

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение высшего  
образования  
«Санкт-Петербургский политехнический университет Петра Великого»  
Институт компьютерных наук и технологий  
**Кафедра «Системы и технологии управления»**

Работа допущена к защите

Заведующий кафедрой СТУ

\_\_\_\_\_ В.П. Шкодырев

«\_\_» \_\_\_\_\_ 2016 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**БАКАЛАВРА**

**Система управления водяными полами на базе Arduino.**

направление 27.03.04 – Управление в технических системах

профиль 27.03.04\_02 – Системы и технические средства автоматизации и  
управления

Выполнила

студентка гр. 43503/5

<

>

Е.Э. Сулима

Научный руководитель

<

>

И.А. Фадеев

Санкт-Петербург

2016

## РЕФЕРАТ

38 стр., 8 таблиц, 19 рисунков, 1 приложение

### ВОДЯНОЙ ТЕПЛЫЙ ПОЛ, СИСТЕМА УПРАВЛЕНИЯ, БЕСПРОВОДНАЯ СВЯЗЬ, ГРАФИЧЕСКИЙ ИНТЕРФЕЙС

Разработана система управления водяным теплым полом. Сформированы требования к системе управления и её основные задачи. Определены алгоритмы и принципы работы. Составлена программа, реализующая алгоритм управления водяным теплым полом в соответствии с поданными входными сигналами. Введена беспроводная связь Wi-Fi для обмена информации Arduino с мобильным устройством. Создан графический интерфейс для мобильного телефона с операционной системой Android.

## THE ABSTRACT

38 pages, 8 tables, 19 pictures, 1 application

### THE WATER FLOORS HEATING, CONTROL SYSTEM, WIRELESS CONNECTION, GRAPHIC INTERFACE

The system water floors heating control was developed. The requirements and tasks for control system were formed. Algorithms and operation principles was defined. The program was compiled, which realize the water floors heating control algorithm according with input signals. Wireless connection Wi-Fi for information exchange between Arduino and mobile device was inserted. Graphic interface for mobile phone based on Android operation system was created.

## Содержание

<b>СОДЕРЖАНИЕ .....</b>	<b>3</b>
<b>СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ.....</b>	<b>4</b>
<b>1. ВВЕДЕНИЕ.....</b>	<b>5</b>
<b>2. ОБЪЕКТ УПРАВЛЕНИЯ.....</b>	<b>6</b>
<b>3. ПОСТАНОВКА ЗАДАЧИ.....</b>	<b>10</b>
<b>4. ВЫБОР ПРИБОРОВ, ОБОРУДОВАНИЯ И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....</b>	<b>12</b>
<b>5. ПРОЕКТИРОВАНИЕ ПРОГРАММНОЙ ЧАСТИ.. ..</b>	<b>24</b>
<b>6. ЗАКЛЮЧЕНИЕ.....</b>	<b>30</b>
<b>7. СПИСОК ИСТОЧНИКОВ.....</b>	<b>31</b>
<b>ПРИЛОЖЕНИЕ 1. КОД ПРОГРАММЫ .....</b>	<b>32</b>

## **Список используемых сокращений.**

ВТП – водяной теплый пол.

СУ – система управления.

ДТП – датчик температуры пола.

ЦО – центральное отопление.

## 1. Введение.

В проекте рассмотрена реализация системы управления водяного пола на базе микропроцессорной системы.

Задачей проекта является создание системы управления (СУ). Объект управления – водяной теплый пол (ВТП). Водяной пол обеспечивает тепло в квартире, может выступать как основным источником отопления, так и дополнительным. Нагревательные элементы располагаются под напольным покрытием, по которым циркулирует теплоноситель – нагретая жидкость (вода или растворы этилен пропиленгликоля). Теплоноситель отдает тепло материалу, окружающему трубы контуров системы теплого пола. Далее тепло передается тепловому покрытию. Каждое чистовое покрытие имеет свое термическое сопротивление, зависящее от материала изготовления и его толщины. От нагретой поверхности пола тепло поднимается вверх, отапливая все помещение.

Самым большим плюсом качественного водяного пола можно признать возможность использовать его как основную систему отопления, то есть при определенных условиях вообще обойтись без радиаторов. Также основными достоинствами отопления на основе ВТП являются: комфорт, уют, современный дизайн интерьеров, надежность, экономичность, перспективность.

Для каждого человека температура комфорта является индивидуальной характеристикой. В процессе эксплуатации ВТП пользователь сам находит для себя наиболее комфортный диапазон температур.

## 2. Объект управления.

В качестве объекта управления выступает ВТП, который включает в себя: радиаторы, насос, теплообменник, датчики температуры, сервоприводы.

Для автоматизации системы используется плата Arduino, которая, в соответствии с заложенной программой, осуществляет управления всеми устройствами автоматики.

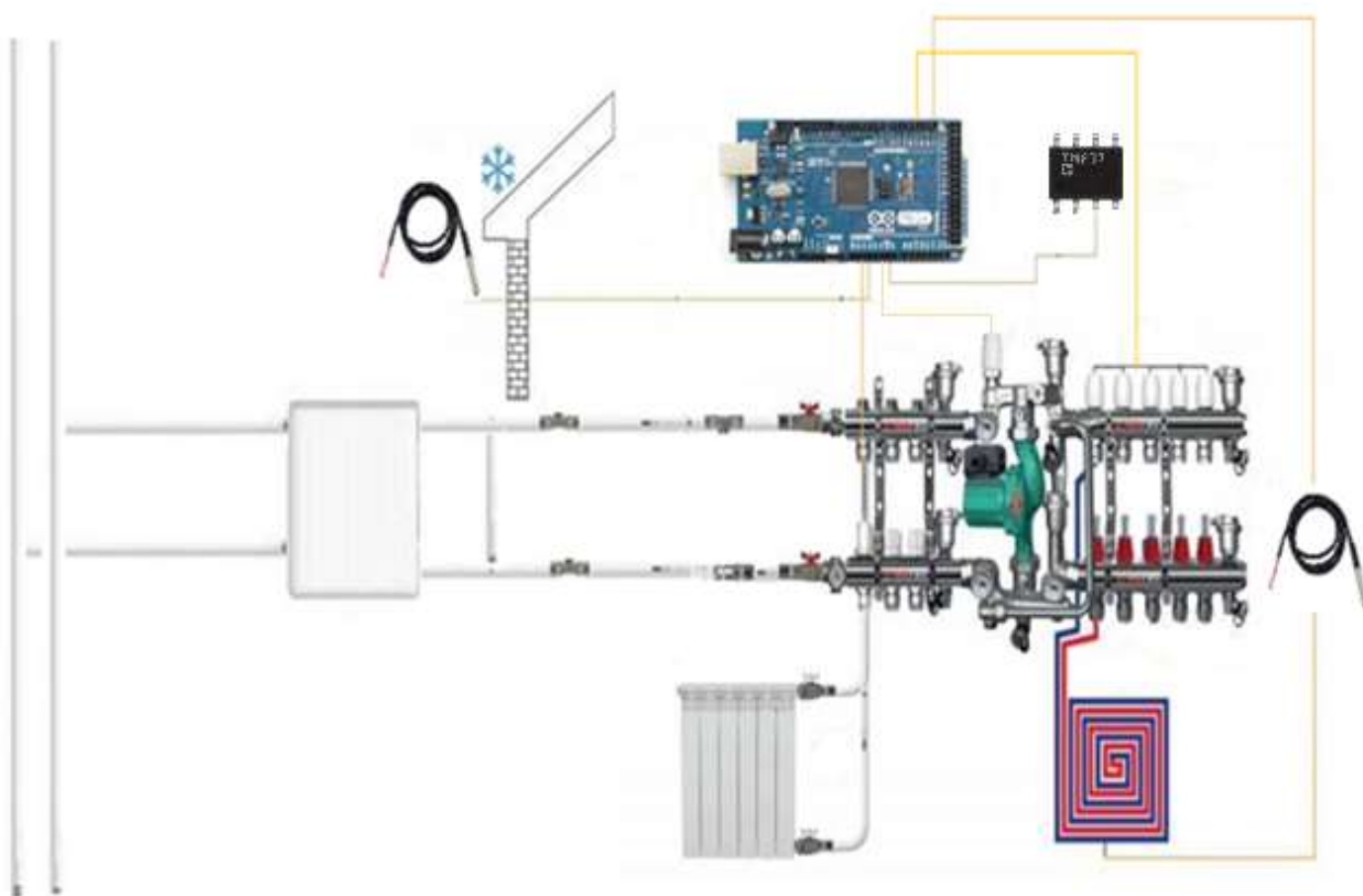


Рис. 2.1 Система автоматического регулирования ВТП.

### Описание схемы.

Распределение теплоносителя по контурам отопления осуществляется коллекторным блоком, в данном случае он состоит из двух коллекторов,

коллектора управления контуром радиаторов и коллектора управления контурами теплого пола. Между коллекторами установлен насос рециркуляции, обеспечивающий регулировку температуры в подающем контуре коллектора теплого пола за счет подмеса холодного теплоносителя с обратной линии контура теплого пола.

#### Контур отопления радиаторами.

В проекте ВТП является основным источником отопления, и для него существует строгое ограничение по температуре: температура пола не может превышать 28 С, иначе человек может обжечься. Поэтому, в случае нехватки отдачи тепла от пола, подключаем дополнительный источник отопления - радиатор.. Также, если поступает сигнал с датчика наружной температуры, то Arduino дает команду на включение дополнительного источника тепла, с помощью сигнала на электротермический сервопривод, который воздействует на термостатический клапан коллекторного блока (открывает/закрывает регулировочный клапан, обеспечивает регулирование расхода теплоносителя через систему отопления).

При достижении в комнате требуемой температуры датчик температуры помещения дает управляющий сигнал сервоприводу на закрытие клапана и тем самым происходит уменьшение подачи теплоносителя в контур отопления.

#### Контур теплого пола.

Контур теплого пола в данном случае имеют регулирование по температуре воздуха с помощью датчиков температуры, работа клапанов управляется контроллером управления.

#### Регулирование температуры теплоносителя.

Контроллер управления получает данные температуры от датчика комнатной температуры и датчика температуры пола. В соответствии с заложенной программой Arduino, обеспечивает необходимую температуру теплоносителя на входе в коллектор контура теплого пола, за счет регулировки расхода теплоносителя с помощью клапана (подает управляющий сигнал сервоприводу клапана на открытие/закрытие) и обеспечивает включение/отключение насоса рециркуляции.

Чтобы лучше разобраться с объектом управления необходимо составить функциональную схему, на которой будут показаны все связи устройств с Arduino. Также будут отображены входные и выходные сигналы.

### Функциональная схема. Автоматика ВТП.

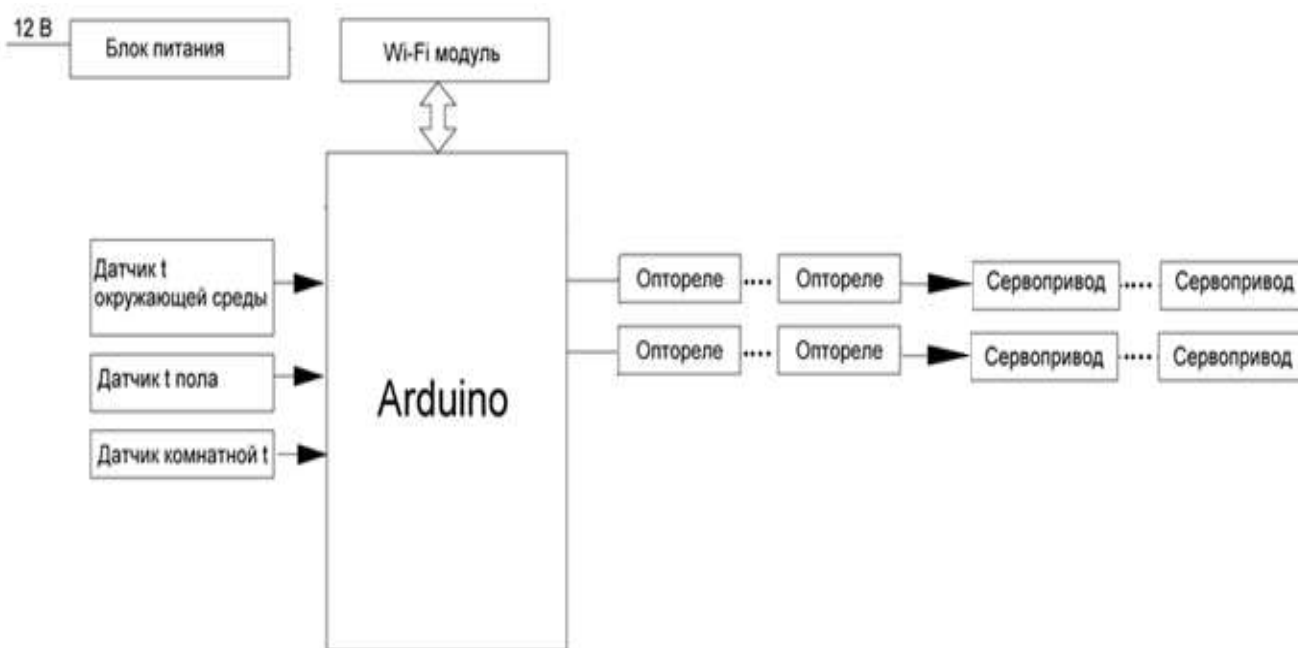


Рис. 2.2. Функциональная схема

### Описание функциональной схемы.

В системе присутствуют 3 датчика: датчик  $t$  окружающей среды, датчик  $t$  пола и датчик комнатной температуры. Вся информация с датчиков отправляется на Arduino.



От Arduino идут управляющие сигналы на сервопривод для открытия/закрытия клапана, так как сервопривод питается от сети, лучше поставить перед ним оптореле, которое будет размыкать/замыкать сеть в зависимости от поступившего сигнала.

В системе присутствует поддержка беспроводного соединения Wi-Fi («Wi-Fi-модуль»). С помощью него осуществляется передача информации.

### **3. Постановка задачи.**

Цель – создать систему, позволяющую регулировать температуру воздуха с помощью ВТП и других нагревательных элементов.

Управление ВТП осуществляется с помощью беспроводной сети. В качестве терминала может выступать компьютер или смартфон. Управляющие сигналы посылаются на контроллер с управляющего устройства с помощью Wi-Fi.

Система должна регулярно проводить опрос датчиков температуры воздуха и датчиков температуры воды для поддержания установленной комнатной температуры и предотвращения возможных аварийных ситуаций. В случае необходимости Arduino отправляет управляющий сигнал на сервопривод для открытия/закрытия клапана.

Из-за большой постоянной времени нет смысла поддерживать строго определенную температуру, поэтому целесообразней реализовать релейный режим. Температура воды и воздуха удерживается в определенном диапазоне. В этом случае погрешность составляет 1-2 С. Но это не особо ощутимо для человека.

Основные задачи СУ:

1. Установка и регулирование комнатной температуры с помощью удаленного доступа.
2. Получение и анализ входных сигналов от температурных датчиков.
3. Поддержание заданной температуры в помещении.
4. Задание температуры ВТП по дням недели.
5. Информирование пользователя о состоянии системы.

6. Предотвращение аварийных ситуаций.

7. Передача информации по Wi-Fi.

Чтобы реализовать удаленное управление, необходимо создать графический интерфейс для телефона/смартфона. При создании графического интерфейса нужно учесть, что любой пользователь должен в нем быстро разобраться. Поэтому он должен быть прост в управлении. Должен включать в себя элементы управления и элементы мониторинга. Устройство управления должно обеспечить пользователю полный контроль над управляемым устройством. Также осуществлять проверку входных и выходных сигналов. Для коммутации между устройствами, необходимо ввести протокол передачи данных.

Для реализации проекта потребуются следующие устройства автоматики:

- Сервопривод
- Отладочная плата Arduino
- Датчик температуры пола
- Комнатный термостат
- Датчик измерения температуры окружающей среды

Все функции, подлежащие реализации, делятся на аппаратную и программную часть. В аппаратную часть входят принципиальная и электрическая схемы. Схемы подключения датчиков температуры, блока питания и сервоприводов с Arduino. Программная часть включает в себя программу для Arduino, которая реализует все необходимые алгоритмы для поддержания комфортной температуры, обрабатывает все входные сигналы, с заданной частотой осуществляет опрос всех датчиков. Управление ВТП осуществляется по интерфейсу Wi-Fi. Необходимо создать программу для смартфона.

#### 4.Выбор приборов, оборудования и программного обеспечения.

##### Отладочная плата Arduino.

Для создания проекта была выбрана плата Arduino Uno. Ее памяти достаточно для реализации, количество входов и выходов хватает с запасом для подключения всех устройств автоматики. Подачу питания можно осуществить, используя внешний источник питания (адаптера AC/DC или батареи), а также можно подключить через USB.

При выборе платы учитывались следующие параметры:

- 3 цифровых входа
- 3 аналоговых выхода
- Последовательный интерфейс UART
- Объем памяти не менее 1 Кб

Данными параметрами обладает плата ArduinoUno - это устройство на основе микроконтроллера ATmega328.

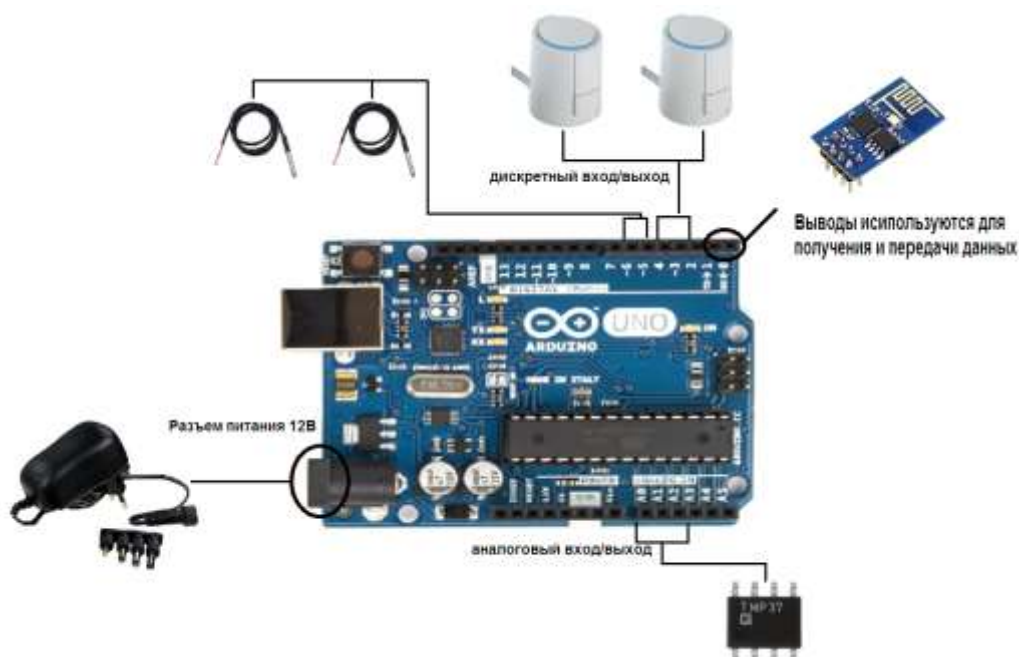


Рис. 4.1 Arduino Uno.

### Технические характеристики.

№	Характеристики	Ед. изм.	Значение
1	Микроконтроллер		ATmega328
2	Рабочее напряжение	В	5
3	Напряжение питания (рекомендуемое)	В	5 ÷ 12
4	Напряжение питания (предельное)	В	6 ÷ 20
5	Цифровые входы/выходы	шт	14
6	Аналоговые входы	шт	6
7	Flash-память	Кб	2
8	Тактовая частота	МГц	16
9	Максимальный ток одного вывода	мА	40
10	Максимальный выходной ток вывода 3.3V	мА	50

### Блок питания для Arduino.

Для работы Arduino на него необходимо подать питание. Напряжение должно быть в районе 12 В. Был выбран блок питания ROBITON EN300S 300мА.

Это импульсный блок питания, а значит стабилизированный. Из-за этого его выходное напряжение остается постоянным вне зависимости от подключенной нагрузки. При подключении нагрузки необходимо убедиться, что соблюдена полярность подключения переходника. Для этого на штекере есть маркировка.

Чтобы питать Arduino, штекер должен быть положительным. Напряжение находится в диапазоне: 7,5...12 В. С помощью переключателя происходит установка выходного напряжения: 3, 4.5, 6, 7.5, 9, 12 В. Были учтены параметры датчиков температуры и оптореле и в соответствии с этими данными, был выбран блок питания с небольшим запасом.



Рис. 4.2 Блок питания для Arduino.

Соединение с Arduino.

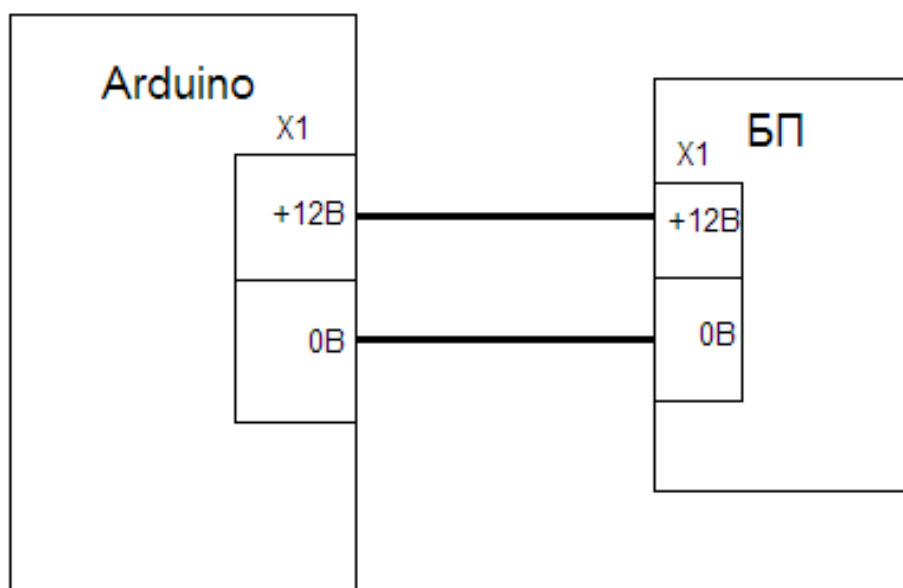


Рис. 4.3 Соединение блока питания с Arduino.

### Сервопривод для ВТП и радиаторного отопления.

Есть нормально закрытые и нормально открытые сервоприводы. У нормально закрытых клапан закрыт по умолчанию, то есть до тех пор, пока на него не поступает сигнал, он пребывает в положении «Клапан закрыт».

У нормально открытых сервоприводов клапан открыт до тех пор, пока на них не поступает сигнал. В данном случае при отсутствии напряжения теплоноситель не проходит через закрытый клапан.

В нашем случае для ВТП необходим сервопривод, нормально-закрытый. А для радиаторного отопления нормально открытый, так как при возникновении аварийной ситуации зимой, ВТП перестанет функционировать и люди будут мерзнуть. Оба устройства должны быть на 220 В. Подойдут сервоприводы фирмы Valtec - VT.TE3040.0.220.



Рис. 4.4 Сервопривод.

Принцип действия привода: В основе работы данного электротермического привода лежит принцип расширения при нагревании. При подаче на привод электрического тока, он проходит через греющий нихромовый проводник, который нагревает армопарафин, который находится в герметической емкости. Расширяющийся от нагревания армопарафин придаёт поступательное движение толкателю, преодолевая сопротивление возвратной пружины. Когда мы снимаем с привода электропитание, армопарафин тут же начинает остывать, и возвратная пружина возвращает толкатель в исходное положение.

Технические характеристики нормально открытого и нормально закрытого сервопривода (для радиаторного отопления).

№	Характеристики	Ед. изм.	Значение
1	Напряжение питания/частоты	В/Гц	230/50
2	Мощность	Вт	1
3	Диапазон температур при работе	С	0 ÷ +60
4	Температура теплоносителя	С	0 ÷ 100
5	Максимальный допустимый ток	мА	550
6	Максимальная допустимая влажность воздуха	%	80
7	Монтажное положение		<b>360°</b>

### Оптореле.

С помощью Arduino можно управлять оптореле, при этом электрическая связь между управляющей электроникой и коммутируемой нагрузкой отсутствует. С помощью реле можно включать и выключать электроприборы, которые подключены к бытовой электросети 220 В. Без реле не обойтись при обустройстве «умного дома».



Преимущества оптореле это малый ток управления, отсутствие электромагнитных помех при коммутации нагрузки, высокое напряжение изоляции, широкий диапазон рабочих температур. Еще малые габариты и большая надежность.

Оптореле срабатывает когда на вход модуля подаётся логическая единица. При этом управление производится через встроенный ключ, что даёт возможность использовать в качестве управляющего сигнала любое напряжение от 3 до 5 вольт.

Было выбрано твердотельное реле ASO 241 R на 1,5 А.



Рис. 4.5 Оптореле ASO 241 R

Технические характеристики.

№	Характеристики	Ед. изм.	Значение
1	Номинальный ток нагрузки	В	12-280
2	Управляющее напряжение	В	4-10
3	Напряжение размыкания	В	1
4	Максимальный ток нагрузки	А	1,5
5	Время включения максимальное	мс	10
6	Время выключения максимальное	мс	10
7	Напряжение изоляции	кВ	2,5
8	Рабочая температура	С	-30...80

Твердотельное реле коммутирует переменное напряжение 220 В. Управляется он напряжением питания от 4-10 В. У него есть встроенный резистор, который ограничивает ток, который идет на светодиод. Потребление на 5 В 15 мА. При поступлении сигнала на оптореле, оно размыкается и идет ток на сервопривод.

Соединение оптореле с Arduino.

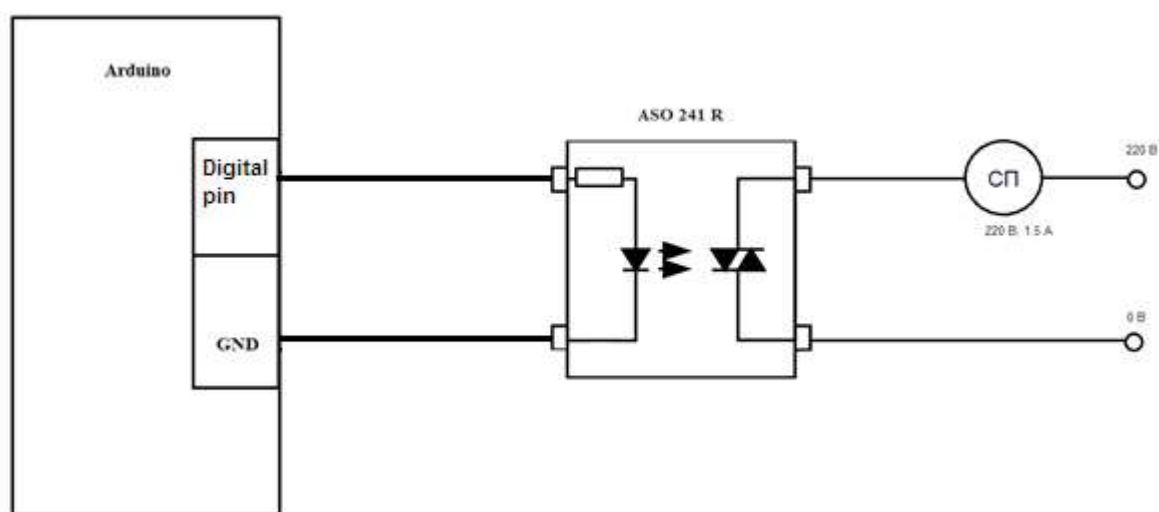


Рис. 4.6 Соединение оптореле с Arduino.

### Датчик окружающей среды и ВТП.

Выбирается на основании диапазона температур, которые будут использоваться при эксплуатации ВТП. Был выбран герметичный датчик температуры DS18B20. Он подходит и для монтирования в стяжку и его можно использовать для измерения температуры на улице и при этом не бояться, что датчик зальёт дождём.



Рис. 4.7 Герметичный датчик температуры DS18B20

Этот датчик температуры основан на популярной микросхеме DS18B20. Датчик измеряет температуру окружающей среды в диапазоне от  $-55^{\circ}\text{C}$  до  $+125^{\circ}\text{C}$  и позволяет получать данные в виде цифрового сигнала с 12-битным разрешением по 1-Wire протоколу. Так как 12 бит, то точность не менее 1 градуса. Протокол позволит подключить огромное количество таких датчиков, используя всего 1 цифровой порт контроллера, и всего 2 провода для всех датчиков: земли и сигнала. В этом случае применяется так называемое «паразитное питание», при котором датчик получает энергию прямо с линии сигнала. Все эти датчики имеют уникальный 64-битный код, который может использоваться микроконтроллером для того, чтобы общаться с определенным сенсором на общей шине. С помощью отдельной команды можно считать код отдельного сенсора.

Соединение датчика DS18B20 с Arduino.

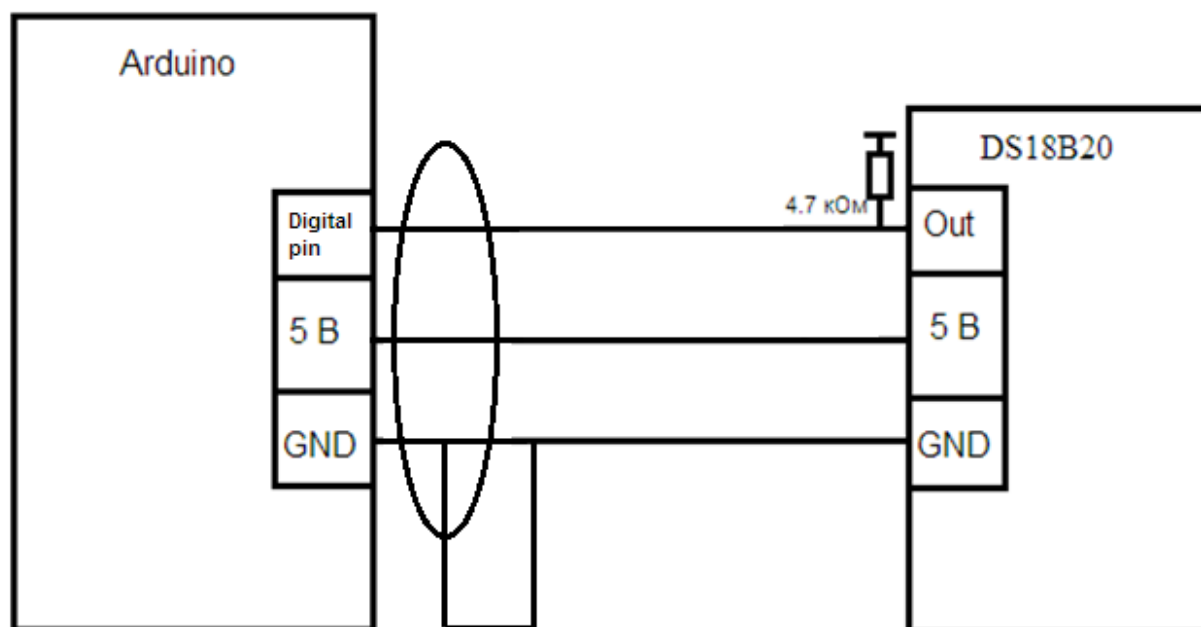


Рис. 4.8 Соединение датчика DS18B20 с Arduino.

Технические характеристики.

№	Характеристики	Ед. изм.	Значение
1	Точность	С	+0,5
2	Напряжение питания	В	3-5,5
3	Диапазон измеряемых температур	С	-55...+125
4	Потребляемый ток при опросе	мА	1
5	Потребляемый ток при бездействии	нА	750

Датчик комнатной температуры.

Температура в комнате не должна падать меньше +5 С и превышать +25 С, иначе становится жарко. По этому критерию подбираем датчик. Нам подходит датчик температуры TMP37 GS



Рис. 4.9 датчик температуры TMP37 GS

У датчик три вывода: 5 В, Out (выход температура) и GND, они приходят на соответствующие pin Arduino. Соединение осуществляется с помощью экранированных кабелей. Так как провода длинные, может наводиться высокое напряжение и чтобы Arduino не вышло из строя необходимо поставить фильтр.

Технические характеристики датчика температуры TMP37 GS.

№	Характеристики	Ед. изм.	Значение
1	Температура считывания	С	5-100
2	Напряжение питания	В	2,7...5,5
3	Точность	С	+1

Соединение Arduino с TMP37 GS.

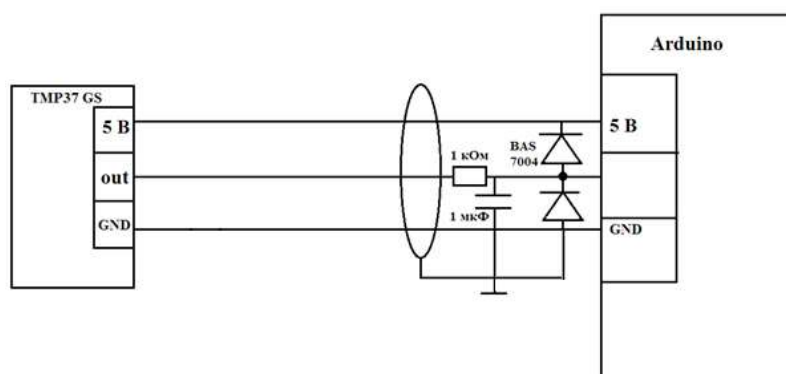


Рис. 4.10 Соединение Arduino с TMP37 GS

## Wi-Fi модуль.

Чтобы реализовать удаленное управление необходимо установить Wi-Fi модуль. Модуль ESP8266 — это простой способ добавить в устройство функции беспроводной связи через Wi-Fi. Wi-Fi модуль подключается по последовательному интерфейсу UART к Arduino. Программное сопряжение осуществляется за счет библиотек Arduino.

Для питания данного модуля подойдет 3,3 В. На плате Arduino есть pin 3,3 В, но также их можно получить с отдельного регулятора напряжения.



Рис. 4.11 Wi-Fi модуль.

## Соединение Arduino с Wi-Fi модулем.

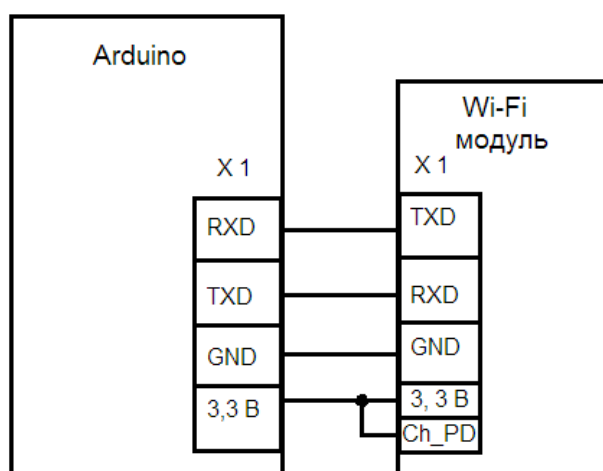


Рис. 4.12 Соединение Arduino с Wi-Fi модулем.

## Технические характеристики.

№	Наименование характеристики	Ед. изм.	Характеристики
1	Модификация		ESP-01 V090
2	Беспроводной интерфейс	ГГц	Wi-Fi 802.11 b/g/n 2,4
3	Максимальная выходная мощность	дБ	19,5
4	Номинальное напряжение	В	3,3
5	Максимальный потребляемый ток	мА	220
6	Портов ввода-вывода свободного назначения		2
7	Частота процессора	МГц	80
8	Объем оперативной памяти	КБ	96

## 5. Проектирование программной части.

Для создания программы СУ использовалась Arduino IDE, так как имеет удобный и понятный интерфейс. В этой среде программа пишется на языке processing. Это язык программирования, основанный на Java. Подходит для программирования анимации и интерфейсов. У Processing все классы являются внутренними классами основного. Это нужно помнить при разработке программы.

Алгоритм программы СУ представлен на рисунке 4.1

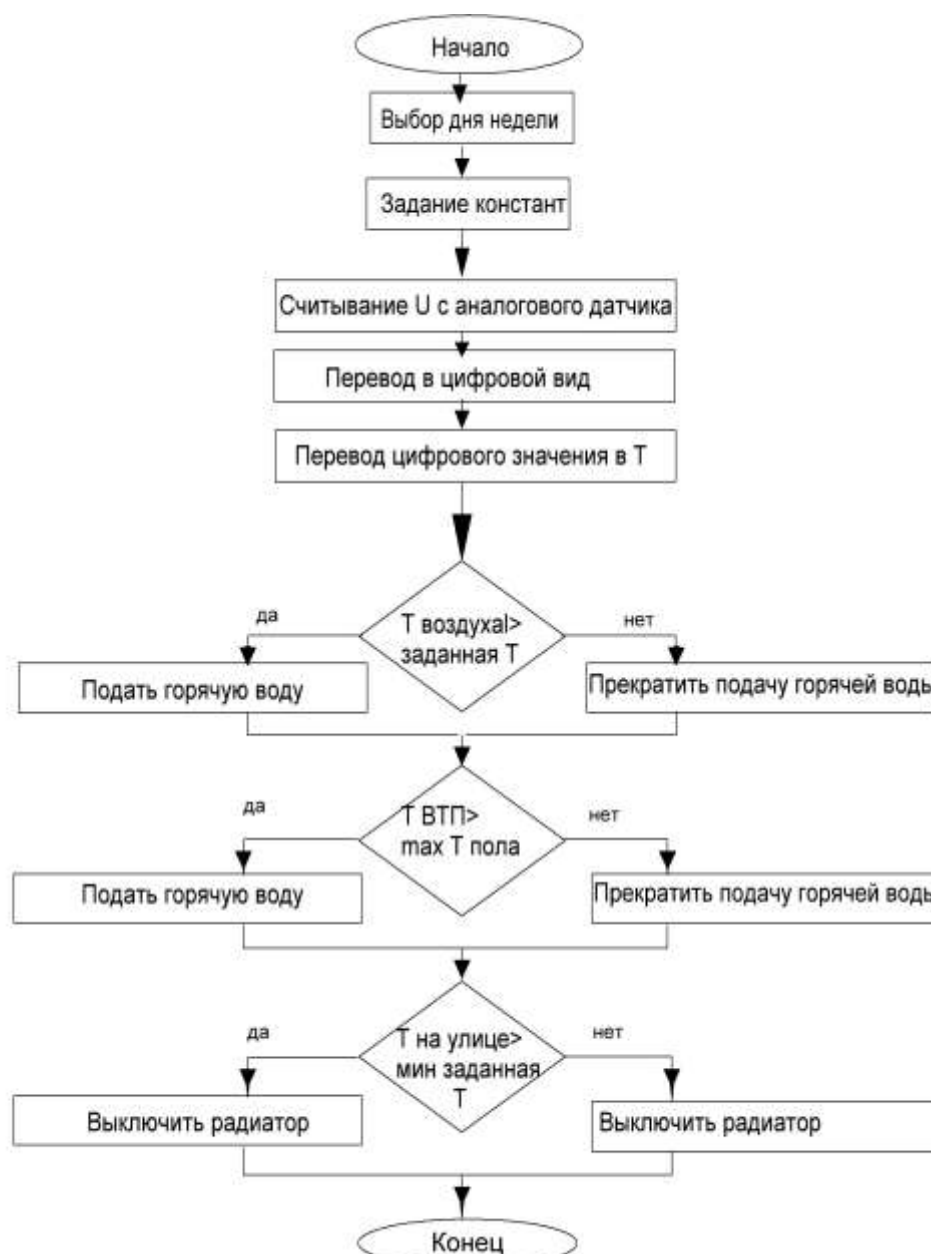


Рис. 5.1. Блок-схема алгоритма СУ.



### **Описание блок-схемы.**

В начале программы выбирается день недели, и задаются константы: константа теплого пола (максимальная температура воздуха в помещении) и константа температуры воздуха на улице, при которой включается радиатор.

Arduino начинает опрашивать датчики температуры. Сначала он опрашивает датчик температуры помещения. Так как это аналоговый датчик, с него считывается напряжение, которое обрабатывается и преобразуется в цифровой вид. После этого цифровое значение переводится в температуру.

В том случае, если температура датчика воздуха выше заданной, включается подача горячего теплоносителя на пол, то есть подается сигнал на выключение реле. Если же температура датчика воздуха ниже заданной температуры, то идет управляющий сигнал на реле, вследствие чего сервопривод начинает пропускать поток теплоносителя.

Может возникнуть следующая ситуация. Мы достигаем требуемую температуру в комнате, но при этом температура полов превышает константу. Поэтому если температура пола слишком высокая, то идет сигнал с Arduino и реле отключает сервопривод от питания. Если температура пола приемлема, то есть лежит в заданном диапазоне, то МК сигнал на реле не отправляет.

Также необходимо регулярно опрашивать датчик температуры, который измеряет температуру окружающей среды, чтобы поддерживать температуру в помещении. Поэтому, если температура воздуха на улице слишком низкая, то подключается еще один источник отопления – радиатор. Идет управляющий сигнал на реле, которое располагается в ветви радиаторного отопления. Если температура больше  $-15\text{ C}$ , то у нас работает только ВТП.

### Таблицы входов и выходов.

Сигналы с датчиков температуры и с реле подаются на аналоговые, и, соответственно дискретные входы МК.

Таблица выходов.

Название переменной	Номер pin
FLOOR_RELAY	Digital 3
RADIATOR_RELAY	Digital 4

В этой таблице располагаются оптореле. Arduino с помощью управляющих сигналов присваивает им определенное состояние.

Таблица входов.

Название переменной	Номер pin
FLOOR_SENS_D_PIN	Digital 1
STREET_SENS_D	Digital 2
AIR_SENS_A	Analog 3

Включает в себя информацию о датчиках температуры. С датчиков показания отправляются на Arduino.

## **Программное обеспечение верхнего уровня.**

Для реализации Wi-Fi-клиента для управления ВТП была использована среда разработки приложений QT.

## **Разработка графического интерфейса.**

В ходе работы был разработан графический интерфейс под операционную систему Android. Для удобства использования было введено несколько графических экранов.

На главном экране пользователь видит главное меню, которое содержит следующие кнопки: «календарь» и «настройка параметров». Если он захочет просмотреть текущую температуру пола, помещения или узнать, какая сейчас температура окружающей среды в его районе, а также установить температуру помещения в своей квартире, то он должен нажать на кнопку «Настройка параметров». После этого действия, он переходит на новый экран, где располагаются все вышеизложенные параметры.

Задание температуры осуществляется с помощью перемещения ползунка «Выход 1». Изменения температуры ограничено и лежит в диапазоне от +5 С до +28 С тепла. Так мы предупреждаем пользователя от ошибки, а именно от нежелательного замораживания или перегрева комнаты.

Далее необходимо нажать на кнопку «Установить». Программа обработает полученные данные и отправит с помощью беспроводной сети их Arduino. После того, как пользователь установит желаемую температуру, он может вернуться в основное меню, нажав на кнопку «Главное меню».

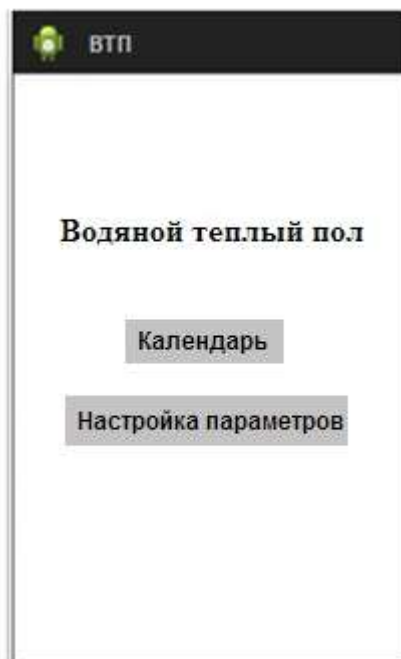


Рис.5.2 Главное меню

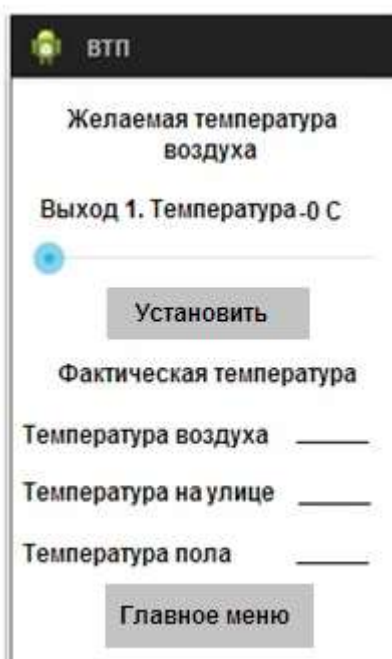


Рис.5.3 Настройка параметров

В главном меню, нажав на кнопку «Календарь», пользователь переходит к новому окну, которое отображает дни недели. Здесь, выбрав любой день, например, понедельник пользователь переходит на новую страницу, которая дает ему возможность настроить температуру по часам.

С помощью этой функции, мы еще больше автоматизируем систему и экономим ресурсы. Так как с понедельника по пятницу человек отсутствует в квартире с 8:00 – 18:00, и в это время не обязательно поддерживать в помещении температуру в районе 26 С. Можно понизить и до 15 С. А к моменту возвращения пользователя домой, снова её поднять. После чего можно снова вернуться в главное меню.

Так можно настроить температуру на всю неделю, после чего настройки сохранятся, и будут поддерживаться до момента их изменения.

В качестве примера приведен экран настройки температуры помещения для понедельника.



Рис.5.4 Календарь

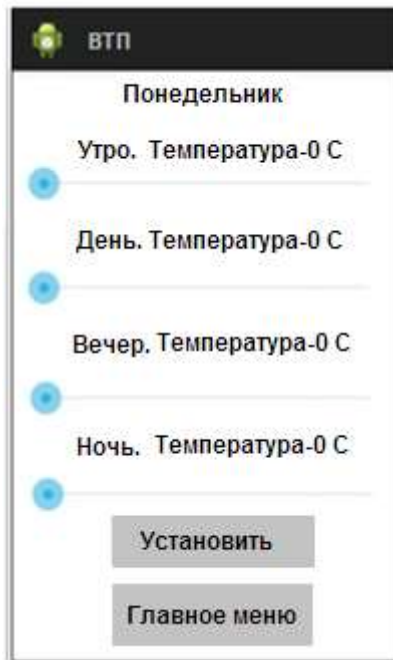


Рис.5.5 Ручная настройка

Данный графический интерфейс очень прост и удобен в использовании. Он предоставляет пользователю большой объем возможностей. Температура воздуха будет всегда находиться в желаемом диапазоне значений.

## **Заключение.**

В результате работы была создана система управления водяным теплым полом, которая корректно выполняет все требуемые функции. Программа была протестирована, ошибок не обнаружено.

Для выполнения поставленной задачи было необходимо разобраться в работе реального объекта и реализовать его функциональную и принципиальную схемы. В ходе работы была максимально оптимизирована аппаратная часть.

Получены новые навыки программирования микроконтроллера Arduino Uno, более досконально изучена среда программирования Arduino IDE. При отладке программы возникали трудности, которые были успешно устранены. Был приобретен полезный опыт, который можно будет применить на практике.

**Список источников.**

Электронные сайты:

1) <http://teplo.guru/teplyy-pol/vodyanoy/vodyanoi-teplyi-pol-v-kvartire.html>

Водяной теплый пол в квартире, установка и принцип работы.

2) <http://mypractic.ru/urok-12-posledovatelnyj-port-uart-v-arduino-biblioteka-serial-otladka-programm-na-arduino.html>

принцип работы последовательного интерфейса UART.

3) <http://arduino.ru/Hardware/ArduinoBoardUno> характеристики контроллера Arduino Uno.

4) <http://developer.alexanderklimov.ru/arduino/arduino-minimum.php>

программирование Arduino Uno.

5.) <http://www.alldatasheet.com/view.jsp?Searchword=Tmp37> datasheet TMP37

6.) <https://avrlab.com/node/131> datasheet DS18B20 datasheet DS18B20

## Приложение 1. Код программы.

```
#include <OneWire.h>
```

```
#include <DallasTemperature.h>
```

```
#define FLOOR_RELAY_PIN 3
```

```
#define RADIATOR_RELAY_PIN 4
```

```
#define FLOOR_SENS_D_PIN 1
```

```
#define STREET_SENS_D_PIN 2
```

```
#define AIR_SENS_A_PIN A3
```

```
OneWire    oneWire_street(STREET_SENS_D_PIN);    //экземпляр    класса
управления шиной по протоколу ONE_WIRE для улицы
```

```
OneWire    oneWire_floor(FLOOR_SENS_D_PIN);    //    экземпляр    класса
управления шиной по протоколу ONE_WIRE для пола
```

```
// Pass our oneWire reference to Dallas Temperature.
```

```
DallasTemperature    street_sensors(&oneWire_street);    //экземпляр    класса
управления цифровым термометром
```

```
DallasTemperature floor_sensors(&oneWire_floor);
```



```
typedef struct {  
  
    int morning_temp;  
  
    int day_temp;  
  
    int evening_temp;  
  
    int night_temp;  
  
} DAY_TEMP_STUCT_T; //создаем структуру  
  
TIME_T cur_time;  
  
DAY_TEMP_STUCT_T week [7];  
  
const int floor_max_temp = 28; //константа горячего пола  
  
const int street_min_temp = -15; //константа температуры воздуха на улице,  
при которой включается радиатор  
  
void get_task_from_phone () { //функция, которая будет получать задание от  
телефона  
  
    //В этой функции мы будем получать температуру задания, и возвращать ее  
как результат функции.  
  
    DAY_TEMP_STUCT_T incame_week [7];  
  
    TIME_T incame_cur_time;  
  
    for (i=0; i<7; i++) { //копируем полученный массив температур по дням  
неделей
```

```

    week [i] = incame_week [i];

}

incame_cur_time=cur_time; //получаем время с телефона
}

char ssid[] = "yourNetwork"; // your network SSID (name)

char pass[] = "secretPassword"; // your network password

int status = WL_IDLE_STATUS; // the Wifi radio's status

void connect_to_wifi () {

    //Функция подключения к Wi-Fi

    if (WiFi.status() == WL_NO_SHIELD) {

        Serial.println("WiFi shield not present");

        // don't continue:

        while(true);

    }

    // attempt to connect to Wifi network:

    while ( status != WL_CONNECTED) {

        Serial.print("Attempting to connect to WPA SSID: ");

        Serial.println(ssid);

```

```
// Connect to WPA/WPA2 network:

status = WiFi.begin(ssid, pass);

}

int get_temp_based_on_time () {

    //основываясь на текущем времени, возвращаем необходимую для
    поддержания температуры

    if (cur_time.hours<11) then return week[cur_time.day].morning; //возвращаем
    температуру в зависимости от времени и дня недели

    if (cur_time.hours<18) then return week[cur_time.day].day;

    if (cur_time.hours<23) then return week[cur_time.day].evening;

    if (cur_time.hours<6) then return week[cur_time.day].night;

    return week[cur_time.day].day;

}

int air_sens_val (int pin) { //Значение датчика температуры воздуха в комнате

    float voltage = 5.0/1024.*analogRead(pin);

    return voltage/0.01-40; //возвращаем пересчитанную темепературу

}

int street_sens_val () { //Значение датчика температуры воздуха на улице
```

```
street_sensors.requestTemperatures(); // посылаем запрос на получение
температуры

return street_sensors.getTempCByIndex(0); //возвращаем результат работы
функции, получающей температуры

}
```

```
int floor_sens_val () { //Значение датчика температуры пола
```

```
floor_sensors.requestTemperatures(); // посылаем запрос на получение
температуры

return floor_sensors.getTempCByIndex(0); //возвращаем результат работы
функции, получающей температуры

}
```

```
void relay_regime (int temp_task) {
```

```
if (air_sens_val (AIR_SENS_A_PIN)>temp_task) { //в случае, если
температура датчика воздуха выше заданной, выключаем подачу горячего
теплоносителя на пол
```

```
digitalWrite(FLOOR_RELAY_PIN, LOW); //выключаем реле на пол

}
```

```
else { //в ином случае - включаем
```

```
digitalWrite(FLOOR_RELAY_PIN, HIGH); //включаем реле на пол

}
```

```
if (floor_sens_val ()>floor_max_temp) { //если температура пола
максимальная
```

```
digitalWrite(FLOOR_RELAY_PIN, LOW); //выключаем реле на пол
```

```
}
```

```
if (street_sens_val ()>street_min_temp) { //если температура воздуха на улице
слишком низкая, то мы включаем радиатор
```

```
digitalWrite(RADIATOR_RELAY_PIN, HIGH); //включаем реле на радиатор
```

```
}
```

```
else { //в ином случае - выключаем
```

```
digitalWrite(RADIATOR_RELAY_PIN, LOW); //выключаем реле на
радиатор
```

```
}
```

```
}
```

```
void setup() { //стандартная функция установки начальных параметров
(запускается один раз при включении ARDUINO)
```

```
// put your setup code here, to run once:
```

```
connect_to_wifi ();
```

```
street_sensors.begin(); // инициализация интерфейса общения с цифровым
датчиком температуры воздуха на улице
```

```
floor_sensors.begin(); // инициализация интерфейса общения с цифровым
датчиком температуры пола
```

```
pinMode(FLOOR_RELAY_PIN, OUTPUT);

pinMode(RADIATOR_RELAY_PIN, OUTPUT);

for (int i=0; i<7; i++) {

    week[i].morning_temp = 27; //задаем температуры дня по умолчанию

    week[i].day_temp = 27;

    week[i].evening_temp = 27;

    week[i].night_temp = 27;

}

}

void loop() {

    get_task_from_phone ();

    int temp_task = get_temp_based_on_time;

    // put your main code here, to run repeatedly:

    relay_regime (temp_task);

    delay (10000); //задержка в 10 секунд

    cur_time.sec+=10;

}
```