

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

Факультет Электротехнический

Кафедра автоматики

Направление подготовки 27.03.04 «Управление в технических системах»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой

\_\_\_\_\_  
Голощапов С. С.

\_\_\_\_\_  
2021 г.

Контроллер температуры с образцом свидетеля и синхронизацией  
с мобильным приложением

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ– 27.03.04.2021.132.00.00ПЗ ВКР

**Директор ЦНПП**

\_\_\_\_\_  
подпись / Усачёв Д.В.  
ФИО  
\_\_\_\_\_  
2021 г.

**Автор ВКР**

студент группы МиЭт- 529  
\_\_\_\_\_  
подпись / Скрябин А.Ю.  
ФИО  
\_\_\_\_\_  
2021 г.

**Руководитель отделения предприятия**

\_\_\_\_\_  
подпись / \_\_\_\_\_  
ФИО  
\_\_\_\_\_  
2021 г.

**Руководитель работы**

д.т.н., профессор кафедры автоматика  
должность  
\_\_\_\_\_  
подпись / Белугин В.Б.  
ФИО  
\_\_\_\_\_  
2021 г.

**Начальник лаборатории 124**

\_\_\_\_\_  
подпись / Кинзин Р.А.  
ФИО  
\_\_\_\_\_  
2021 г.

**Консультант**

инженер \_\_\_\_\_  
должность  
\_\_\_\_\_  
подпись / Волокитин Н.А.  
ФИО  
\_\_\_\_\_  
2021 г.

**Нормоконтроль**

старший преподаватель  
должность  
\_\_\_\_\_  
подпись / Елисеев В.П.  
ФИО  
\_\_\_\_\_  
2021 г.

## АННОТАЦИЯ

Скрябин А.Ю. Контроллер температуры с образцом свидетеля и синхронизацией с мобильным приложением – Миасс: ЮУрГУ ЭТФ; 2021, 77 с. 26 ил., библиогр. список - 20 наименований, 4 прил., 1 лист чертеж ф. А2.

После анализа технического задания были изучены отечественные и зарубежные аналоги устройств контролеров температуры. Разработана функциональная схема системы управления устройством контроля температуры для вакуумной установке УВН 71П-3.

Разработана электрическая принципиальная схема и подобраны элементы, разработано программное обеспечение для микроконтроллера и для мобильного приложения.

Данное устройство позволит задавать и отслеживать температуру в вакуумной камере при нанесении гальванических покрытий и снимать показания с образца свидетеля для контроля толщин нанесения покрытия.

					<b>27.03.04.2021.132.00.00 ВКР ПЗ</b>			
<i>Изм.</i>	<i>Лист</i>	<i>№ Докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>		Скрябин А.Ю			<i>Контроллер температуры с образцом свидетеля и синхронизацией с мобильным приложением</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>		Волокитин Н.А						
<i>Реценз.</i>								
<i>Н. Контр.</i>		Елисеев В.П.						
<i>Утверд.</i>								
						<b>ЮУрГУ Кафедра ЭТФ</b>		

	Оглавление	
	ВВЕДЕНИЕ .....	5
1	ОБЗОР АНАЛОГОВ РАЗРАБАТЫВАЕМОГО УСТРОЙСТВА .....	6
	1.1 Обзор ПИД-регуляторов.....	6
	1.2 Выводы по разделу один.....	12
2	ОСНОВНОЙ МАТЕРИАЛ ПО СПЕЦИАЛЬНОЙ ЧАСТИ.....	13
	2.1 Разработка функциональной схемы системы управления.....	13
	2.2 Выбор элементной базы.....	15
	2.2.1 Выбор микроконтроллера.....	15
	2.2.2 Выбор устройства ввода и вывода.....	18
	2.2.3 Выбор устройства вывода информации .....	21
	2.2.4 Выбор светодиодов .....	25
	2.2.5 Выбор модуля передачи данных на мобильное приложение .....	26
	2.2.6 Выбор элементов входящих в блок управления нагрузкой.....	28
	2.2.7 Расчет источника питания .....	29
	2.3 Построение схемы электрической принципиальной .....	34
	2.4 Вывод по разделу два.....	36
3	РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	37
	3.1 Выбор среды разработки.....	37
	3.2 Расчет ПИД- регулятора .....	38
	3.3 Разработка программного обеспечения.....	42
	3.4 Вывод по разделу три.....	43
	ЗАКЛЮЧЕНИЕ.....	44
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	45
	ПРИЛОЖЕНИЕ А. ФУНКЦИОНАЛЬНАЯ СХЕМА УСТРОЙСТВА.....	47
	ПРИЛОЖЕНИЕ Б. СХЕМА ЭЛЕКТРИЧЕСКАЯ ПРИНЦИПАЛЬНАЯ.....	48
	ПРИЛОЖЕНИЕ В. ЛИСТИНГ ПРОГРАММЫ МК .....	49
	ПРИЛОЖЕНИЕ Г. ЛИСТИНГ ПРОГРАММЫ МОБИЛЬНОГО ПРИЛОЖЕНИЯ.....	74

## ВВЕДЕНИЕ

Управление технологическими процессами с помощью регуляторов, работающих по пропорционально-интегрально-дифференциальному закону (ПИД-регуляторов), позволяет поддерживать требуемую технологическую величину с достаточно высокой точностью и приемлемой помехоустойчивостью к внешним возмущающим воздействиям. [1].

ПИД регуляторы находят широкое применение в современных системах точного контроля, таких как управление термосистемами и системами позиционирования. Использование ПИД регуляторов помогает уменьшить энергетические потери на настройку системы и обеспечивают более быстрый выход на требуемые параметры.

Целью данной ВКР является модернизация вакуумной установки УВН 71П-3 с помощью микроконтроллера (МК) который будет контролировать температуру в вакуумной камере для качественного нанесения вакуумного покрытия, также с помощью МК будут сниматься показания с образца свидетеля для контроля толщины нанесенного покрытия и передачи этих данных на мобильное приложение в реальном времени, что бы не было нужды находиться у вакуумной установки на время отработки технологического процесса.

При создании контролера температуры с образцом свидетеля и синхронизацией с мобильным приложением был использован готовый модуль платы разработки системы STM32F103C8T6.

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

# 1 ОБЗОР АНАЛОГОВ РАЗРАБАТЫВАЕМОГО УСТРОЙСТВА

## 1.1 Обзор ПИД-регуляторов

### 1.1.1 Термоконтроллер ПИД регулятор с релейным выходом REX-C100FK02-MAN

Термоконтроллер ПИД регулятор REX-C100FK02-MAN прибор предназначенный для управления процессом нагрева и обеспечивает точное и быстрое регулирование температуры. Одноканальный ПИД-регулятор с функцией автоматической настройки. (Общий вид - рисунок 1.1)



Рисунок 1.1– Термоконтроллер ПИД регулятор с релейным выходом REX-C100FK02-MAN

Основные характеристики C100FK02-MAN представлены в таблице 1.1

Таблица 1.1 – Основные характеристики C100FK02-MANX [10]

Наименование	Описание
Входной сигнал	Термопара типа К (ХА)
Диапазон измерения температуры	0-400°C

Продолжение таблицы 1.1

Относительная погрешность измерения температуры	0,5 %
Вывод информации	2 светодиодных четырёхзначных индикатора
Количество управляющих выходов	1
Выходное устройство (на управление)	реле 250В, 3А
Тип управления	ПИД с ручной и автоматической настройкой (на выбор) либо двухпозиционный
Питание прибора	~85 -265 в 50 Гц
Потребляемая мощность	Не более 10 ВА
Сопротивление изоляции	Более 500 МОм (при тестовом напряжении=500 В)
Условия окружающей среды	Атмосферное давление 86-106 кПа, температура 0-50°С, относительная влажность 30-85 %, отсутствие в воздухе агрессивны газов и паров
Габаритные размеры	48 × 48 × 100 мм

1.1.2 Температурный контроллер с ПИД-регулятором TC4S-14R

Температурный контроллер серия TC4 с ПИД-регулятором.

Обеспечивает оптимальное регулирование температуры за счет использования

										Лист
										7
Изм.	Лист	№ докум.	Подпись	Дата						

27.03.04.2021.132.00 ПЗ

нового алгоритма ПИД-регулирования, а также короткого интервала измерений (100 мс) Один дисплей с возможностью индикации рассогласования между текущим значением и установкой. Питание: 100...240 В~ 50/60 Гц Тип дисплея: 7-сегментный Высота знаков до 22 мм (зависит от модели) Тип входа: ТС (К и J типа), DIN Pt 100 Ом; термопара К(СA), J(IC) Управляющий выход: релейный 250 В переменного тока, 3А; SSR (ТТР) 12 В пост. тока, ±2В, 20 мА макс. Пределы индикации: 0.1 ~ 999.9°С TC4SP подключается в круглую 11-ти контактную колодку. Тип термодатчика К, J, Pt100 (Общий вид - рисунок 1.2)



Рисунок 1.2 - Температурный контроллер с ПИД-регулятором  
TC4S-14R

Основные характеристики TC4S-14R представлены в таблице 1.2

										Лист
										8
Изм.	Лист	№ докум.	Подпись	Дата	27.03.04.2021.132.00 ПЗ					

Таблица 1.2 – Основные характеристики TC4S-14R

Наименование	Описание
Дисплей	7-сегментный, светодиодный
Тип регулирования	вкл/выкл, P, PI, PID, PD
Способ настройки	сенсорный переключатель
Выход управления	релейный + выход ТТРФУ ( выход твердотельного реле с функцией фазового и циклического управления)
Выход сигнализации	выход сигнализации 1
Период измерения	100 мс
Источник питания	100-240 VAC, 50/60Гц
Габаритные размеры	48x48x70,5 мм (с блоком зажимов)
Рабочая температура	-10°C ... +50°C

### 1.1.3 Терморегулятор ОВЕН ТРМ10

Терморегулятор ОВЕН ТРМ10 предназначен для измерения температуры или другой физической величины (веса, давления, влажности и т. п.), импульсного или аналогового регулирования по ПИД закону, а также для формирования дополнительного сигнала, который может быть использован для сигнализации о выходе параметра за установленные границы или для двухпозиционного регулирования.[9]

#### Основные функции ПИД-регулятора ОВЕН ТРМ10

1) Универсальный вход для подключения широкого спектра датчиков температуры, давления, влажности, расхода, уровня и т.п.

									Лист
									9
Изм.	Лист	№ докум.	Подпись	Дата	27.03.04.2021.132.00 ПЗ				



2) ПИД-регулирование измеренной величины с использованием «нагревателя» или «холодильника»

3) Автонастройка ПИД-регулятора по современному эффективному алгоритму

4) Дополнительный выход для сигнализации о выходе регулируемой величины за установленные границы (или для двухпозиционного регулирования)

5) Регулирование мощности (например, для управления инфракрасной лампой) в модификации с аналоговым выходом 4...20 мА, совместно с прибором ОВЕН БУСТ

6) Возможность управления трехфазной нагрузкой

7) Возможность управления трехфазной нагрузкой (модиф. по типу выхода С3)

8) Импульсный источник питания 90...245 В 47...63 Гц

9) Встроенный источник питания 24 В для активных датчиков, выходных аналоговых устройств (ЦАП) и др.

#### 1.1.4 Многоканальный регистратор ЭНИ-702

Видеографический экранный самописец — регистратор ЭНИ-702 многоканальный предназначен для:

1) измерения аналоговых сигналов с датчиков расхода, давления, температуры, уровня, и т.д.;

2) отображения измеренных значений в виде графиков, гистограмм и таблиц на ЖКИ;

3) ведения архива измеренных параметров;

Особенности и преимущества регистратора ЭНИ-702:

										Лист
										10
Изм.	Лист	№ докум.	Подпись	Дата						

- 1) компактный корпус панели индикации;
- 2) модульная архитектура построения;
- 3) до 24 гальванически развязанных универсальных каналов измерения с индивидуальной настройкой;

Внешний вид многоканального регистратора ЭНИ-702 предоставлен на рисунке 1.3



Рисунок 1.3 – Регистратор ЭНИ 702

Технические характеристики ЭНИ -702 предоставлены в таблице 1.3

Таблица 1.3 – Технические характеристики ЭНИ -702 [12]

Параметры	Значение
Диапазон напряжения питания переменного тока, В	110...265
Частота напряжения питания переменного тока, Гц	49...51
Потребляемая мощность, В·А	не более 5

Продолжение таблицы 1.3

Периодичность записи данных в архив, с	1...60
Интервал времени хранения архива во встроенной памяти (при периодичности записи данных в архив 1 с), суток	не менее 30
Интерфейсы обмена данными	USB-Host, RS-485
Конструктивное исполнение	пластмассовый корпус для монтажа в щит
Степень защиты по ГОСТ 14254	IP30
Средний срок службы, лет	12
Масса, кг	не более 0,7

### 1.2 Выводы по разделу один

Рассмотренные выше регуляторы являются высокотехнологичными многофункциональными модулями способными выполнять контроль температуры в заданных в техническом задании параметров. ПИД - регуляторы температуры контролирующие температуру нагрева в вакуумной камере, значительно повышают безопасность и надежность работы с установкой и прохождения технологического процесса.

Однако для решения задачи данной в техническом задании эти системы подходят только по одному параметру, контролю температуры, а так как есть необходимость съема данных сопротивления с образца свидетель и передачи этих данных на мобильное приложение, то все данные устройства не подходят под заданные цели.

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

## 2 ОСНОВНОЙ МАТЕРИАЛ ПО СПЕЦИАЛЬНОЙ ЧАСТИ

### 2.1 Разработка функциональной схемы системы управления

#### 2.1.1 Функциональная схема системы управления

Функциональная схема системы управления представлена на рисунке 2.1.

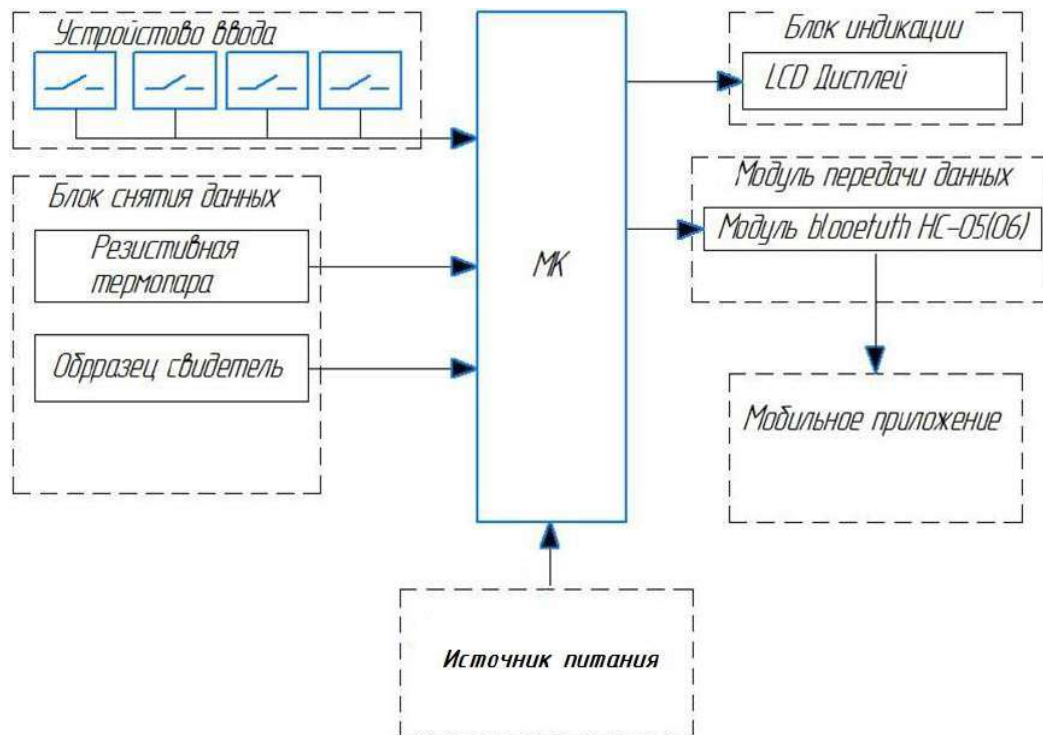


Рисунок 2.1 - Схема функциональная системы управления устройством

#### 2.1.2 Принцип работы функциональной схемы

1) Устройство ввода состоит 4х тактовых кнопок подключенных к входу МК и служит для задания температурного режима.

**Первая кнопка** отвечает за задание входа в настройку режима, при повторном нажатии происходит возвращение к показанию реальных данных снятых с устройства.

**Вторая кнопка** служит для того что бы задать выставленную температуру, после ее нажатия устанавливается режим на нужную температуру и возвращает нас на дисплей с фактическими параметрами.

**Третья кнопка** выполняет роль задания температуры с положительную сторону с шагом в 1 градус, при зажатии кнопки происходит смена температуры с шагом в 10 градусов.

**Четвертая кнопка** так же выполняет роль задания температуры только в отрицательную сторону с шагом в 1 градус, при зажатии кнопки происходит смена температуры с шагом в 10 градусов.

2) Блок снятия температуры состоит из двух датчиков.

**1-й датчик** – датчик температуры, основанный на РТС термисторе (с положительным температурным коэффициентом). Сопротивление РТС термистора с ростом температуры увеличивается.

**2-й датчик** – образец свидетель, представляющий собой маленькую пластинку из поликоровой подложки размерами 48 мм на 79 мм толщиной 1 или 2 мм. По росту толщины напыления тантала его поверхностное сопротивление растет сопротивление. Таким образом, контролируется толщина нанесенного активного покрытия.

3) Блок индикации представляет собой монохромный графический дисплей. На котором в 2 строки выводится информация о текущей температуре и сопротивление.

4) Блок передачи данных служит для передачи данных с МК на мобильное приложение посредством блютуз модуля.

5) Мобильное приложение служит для отображения полученных данных с МК, что позволяет отслеживать температуру в вакуумной камере и контролировать процесс нанесения покрытия из тантала по средством контроля поверхностного сопротивления не подходя к установке.

6) Блок питания обеспечивает питанием 5В, 3,3В функциональные элементы схемы.

					27.03.04.2021.132.00 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

## 2.2 Выбор элементной базы

### 2.2.1 Выбор микроконтроллера

Микроконтроллер это микросхема, предназначенная для управления периферийными устройствами, элементами.

В настоящее время предложение от производителей микроконтроллеров представлено широкой номенклатурой изделий предназначенных для решения различных задач.

В выборе контроллера было отдано предпочтение STM32, из-за более дешевых отладочных плат на рынке. Так же малопотребляющие контроллеры семейств STM32 компании STMicroelectronics зарекомендовали себя с наилучшей стороны при создании высокоэффективных устройств с малым потреблением. STM32F имеют рекордно низкие цены и, сохраняя преимущества базовых семейств, позволяют значительно сократить стоимость конечного изделия.[16]

Основные характеристики STM32f103c8T6 приведены в таблице 2.1

Таблица 2.1 – Основные характеристики STM32f103c8T6

Ядро	ARM Cortex-M3
Тактовая частота:	72 МГц
Размер программной памяти (Flash) (Программная память):	64 кБ
Память ОЗУ (SRAM) (RAM):	20 кБ
Кол-во 16-бит таймеров (IC/OC/PWM) (16-бит таймер)	4

Продолжение таблицы 2.1

Дополнительные функциональные возможности таймера (Дополнительные функции таймера):	2 x WDG, RTC, 24-bit down counter
Разрешение и количество каналов аналого-цифрового преобразователя (АЦП) (АЦП)	10x12-bit
Количество линий ввода-вывода (I/O):	36
Типы последовательных интерфейсов и кол-во каналов (Последовательные интерфейсы):	2xSPI;2xI2C;3xUSART(IrDA, ISO 7816);USB;CAN
Напряжение питания (min) (Uпит (min)):	2 В
Напряжение питания (max) (Uпит (max)):	3.6 В
Ток потребления в режиме пониженного энергопотребления (Icc(h)):	1.7 мкА
Ток потребления в рабочем режиме (на МГц) (Icc(run)):	373 мкА
Диапазон рабочей температуры	-40 °C до 85 °C

Схема расположения выводов STM32F103C8T6 представлена на рисунке 2.2.[17]

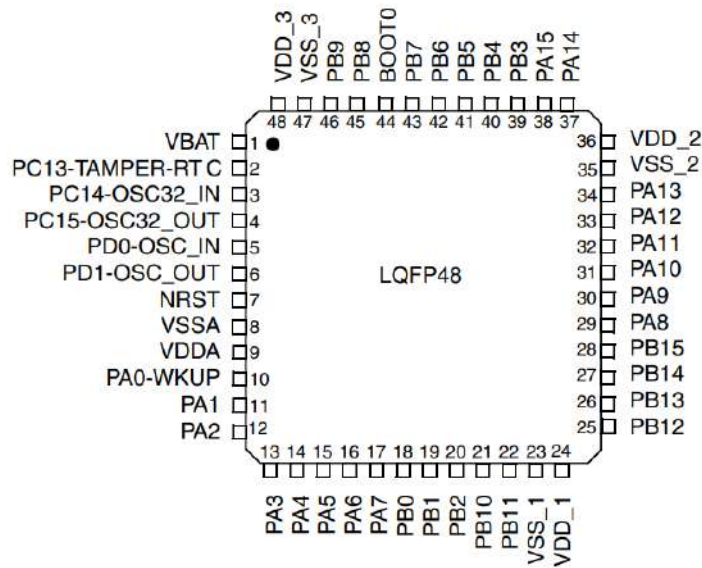


Рисунок 2.2 - Схема расположения выводов STM32F103C8T6

Для более наглядного и понятного проектирование схемы электрической на рисунке 2.3 изображена отладочная плата с более подробным описанием каждого вывода.[18]

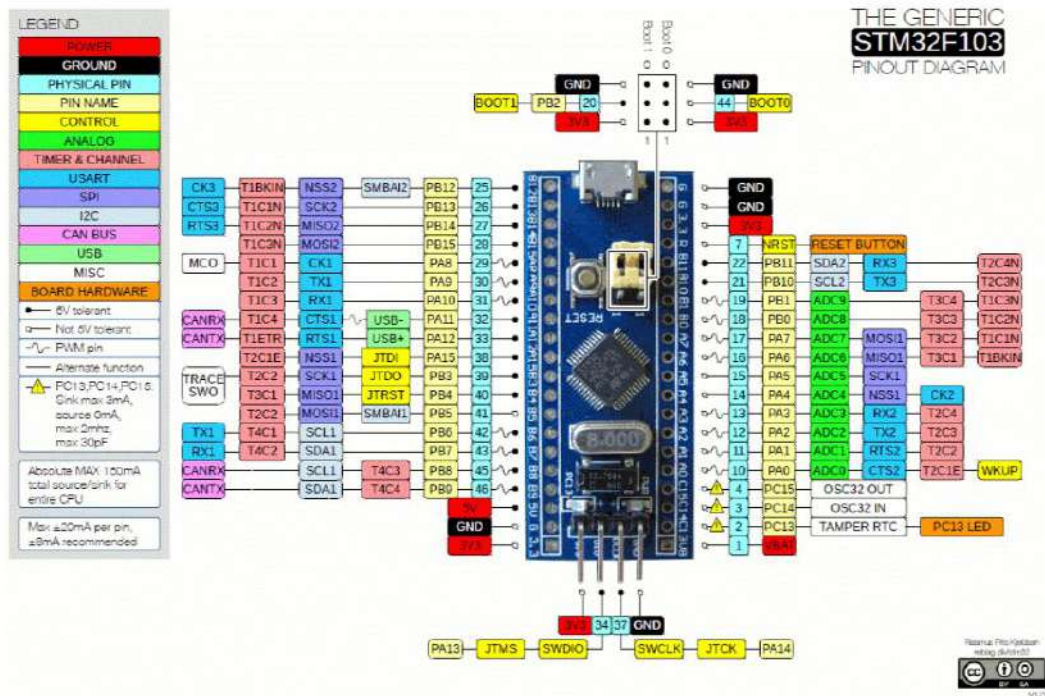


Рисунок 2.3 – Диаграмма выходов МК STM32F106C8T6

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



Общий вид отладочной платы предоставлен на рисунке 2.4

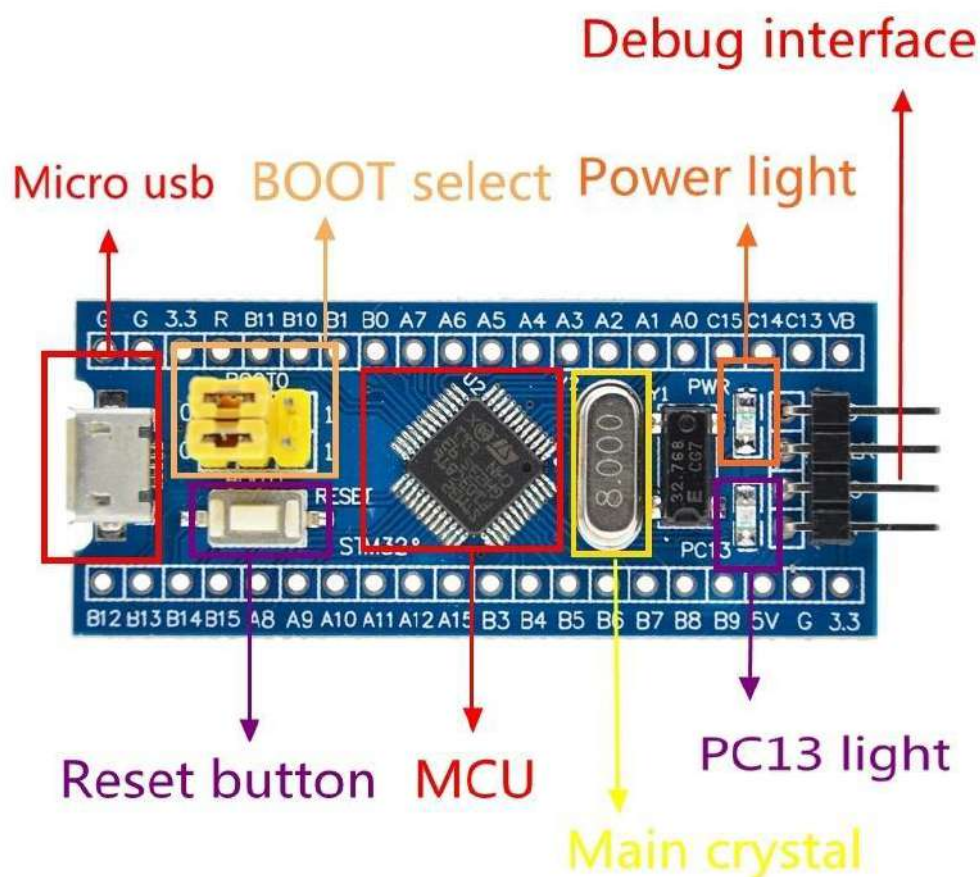


Рисунок 2.4 – Общий вид отладочной платы STM32F103C6T8

## 2.2.2 Выбор устройства ввода и вывода

### 2.2.2.1 Выбор устройства ввода

Устройство ввода предназначено для управления режимами устройства, началом и концом режима работы устройства, реализуется с помощью тактовых кнопок. Рассмотрим несколько видов тактовых кнопок и выберем подходящее.

### 2.2.2.1.1 Тактовый переключатель KLS7-TS1202-4.3-180

Вид тактовой кнопки представленных на рисунке 2.5[13]



Рисунок 2.5 – вид тактовой кнопки KLS7-TS1202-4.3-180

Основные характеристики KLS7-TS1202-4.3-180 предоставлены в таблице 2.2

Таблица 2.2 – Основные характеристики KLS7-TS1202-4.3-180 [13]

Производство	KLS
Тип	прямая
Способ монтажа:	в отверстия на плату
Рабочее напряжение, В	12
Рабочий ток, А	0.05
Высота, мм	4.3
Типоразмер	12x12

### 2.2.2.1.2 Кнопка тактовая с заземлением ТС-0119

Вид тактовой кнопки представленных на рисунке 2.6



Рисунок 2.6 – Кнопка тактовая с заземлением ТС-0119

Основные характеристики кнопки ТС-0119 приведены в таблице 2.3

Таблица 2.3 – Основные характеристики кнопки ТС-0119 [13]

Производство	Wealth Metal
Тип	прямая
Способ монтажа	в отверстия на плату
Рабочее напряжение, В	12
Длина исполнительного механизма, мм	9
Рабочий ток, А	0.05
Высота, мм	13
Типоразмер:	6x6

### 2.2.2.1.3 Тактовая кнопка В3F-1070

Вид тактовой кнопки В3F-1070 показан на рисунке 2.7



Рисунок 2.7 – тактовая кнопка В3F-1070

Технические характеристики В3F-1070 предоставлены в таблице 2.4

Таблица 2.4 – Технические характеристики В3F-1070 [13]

Тип исполнительного механизма	Плунжер
Производитель	Omron
Номинальный ток контактов	50 мА при 24 В пост. тока
Длина исполнительного механизма	6.1мм
Размеры	6 x 6 x 9.5мм
Конфигурация контактов	SPST-NO
Воздействующая сила	0.98

Из всех описанных выше тактовых кнопок для данного устройства подходит кнопка с большим размером исполнительного механизма, такой кнопкой является ТС-0119. Выбор устройства вывода информации

### 2.2.3.1 Символьный ЖК-дисплей WH1602

Вид ЖК-дисплей WH1602 показан на рисунке 2.8

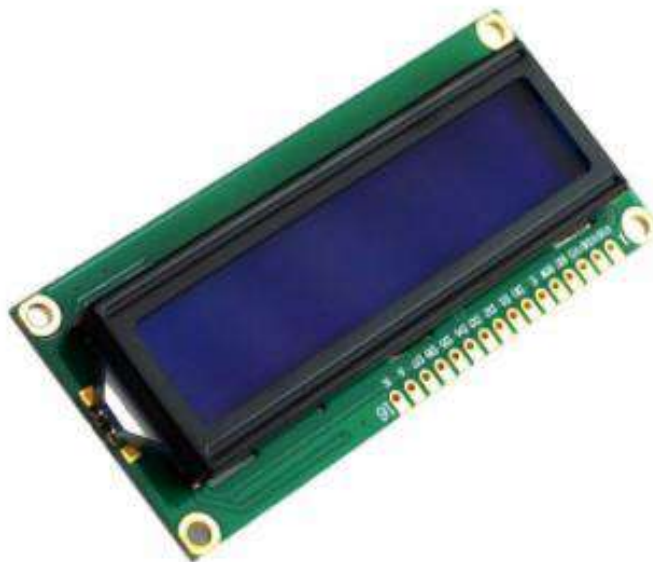


Рисунок 2.8 – общий вид дисплея WH1602

Основные характеристики дисплея WH1602 приведены в таблице 2.5

Таблица 2.5 – Основные характеристики дисплея WH1602

Размеры модуля, мм	80,0x36,0x13,5
Активная область, мм	56,20x11,5
Напряжение питания, В	5
Ток питания, мА	1,5

### 2.2.3.2 LCD дисплей Nokia 5110

Вид LCD дисплея Nokia 5110 показан на рисунке 2.9



Рисунок 2.9 – LCD дисплей Nokia 5110

Основные характеристики LCD дисплея Nokia 5110 представлены в таблице 2.6

Таблица 2.6 – Основные характеристики LCD дисплея Nokia 5110

Размеры модуля, мм	43,6 x 43,1
Разрешение экрана	48x48
Напряжение питания, В	3,3
Ток питания, мА	1,5

Для данного устройства мной был выбран LCD дисплей Nokia 5110. Рассмотрим подробнее LCD дисплей Nokia 5110 представляющий собой монохромный графический дисплей с разрешением 48×84 пикселей, который



Матрица дисплея условно поделена на столбцы, представляющие собой 8 пикселей, то есть когда мы отправляем 1 байт данных на дисплей, он отображается в этом столбце. По ширине (горизонтали) дисплея расположены 84 столбца, по высоте (вертикали) расположены 6 столбцов (то есть всего 6 строк). Адрес необходимого столбца (адресный указатель) задается командами X address, Y address. Ниже на картинке 2.10 представлена структура матрицы.[19]

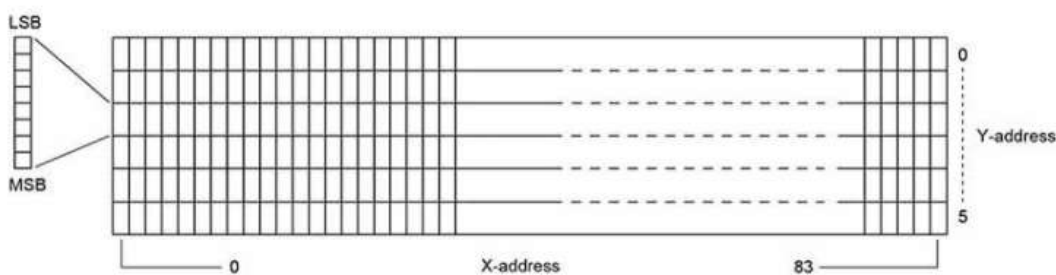


Рисунок 2.10 Структура матрицы контроллера PCD8544

На картинке 2.11 показана схема подключения дисплея Nokia 5110 к микроконтроллеру.

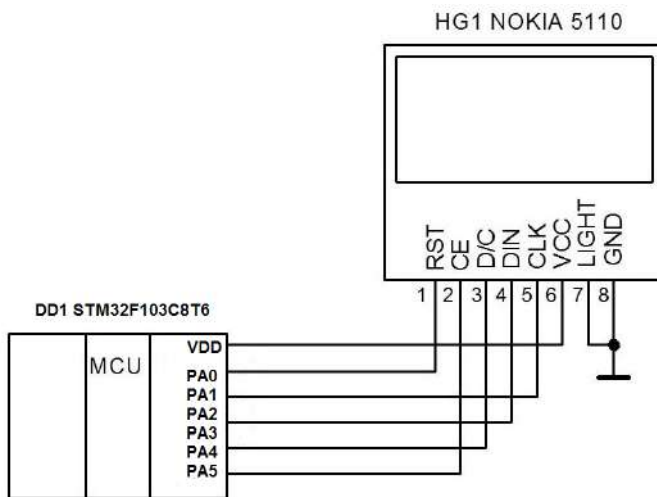


Рисунок 2.11 – Схема подключения дисплея к МК

#### 2.2.4 Выбор светодиодов

В качестве световых индикаторов будут применены имеющиеся в наличии светодиод АЛ307В зеленого спектра излучения и светодиод АЛ307Л красного спектра излучения

Характеристики светодиодов представлены в таблице 2.8.



Таблица 2.8 - Технические характеристики светодиодов [15]

Наименование	Спектр излучений	Длина волны	Сила света	Ток	Прямое падение напряжения
АЛ307В	зеленый	567 нм	0,4 мкд	20 мА	2,8 В
АЛ307Л	красный	655нм	6 мкд	10 мА	2 В

Светодиоды должны подключаться в цепи последовательно с резисторами для ограничения тока протекающего через диод. Рассчитаем величину номиналов токоограничительных резисторов согласно формуле:

$$R = \frac{(U_0 - U_1)}{I}, \quad (1)$$

где  $U_0$  – напряжение питания;

$U_1$  – прямое падение напряжения;

$I$  – ток светодиода.

$$АЛ307В = (5В-2,8В)/20мА=110 \text{ Ом.}$$

$$АЛ307Л = (5В-2В)/10мА=300 \text{ Ом.}$$

Выбираем резисторы С2-33Н-0,5-300 Ом  $\pm 2\%$  А-В-В ОЖО.476.093 ТУ (1 шт.) и С2-33Н-0,5-110 Ом  $\pm 2\%$  А-В-В ОЖО.476.093 ТУ (1 шт.)

### 2.2.5 Выбор модуля передачи данных на мобильное приложение

Для передачи данных с МК был выбран bluetooth модуль HC-05 он отличается от аналогов не дорогой стоимостью и обширностью информации о его подключение к МК. Он необходим для передачи данных на мобильное приложение под управление ОС Андроид. Общий вид bluetooth HC-05 предоставлен на рисунке 2.12.[10]

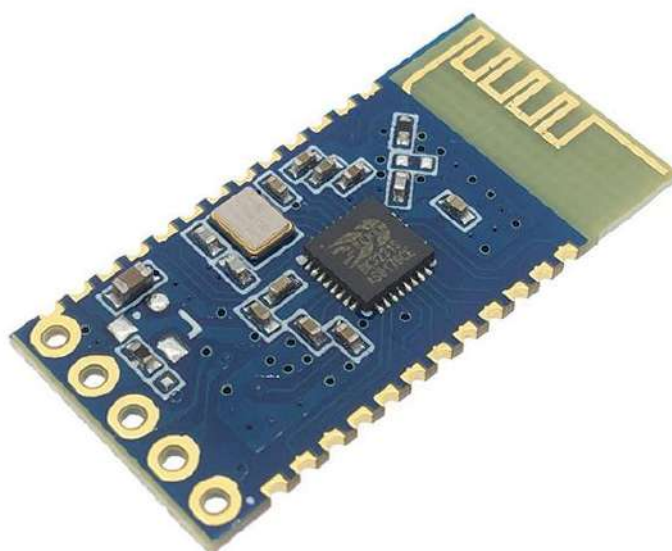


Рисунок 2.12 – bluetooth HC-05

Значение выводов HC-05 предоставлена в таблице 2.8

Таблица 2.9 – Значение выводов HC-05 [19]

Вывод	Значение
EN	включение/выключение модуля
VCC	питание +5В
GND	земля
TXD, RXD	UART интерфейс для общения с контроллером
STATE	индикатор состояния
KEY	нога для входа в режим AT-команд

Характеристики bluetooth модуля HC-05:

- 1) Активные частоты радиосвязи находятся в диапазоне 2.4-2.48 ГГц
- 2) Канал приспособлен к адаптивному переключению

3) Условная дальность связи – 10 метров, но в этой цифре не учитываются преграды и помехи

4) Максимальная скорость при обмене информацией – 115300 бод (1 бод = 0.8 бит/сек.)

5) Хранить чип можно в температурном диапазоне от - 40 до +85 градусов, а вот использовать – от -20 до +70

6) Для работы необходимо будет напряжение в 3.3 В, что мы рассмотрим чуть ниже.

### 2.2.6 Выбор элементов входящих в блок управления нагрузкой

Подключение нагрузки напрямую к микроконтроллеру невозможно, необходимо использовать дополнительный элемент с гальванической развязкой. Такой элемент под имеющийся источник питания нагрузки является и служащий гальванической развязкой между нагрузкой и МК является оптопара с симисторным выходом МОС3021. Технические характеристики приведены в таблице 2.10

Таблица 2.10

Тип выхода	Симистор
Детектор перехода через ноль	нет
Количество каналов	1
Напряжение изоляции (RMS), В	4170
Максимальное напряжение в закрытом состоянии, В	400
Отпирающий постоянный ток управления, мА	15
Ток удержания, мА	0.1
Рабочая температура, °С	от -40 до +80

Используя данную схему можно управлять включением и отключением нагрузки.

Принцип подключения нагрузки к МК через МС3021 показан на рисунке 2.13

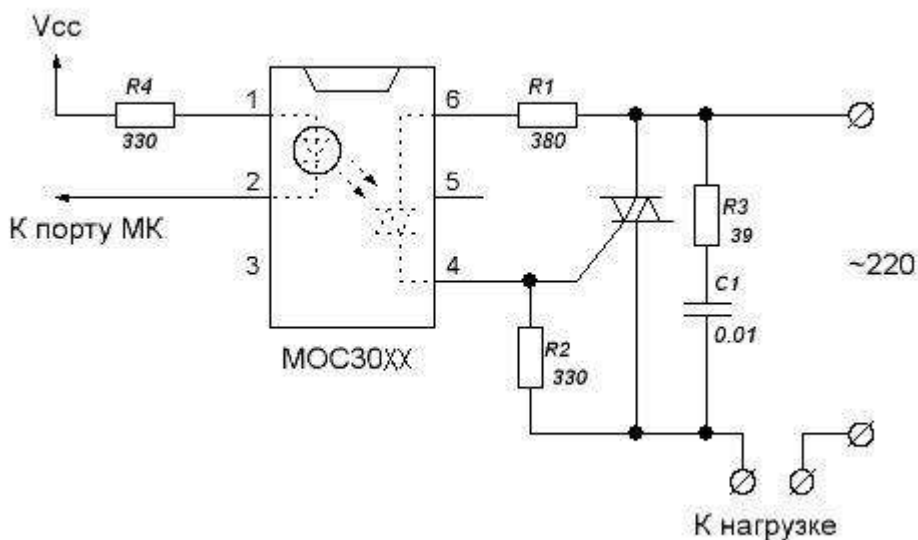


Рисунок 2.13 – Подключения нагрузки к МК через МОС3021

### 2.2.7 Расчет источника питания

Источник питания должен преобразовывать переменное напряжение 220 В в напряжение 5 В для питания элементов схемы контроллера температуры с образцом свидетеля.

- а) оценим максимальный ток потребляемый элементами схемы.

$$I_{\text{общ}} = I_1 + I_2 + \dots + I_n, \quad (2)$$

Максимальный ток потребляемый элементами, использующими питание 5 В:

$$I_{\text{общ}} = I_1 + I_2 + I_3 + I_4 + I_5 = 0,5 \text{ А}$$

где:  $I_1$  – ток потребляемый микроконтроллером;

$I_2$  – ток потребляемый экраном;

$I_3$  - ток потребляемый светодиодом;

$I_4$  - ток потребляемый МОС3021;

$I_5$  - ток потребляемый Bluetooth модулем HC-05

1) используя значения потребляемого тока и значения выходных напряжений подберем трансформатор.

Трансформатор выбираем из линейки унифицированных трансформаторов отечественного производства серии ТПП для питания устройств на полупроводниковых приборах, при условии:

- для напряжения питания 5 В:
- ток первичной обмотки, А:  $\geq 0,5$ ;
- напряжение на вторичной обмотке, В: 7,5.

Электрическая принципиальная схема трансформатора представлена на рисунке 2.14

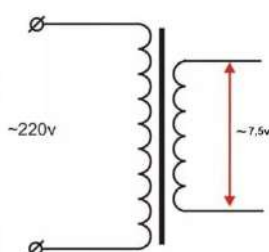


Рисунок 2.14 - Электрическая принципиальная схема трансформаторов ТПП на 50 Гц, 220 В на броневых сердечниках ШЛ и ШЛМ

Электрические характеристики трансформаторов подходящих для источника питания системы управления автомата для лужения изделий представлены в таблице 2.10.

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

Таблица 2.10 - Электрические параметры трансформаторов ТПП

Номер типонаминала	Мощность, В·А. Тип и размеры сердечника, мм	Ток первичной обмотки, А	Напряжение вторичных обмоток, В			Допустимый ток вторичных обмоток, А
			1-12, 13-14	5-16, 17-18	9-20, 21-22	
ТПП 253	22,0 ШЛМ 20×25	0,250/0,145	5,05	10,0	2,58	0,610
ТПП 259	31,0 ШЛМ 25×32	0,340/0,190	5,0	20,0	1,34	0,590
ТПП 266	57,0 ШЛМ 25×25	0,615/0,360	2,48	10,0	2,57	1,89
ТПП 269	57,0 ШЛМ 25×25	0,615/0,360	4,98	20,0	1,34	1,080

Выбираем трансформатор ТПП 253.

2) переменное напряжение необходимо выпрямить, используем мостовую схему для решения этой задачи.

По току нагрузки определим максимальный ток, протекающий через каждый диод выпрямительного моста

$$I_{\text{пр}} = 0,5 \times 1,9 \times I_{\text{н}}, \quad [16] \quad (3)$$

$$I_{\text{пр}} = 0,475 \text{ А.}$$

Подсчитаем обратное напряжение, которое будет приложено к каждому диоду выпрямителя:

$$U_{обр}=1,5 \times U_n, \quad (9)$$

$U_{обр}=7,5$  В для питания 5 В.

Выбираем диодные мосты у которых  $I_{пр} \geq 0,475$  А,  $U_{обр} \geq 7,5$  В.  
Выбираем диодный мост отечественного производства КЦ412В.

Технические характеристики КЦ412В:

$U_{обр}=200$  В;

$I_{пр}=1000$  мА;

$f_d=5$  кГц;

$U_{пр} < 1,2$ В при  $I_{пр}=500$  мА;

$I_{обр}= 50$  мкА

3) рассчитаем емкость конденсатора, используемого в качестве сглаживающего фильтра:

$$R_n = U_{вых} / I_{вых}, \quad (4)$$

$R_n=10$  Ом для питания 5 В.

$$C_{min} = 10^6 / f \times R_n, \quad (5)$$

$C_{min} = 2000$  мкФ для питания 5 В.

Выбираем конденсатор К50-35-16В-2200мкФ  $\pm 10\%$  ОЖО.464.156 ТУ

4) для обеспечения стабильной выходной величины напряжения 5 В установим после выпрямителя стабилизатор напряжения. Зная значения  $U_{вх}=7,63$  В,  $U_{вых}=5$  В,  $I_{вых} \leq 0,5$  А выбираем стабилизатор отечественного производства 142ЕН5А.

Технические характеристики 142ЕН5А:

$U_{вх}=7,5 - 15$  В;

					27.03.04.2021.132.00 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

$U_{\text{вых}}=4,9 - 5,1 \text{ В};$

$K_U= 0,05\% / \text{В};$

$K_I= 1\% / \text{А};$

диапазон рабочих температур от  $-45^{\circ}\text{C}$  до  $+70^{\circ}\text{C};$

тип корпуса КТ-28-2:

5) Разработаем электрическую принципиальную схему источника питания, используя выбранные элементы.

Электрическая принципиальная схема источника питания контроллера температуры с образцом свидетеля представлена на рисунке 2.15.

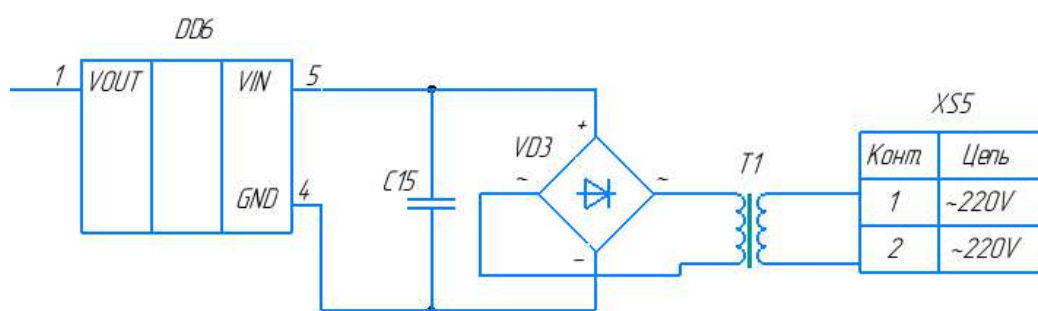


Рисунок 2.15 - Электрическая принципиальная схема источника питания контроллера температуры с образцом свидетеля



## 2.3 Построение схемы электрической принципиальной

### 2.3.1 Схема электрическая принципиальная

Схема электрическая принципиальная создана в программе COMPAS-3D с учетом всех условных значений графических элементов предоставлена на рисунке 2.16

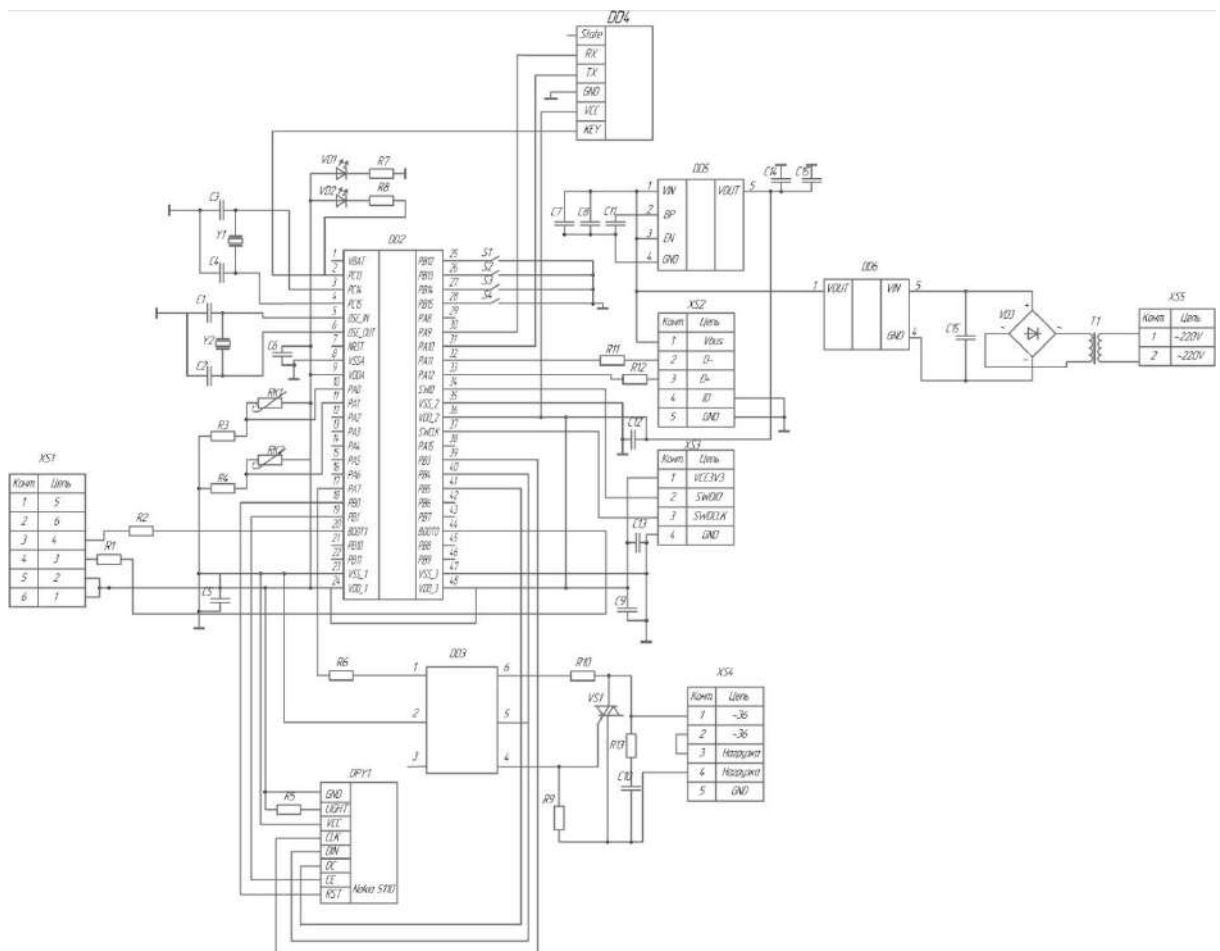


Рисунок 2.16 – Схема электрическая принципиальная контроллера температуры с образцом свидетеля и синхронизацией с мобильным приложением

### 2.3.2 Описание принципа работы схемы электрической

В схеме изображенной на рисунке 2.16 к портам микроконтроллера STM32F103C8T6 (DD1) к портам PB12, PB13, PB14, PB15 подключены четыре тактовые кнопки которые служат для управления устройством нагрева.

Схема стабилизации напряжения для МК реализована на микросхеме RT9193-33 (DD4), оно преобразует и стабилизирует входное напряжение 5 В в 3,3 В.

На схема также предусмотрены светодиоды для индикации работы платы, это светодиоды VD1 и VD2, так же на них стоят токоограничительные резисторы R7 и R8 на 510 Ом.

Контроль сопротивления и температуры вакуумной установки УВН71П-3 происходит с помощью терморезисторов выраженных на схеме элементами RK1 и RK2. Первые выводы подключен к питанию 3,3 В, вторые выводы подключенный к АЦП микроконтроллера исходит из средней точки делителя напряжения, третьи выводы от резисторов R3 и R4 идут на заземление.[6]

Так же схема МК имеет два встроенных кварцевых резонатора Y1 и Y2, они служат для задавания тактирования МК. Чем больше тактовая частота МК тем большее количество операций он может выполнить за 1 секунду.

На разъеме XS2 изображен разъем питания платы. Он представляет собой micro-usb разъем.

На служебном разъеме XS3 изображены выводы для программирования МК.

Для вывода полученных данных с терморезисторов предусмотрен LCD-Display Nokia 5110 (DPY1).

Так же на схеме отображен блютуз-модуль HC-05 (DD3) он необходим для синхронизации МК с мобильным приложением, написанным под операционную систему андроид.

Для подключения высокоточной нагрузки на схеме предусмотрена гальваническая развязка, основанная на MOC3021 (DD2).

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

## 2.4 Вывод по разделу два

В ходе работы по разделу два была выбрана элементная база разрабатываемого устройства, рассчитаны параметры источника питания всего устройства и составлена схема электрическая принципиальная с учетом выбранных элементов и разработан перечень элементов для данной схемы.

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

### 3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

#### 3.1 Выбор среды разработки

Для разработки программного обеспечения МК мной была использована официальная бесплатная программа STM32Cube IDE, от аналогов она отличается отличным визуальным интерфейсом и удобным подключением периферии, в отличие от аналогов сред разработки там не надо прописывать вручную активацию АЦП, таймеров и тому подобного.

На рисунке 3.1 наглядно показано как можно настраивать данный МК с помощью программы STM32Cube IDE.

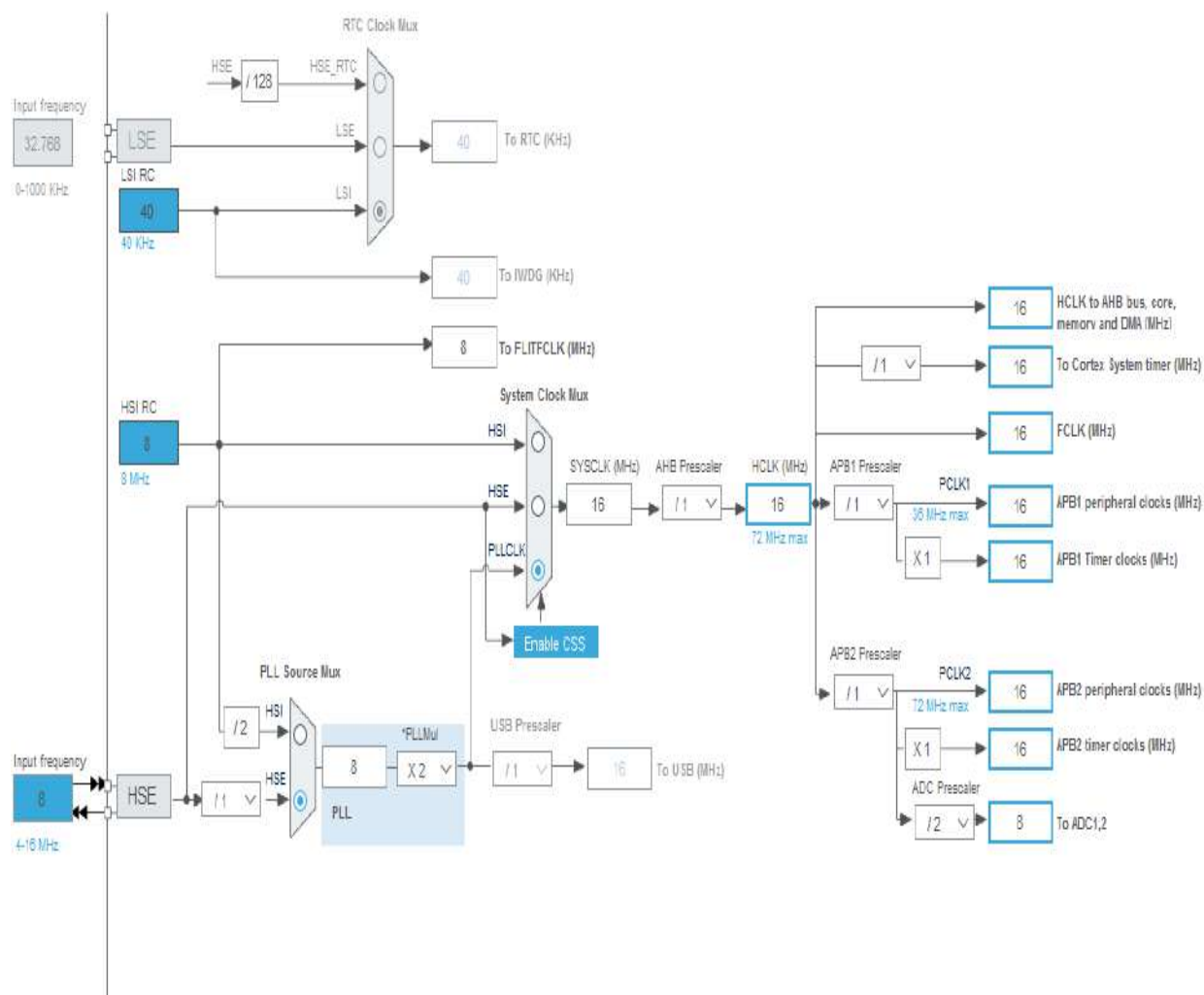


Рисунок 3.1 – Конфигурация МК

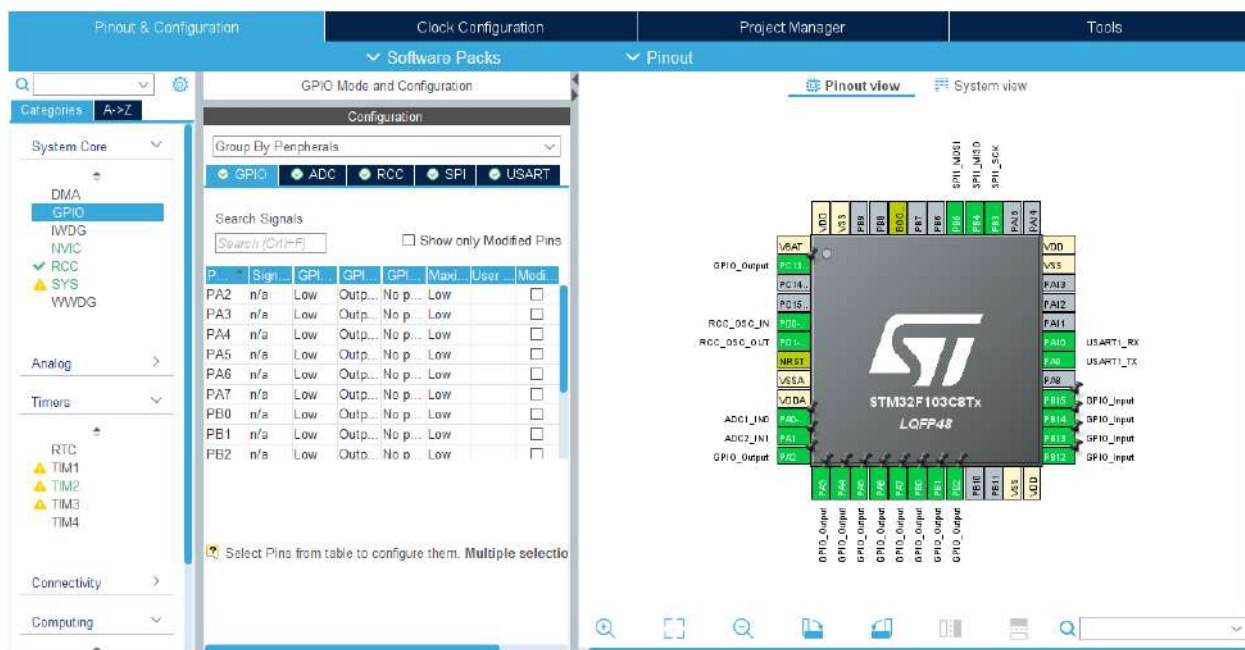


Рисунок 3.2 – Конфигурация выходов МК

### 3.2 Расчет ПИД- регулятора

Для регулирования контроллера температуры в первую очередь необходимо определить динамические характеристики объекта, которым мы будем управлять, путем определения передаточной функции.

Определим передаточную функцию экспериментальным методом: Экспериментальный метод для системы заключается в том, что на вход объекта управления подается управляющее воздействие со стороны схемы управления, и идет отслеживание данных. В моем случае объектом управления является нагревательный элемент, находящийся в вакуумной установке УВН-71П-3.

Для того чтобы определить значения передаточной функции с помощью цифрового вольтметра снимем показания с термопары, и построим зависимость температуры от времени. Считывание данных происходило с интервалом 30 секунд.

Для дальнейших расчетов из паспорта на нагреватель было определено, что он имеет мощность 500 Вт.

Экспериментальное возмущение возьмём  $x_{0вх} = 6\%$  от максимальной мощности  $P_n$  нагревателя.

$$P_{нм} = 500 \cdot 0,06 = 30 \text{ Вт} \quad (3)$$

Вычислим входное напряжение источника питания для 6% от мощности нагревателя

$$U_{6\%} = \sqrt{3 \cdot 30} = 9,5 \text{ В} \quad (4)$$

В результате проведенного эксперимента был получен график переходной характеристики объекта, который изображен на рисунке 3.3.

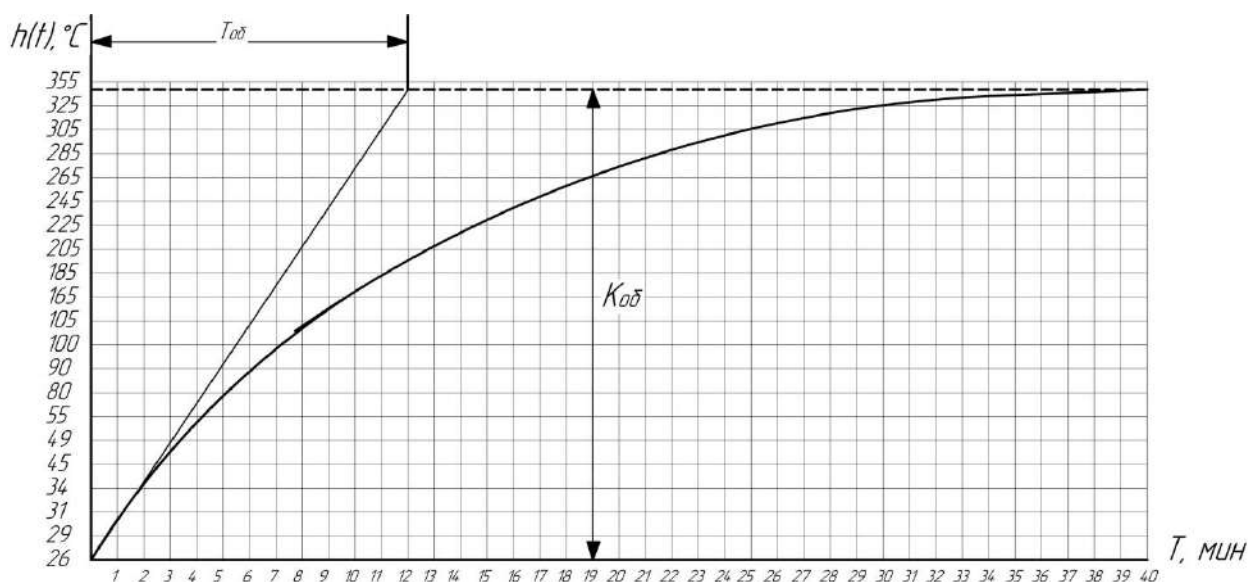


Рисунок 3.3 – График зависимости температуры от времени

Так как датчик близко расположен к нагревателю то временем запаздывания можно пренебречь, из-за достаточно малого времени переходного процесса.

Исходя из зависимости температуры от времени предоставленных на рисунке 3.3 можно вычислить данные необходимые для построение апериодического звена 1-го порядка, передаточная функция звена предоставлена в формуле 5.

$$W_{об}(p) = \frac{k_{об}}{T_{об}p + 1} \quad (5)$$

Начальная температура в вакуумной камере составляла

$$t_0 = 5 \text{ }^\circ\text{C}$$

Установившееся значение температуры после окончания процесса

$$t_{уст} = 353 \text{ }^\circ\text{C}$$

Время переходного процесса 30 минут

Значение постоянной времени  $T_{об}$  и коэффициента передачи  $k_{об}$  определены из графика зависимости температуры от времени, рисунок 3.3

$$k_{об} = 350 - 26 = 324 \text{ }^\circ\text{C}$$

$$T_{об} = 720 \text{ с}$$

Искомая передаточная функция:

$$W_{об}(p) = \frac{324}{720p + 1}$$

Для дальнейшего построение и настройка Пид-регулятора реализована в среде графического программирования Simulink, на основе MATLAB для моделирования и анализа динамических систем Simulink. Моделирование объекта описанного апериодическим звеном первого порядка приведено на рисунок 3.4.

					27.03.04.2021.132.00 ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		



Рисунок 3.4 – Схема подключения объекта управления

Подключение блока ПИД-регулятора предоставлено на рисунке 3.5

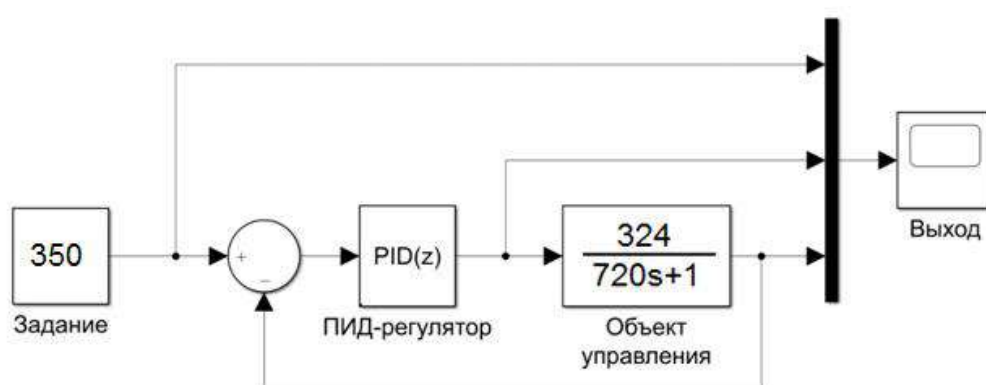


Рисунок 3.5 – Схема подключения ПИД-регулятора  
к объекту управления в Simulink

Далее переходим к настройке ПИД-регулятора с помощью программы Matlab и встроенного инструмента PID Tuner. Данный инструмент позволяет линеаризовать систему и автоматически вычислить коэффициенты для обеспечения качественного переходного процесса управления. Результаты работы показаны на рисунке 3.6



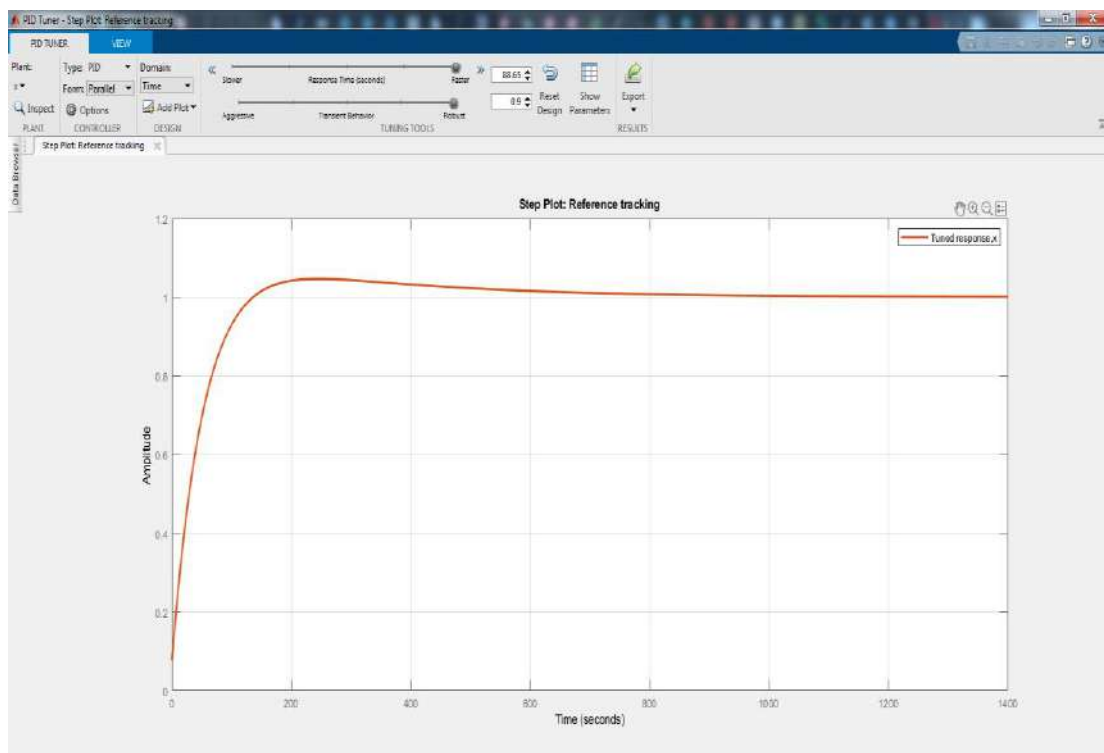


Рисунок 3.6 – График настройки ПИД-регулятора с помощью инструмента PID Tuner

Параметры настройки ПИД-регулятора полученные из PID Tuner :

Время переходного процесса 87,3 с;

Перерегулирование 4.7%, до 353 °С;

$$K_p = 43,32;$$

$$K_i = 23,13;$$

$$K_d = 20;$$

### 3.3 Разработка программного обеспечения

#### 3.3.1 Алгоритм работы МК

Принцип работы системы управления устройством лужения изделий заключается в следующем:

- 1) Инициализация настроек микроконтроллера.

- 2) Задаем при помощи тактовых кнопок режим работы нагревателей
- 3) Сканируем нажатые клавиши
- 4) Выводим на LCD дисплей параметры датчиков.
- 5) Проверка на предмет нажатия кнопки ПУСК.
- 6) Включаем МОС (запуск нагревателей)
- 7) Проверка на предмет нажатия кнопки СТОП
- 8) Отключаем МОС (отключение нагревателей)
- 9) Отправляем данные на мобильное приложение

Согласно принципу работы контроллера температуры с образцом свидетеля и синхронизацией с мобильным приложением составлен алгоритм программы микроконтроллера

### 3.3.2 Разработка программы МК и мобильного приложения

Согласно алгоритму, описанному выше напишем программу микроконтроллера. Программа написана на языке С, код программы для МК представлен в приложении В. Код программы для мобильного приложения представлен в приложении Г

### 3.4 Вывод по разделу три

1) проанализированы и выбраны способы реализации контроллера температуры с образцом свидетеля и синхронизацией с мобильным приложением в соответствии с техническим заданием;

2) разработан код программы микроконтроллера (Приложение В).

3) разработан код программы мобильного приложения (Приложение Г)

					27.03.04.2021.132.00 ПЗ	Лист
						43
Изм.	Лист	№ докум.	Подпись	Дата		

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы по выбранной теме были изучены аналоги схожих устройств, разработано техническое задание на контроллер температуры с образцом свидетелем и синхронизацией с мобильным приложением, разработана схема электрическая принципиальная и написана программа для МК и мобильного приложения под ОС андроид.

Использование контроллера температуры с образцом свидетеля и синхронизацией с мобильным приложением для вакуумной установки УВН71П-3 позволит снизить нахождение оператора за установкой, что позволит увеличить среднюю производительность труда.

Техническое задание ВКР работы выполнено в полном объеме.

В процессе выполнения ВКР мной были использованы знания полученные в ходе обучения в ЮУрГУ, по курсам:

- курс микропроцессоры, микроконтроллеры и вычислительная техника
- курс автоматических систем управления
- курс информатика и программирование
- курс программирование и основы алгоритмизации
- курс теоретические основы электротехники
- курс основы микроэлектроники
- курс переходные процессы

При выполнении ВКР были отработаны навыки работы с технической литературой, действующими стандартами единой системы конструкторской документации, получен навык работы со средами разработки, и работой в программах систем автоматического проектирования.

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		44

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

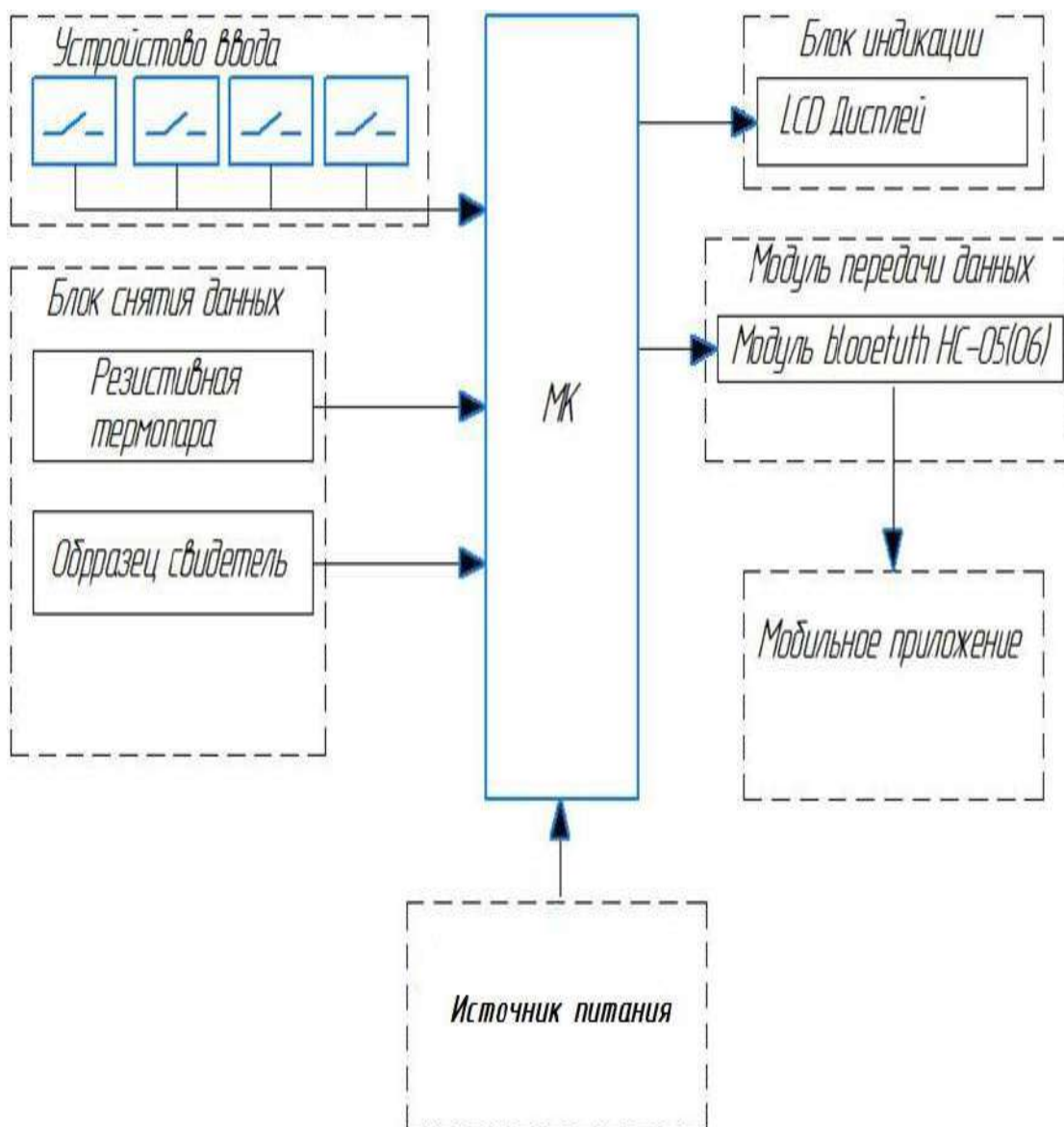
1. Клюев А.С., Лебедев А.Т., Клюев С.А., Товарнов А.Г. Наладка средств автоматизации и автоматических систем регулирования
2. Денисенко, В. ПИД – регуляторы: принципы построения и модификации. Ч. 1. Современные технологии автоматизации. – 2006. - №4. – с. 66-74.
3. Бессекерский В.А., Попов Е.П. Теория систем автоматического управления / В.А. Бессекерский, Е.П. Попов — Изд. 4-е, перераб. И доп. — Спб., Профессия, 2007. — 752с.
4. Датчики: Справочное пособие / В.М. Шарапов, Е.С. Полищук, Н.Д. Кошевой, Г.Г. Ишанин, И.Г. Минаев, А.С. Совлуков. — Москва: Техносфера, 2012. — 624 с.
5. Г. Виглеб. Датчики. Устройство и применение. Москва. Издательство «Мир», 1989
6. Современные датчики. Справочник. ДЖ. ФРАЙДЕН Перевод с английского Ю. А. Заболотной под редакцией Е. Л. Свинцова ТЕХНОСФЕРА Москва Техносфера-2005
7. ГОСТ 21962-76 Соединители электрические. Термины и определения. – М.: Издательство стандартов, 2002. - 12с.
8. Резисторы, конденсаторы, трансформаторы, дроссели, коммутационные устройства РЭА: справочник/Н.Н Акимов, Е.П. Ващуков, В.А. Прохоренко, Ю.П. Ходоренок – Мн.: Беларусь, 1994. – 591с.
9. <https://owen.ru/product/trm10> Технические характеристики контроллера температуры ОВЕН ТРМ 10.

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		45

10. <https://www.prostanki.com/board/item/254378> Технические характеристики регулятора температуры REX-C100FK02-M AN.
11. <https://robototehnika.ru/e-store/catalog/413/2093/> Технические характеристики блютуз модуля HC-05.
12. <https://eni-bbmrv.ru/index.php/ru/catalog/item/8-registratory/72-mnogokanalnyj-registrator-eni-702> Технические характеристики многоканального регистратора ЭНИ-702.
13. <http://www.chipdip.ru/> Технические характеристики тактических кнопок
14. <https://radioprogram.ru/shop/merch/48> Технические характеристики LCD Display Nokias 5110
15. [www.litagents.ru](http://www.litagents.ru) Технические характеристики светодиодов.
16. <https://kit-e.ru/bez-rubriki/nizkoe-energopotreblenie-nizkaya-cena-mikrokontrollery-stmicroelectronics-dlya-ustrojstv-s-batarejnym-pitaniem-2/> Характеристика отладочных плат семейства stm32f
17. <https://www.st.com/resource/en/datasheet/stm32f103tb.pdf> Описание отладочной платы stm32f103C8
18. <https://circuitdigest.com/microcontroller-projects/hc-05-bluetooth-module-interfacing-with-STM32F103C8-blue-pill> Методы подключения модуля блютуз HC-05 к плате stm32f103C8
19. <https://radiolaba.ru/microcotrollers/display-nokia-5110-podklyuchenie-k-mikrokontrolleru.html> Методы подключения LCD экрана Nokia 5110
20. <https://habr.com/ru/post/145991/> Расчет ПИД-регулятора

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		46

## ПРИЛОЖЕНИЕ А. ФУНКЦИОНАЛЬНАЯ СХЕМА УСТРОЙСТВА



Изм.	Лист	№ докум.	Подпись	Дата

27.03.04.2021.132.00 ПЗ

Лист

47

ПРИЛОЖЕНИ Б. СХЕМА ЭЛЕКТРИЧЕСКАЯ ПРИНЦИПИАЛЬНАЯ

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		48

ПРИЛОЖЕНИЕ В. ЛИСТИНГ ПРОГРАММЫ МК

```
****                               Main.c
*****
/*
Last committed:  $Revision: 00 $
Last changed by:  $Author: $
Last changed date: $Date: $
ID:              $Id: $
*****
#include "stm32f0xx_conf.h"

#define HSE_VALUE ((uint32_t)8000000)

#include "pcd8544.h"
#include "eeprom.h"
#include "stdio.h"
#include "string.h"
#include "k_therm.h"
#include "pid.h"

typedef struct
{
    void (*exec)();
    void (*on_key)(int key_count, int rotation);
} activity_t;

typedef enum {TARGET_ACTIVITY, TARGET_UINT, TARGET_PROC}
e_target;
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49



```
typedef struct
{
    char *caption;
    e_target target;
    void *pTarget;
    uint32_t divider;
} menuitem_t;

// Переменные и структуры
volatile unsigned int delay_val;    // счетчик для delay_ms()
volatile unsigned int counter_frame; // счетчик интервалов (такт 100 мкс)

__IO uint16_t VAR_ADC[4];
uint32_t VAR_ADC_int[4]= {0,0,0,0};
uint16_t VAR_ADC_mid[4]= {0,0,0,0};

uint32_t adc_count=1;

/* Счетчик времени удержания кнопки */
int button_hold_count;

unsigned char temp_str[16]; //Строка для printf
unsigned char pid_str[16]; ///Строка для параметров P,I,D

void main_onkey(int key_count, int rotation);
void main_exec();
void graph_onkey(int key_count, int rotation);
```

```
void graph_exec();
void reset_pid();
void menu_onkey(menuitem_t *menu,int key_count, int rotation);
void menu_exec(menuitem_t *menu);
void pid_menu_onkey(int key_count, int rotation);
void pid_menu_exec();
void calib_menu_exec(menuitem_t *menu);
void calib_menu_onkey(int key_count, int rotation);
void calib_save();

const activity_t main_activity= {&main_exec,&main_onkey};
const activity_t graph_activity= {&graph_exec,&graph_onkey};
const activity_t pid_menu_activity= {&pid_menu_exec,&pid_menu_onkey};
const activity_t calib_menu_activity= {&calib_menu_exec,&calib_menu_onkey};
const activity_t *current_activity=&main_activity;
/* PID */
pidData_t pid;
int32_t pid_data;
uint32_t P,I,D;
uint32_t pid_enabled;

//uint32_t Uint=1227000;
volatile int adc5mv=1736;
volatile int adc10mv=3164;
volatile int adc_base_temp=1729;
volatile int base_temp=277;
volatile int32_t adc_start_temp;
```

```
volatile int32_t avg_slope=4468;
const menuitem_t pid_menu[]=
{
    ">ДАЛЕЕ<",TARGET_ACTIVITY, &calib_menu_activity,0,
    "P=",TARGET_UINT, &P,1000,
    "I=",TARGET_UINT, &I,1000,
    "D=",TARGET_UINT, &D,1000,
    "СБРОС",TARGET_PROC, &reset_pid,0,
    0,0,0,0
};

const menuitem_t cali_menu[]=
{
    ">ДАЛЕЕ<",TARGET_ACTIVITY, &main_activity,0,
    "OKP=",TARGET_UINT, &base_temp,10,
    "SLOPE=",TARGET_UINT, &avg_slope,1000,
    "ADC5=",TARGET_UINT, &adc5mv,1,
    "ADC10=",TARGET_UINT, &adc10mv,1,
    "СОХРАНИТЬ",TARGET_PROC, &calib_save,0,
    0,0,0,0
};

uint32_t menu_pos=0;
uint32_t menu_selected=0;

char pid_divider=0;
int target_temp=50;
```

```
int32_t amb_tempA,amb_temp,heat_temp;
int32_t mVolt;
/* History */
#define HIST_DELAY 10
signed char temp_history[512];
short hist_x=sizeof(temp_history)-86;
int hist_count=0;

// прерывание SysTick - отсчет 100 мкс интервалов
void SysTick_Handler(void)
{
    if (delay_val>0) delay_val--;
    counter_frame++;
}

// реализация задержки на delay_value миллисекунд
void delay_ms(unsigned int delay_value)
{
    delay_val=delay_value*10;
    while (delay_val!=0)__WFI();
}

// реализация задержки, выравнивание по фрейму
void delay_frame_ms(unsigned int delay_value)
{
    delay_val=delay_value*10;
    while (counter_frame % delay_val)__WFI();
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		53

```

}
// реализация задержки на delay_value в * 10 мкс
void delay_10us(unsigned int delay_value)
{
    delay_val=delay_value;
    while (delay_val!=0);
}

static void DMA_Config(void)
{
    DMA_InitTypeDef DMA_InitStructure;
    /* DMA1 clock enable */
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1 , ENABLE);

    /* DMA1 Channel1 Config */
    DMA_DeInit(DMA1_Channel1);
    DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)&ADC1->DR;
    DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)VAR_ADC;
    DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
    DMA_InitStructure.DMA_BufferSize = 3;
    DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
    DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
    DMA_InitStructure.DMA_PeripheralDataSize =
DMA_PeripheralDataSize_HalfWord;
    DMA_InitStructure.DMA_MemoryDataSize =
DMA_MemoryDataSize_HalfWord;
    DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;

```

```
DMA_InitStructure.DMA_Priority = DMA_Priority_High;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel1, &DMA_InitStructure);

DMA_ITConfig(DMA1_Channel1, DMA_IT_TC, ENABLE);

NVIC_InitTypeDef NVIC_InitStructure;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_InitStructure.NVIC_IRQChannel = DMA1_Channel1_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPriority = 3;
NVIC_Init(&NVIC_InitStructure);

/* DMA1 Channel1 enable */
DMA_Cmd(DMA1_Channel1, ENABLE);
}

void DMA1_Channel1_IRQHandler(void) /* Read ADC values */
{
    if (DMA_GetITStatus(DMA1_IT_TC1))
    {
        for(int i=0; i<3; i++)
        {
            VAR_ADC_int[i]+=VAR_ADC[i];
            if(adc_count>=4096) VAR_ADC_mid[i]=VAR_ADC_int[i]/adc_count;
        }
        if(adc_count>=4096)
        {
```

```
adc_count=1;
    VAR_ADC_int[0]=0;
    VAR_ADC_int[1]=0;
    VAR_ADC_int[2]=0;
    VAR_ADC_int[3]=0;
}
else
    adc_count++;
}
DMA1->IFCR = DMA1_IT_GL1;
}

static void ADC_Config(void)
{
    ADC_InitTypeDef  ADC_InitStructure;
    GPIO_InitTypeDef GPIO_InitStructure;
    /* ADC1 DeInit */
    ADC_DeInit(ADC1);

    /* GPIOC Periph clock enable */
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOA, ENABLE);

    /* ADC1 Periph clock enable */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);

    /* Configure ADC channel1 as analog input */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 ;
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		56

```
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AN;  
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL ;  
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 ;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;  
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP ;  
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

```
/* Initialize ADC structure */
```

```
ADC_StructInit(&ADC_InitStructure);
```

```
/* Configure the ADC1 in continuous mode with a resolution equal to 12 bits */
```

```
ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;
```

```
ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
```

```
ADC_InitStructure.ADC_ExternalTrigConvEdge =
```

```
ADC_ExternalTrigConvEdge_None;
```

```
ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
```

```
ADC_InitStructure.ADC_ScanDirection = ADC_ScanDirection_Upward;
```

```
ADC_Init(ADC1, &ADC_InitStructure);
```

```
ADC_ClockModeConfig(ADC1, ADC_ClockMode_SynClkDiv4);
```

```
ADC_ChannelConfig(ADC1, ADC_Channel_1,  
ADC_SampleTime_239_5Cycles);
```



```
/* Convert the ADC1 ADC_Channel_9 (Temp diodes) with 239_5 Cycles as  
sampling time */
```

```
    //ADC_ChannelConfig(ADC1, ADC_Channel_9 ,  
ADC_SampleTime_239_5Cycles);
```

```
    ADC_ChannelConfig(ADC1, ADC_Channel_TempSensor ,  
ADC_SampleTime_239_5Cycles);
```

```
    ADC_TempSensorCmd(ENABLE);
```

```
/* Convert the ADC1 Vref with 239_5 Cycles as sampling time */
```

```
    ADC_ChannelConfig(ADC1, ADC_Channel_Vrefint ,  
ADC_SampleTime_239_5Cycles);
```

```
    ADC_VrefintCmd(ENABLE);
```

```
/* ADC Calibration */
```

```
ADC_GetCalibrationFactor(ADC1);
```

```
/* ADC DMA request in circular mode */
```

```
ADC_DMAResultModeConfig(ADC1, ADC_DMAMode_Circular);
```

```
/* Enable ADC_DMA */
```

```
ADC_DMACmd(ADC1, ENABLE);
```

```
/* Enable the ADC peripheral */
```

```
ADC_Cmd(ADC1, ENABLE);
```

```
/* Wait the ADRDY flag */
```

```
while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_ADRDY));
```

```

/* ADC1 regular Software Start Conv */
ADC_StartOfConversion(ADC1);
}

/* Temperature sensor calibration value address */
//Temperature sensor raw value at 30 degrees C, VDDA=3.3V
#define TEMP30_CAL_ADDR ((uint16_t*) ((uint32_t) 0x1FFFF7B8))
#define VDD_CALIB ((int32_t) (3300))
#define VDD_APPLI ((int32_t) (3296))
#define AVG_SLOPE ((int32_t) (4468)) //AVG_SLOPE in ADC conversion step
//( @3.3V)/°C multiplied by 1000 for precision on the division

int32_t ComputeTemperature(int32_t measure)
{
    int32_t temperature;

    temperature=(adc_base_temp - measure) * 10000;
    temperature = (temperature / avg_slope) + base_temp;
    return(temperature);
}

void Encoder_Init_TIM3()
{
    GPIO_InitTypeDef  GPIO_InitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;

    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOA, ENABLE);

```

```
/* For encoder */
```

```
GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_2MHz;
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF;
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7;
GPIO_Init(GPIOA, &GPIO_InitStruct);
GPIO_PinAFConfig(GPIOA, GPIO_PinSource6, GPIO_AF_1);
GPIO_PinAFConfig(GPIOA, GPIO_PinSource7, GPIO_AF_1);
```

```
/* For button */
```

```
GPIO_InitStruct.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_InitStruct.GPIO_Speed = GPIO_Speed_2MHz;
GPIO_InitStruct.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStruct.GPIO_Pin = GPIO_Pin_10;
GPIO_Init(GPIOA, &GPIO_InitStruct);
```

```
RCC_APB1PeriphClockCmd(RCC_APB1ENR_TIM3EN,ENABLE);
```

```
TIM_TimeBaseStructInit(&TIM_TimeBaseStructure);
//TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
TIM_EncoderInterfaceConfig(TIM3, TIM_EncoderMode_TI12,
TIM_ICPolarity_Falling, TIM_ICPolarity_Rising);
TIM3->CCMR1 |=TIM_CCMR1_IC1F | TIM_CCMR1_IC2F;
TIM_Cmd(TIM3, ENABLE);
}
```

									Лист
									60
Изм.	Лист	№ докум.	Подпись	Дата	27.03.04.2021.132.00 ПЗ				

```
/*
signed char enc_a[86];
signed char enc_b[86];
void encoder_test()
{
    lcd8544_clear();

    for(int i=0; i<85; i++)
    {
        lcd8544_line(i,enc_a[i],i+1,enc_a[i+1],1);
        lcd8544_line(i,enc_b[i],i+1,enc_b[i+1],1);
    }
    for(int i=0; i<85; i++)
    {
        enc_a[i]=enc_a[i+1];
        enc_b[i]=enc_b[i+1];
    }
    enc_a[85]=(GPIOA->IDR & (1 << 6)) ? 10 : 20;
    enc_b[85]=(GPIOA->IDR & (1 << 7)) ? 30 : 40;

    lcd8544_refresh();
}
*/
void reset_pid()
{
    P=2000;
    I=2;
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

```
D=1000;
}
void calib_save()
{
    eeprom_write('aslp',avg_slope);
    if(base_temp != eeprom_read('ambt'))
    {
        eeprom_write('ambt',base_temp);
        adc_base_temp = adc_start_temp;
        eeprom_write('amad',adc_base_temp);
    }

    eeprom_write('adc5',adc5mv);
    eeprom_write('ad10',adc10mv);
    current_activity=&main_activity;
}

void calib_menu_exec(menuitem_t *menu)
{
    if(menu_selected && (VAR_ADC_mid[0] < 4040))
    {
        if(menu_pos==3) adc5mv=VAR_ADC_mid[0];
        if(menu_pos==4) adc10mv=VAR_ADC_mid[0];
    }

    menu_exec((void*)&cali_menu);
}
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		62

```
pid_enabled=0;
```

```
}
```

```
void calib_menu_onkey(int key_count, int rotation)
```

```
{
```

```
    menu_onkey(&cali_menu,key_count,rotation);
```

```
}
```

```
void pid_menu_exec(menuitem_t *menu)
```

```
{
```

```
    menu_exec(&pid_menu);
```

```
    pid_enabled=0;
```

```
    pid_Init(P,I,D,&pid);
```

```
    pid_Reset_Integrator(&pid);
```

```
}
```

```
void pid_menu_onkey(int key_count, int rotation)
```

```
{
```

```
    menu_onkey(&pid_menu,key_count,rotation);
```

```
}
```

```
void menu_exec(menuitem_t *menu)
```

```
{
```

```
    lcd8544_clear();
```

```
    lcd8544_rect(0,0,83,47,1);
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

```
for(int i=0; menu[i].caption!=0; i++)
{
    if(i==menu_pos && !menu_selected)
        lcd8544_putstr(3,2+i*7,menu[i].caption,1);
    else
        lcd8544_putstr(3,2+i*7,menu[i].caption,0);

    if(menu[i].target==TARGET_UINT)
    {
        uint32_t d=*(uint32_t*)menu[i].pTarget;

        if(menu[i].divider==1000)
            sprintf(temp_str,"%d.%03d",d/1000,(d % 1000)) ;
        if(menu[i].divider==100)
            sprintf(temp_str,"%d.%02d",d/100,(d % 100)) ;
        if(menu[i].divider==10)
            sprintf(temp_str,"%d.%d",d/10,(d % 10)) ;
        if(menu[i].divider==1)
            sprintf(temp_str,"%d",d) ;

        if(menu_selected && (menu_pos == i))
            lcd8544_putstr(46,2+i*7,temp_str,1);
        else
            lcd8544_putstr(46,2+i*7,temp_str,0);
    }
}
```

```
lcd8544_refresh();
}

void menu_onkey(menuitem_t *menu,int key_count, int rotation)
{
    if(!menu_selected)
    {
        if(rotation>0)menu_pos++;
        if(rotation<0 && (menu_pos))menu_pos--;
        if(!menu[menu_pos].caption)menu_pos--;
    }
    else if(menu[menu_pos].target==TARGET_UINT)
    {
        if(abs(rotation)<4)
            (*(uint32_t*)menu[menu_pos].pTarget)+=rotation;
        else
            (*(uint32_t*)menu[menu_pos].pTarget)+=rotation*10;

        if((*int32_t*)menu[menu_pos].pTarget)<0)
            (*(uint32_t*)menu[menu_pos].pTarget)=0;
    }

    if(key_count)
    {
        menu_selected=!menu_selected;
        if(menu[menu_pos].target==TARGET_ACTIVITY)
        {
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		65



```
    eeprom_write('P===',P);
        eeprom_write('I===',I);
        eeprom_write('D===',D);
        menu_selected=0;
        current_activity=(activity_t*)menu[menu_pos].pTarget;
    }

    if(menu[menu_pos].target==TARGET_PROC)
    {
        ((void(*)())menu[menu_pos].pTarget());
        menu_selected=0;
    }

}

void graph_exec()
{
    lcd8544_clear();
    lcd8544_rect(0,0,83,47,1);

    sprintf(temp_str,"%dC",target_temp);
    lcd8544_putstr(3,17,temp_str,0);

    for(int i=0; i<84; i++)
    {
```

```
lcd8544_line(i,-temp_history[i+hist_x]+24+target_temp,i+1,-
temp_history[i+hist_x+1]+24+target_temp,1);
    if(!((i+hist_x-2) % 30))
    {
        lcd8544_line(i,0,i,47,1);
        sprintf(temp_str,"%dS",sizeof(temp_history)-hist_x-i);
        lcd8544_putstr(i+2,40,temp_str,0);
    }

}

lcd8544_line(0,24,85,24,1);
if(pid_data>0)lcd8544_putstr(3,39,"*",0);
lcd8544_refresh();
pid_enabled=1;
}

void graph_onkey(int key_count, int rotation)
{
    hist_x+=rotation;
    if(hist_x>(sizeof(temp_history)-86))hist_x=sizeof(temp_history)-86;
    if(hist_x<0)hist_x=0;

    if(key_count)
    {
        menu_selected=0;
        menu_pos=0;
    }
}
```

```
current_activity=&main_activity;
}
}

void main_onkey(int key_count, int rotation)
{
    target_temp+=rotation;
    if(target_temp<30)target_temp=30;
    if(target_temp>200)target_temp=200;
    eeprom_write('temp',target_temp);
    eeprom_write('P===',P);
    eeprom_write('I===',I);
    eeprom_write('D===',D);

    if(key_count)
        if(key_count<15)
            current_activity=&graph_activity;
        else
            current_activity=&pid_menu_activity;
}

void main_exec()
{
    lcd8544_clear();
    lcd8544_rect(0,0,83,47,1);
    sprintf(temp_str,"ОКРУЖ: %d.%dC",amb_temp/10,amb_temp%10);
    lcd8544_putstr(3,3,temp_str,0);
}
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		68

```

sprintf(temp_str,"УСТАН: %dC",target_temp);
    lcd8544_putstr(3,13,temp_str,1);
    sprintf(temp_str,"НАГРЕВ: %dC",heat_temp/1000);
    lcd8544_putstr(3,23,temp_str,0);
    lcd8544_putstr(3,33,pid_str,0);

    if(pid_data>0)lcd8544_putstr(3,39,"*",0);
    lcd8544_refresh();
    pid_enabled=1;
    menu_pos=0;
    menu_selected=0;
}

void SetCPU48MHz()
{
    RCC->CFGR &= (uint32_t) (~RCC_CFGR_SW);
    while ((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_HSI);
    RCC->CR &= (uint32_t)(~RCC_CR_PLLON);
    while((RCC->CR & RCC_CR_PLLRDY) != 0);
    RCC->CFGR = (RCC->CFGR & (~RCC_CFGR_PLLMUL)) |
(RCC_CFGR_PLLMUL12);
    RCC->CR |= RCC_CR_PLLON; /* (7) */
    while((RCC->CR & RCC_CR_PLLRDY) == 0);
    RCC->CFGR |= (uint32_t) (RCC_CFGR_SW_PLL);
    while ((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_PLL);
}

```

```
int main(void)
{
    //SystemInit();
    SetCPU48MHz();

    SysTick_Config(48000000/10000); // запуск systick (100 мкс)

    eeprom_init();
    DMA_Config();
    ADC_Config();

    lcd8544_init(); // запуск модуля LCD
    lcd8544_clear();
    // memcpy lcd8544_buff,EniLogo,sizeof(EniLogo));
    lcd8544_refresh();
    delay_ms(1000);
    adc_start_temp = VAR_ADC_mid[1];

    reset_pid();

    if(eeprom_read('P===')!=-1)
    {
        P=eeprom_read('P===');
        I=eeprom_read('I===');
        D=eeprom_read('D===');
        avg_slope=eeprom_read('aslp');
        base_temp=eeprom_read('ambt');
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		70

```
adc_base_temp=eeprom_read('amad');
    adc5mv=eeprom_read('adc5');
    adc10mv=eeprom_read('ad10');
    target_temp= eeprom_read('temp');
}
Encoder_Init_TIM3();

pid_Init(P,I,D,&pid);
pid_Reset_Integrator(&pid);

memcpy(VAR_ADC_mid,VAR_ADC,sizeof(VAR_ADC_mid));

while(1)
{
    amb_tempA=(adc_base_temp - (int)VAR_ADC_mid[1]) * 10000;
    amb_temp = (amb_tempA / avg_slope) + base_temp;
    mVolt=((int)VAR_ADC_mid[0]-(int)adc5mv)*5000/(adc10mv-
adc5mv)+5000;

heat_temp=thermocoupleConvertWithCJCompensation(mVolt,amb_temp*100);

    /* EXEC CODE */
    current_activity->exec();

    if(pid_enabled)
    {
        /* PID CONTROLLER */
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		71

```
pid_divider++;
    if(pid_divider>=5)
    {
        pid_data=pid_Controller(target_temp*10,heat_temp/100,&pid,pid_str);
        if(pid_data>0)
            GPIOA->ODR=GPIOA->IDR & ~ 1;
        else
            GPIOA->ODR=GPIOA->IDR | 1;
        pid_divider=0;
    }
}
else
    GPIOA->ODR=GPIOA->IDR | 1;
/* History */
hist_count++;
if(hist_count>=HIST_DELAY)
{
    for(int i=0; i<(sizeof(temp_history)-1); i++)
        temp_history[i]=temp_history[i+1];
    temp_history[sizeof(temp_history)-1]=heat_temp/1000;
    hist_count=0;
}
delay_frame_ms(100);
if(!(GPIOA->IDR & (1 << 10)))
    button_hold_count++;
else if (button_hold_count)
{
```

```

if(button_hold_count>1)current_activity->on_key(button_hold_count,0);
button_hold_count=0;
}
if(TIM3->CNT)
{
if((signed short)(TIM3->CNT)/4)
current_activity->on_key(button_hold_count,(signed short)(TIM3-
>CNT)/4);
TIM3->CNT=((signed short)(TIM3->CNT)) % 4;
}
}
}

```



ПРИЛОЖЕНИЕ Г. ЛИСТИНГ ПРОГРАММЫ МОБИЛЬНОГО  
ПРИЛОЖЕНИЯ

```
privateclass ConnectThreadextends Thread{

private final BluetoothSocket mmSocket;

private final BluetoothDevice mmDevice;

public ConnectThread(BluetoothDevice device){

// используем вспомогательную переменную, которую в дальнейшем

BluetoothSocket tmp=null;

mmDevice= device;

// получаем BluetoothSocket чтобы соединиться с BluetoothDevice

try{

// MY_UUID это UUID, который используется и в сервере

tmp= device.createRfcommSocketToServiceRecord(MY_UUID);

} catch(IOException e){}

mmSocket= tmp;

}

public void run()

{
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		74

```
// Отменяем сканирование (это нужно, чтобы не тормозить процесс, можно  
убрать при желании)
```

```
mBluetoothAdapter.cancelDiscovery();
```

```
try{
```

```
// Соединяемся с устройством через сокет.
```

```
// Метод блокирует выполнение программы до
```

```
// установки соединения или возникновения ошибки
```

```
mmSocket.connect();
```

```
} catch(IOException connectException){
```

```
// Невозможно соединиться. Закрываем сокет и выходим.
```

```
try{
```

```
mmSocket.close();
```

```
} catch(IOException closeException){ }
```

```
return;
```

```
}
```

```
// управляем соединением (в отдельном потоке)
```

```
manageConnectedSocket(mmSocket);
```

```
}
```

```
/** отмена ожидания сокета */
```

```
public void cancel(){
```

```
try{
```

```
mmSocket.close();
```

```
} catch(IOException e){}
```

```
}
```

```
}
```

```
    // принимаем данные
```

```
try{
```

```
tmpIn= socket.getInputStream();
```

```
tmpOut= socket.getOutputStream();
```

```
} catch(IOException e){}
```

```
public void REC(){
```

```
byte[] buffer=new byte[32];// буферный массив
```

```
int bytes;// bytes returned from read()
```

```
    // Прослушиваем InputStream пока не произойдет исключение
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		76

```
while(true){  
  
try{  
  
// читаем из InputStream  
  
bytes= tmpIn.read(buffer);  
  
//прочитанные байты  
  
mHandler.obtainMessage(MESSAGE_READ,bytes,  buffer).parsing(«.».,buffer[15],  
buffer[31]);  
  
} catch(IOException e){  
  
break;  
  
}  
  
}  
  
// отображаем на экране  
  
Label1.ValueStr = Func1(buffer, 0, 15);  
  
Label2.ValueStr = Func2(buffer, 16, 31);  
  
}
```

					27.03.04.2021.132.00 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		77