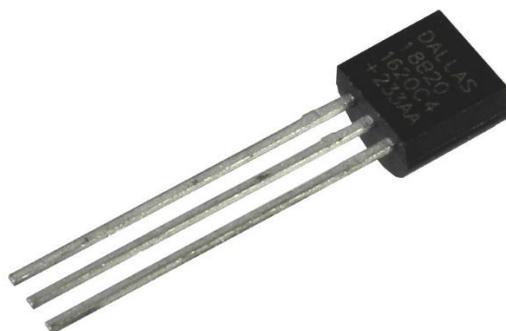


Рисунок 10 – Цифровой термометр DS18B20

Аналоговый датчик имеет более простую конструкцию и меньшую стоимость, но требует хорошую точность измерения и калибровку. В основе используется полупроводниковый прибор, электрическое сопротивление которого изменяется в зависимости от его температуры. Пример термистора на рисунке 11. Большая часть термосопротивлений (термисторов) имеет довольно широкие границы допустимых температур – от -50°C до $+260^{\circ}\text{C}$.



Рисунок 11 – Термистор

Будет выбран именно аналоговый датчик, так как в периферии у STM32F767ZI имеется 12-битный аналого-цифровой преобразователь (АЦП) [5], который позволит выполнять точное измерение напряжения, с последующим расчетом температуры, без необходимости дополнительной программной нагрузки в виде протокола обмена информацией 1-Wire.

В данном подразделе был выбран аналоговый датчик, а также выполнено сравнение с цифровыми датчиками температуры.

3.3 Выбор токового шунта

Современные датчики можно разделить на следующие типы: резистивные датчики (токовые шунты), датчики тока на эффекте Холла и трансформаторы тока.

Для измерения постоянного и переменного тока активно применяется резистивный датчик тока, пример на рисунке 12. Данный датчик является самым простым в устройстве, но обладает хорошей точностью измерения. Главный недостаток таких датчиков состоит в гальванической связи с измерительными цепями, что может сильно ограничить применение.

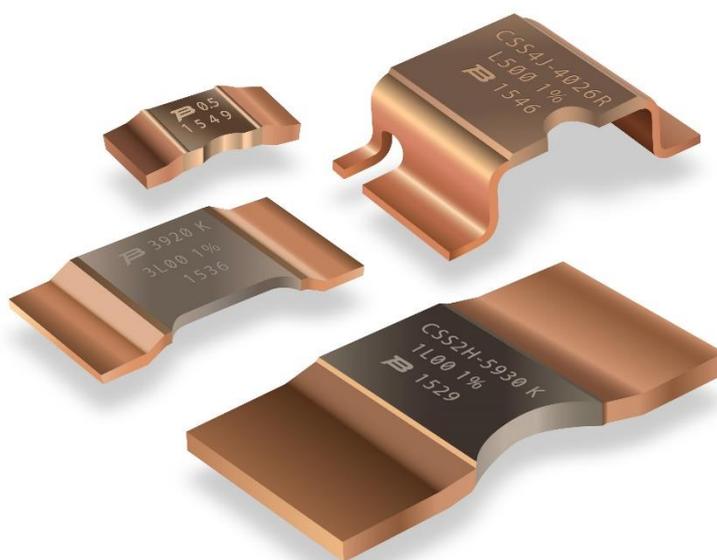


Рисунок 12 – Резистивный токовый шунт

Принцип действия беспроводного измерения основан на том, что ток, протекающий в проводе создает вокруг себя магнитное поле. Если это поле

переменное, то возникает ЭДС, которая с некоторыми условиями будет пропорциональна измеряемому току. Если измерить напряжение на таком проводнике, то можно определить величину измеряемого тока.

Для целей работы отлично подойдет резистивный датчик тока, который имеет хорошую точность, при этом гальваническая изоляция не требуется [9].

В данном подразделе был выбран датчик тока для устройства мониторинга состояния блока питания.

3.4 Выбор блока отключения нагрузки

Для подключения и отключения нагрузки возможно применение электромеханического реле или силового полупроводникового ключа.

Электромеханическое реле на большую мощность имеет относительно большие размеры и заметно большую цену. При этом, на основе реле нельзя сделать быстродействующую защиту от превышения тока, что обусловлено механической конструкцией. Так же, у реле существует ресурс замыканий, что может быть критично в плане срабатывания защиты блока мониторинга.

Для коммутации нагрузки очевидным выбором станет использование силового MOSFET ключа, который лишен всех недостатков реле.

Требования для ключа можно определить из исходных данных устройства. Ток коммутации – 10А, напряжение коммутации – 200В.

Выбираем транзистор IRF644PBF – это N-канальный MOSFET в корпусе TO-220AB, его внешний вид представлен на рисунке 13.

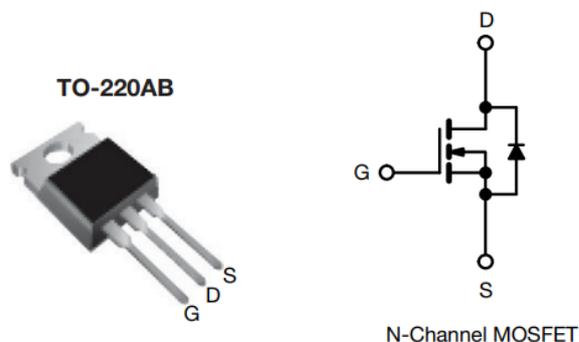


Рисунок 13 – Внешний вид и условное графическое обозначение транзистора IRF644PBF

Краткие характеристики транзистора представлены в таблице 4 [16].

Таблица 4 – Характеристики IRF644PBF

Параметр	Значение
Макс. напряжения исток-сток (V_{DS})	250 В
Сопротивление открытого канала ($R_{DS(on)}$)	0.28 Ом
Допустимый максимальный ток стока (I_D)	14 А
Пороговое напряжение открытия ($V_{GS(th)}$)	2-4 В
Входная емкость (C_{iss})	1300 пФ

Транзистор IRF644PBF является актуальной моделью и может долгое время закупаться для производства устройства.

В данном подразделе был выбран ключ для коммутации нагрузки и приведены его краткие характеристики.

3.5 Выбор регулятора напряжения

Главная задача регулятора напряжения – поддержание стабильного фиксированного выходного напряжения. Регуляторы можно разделить на несколько типов.

Линейные стабилизаторы – устройства, имеющие низкий КПД, так как лишнее напряжение выделяется в виде тепла. Применяются в цепях с низким потреблением.

Импульсные преобразователи – устройства с высоким КПД, что достигается за счет использования контроллера широтно-импульсной модуляции и ключевого режима работы схемы. Применяются в цепях с высоким потреблением, где необходима большая сила тока.

В схеме устройства регулятор напряжения необходим только для питания логической части, которая является цепью с низким потреблением. Рациональнее всего будет использование линейного стабилизатора напряжения, что упростит схемотехнику и удешевит итоговое устройство.

Из всего множества моделей выбираем AMS1117, как один из самых распространенных линейных стабилизаторов. Краткие характеристики AMS1117 представлены в таблице 5 [7].

Таблица 5 – Характеристики AMS1117

Параметр	Значение
Фиксированное выходное напряжение	1.5В, 1.8В, 2.5В, 2.85В, 3.3В, 5.0В
Максимальный ток	800 мА

В данном подразделе был выбран стабилизатор AMS1117, который имеет подходящие для устройства параметры.

3.6 Выбор LCD дисплея

Согласно исходным данным на цифровую плату мониторинга, ЖК дисплей должен отображать:

- напряжение,
- ток,
- температуру,
- состояние выхода,
- режим работы вентилятора,
- скользящий график.

Минимальный дисплей, который позволит реализовать данный список – OLED матрица на базе SSD1306. Внешний вид дисплея на рисунке 14.

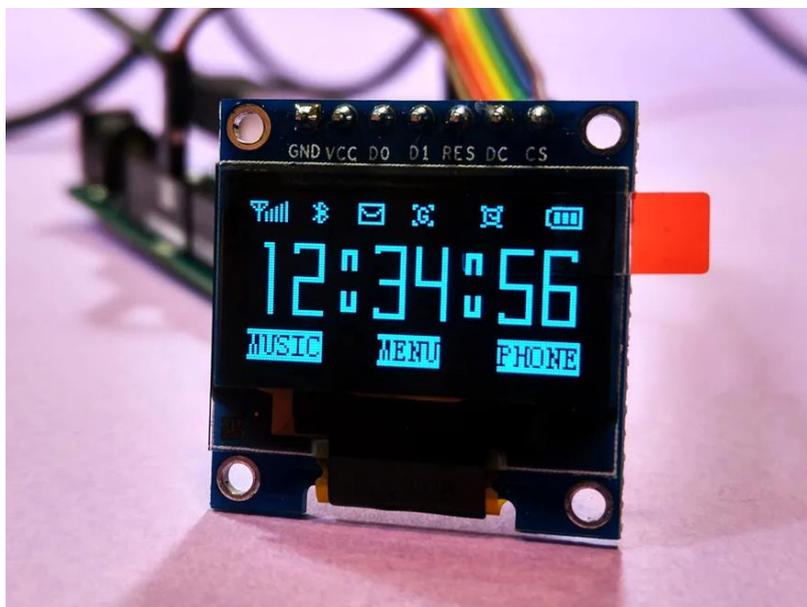


Рисунок 14 – OLED матрица

Данный дисплей имеет разрешение 128x64 точки. На контроллере SSD1306 существуют экраны только с малой диагональю от 1 до 2 дюймов, поэтому мелкая графика может тяжело восприниматься зрительно.

Для больших диагоналей следует выбрать TFT LCD матрицу. Например, TFT дисплей на базе контроллера ILI9341. Внешний вид дисплея на рисунке 15.



Рисунок 15 – TFT LCD на ILI9341

Этот дисплей цветной и значительно больше по диагонали – 2,4 дюйма, а также имеет разрешение 320x240 точек. Имеет SPI подключение к микроконтроллеру и возможность управления подсветкой.

Для прототипа будет выбран дисплей Waveshare 2.8inch Touch LCD, его внешний вид изображен на рисунке 16.



Рисунок 16 – Waveshare 2.8inch Touch LCD

Краткие характеристики дисплея представлены в таблице 6 [14].

Таблица 6 – Характеристики Waveshare 2.8inch Touch LCD

Параметр	Значение
Тип LCD	TFT
Интерфейс подключения	SPI
Контроллер	ST7789
Формат цвета	RGB888
Разрешение	320x240
Напряжение питания	3.3V/5V

Это аналог упомянутого выше дисплея на ILI9341, но выполненный на другой печатной плате и имеющий чуть большую диагональ. Функционал сенсорного экрана использоваться не будет.

В данном подразделе был выбран LCD дисплей, а так же приведены другие возможные дисплеи для индикации параметров.

4 Разработка принципиальной схемы

4.1 Регулятор напряжения

На основе структурной схемы устройства и выбранных компонентов можно приступить к разработке принципиальной схемы.

Воспользуемся бесплатной интегрированной средой разработки EasyEDA.

Начнем с блока регулятора напряжения, выполненного на линейном стабилизаторе AMS1117. Для дополнительных данных обратимся к технической документации (datasheet) на микросхему.

В разделе «Application hints» изображена схема включения стабилизатора, продублирована на рисунке 17, а в разделе «Stability» можно узнать о рекомендуемых номиналах развязывающих конденсаторов.

«The circuit design used in the AMS1117 series requires the use of an output capacitor as part of the device frequency compensation. The addition of 22uF solid tantalum on the output will ensure stability for all operating conditions.» [7]

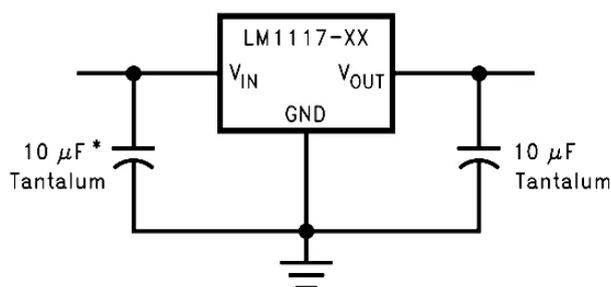


Рисунок 17 – Схема включения из технической документации AMS1117

Принципиальная схема регулятора напряжения изображена на рисунке 18.

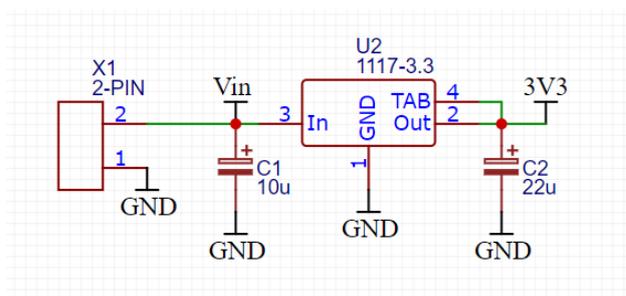


Рисунок 18 – Принципиальная схема регулятора напряжения

На вход стабилизатора поступает постоянное напряжение +12В, поэтому входной конденсатор следует выбирать под соответствующее допустимое напряжение [24].

В данном подразделе была выполнена схема регулятора напряжения, а также определены номиналы.

4.2 Датчик температуры

В схеме датчика температуры применяется NTC (отрицательный температурный коэффициент сопротивления) термистор [1]. Его схема подключения довольно проста, так как она является обычным делителем напряжения. Схема представлена на рисунке 19.

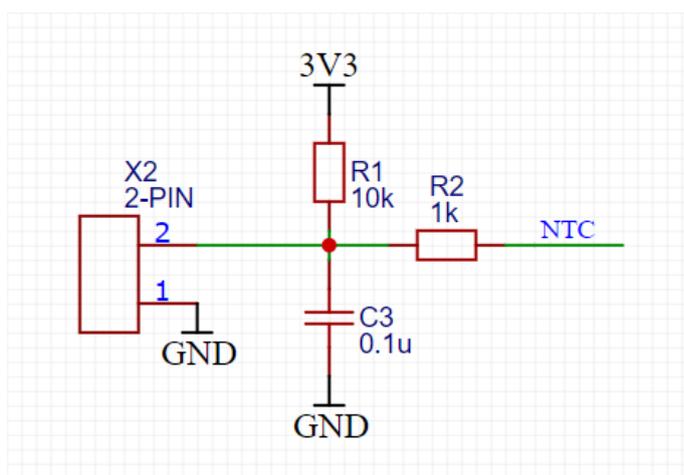


Рисунок 19 – Схема NTC термистора

На схеме резистор R1 и термистор, подключенный к X2, образуют делитель сопротивления, а R2 и C3 выступают в качестве RC фильтра для повышения точности оцифровывания [17].

В данном подразделе была выполнена схема датчика температуры.

4.3 Схема токового шунта и блока отключения нагрузки

Схему токового шунта и блока отключения нагрузки можно представить на одной схеме, которая представлена на рисунке 20.

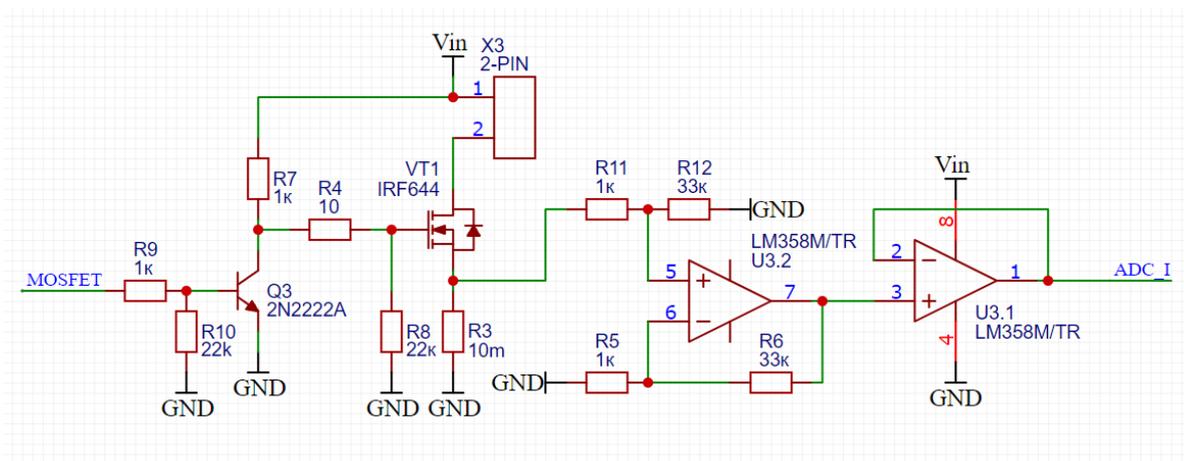


Рисунок 20 – Схема токового шунта и блока отключения нагрузки

Цепь отключения нагрузки выполнена на ключе VT1, драйвер которого собран на NPN транзисторе общего назначения [2]. Драйвер применяется для возможности управления ключом от порта ввода-вывода микроконтроллера. В противном случае, максимальное напряжение порта +3,3В было бы недостаточно [25].

Резистор R9 ограничивает ток базы ключа Q3, а R4 ограничивает ток разряда затвора VT1. Резистор R3 служит в роли токового шунта [9], напряжение падения на котором усиливается с помощью U3 – ОУ LM358 [8]. Первая часть усилителя используется как повторитель напряжения для

буферизации выходного сигнала, а вторая часть непосредственно для усиления напряжения шунта [18]. Нагрузка подключается к разъему X3.

В данном подразделе была выполнена схема токового шунта и блока отключения нагрузки.

4.4 Схемы LCD дисплея и кнопок

Схемы LCD дисплея и кнопок так же представим на одной схеме на рисунке 21.

Кнопки SW1 – SW3 имеют подтягивающие резисторы R13 – R15, что дает активный низкий сигнал управления. Цепи передачи данных дисплея подключены к аппаратному блоку SPI микроконтроллера.

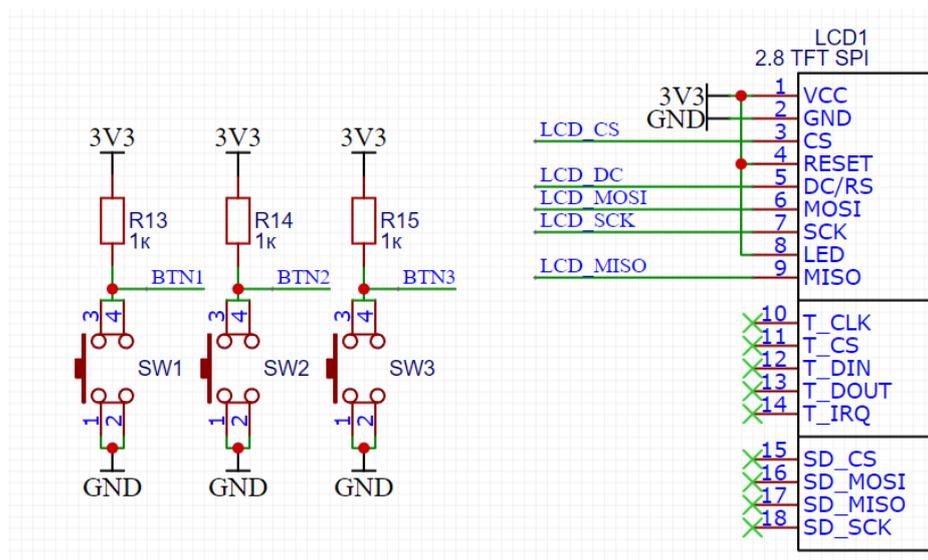


Рисунок 21 – Схема LCD дисплея и кнопок

Контакты SD карты и дублирующие выводы передачи данных не используются [4].

В данном подразделе была выполнена схема LCD дисплея и кнопок.

4.5 Схема вентилятора

Схема вентилятора выполняет управление оборотами крыльчатки за счет регулирования скважности (ШИМ) питающего напряжения. Схема на рисунке 22.

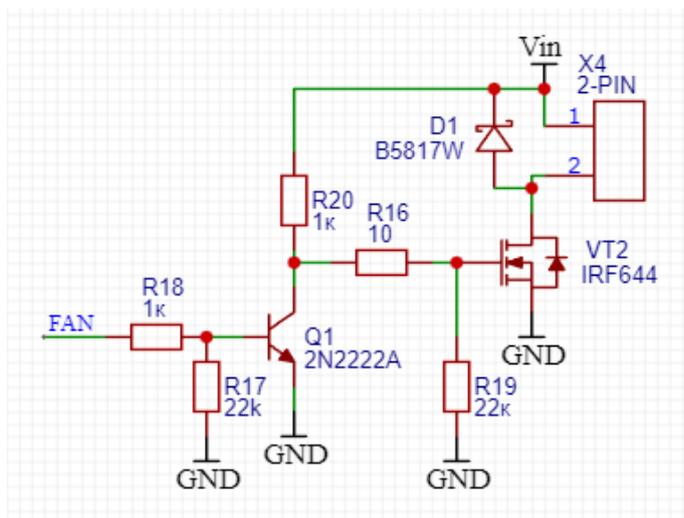


Рисунок 22 – Схема управления вентилятором

Схемотехнически она повторяет схему блока отключения нагрузки и работает по аналогичному принципу. Единственное отличие – защитный диод D1, который добавлен для гашения возможных индуктивных выбросов от вентилятора [20].

В данном подразделе была выполнена схема управления вентилятором, с помощью ШИМ сигнала.

4.6 Схема микроконтроллера

Схема микроконтроллера включает в себя исполнительные цепи всех других блоков. Схема на рисунке 23. STM32F767ZI имеет довольно большой корпус LQFP-144 с большим количество выводов, поэтому будет представлен фрагмент с активными выводами [4].

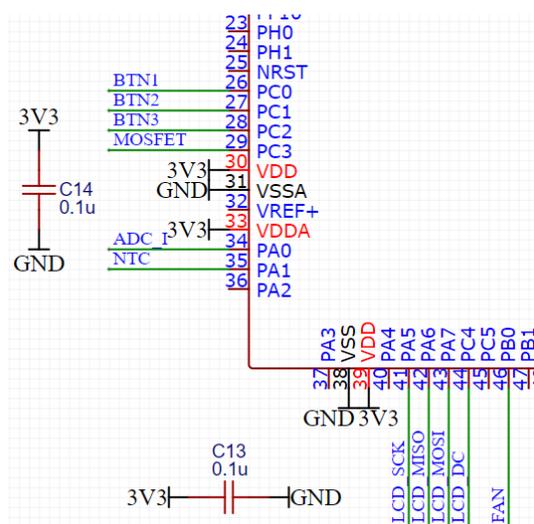


Рисунок 23 – Схема микроконтроллера

На каждую пару выводов 3V3 и GND подключен развязывающий конденсатор для стабилизации питания [6].

В данном подразделе была выполнена схема микроконтроллера.

5 Компоновка и конструирование платы управления

После подготовки схемы устройства можно приступить к трассировке печатной платы устройства. Все SMD компоненты располагаются на слое «Bottom Layer». На слое «Top Layer» разъемы и элементы управления. Кнопки и дисплей должны быть доступны для пользователя.

Для минимизации габаритов печатной платы используем преимущественно SMD монтаж компонентов. Типоразмер для резисторов и конденсаторов – 0805.

При разработке печатной платы зададим следующие нормы и правила:

- ГОСТ Р 55490-2013 (Правила приема электронных сборок);
- выводные компоненты должны быть расположены только на верхней стороне печатной платы;
- минимальная ширина проводников 0,2мм;
- минимальная ширина линий питания 0,5мм;
- расстояние компонентов и токоведущих частей платы от контура платы и крепежных отверстий не менее 0,5 мм;
- сохранять свободный доступ к элементам, предназначенным для управления, индикации, подключения внешних элементов, настройки и регулировки;
- отступы шелкографии от контактных площадок и контура платы не менее 0.1 мм;
- расстояние между линиями шелкографии соседних компонентов не менее 0.2 мм;
- надписи текста шелкографии должны быть видны после сборки печатной платы (не находиться под компонентами).

Для подключения питания, нагрузки, вентилятора и температурного датчика применим клеммные колодки, что даст удобство подключения и возможность автономного применения устройства.

Создадим файл печатной платы и перенесем компоненты и цепи из схемы. Фрагмент редактора представлен на рисунке 24.

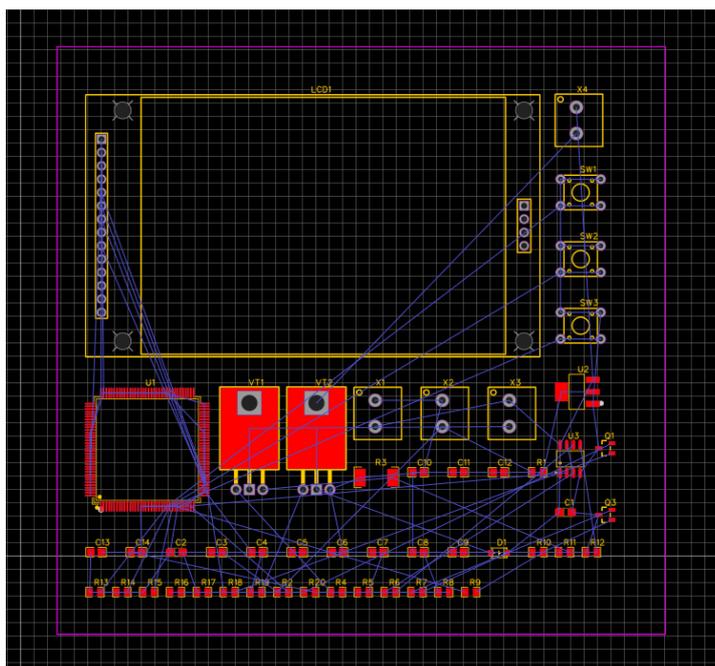


Рисунок 24 – Окно редактора печатной платы

Выполним трассировку печатной платы, уделяя особое внимание нагруженным участкам цепей, как на рисунке 25, и расположению развязывающих конденсаторов [22].

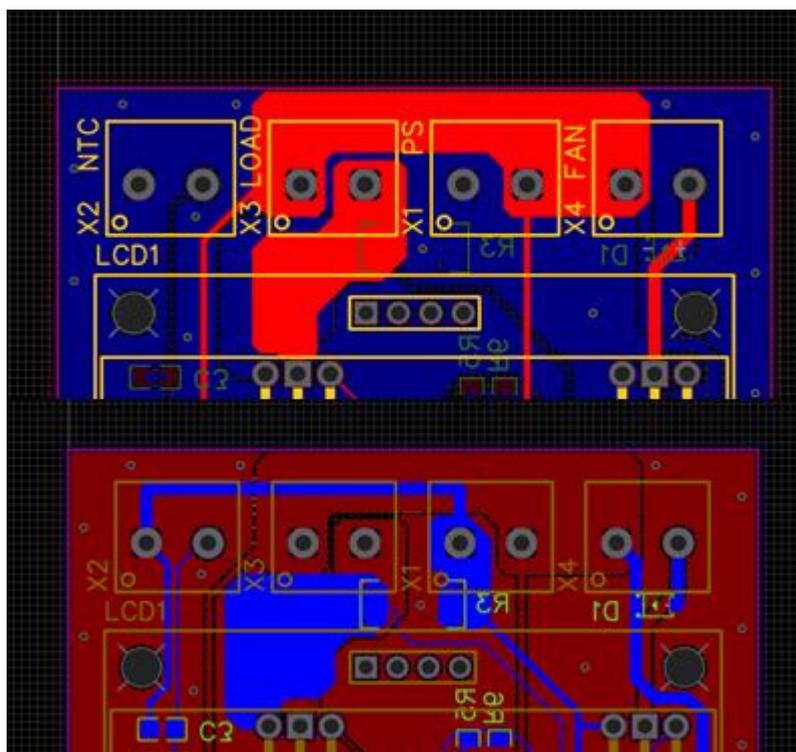


Рисунок 25 – Массивные полигоны нагруженных цепей

Согласно ГОСТ Р 55490-2013, для повышения электромагнитной совместимости расположим на всей площади печатной платы полигоны GND на двух слоях. Это обеспечит оптимальное распределение полей и снизит паразитные перекрестные помехи между цепями.

Для повышения точности измерения тока трассировку токового шунта так же следует выполнить особым образом. Измерительная цепь должна располагаться не на пути протекания тока и как можно ближе к посадочному месту, устраняя влияние сопротивлений дополнительных участков меди [15]. Выполненная трассировка шунта изображена на рисунке 26.

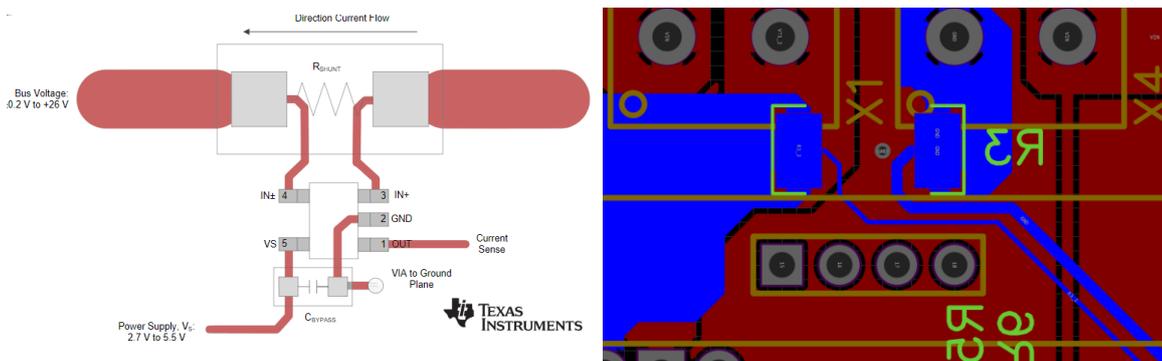


Рисунок 26 – Рекомендации трассировки токового шунта и выполненная трассировка

Так же вся плата имеет множество переходных отверстий, которые соединяют полигоны двух слоев [21].

После проверки расположения компонентов на 3D виде итоговый прототип печатной платы на рисунке 27 нужно подготовить к производству.

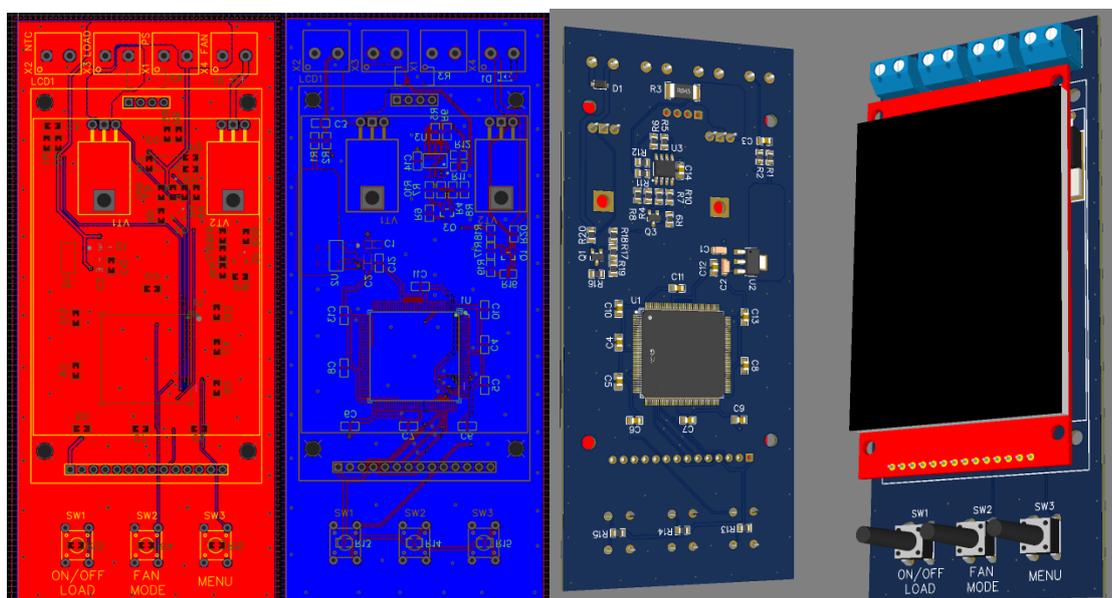


Рисунок 27 – Окно редактора печатной платы и её 3D вид

Для оформления производства печатной платы на производстве создадим выходные GERBER файлы. Полученные файлы и их назначение перечислим в таблице 7.

Таблица 7 – GERBER файлы для производства

Файл	Назначение
Gerber_BoardOutlineLayer.GKO	Контур печатной платы
Gerber_TopLayer.GTL	Верхний слой проводников
Gerber_TopSilkscreenLayer.GTO	Верхний слой шелкографии
Gerber_TopSolderMaskLayer.GTS	Верхний слой паяльной маски
Gerber_BottomLayer.GBL	Нижний слой проводников
Gerber_BottomSilkscreenLayer.GBO	Нижний слой шелкографии
Gerber_BottomSolderMaskLayer.GBS	Нижний слой паяльной маски
Drill_PTH_Through.DRL	Сквозные монтажные отверстия
Drill_PTH_Through_Via.DRL	Сквозные переходные отверстия

Стандарт GERBER широко распространен и используется на всех предприятиях производства печатных плат.

В данном разделе была выполнена трассировка печатной платы и подготовлены файлы для производства.

6 Эскизная компоновка корпуса изделия

Для завершения изделия необходимо поместить печатную плату устройства в корпус. Корпус будет служить сразу нескольким целям. Первая – защита от прикосновений к участкам платы под напряжением, что повысит безопасность использования. Вторая – эстетичный внешний вид, который может привлечь внимание.

Эскиз корпуса изделия выполним в САПР Autodesk Fusion 360 с бесплатной лицензией. С помощью встроенного функционала EasyEDA получим 3D модель печатной платы с компонентами [23]. Импортируем модель в Autodesk Fusion 360, что изображено на рисунке 28.

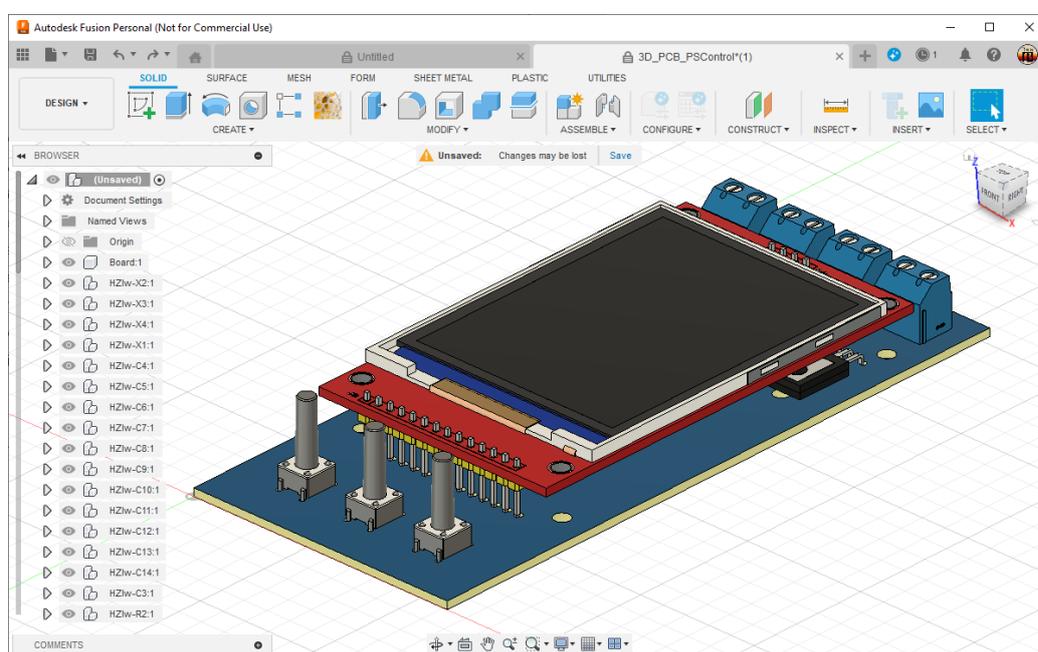


Рисунок 28 – Окно редактора Autodesk Fusion 360 и 3D вид импортированной платы

Зная расположение всех крепежных и монтажных отверстий, можно приступить к разработке корпуса.

Готовый корпус изображен на рисунке 29, с расположенной внутри печатной платой.



Рисунок 29 – 3D вид корпуса

Исходные файлы корпуса можно отправить на производство и получить готовое изделие.

В данном разделе было выполнено проектирование корпуса для устройства мониторинга состояния блока питания и подготовлен эскизный чертеж.

7 Разработка управляющей программы

7.1 Создание и настройка проекта

Для написания управляющей программы необходимо подготовить проект для микроконтроллера STM32F767ZI. Для настройки микроконтроллера и генерации проекта будет использована программа STM32 CubeMX, а для написания исходного кода программа STM32 Cube IDE.

На рисунке 30 изображено создание нового проекта под отладочную плату Nucleo-F767ZI.

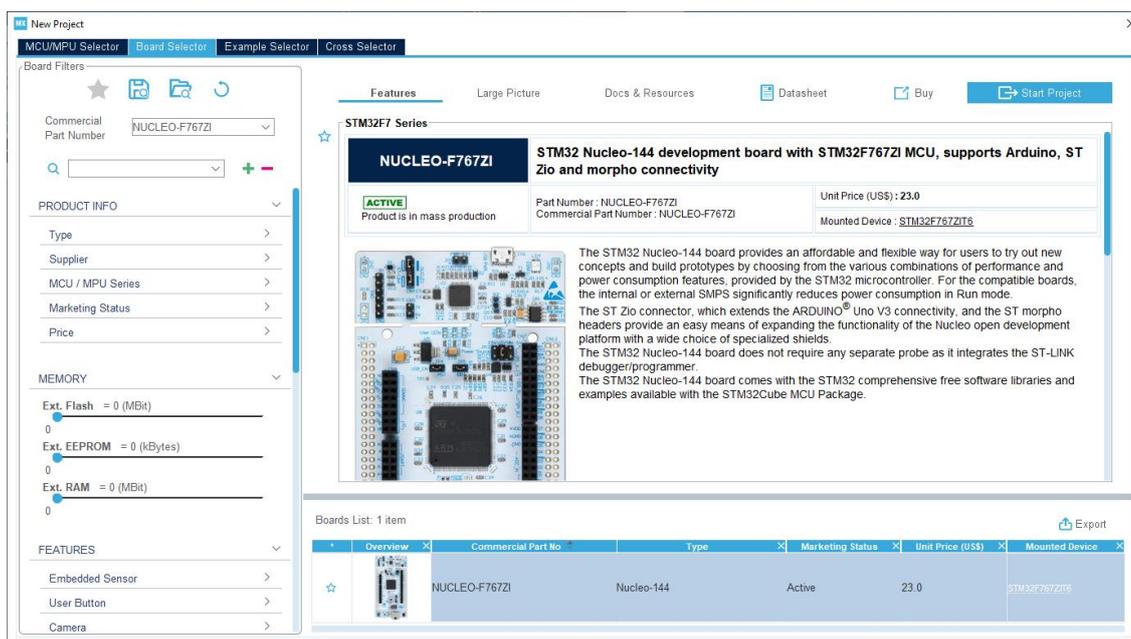


Рисунок 30 – Создание проекта

В первую очередь настроим тактирование во вкладке «Clock Configuration». Выберем источник тактирования «HSE» (High Speed External) — внешний кварцевый резонатор на 8 МГц. У микроконтроллеров STM32 есть аппаратный блок умножения и фазовой автоподстройки частоты, который позволяет получить любую необходимую частоту на основе исходной. Настроим блок так, чтобы на выходе получить частоту 96 МГц, результат на рисунке 31.

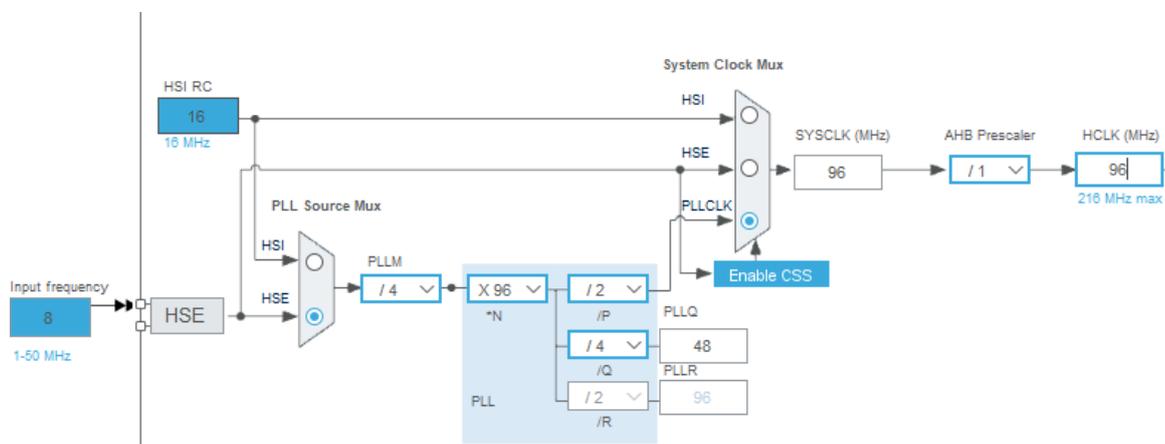


Рисунок 31 – Настройка тактирования

Далее для LCD дисплея активируем и настроим блок SPI (Serial Peripheral Interface) — последовательный синхронный стандарт передачи данных [4]. Схема подключения LCD дисплея по SPI изображена на рисунке 32.

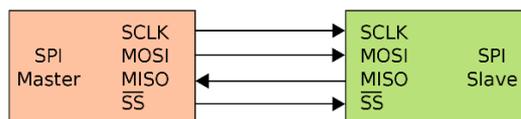


Рисунок 32 – Подключение дисплея

Выбираем полнодуплексный режим работы, размерность данных 8 бит, перед делитель тактовой частоты 8, результат на рисунке 33. Получаем скорость передачи данных 12,0 Мбит/с, что является рекомендуемой для контроллера дисплея.

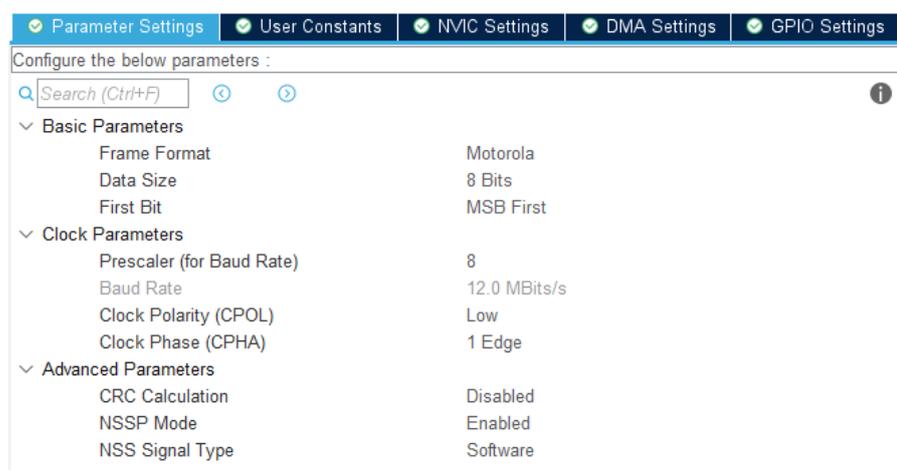


Рисунок 33 – Настройка блока SPI

Далее активируем три канала АЦП для оцифровки напряжения, тока и температуры. Чтобы упростить получение данных настроим каналы в режим «Injected Conversion Mode», в котором под результат оцифровывания для каждого канала присваивается свой адрес в памяти, результат на рисунке 34. Каждому каналу присвоим свой «Injected Rank».

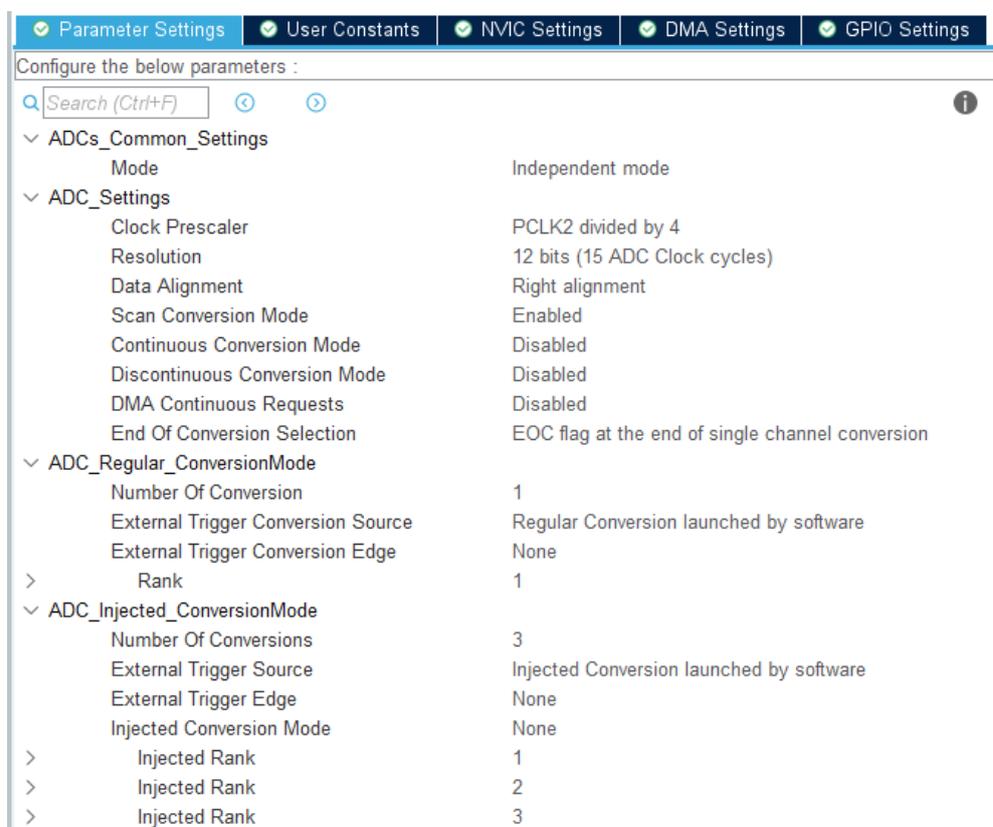


Рисунок 34 – Настройка блока АЦП

Для управления вентилятором необходимо включить таймер TIM3. Согласно схеме, вентилятор подключается к выводу PB0, который можно настроить на генерацию ШИМ сигнала (канал 3). Частоту управления выберем

за границами слышимого диапазона, чтобы избежать высокочастотного звона обмоток мотора вентилятора.

Подойдут значения 20 кГц и выше [19]. Для этого установим делитель частоты «Prescaler» на значение 31, а счетный период «Counter Period» на 99, что даст нам частоту ШИМ равную 30 кГц, что представлено на рисунке 35. Такие значения «Counter Period» так же позволит напрямую устанавливать заполнение от температуры без необходимости дополнительных вычислений.

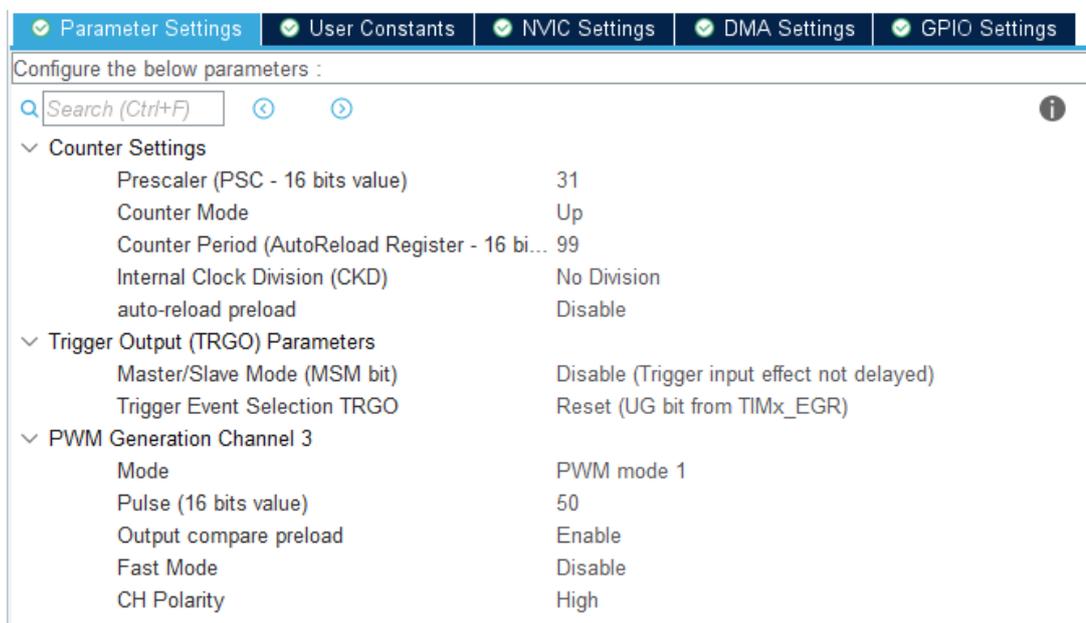


Рисунок 35 – Настройка таймера в режиме ШИМ

Для всех остальных необходимых портов ввода-вывода вся настройка заключается лишь в выборе режимы работы как вход (GPIO_Input) или выход (GPIO_Output). Итоговое окно с настроенными портами ввода-вывода изображено на рисунке 36.

Pin Name	Signal on Pin	GPIO output le...	GPIO mode	GPIO Pull-up/...	Maximum outp...	Fast Mode	User Label	Modified
PB7	n/a	Low	Output Push Pull	No pull-up and ...	Low	Disable	LD2 [Blue]	<input checked="" type="checkbox"/>
PB14	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	LD3 [Red]	<input checked="" type="checkbox"/>
PC0	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	BTN1	<input checked="" type="checkbox"/>
PC1	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	BTN2	<input checked="" type="checkbox"/>
PC2	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	BTN3	<input checked="" type="checkbox"/>
PC3	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	MOSFET	<input checked="" type="checkbox"/>
PC13	n/a	n/a	External Interru...	No pull-up and ...	n/a	n/a	USER_Btn [B1]	<input checked="" type="checkbox"/>
PD14	n/a	Low	Output Push Pull	No pull-up and ...	Medium	n/a	TFT_CS	<input checked="" type="checkbox"/>
PD15	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	TFT_BL	<input checked="" type="checkbox"/>
PE11	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	SD_CS	<input checked="" type="checkbox"/>
PE13	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	TP_IRQ	<input checked="" type="checkbox"/>
PF13	n/a	Low	Output Push Pull	No pull-up and ...	Medium	n/a	TFT_DC	<input checked="" type="checkbox"/>
PF14	n/a	Low	Output Push Pull	No pull-up and ...	Medium	n/a	TP_CS	<input checked="" type="checkbox"/>
PG6	n/a	Low	Output Push Pull	No pull-up and ...	Low	n/a	USB_PowerSwi...	<input checked="" type="checkbox"/>
PG7	n/a	n/a	Input mode	No pull-up and ...	n/a	n/a	USB_OverCurre...	<input checked="" type="checkbox"/>

Рисунок 36 – Настройка GPIO

После внесения всех необходимых изменений можно сгенерировать настроенный проект кнопкой «Generate Code».

В данном подразделе был создан проект для микроконтроллера и выполнена инициализация периферии.

7.2 Описание исходного кода программы

Файл main.c устроен так, что пользовательский код допускается только в специальных секциях «/* USER CODE BEGIN */». Сделано это для возможности исправления настроек проекта через CubeMX, так как код инициализации располагается в том же файле main.c.

Далее необходимо определить алгоритм программы, чтобы на основе него нарисовать необходимые функции и различные ветвления. Составим предварительную блок-схему программы на рисунке 37.

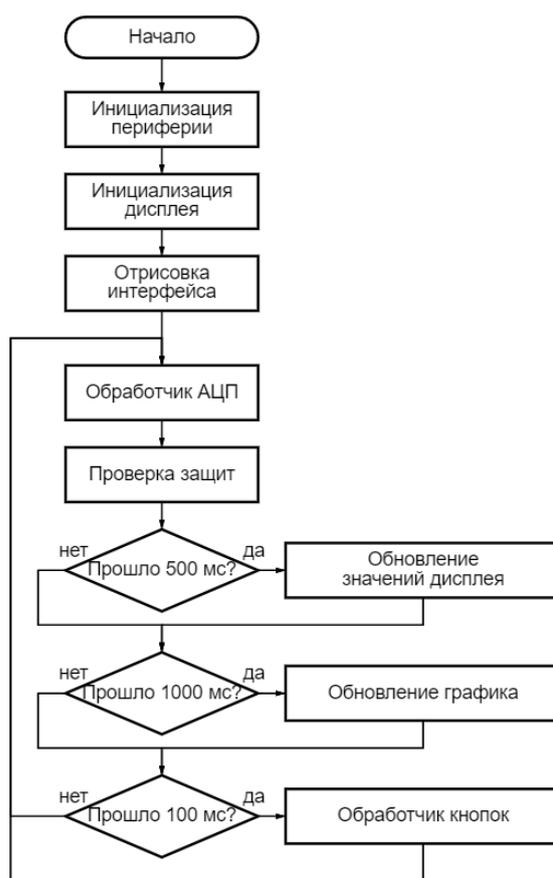


Рисунок 37 – Предварительная блок-схема программы

Инициализацию периферии выполняет CubeMX, поэтому, открыв проект, можно увидеть, что эта часть уже готова (листинг 1).

Далее необходимо выполнить инициализацию дисплея и отобразить пользовательский интерфейс. Дополнительно можно запустить счет таймера TIM3. Интерфейс выполнен в виде текста, который дополняется линиями и прямоугольниками, используя функции `lcd_draw_text()`, `lcd_draw_line()` и `lcd_fill_rect()` (листинг 2). Все используемые графические функции объявлены и описаны в драйвере LCD дисплея от производителя [14].

Листинг 1 - Инициализация периферии

```
/* MCU Configuration-----*/
/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
HAL_Init();
/* Configure the system clock */
SystemClock_Config();
/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_SPI1_Init();
MX_USART3_UART_Init();
MX_RNG_Init();
MX_ADC1_Init();
MX_TIM3_Init();
```

Листинг 2 – Графический интерфейс

```
/* USER CODE BEGIN 2 */
lcd_init(); //инициализация дисплея
lcd_clear_screen(WHITE); // очистка дисплея
//рисуем интерфейс
lcd_draw_text("U    V", 13, 10, WHITE, 4, RED_PS);
lcd_draw_text("I    A", 13, 42, WHITE, 4, GREEN_PS);
lcd_draw_text("t    'C", 13, 74, BLACK, 4, WHITE);
lcd_fill_rect(10, 116, 220, 100, WHITE);
lcd_fill_rect(9, 116, 5, 5, BLACK);
lcd_fill_rect(10, 116, 3, 100, BLACK);
lcd_fill_rect(10, 216, 220, 3, BLACK);
lcd_fill_rect(225, 215, 5, 5, BLACK);
```

```
lcd_draw_text("FAN:", 10, 226, BLACK, 3, WHITE);  
lcd_draw_text("LOAD:", 10, 250, BLACK, 3, WHITE);
```

```
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_3); //запускаем ШИМ  
вентилятора
```

```
/* USER CODE END 2 */
```

Далее программа попадает в основной цикл, где выполняются все остальные действия (листинг 3).

Раз в 500мс выполняется обновление значений на дисплее. Каждую секунду обновляется график температуры блока питания и применяется новое значение управления вентилятором. Отображается температура за последние 50 секунд. Временные границы могут быть настроены изменением частотой вызова функции `draw_graph()`. На графике помещается 50 точек. Функция `fan_control()` принимает на вход процент от максимального значения оборотов вентилятора. Раз в 100мс выполняется опрос кнопок.

Листинг 3 – Основной цикл

```
/* USER CODE BEGIN WHILE */  
while (1)  
{  
  adc_handler(); // как можно чаще запускаем АЦП  
  protection(); // и проверяем защиты  
  if(HAL_GetTick() - t0 > 500) { // каждые 500мс  
    t0 = HAL_GetTick();  
    update_values(); //обновляем значения на дисплее  
  }  
  if(HAL_GetTick() - t1 > 1000) { // каждую 1с  
    t1 = HAL_GetTick();  
    draw_graph(); // обновляем график  
    fan_control(map(ps_temperature, 0, 100, 0, 100)); // управляем вентилятором  
  }  
  if(HAL_GetTick() - t2 > 100) { //каждые 100мс  
    t2 = HAL_GetTick();  
    button_handler(); //опрашиваем кнопку  
  }  
}  
/* USER CODE END WHILE */
```

Далее рассмотрим подробнее реализацию функций основного цикла.

Запуск АЦП выполняется с помощью функции HAL_ADCEX_InjectedStart(). После ожидания завершения преобразования функцией HAL_ADCEX_InjectedGetValue() происходит чтение значений в соответствующие переменные для дальнейшего пересчета (листинг 4) [19].

Чтобы обеспечить быстрое срабатывание защит, преобразование не привязано к периоду во времени и выполняется с максимальной возможной скоростью.

Листинг 4 – Обработка АЦП

```
void adc_handler() {
    HAL_ADCEX_InjectedStart(&hadc1); //запуск преобразования
    HAL_ADCEX_InjectedPollForConversion(&hadc1, 100); //ожидание завершения
    adc_current_val = HAL_ADCEX_InjectedGetValue(&hadc1,
    ADC_INJECTED_RANK_1); //чтение значений
    adc_temperature_val = HAL_ADCEX_InjectedGetValue(&hadc1,
    ADC_INJECTED_RANK_2);
    adc_voltage_val = HAL_ADCEX_InjectedGetValue(&hadc1,
    ADC_INJECTED_RANK_3);
}
```

Соответственно, функция защит protection() так же не привязана к периоду. Функция проверяет текущие значения с заранее установленными лимитами (листинг 5). Если обнаружено превышение, то выполняется быстрое отключение нагрузки функцией load_control().

Листинг 5 – Функции защит

```
void load_control(bool state) {
    HAL_GPIO_WritePin(MOSFET_GPIO_Port, MOSFET_Pin, !state);
}
void protection() {
    if(ps_voltage > OV_LIMIT) { //если больше допустимого
        load_control(RESET); //откл. нагрузку
        load_state = OFF; //меняем режим
    }
}
```

```

    protect_flag = OV; //поднимаем флаг
}
if(ps_current > OC_LIMIT) { //аналогично
    load_control(RESET);
    load_state = OFF;
    protect_flag = OC;
}
if(ps_temperature > OT_LIMIT) { //аналогично
    load_control(RESET);
    load_state = OFF;
    protect_flag = OT;
}
}
}

```

update_values() – функция обновления значений на дисплее. Сначала функцией map() выполняется пересчет результата АЦП в действующие значения. С помощью стандартной функции sprintf() подготавливается строка с нужным форматом, а функция lcd_draw_text() отображает строку на дисплее. Таким образом обновляются значения всех значений: напряжение, ток, температура, режим работы вентилятора, состояние нагрузки и сообщение защиты (листинг 6).

Листинг 6 – Обновление значений

```

float map(float x, float in_min, float in_max, float out_min, float out_max) {
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}
void update_values() {
    float logRt;
    ps_voltage = map(adc_voltage_val, 0, 4095, 0, 200.0); //считаем напряжение
    ps_current = map(adc_current_val, 0, 4095, 0, 10.0); //считаем ток
    logRt = log(10000.0 * (4095.0 / adc_temperature_val - 1)); //считаем температуру
    ps_temperature = 1.0 / (a + b * logRt + c * pow(logRt, 3)); // по уравнению
    Стейнхарта-Харта
    sprintf(lcd_str, "%5.1f", ps_voltage + add_V); //формируем строки
    lcd_draw_text(lcd_str, 58, 10, WHITE, 4, RED_PS); // и выводим на экран
    sprintf(lcd_str, "%4.1f", ps_current);
    lcd_draw_text(lcd_str, 82, 42, WHITE, 4, GREEN_PS);
    sprintf(lcd_str, "%5.1f", ps_temperature);
    lcd_draw_text(lcd_str, 58, 74, BLACK, 4, WHITE);
}

```

```

switch(fan_mode) { //пишем режим вентилятора
  case AUTO: lcd_draw_text(" AUTO", 136, 226, BLACK, 3, WHITE); break;
  case MAX: lcd_draw_text(" MAX", 136, 226, BLACK, 3, WHITE); break;
}
switch(load_state) { //пишем режим нагрузки
  case ON: lcd_draw_text(" ON", 172, 250, BLACK, 3, WHITE); break;
  case OFF: lcd_draw_text("OFF", 172, 250, BLACK, 3, WHITE); break;
}
switch(protect_flag) { // если сработала защита, то пишем сообщение
  case OV: lcd_draw_text("OVERVOLTAGE!", 10, 284, WHITE, 3, RED); break;
  case OC: lcd_draw_text("OVERLOAD!", 10, 284, WHITE, 3, RED); break;
  case OT: lcd_draw_text("OVERHEAT!", 10, 284, WHITE, 3, RED); break;
}
}

```

draw_graph() – функция обновления графика температуры (листинг 7). История реализована на большом массиве temperature_history[], который сдвигается при каждом вызове draw_graph(), а новое значение записывается в последний элемент.

Далее температура из каждого элемента temperature_history[] пересчитывается в координаты точек на дисплее, учитывая смещение и порядковый номер элемента. Полученный набор точек хранится в локальном массиве graph_dots_new[] в формате [X0][Y0][X1][Y1][X2][Y2] и т.д. После получения новых точек выполняется стирание старого графика на основе старых точек из массива graph_dots[] и рисование нового из данных graph_dots_new[]. Стирание и рисование выполняется функцией lcd_draw_line(). После завершения рисования нового графика необходимо присвоить массиву старых точек значения массива новых точек [3].

Листинг 7 – Обновление графика

```

void draw_graph() {
  uint16_t graph_dots_new[104];
  for(uint8_t i = 0; i <= 50; i++) { // сдвигаем весь массив влево
    temperature_history[i] = temperature_history[i + 1];
  }
}

```

```

temperature_history[51] = (int8_t)ps_temperature; // новое значение температуры
for(uint8_t i = 0; i <= 50; i++) { // считаем новые координаты линий
    graph_dots_new[i * 2] = 13 + (i * 4); // точка X, новая точка каждые 4 px
    graph_dots_new[i * 2 + 1] = (uint16_t)((235 - temperature_history[i]) * 0.9); //
точка Y
}
for(uint8_t i = 0; i <= 49; i++) { // стираем старый график и рисуем новый
    lcd_draw_line(graph_dots[i * 2], graph_dots[i * 2 + 1], graph_dots[i * 2 + 2],
graph_dots[i * 2 + 3], WHITE);
    lcd_draw_line(graph_dots_new[i * 2], graph_dots_new[i * 2 + 1],
graph_dots_new[i * 2 + 2], graph_dots_new[i * 2 + 3], BLACK);
}
for(uint8_t i = 0; i <= 103; i++) { // обновляем старые точки
    graph_dots[i] = graph_dots_new[i];
}
}

```

button_handler() – функция опроса и обработки кнопки (листинг 8). Алгоритм основан на проверке старого и нового состояния. Для этих целей были объявлены флаги button_new, button_old, button_hold_flag. Функция button_press() срабатывает только при условии button_new=1 и button_old=0, что соответствует моменту нажатия. button_release() работает аналогичным образом, но с обратными значениями. При удержании кнопки происходит инкремент счетчика button_cnt и если он превышает порог, то разово выполняется функция button_hold(). Кратковременное нажатие кнопки переключает состояние нагрузки, а долгое удержание переключает режим работы вентилятора.

Листинг 8 – Обработка кнопки

```

void button_handler() {
    bool button_new = HAL_GPIO_ReadPin(USER_Btn_GPIO_Port,
USER_Btn_Pin);
    if (button_new && !button_old) { // нажатие
        button_old = true;
        button_press();
    }
    if (!button_new && button_old) { // отпускание
        button_old = false;
    }
}

```

```

    button_hold_flag = false;
    button_cnt = 0;
    button_release();
}
if(button_new == GPIO_PIN_SET) { //если держим
    if(button_hold_flag == false) button_cnt++; //прибавляем счетчик
    if(button_cnt >= 3) { //если держим долго
        button_hold_flag = true;
        button_cnt = 0;
        button_hold(); //удержание
    }
}
}
void button_press() { // нажатие кнопки
    if((load_state == OFF) && (protect_flag != NORMAL)) {
        lcd_draw_text("      ", 10, 284, WHITE, 3, WHITE);
        protect_flag = NORMAL;
    }
    if(load_state == ON) {
        load_control(RESET);
        load_state = OFF;
    } else {
        load_control(SET);
        load_state = ON;
    }
}
void button_release() { //отпускание кнопки
}
void button_hold() { //удержание кнопки
    if(fan_mode == AUTO) fan_mode = MAX;
    else fan_mode = AUTO;
}

```

fan_control() – функция управления вентилятором, принимающая значение в процентах (листинг 9). В зависимости от текущего режима работы вентилятора (переменная fan_mode) выполняется макрос __HAL_TIM_SET_COMPARE(), который записывает новое значение в регистр сравнения таймера, что изменяет заполнение ШИМ сигнала.

Предусмотрено два режима: автоматический (относительно температуры) и непрерывный (максимальные обороты). Автоматический режим имеет линейную зависимость, но не менее 30% от максимального значения оборотов вентилятора.

Листинг 9 – Функция вентилятора

```
void fan_control(uint8_t percent) {
    percent = constrain(percent, 30, 100); //ограничиваем значение
    switch (fan_mode) { //относительно режима устанавливаем новое значение
        ШИМ
        case AUTO: __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_3,
percent); break;
        case MAX: __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_3, 99);
break;
    }
}
```

В данном подразделе была описана управляющая программа для микроконтроллера STM32F767ZI. Описан алгоритм работы и представлены используемые функции.

Заключение

В данной работе была выполнена разработка прототипа автономного устройства цифрового мониторинга состояния источника питания.

Устройство включает в себя функции отображения параметров, их хранение, контроль характеристик, а также управление исполнительными элементами. Осуществляется вывод значений на ЖК-дисплей. Пользователь может менять режимы работы устройства, воздействуя на встроенные элементы управления.

Результатом выпускной квалификационной работы стало готовое для опытного производства устройство с полным набором необходимой конструкторской документации. Был выполнен обзор аналогичных решений, выбраны комплектующие и на основе них разработана принципиальная схема. Далее была выполнена компоновка и трассировка печатной платы, а также разработан корпус изделия. В основе устройства был применен микроконтроллер STM32F767ZI, который обладает хорошей производительностью. Для выполнения логики работы устройства была написана управляющая программа под все требования задания.

По разработанной документации устройство является опытным образцом. Областью применения данного устройства являются блоки питания любого типа, что реализовано за счет автономного исполнения. Применение устройства благоприятно сказывается на качестве питающего напряжения, так как в функционал включены защиты по превышению тока, напряжения и температуры.

При выполнении задания были отработаны навыки прототипирования, а также получены практические навыки работы в интегрированных средах разработки.

Список используемой литературы и используемых источников

1. Болдырев, И. А. Схемотехническое проектирование систем управления / И. А. Болдырев, М. И. Герасимов, А. С. Кожин; под редакцией В. Л. Бурковского. — Воронеж: Воронежский государственный технический университет, 2019. — 72 с.
2. Болдырев, И. А. Схемотехническое проектирование систем управления / И. А. Болдырев, М. И. Герасимов, А. С. Кожин; под редакцией В. Л. Бурковского. — Воронеж: Воронежский государственный технический университет, 2019. — 159 с.
3. Дорогов В. Г. Основы программирования на языке С: учеб. пособие / В. Г. Дорогов, Е. Г. Дорогова; под ред. Л. Г. Гагариной. - Москва, 2019. - 224 с.
4. Наумкина, Л. Г. Цифровая схемотехника: Москва, МГУ, 2008. – 189 с.
5. Техническая документация STM32F767ZIT6 «Product Specifications» (DS11532), STMicroelectronics. [Электронный ресурс] – Режим доступа: URL: <https://www.st.com/resource/en/datasheet/stm32f767zi.pdf> (дата обращения 22.04.24).
6. Техническая документация STM32F767ZIT6 «Reference Manuals» (RM0410), STMicroelectronics. [Электронный ресурс] – Режим доступа: URL: https://www.st.com/resource/en/reference_manual/rm0410-stm32f76xxx-and-stm32f77xxx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf (дата обращения 22.04.24).
7. Техническая документация AMS1117-3.3 «Advanced Monolithic Systems – Low Dropout Voltage Regulator», Advanced Monolithic Systems, Inc., Sierra Lane, Dublin, CA. [Электронный ресурс] – Режим доступа: URL: <https://oskolchip.ru/datasheet/AMS1117.pdf> (дата обращения 18.04.24).

8. Техническая документация LM358 «Single Supply Dual Operational Amplifier», Semiconductor Components Industries, [Электронный ресурс] – Режим доступа: URL: <https://www.onsemi.com/pdf/datasheet/lm358-d.pdf> (дата обращения 25.04.24).
9. Техническая документация CRS SMD Shunt Resistor, ARCOL UK Limited, [Электронный ресурс] – Режим доступа: URL: <https://www.farnell.com/datasheets/2026270.pdf> (дата обращения 19.04.24).
10. Техническая документация ESP8266 «Technical Reference», ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD, [Электронный ресурс] – Режим доступа: URL: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf (дата обращения 25.04.24).
11. Техническая документация ATmega328/P «low-power CMOS 8-bit microcontroller», Atmel, [Электронный ресурс] – Режим доступа: URL: <https://voltiq.ru/datasheets/ATMEGA328P.pdf> (дата обращения 24.04.24).
12. Техническая документация STM32 Nucleo-64 «Product Specifications», STMicroelectronics. [Электронный ресурс] – Режим доступа: URL: https://www.st.com/resource/en/data_brief/nucleo-f401re.pdf (дата обращения 22.04.24).
13. Техническая документация DS18B20 «Programmable Resolution 1-Wire® Digital Thermometer», Dallas Semiconductor. [Электронный ресурс] – Режим доступа: URL: <https://voltiq.ru/datasheets/ds18b20.pdf> (дата обращения 21.04.24).
14. Техническая документация ST7789V «240RGB x 320 dot 262K Color with Frame Memory Single-Chip TFT Controller», Sitronix Technology. [Электронный ресурс] – Режим доступа: URL: <https://newhavendisplay.com/content/datasheets/ST7789V.pdf> (дата обращения 20.04.24).

15. Техническая документация ОРАx277 «High Precision Operational Amplifiers», Texas Instruments. [Электронный ресурс] – Режим доступа: URL: <https://www.elmatrix.ru/upload/docs/OPA2277.pdf> (дата обращения 26.04.24).
16. Техническая документация РТН08Т230W «ADJUSTABLE POWER MODULE», Texas Instruments. [Электронный ресурс] – Режим доступа: URL: <https://voltiq.ru/datasheets/pth08t230w.pdf> (дата обращения 29.04.24).
17. Титце Ульрих, Шенк Кристоф Полупроводниковая схемотехника. Том I; пер. Г. С. Карабашев. — Саратов: Профобразование, 2017. — 826 с.
18. Титце Ульрих, Шенк Кристоф Полупроводниковая схемотехника. Том II; пер. Г. С. Карабашев. — Саратов: Профобразование, 2017. — 237 с.
19. Carmine Noviello Mastering STM32: A Step-by-step Guide to the Most Complete ARM Cortex-M Platform, Second Edition, Leanpub, 2016. – 57-62 с.
20. David Harris and Sarah L. Harris, Digital Design and Computer Architecture, Morgan Kaufmann, 2007. – 368 – 372 с.
21. Douglas Brooks, Johannes Adam PCB Design Guide to Via and Trace Currents and Temperatures, Artech House, 2021. – 85-92.
22. Hamilton, Scott An analog electronics companion: basic circuit design for engineers and scientists, Cambridge ; New York, 2003. – 125-131 с.
23. Jake O Sugden, Joshua Manley Mastering Fusion 360 - SliceForm LLC, 2023 – 204 – 2015 с.
24. Paul Horowitz, Winfield Hill The Art of Electronics (3rd ed.), Cambridge University Press, 2015. – 98-101 с.
25. Peter Wilson The Circuit Designer's Companion – Newnes, 2017. – 189 – 193 с.