

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ

\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2018 г.

Аппаратный компонент системы профилактики образования наледи

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ  
ЮУрГУ-090301.2018.171 ПЗ ВКР

Руководитель работы,  
к.т. н., доцент

\_\_\_\_\_ П.О. Шабуров  
«\_\_» \_\_\_\_\_ 2018 г.

Автор работы

студент группы КЭ – 484

\_\_\_\_\_ О.С. Степанова  
«\_\_» \_\_\_\_\_ 2018 г.

Нормоконтролёр, ст. преп. каф  
«Электронные вычислительные  
машины»

\_\_\_\_\_ В.В. Лурье  
«\_\_» \_\_\_\_\_ 2018 г.

## АННОТАЦИЯ

Степанова О.С. Аппаратный компонент системы профилактики образования наледи. – Челябинск: ФГАОУ ВО «ЮУрГУ» (НИУ) ВШЭКН; 2018, 98 с., 38 ил., библиогр. список – 28 наим., 5 прил., 1 лист графического материала ф. А3.

В ходе анализа поставленной задачи был разработан аппаратный комплекс, состоящий из различных датчиков с внедрением искусственного интеллекта.

Система антиобледенения предназначена для предотвращения наледи и растапливания снега в зимнее время на открытых поверхностях и кровли. На сегодняшний день, монтаж данной системы является самым эффективным и простым способом обеспечения безопасности жителей в зимний период времени.

Пояснительная записка отражает все этапы разработки устройства: изучение процесса снеготаяния, проектирование, разработка макета.

Программирование системы выполнено в IDE Arduino на языке C++. Моделирование и нейронной сети выполнено в прикладном пакете MATLAB neural network toolbox. Обучение нейросети произведено на языке C#. Пояснительная записка к выпускной квалификационной работе оформлена в текстовом редакторе Microsoft Office Word 2007.

					<b>ЮУрГУ-09.03.01.2018.171 ПЗ</b>		
<b>Изм.</b>	<b>Лист</b>	<b>№ докум.</b>	<b>Подпись</b>	<b>Дата</b>			
Разраб.		О.С. Степанова					
Пров.		П.О. Шабуров					
Н. Контр.		В.В. Лурье					
Утв.		Г.И. Радченко					
<b>Аппаратный компонент системы профилактики образования наледи</b>					<b>Лит.</b>	<b>Лист</b>	<b>Листов</b>
					Д		3
					ФГАОУ ВО «ЮУрГУ (НИУ)» Кафедра ЭВМ		
					<b>2</b>		

## ОГЛАВЛЕНИЕ

	ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	5
	ВВЕДЕНИЕ.....	6
1	АНАЛИЗ РЕШАЕМОЙ ЗАДАЧИ .....	7
1.1	Актуальность задачи .....	7
1.2	Цель выпускной квалификационной работы .....	8
1.3	Обзор существующих аналогов.....	8
1.4	Обзор литературы.....	10
2	ИССЛЕДОВАНИЕ ЛЕДООБРАЗОВАНИЯ.....	12
2.1	Общие сведения.....	12
2.2	Обледенение поверхностей.....	13
2.3	Конструкция системы водоотведения .....	15
3	РЕАЛИЗАЦИЯ СИСТЕМЫ ОБОГРЕВА.....	17
3.1	Выбор греющего кабеля.....	17
3.2	Способы обогрева поверхностей .....	20
3.3	Алгоритм работы системы .....	24
3.4	Управление системой с помощью искусственного интеллекта.....	25
3.5	Моделирование нейронной сети.....	31
4	КОНСТРУИРОВАНИЕ СИСТЕМЫ ОБОГРЕВА .....	34
4.1	Разработка системы.....	34
4.2	Выбор объекта управления .....	38
4.2.1	Raspberry pi .....	38
4.2.2	Контроллер Arduino.....	40
4.2.3	Обоснование использования контроллера .....	42
4.2.4	Выбор модификации Arduino .....	44
4.3	Функциональный состав .....	44
4.3.1	Датчик температуры и влажности DHT22 .....	45
4.3.2	Датчик тока .....	48
4.3.3	Модуль часов реального времени.....	50
4.3.4	Датчик для измерения температуры почвы .....	53
4.3.5	Ethernet shield.....	56
4.3.6	Силовой двухканальный релейный модуль .....	58

4.4	Подключение элементов .....	60
4.5	Программирование в Arduino IDE.....	63
4.6	Взаимодействие с сервисом.....	66
5	РАСЧЕТ МАТЕРИАЛЬНЫХ ЗАТРАТ .....	68
	ЗАКЛЮЧЕНИЕ .....	70
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	71
	ПРИЛОЖЕНИЯ .....	74
	Приложение А.....	74
	Приложение Б .....	82
	Приложение В.....	86
	Приложение Г .....	90
	Приложение Д.....	88

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

- EEPROM (Electrically Erasable Programmable Read-Only Memory) – электрически стираемая программируемая постоянная память;
- HTTP (HyperText Transfer Protocol) – протокол передачи данных;
- IDE (Integrated Development Environment) – интегрированная среда разработки;
- MISO (Master In Slave Out) – вход ведущего, выход ведомого;
- MOSI (Master Out Slave In) – выход ведущего, вход ведомого;
- MSE (mean squared error) – средняя квадратичная ошибка;
- SCK (Serial Clock) – последовательный тактовый сигнал;
- SPI (Serial Peripheral Interface) – последовательный периферийный интерфейс;
- SREG (Status Register) – регистр статуса, содержит набор флагов показывающих текущее состояние микроконтроллера;
- SS (Slave Select) – выбор ведомого;
- USB (Universal Serial Bus) – универсальная последовательная шина;
- АЦП – аналогово-цифровой преобразователь;
- ГОСТ – государственный стандарт;
- ИНС – искусственная нейронная сеть;
- МК – микроконтроллер;
- ОЗУ – оперативное запоминающее устройство;
- ОС – операционная система;
- ПЗУ – постоянное запоминающее устройство;
- ПОС 61 – припой оловянно-свинцовый;
- ПРГ – противогололедный реагент;
- УР-231 – полиуретановый лак;
- ШИМ – широтно-импульсная модуляция, это способ кодирования аналогового сигнала путем изменения ширины (длительности) прямоугольных импульсов несущей частоты;
- ЭДС – электродвижущая сила. Энергетическая характеристика источника.

## ВВЕДЕНИЕ

С каждым годом, в зимний период времени, новостные ленты по всей России пестрят новостями о том, что коммунальные службы не справляются со своими обязанностями – уборкой снега, а также не хватает техники. За день снег тает, а к вечеру тротуар превращается в «бугристый коток». Гололед в городах, на который активно жалуются жители, привел к традиционному для этого периода росту травм.

Хочется отметить, что на сайте <https://www.change.org> создано множество петиций, в которых граждане просят властей запретить использовать противогололедные реагенты (далее ПРГ) и реагенты, состоящие из соли и песка. Одну из них, которая написана жителем Санкт-Петербурга, подписали более 107 тысяч человек.

Обогрев пешеходной зоны на улицах современных городов во всем мире становится нормой. Простой и надежный обогрев тротуара, крыш, террас, ступеней лестниц, въездов в гараж и др. в зимнее время позволяет содержать их поверхность теплой и сухой, исключая образование льда и скопление снега. Установка такой системы значительно снижает риск травматизма людей, вероятность аварийной ситуации из-за гололедицы, уменьшает расходы на обслуживание и ремонт дорожного покрытия, улучшает эксплуатационные характеристики уличных поверхностей в зимнее время года, т.к. нет необходимости убирать лёд ломом и лопатами, а это значит, что поверхность сохраняет свой внешний вид долгое время.

Стоит отметить, что ежегодно на закупку противогололедного средства тратятся миллионы и миллиарды рублей. Так, прошлой зимой Москва потратила 6,6 млрд. рублей, Екатеринбург 19 млн. рублей, Красноярск 11,7 млн. рублей.

# 1 АНАЛИЗ РЕШАЕМОЙ ЗАДАЧИ

## 1.1 Актуальность задачи

С 1930 г. дворники применяли в качестве противогололедного средства – смесь технической соли (NaCl) и песка. В 1960 г. власти поняли, что стране приходится тратить огромные деньги весной на очистку в городе водостоков, тротуаров и газонов от песка. Поэтому, в 1996 г. было принято решение отказаться от песка, и использовать только техническую соль. Уже через 5 лет, в 2001 г., коммунальные службы перешли к использованию новых ПРГ, где в составе в основном содержались твердые хлориды. В то же время, ПРГ теряют свою эффективность в городах с низкими температурами в холодное время года. Но в 2006 г. правительство страны запретило применять хлористый магний из-за накопления аниона магния в почвах и водах.

Поэтому, с 2007г. дороги и тротуары посыпаются смесью соли и песка. Следовательно, проблема гололеда так и не решилась за эти годы. Замёрзшие ухабы никуда не пропадают, а смесь NaCl лишь портит обувь и одежду жителей города, ускоряет коррозию автомобилей, а также, наносит вред экологии и окружающей среде, попадая в подземные воды, почву и реки. Такой метод борьбы приводит к разьеданию проводов, проложенных под землей, вызывает аллергии [1]. В связи с этим, было принято решение разработать автоматизированную систему антиобледенения, которая полностью покроеет цели и задачи, поставленные заказчиком, будет иметь превосходство перед старыми, традиционными для России средствами борьбы с гололедом и обледенением, а также, обладать линейкой преимуществ перед своим аналогом. Она может использоваться не только для тротуаров, но и для очистки крыш, дорог, ступеней лестниц, спортивных площадок, автомобильных стоянок, дорог с наклоном, взлетных полос и др.

Работает она в угрожаемые периоды – в то время, когда происходят суточные перепады температур и вероятность образования наледи наиболее высока.

## 1.2 Цель выпускной квалификационной работы

Целью выпускной квалификационной работы является разработка аппаратного компонента системы антиобледенения на основе контроллера, с применением греющих кабелей для стаивания снега и растапливания льда на крышах, водостоках и открытых поверхностях, таких, как: крыльцо, тротуар, ступени. Разумеется, похожие системы уже были разработаны ранее и продолжают разрабатываться.

Вышеуказанная цель будет достигнута решением следующих задач:

1. Исследование принципа работы проектируемой системы;
2. Исследование физики ледообразования;
3. Разработка самой системы управления на основе данных по физике ледообразования;
4. Обеспечение системы дистанционным управлением;
5. Внедрение нейросети;
6. Изучение характеристик комплектующих и их выбор.

Исследование и выполнение ряда поставленных задач основывается на общеизвестных теоретических понятиях о ледообразовании, а также, использовании современного и общедоступного инструмента для проектирования электронных устройств, который плотно взаимодействует с окружающей средой посредством различных датчиков и модулей.

## 1.3 Обзор существующих аналогов

В столице Исландии уже существует подобная система. Только нагревательными элементами служат трубы с горячей водой, так как Рейкьявик славится своими термальными источниками, которые и являются «источником энергии» для системы. Но эта система не автоматизирована. Коммунальные службы сами включают нагревательные элементы, как только видят ледообразование.



В авиации против обледенения деталей конструкций летательных аппаратов изготавливают такой формы, чтобы они имели минимальное лобовое столкновение (аэродинамика тела). Помимо этого для предотвращения обледенения на земле перед полётом проводится противообледенительная обработка летательных аппаратов. В полёте обледенения удаляют, в основном, с помощью нагрева критичных поверхностей электрическим током или горячим воздухом от двигателей. Преимуществом приведенного решения в авиации являются минимальные затраты, а недостатком то, что этот способ не может применяться при обогреве пешеходной зоны.

Системы антиобледенения используются и в России, например, в Москве или Казани для крыш и водосточных труб. Однако они имеют ручное управление, в них нет каких-либо датчиков, только самонагревающаяся кабель. Стоимость довольно высокая (от 90 тыс. рублей).

Наиболее близким из аналогов считается разработка Шведской компании «Thermo», которая находится в Стокгольме, и занимается производством и продажей систем кабельного электрического отопления. В их продукции можно встретить как греющие кабели, так и терморегуляторы. К примеру, thermoreg ЕТО – это терморегулятор, который включает в себя датчик температуры воздуха и влажности, он считывает погодные условия, и, при неблагоприятной погоде включает систему обогрева. Данный терморегулятор стоит довольно дорого, 22 774 рубля. Его можно считать аналогом проектируемой системы, однако, он имеет ряд своих недостатков. Это высокое потребление электроэнергии, его дороговизна, отсутствие возможности принудительного управления системой удаленно, просмотр данных с датчика на веб-сайте или приложении для on-line мониторинга таких показателей как: температура и влажность воздуха, количество потребленного электричества за сутки. Поэтому и необходимо разработать такую систему, которая позволит устранить данные недостатки.

В связи с этим, необходимо создать автоматизированную систему с возможностью дистанционного управления, которая будет иметь превосходство перед старыми, традиционными для России средствами борьбы с гололедом и

обледенением, а также, обладать линейкой преимуществ перед своими аналогами. Она должна обеспечивать очистку ото льда не только тротуаров, но и для крыш, дорог, ступеней лестниц, спортивных площадок, автомобильных стоянок, дорог с наклоном, взлетных полос и др. Пользователю должна быть предоставлена возможность дистанционного контроля и мониторинга всех процессов, в том числе температуры и влажности воздуха, суточного потребления электроэнергии и др.

#### 1.4 Обзор литературы

При написании выпускной квалификационной работы были использованы статьи в периодических изданиях Российской Федерации, научная и учебно-методическая литература, техническая литература, справочная документация. Данная разработка включает в себя несколько разделов науки и техники.

Основными источниками, раскрывающими теоретические основы физики ледообразования, явились работы Пермякова Г.Н., Литвинова И.В., Любушкина С.Г., Кошевого В.А. В данных источниках подробно рассмотрены понятия о гололеде, гололедице, принципе обледенения поверхностей, а также, о структуре атмосферных осадков. Работа Заморского А.Д. «Атмосферный лед: иней, гололед, снег, град» описывает состояние воды в виде атмосферных осадков, и, определяет, при каких показателях они образуются.

В книге Бляхмана Л.С. «Глобальные, региональные и национальные тенденции развития экономики России в XXI веке» описаны проблемы загрязнения воздуха, почвы и вод при использовании вредных реагентов против обледенения тротуаров.

В работе «отопление загородного дома» Лещинской Л.В и Малышева А.А можно найти сведения о современных отопительных приборах. Это и водяное отопление, и популярное во всем мире электрическое отопление. В данной книге особое внимание уделяется разделу, посвященному системе кабельного обогрева, который используется в данном проекте.

Для разработки системы, а именно, подключения различных модулей, а также, программирования, использованы работы Соммера У. «Программирование микроконтроллерных плат Arduino/Freduino», Иго Т. «Arduino, датчики и сети для связи устройств», Петина В.А. «Arduino и Raspberry Pi в проектах Internet of Things». В данных изданиях рассмотрены примеры создания устройств, показана установка и настройка среды разработки, описаны технические возможности, особенности подключения и взаимодействия датчиков.

Реализация нейронной сети и ее обучение рассмотрено на основе работ Николенко С.И. «Самообучающиеся системы» и «Глубокое обучение: погружение в мир нейронных сетей», Осовского С.А. «Нейронные сети для обработки информации», Галушкина А.И. «Нейронные сети: основы теории», Головки В.А. «Нейронные сети: обучение, организация и применение». Наглядный пример моделирования нейронной сети описан в книге Медведева В.С. «Нейронные сети. MATLAB».

## 2 ИССЛЕДОВАНИЕ ЛЕДООБРАЗОВАНИЯ

### 2.1 Общие сведения

Появление наледи на поверхностях в зимнее время года является неизбежным явлением, которое наблюдается во многих регионах России от 2 до 5 месяцев, а в отдельных регионах это число становится равным 6-7 месяцам.

Существуют такие явления, как гололедица и гололед. В профессиональной речи синоптиков эти два понятия значительно различаются. Но эти два процесса одинаково опасны и представляют большую угрозу жизни и здоровью человека.

Итак, гололедица – это процесс обледенения поверхностей, который протекает при отрицательной температуре воздуха и его высокой влажности. Или же, в узком смысле – корка льда на поверхности, образовавшаяся в результате понижения температуры воздуха и замерзании дождевой или талой воды.

Гололед, в свою очередь – плотный слой льда, который зависит не только от погодных условий, но и от температуры поверхности, на которой образуется лёд. Если температура поверхности ниже нуля, а температура воздуха 0 или ниже градусов, то при выпадении и замерзании жидких осадков, таких как: дождь, морось, туман, капель, на ней образуется лёд. Появляется тонкая прозрачная пленка с высокой степенью скольжения [2].

Стоит отметить, что скорость и сила ветра в процессе ледообразования участвуют косвенно. Он не оказывает влияния на восприятие температуры термометром, но человека «обманывает» на несколько градусов. То есть, ветер, в физическом смысле, влияет на температуру поверхности, отсюда следует, что можно не замерять параметры ветра, т.к. если температура ветра будет ниже температуры воздуха, то и пропорционально этому, температура поверхности понизится.

Итак, если бороться одновременно с гололедом и гололедицей, то стоит учитывать параметры, которые характеризуют и тот и другой процесс, а именно: температура и влажность воздуха, температура поверхности. Механическое воздействие со стороны живого существа или транспортного средства, в случае с

тротуаром, такое как: протаптывание тропинок, скалывания части льда подошвой и т.д – не учитываются, поскольку, рабочей плоскостью считается вся поверхность, а не ее часть.

## 2.2 Обледенение поверхностей

Теперь определим, при каких численных показателях проходит процесс обледенения. Всем известно еще из курса школьной физики, что кристаллизация жидкости происходит при температуре воздуха от 0 °С и ниже. Также, из этого можно сделать вывод, что вода, попадая на поверхность, будет так же замерзать, если температура поверхности будет меньше или равна 0 °С, даже если температура воздуха при этом будет положительной, а выпадаемые осадки будут считаться, как дождь или туман.

Очень важно, знать значение относительной влажности воздуха, т.к именно она говорит нам о наличии осадков. Относительная влажность воздуха — это отношение массовой доли воды, которая содержится в воздухе к максимально возможной при данной температуре, измеряется данная величина в процентах. Так, влажность в 95% говорит о том, что воздух насыщен водяными парами на 95% относительно максимально возможного насыщения при данной температуре. Стоит отметить, что снег образуется от группировки частиц влаги [3].

Воздух в природе не бывает без водяных паров, в нем всегда находится какое-то их количество. На Земле нет мест, где была бы зарегистрирована нулевая относительная влажность. Наибольшая относительная влажность воздуха – 100 процентов – при тумане. Воздух считается умеренно влажным и без осадков, при относительной влажности 50-75%.

Ни в одном источнике не указан точный показатель влажности во время таких явлений, как дождь, снегопад и морось. Но, с помощью наблюдений в течение ноября-марта месяцев 2017-2018 года, было замечено, что влажность воздуха при дожде не опускается ниже 85-89%, при слабой снежной крупе – 90%, а при

снегопаде – 95%. На рисунке 1 представлены показатели во время наблюдений в течение 5 месяцев, а также, приблизительный отклик системы.

Дата	Время суток	Влажность воздуха	Температура воздуха	Обогрев
20.11.2017	утро	90	-6	да
	день	81	-2	нет
	вечер	74	-2	нет
	ночь	82	-3	нет
18.12.2017	утро	93	-15	да
	день	73	-8	нет
	вечер	68	-12	нет
	ночь	75	-22	нет
24.01.2018	утро	58	-27	нет
	день	66	-23	нет
	вечер	71	-23	нет
	ночь	65	-26	нет
20.02.2018	утро	95	-12	нет
	день	91	-8	да
	вечер	77	-8	нет
	ночь	90	-12	да
25.03.2018	утро	64	-13	нет
	день	61	2	нет
	вечер	68	-1	нет
	ночь	69	-6	нет

Рисунок 1 – Результат наблюдений

Итак, продолжительное выпадение осадков, сопровождающихся высокой влажностью и низкой температурой, вызывают обледенение поверхностей, особенно, если и ее температура отрицательна. Для наглядного представления показатели этих параметров приведены в таблице 1.

Таблица 1 – Диапазон показаний, при котором происходит ледообразование

Показатель влажности	Показатель температуры воздуха	Показатель температуры почвы
$\geq 89\%$	$Min_{в} < t \leq 0$	$Min_{п} < t_{п} \leq 0$

где  $Min_{в}$  – минимум из диапазона измерения датчика температуры воздуха,  $Min_{п}$  – минимум из диапазона измерения датчика температуры почвы.

### 2.3 Конструкция системы водоотведения

Так как система считается универсальной, то есть, может использоваться для крыш и пешеходных поверхностей, то рассмотрим следующие случаи системы водоотведения:

1. Сегодня на рынке представлено множество систем линейного поверхностного водоотведения, которые обладают различными достоинствами или недостатками. На рисунке 2 представлена одна из них. Водоотведение с крыш будет осуществляться по желобам, которые, в свою очередь, будут обогреваться. Далее талая вода по водостоку сливается в дождеприемник и проходит по системе линейного поверхностного водоотвода, куда затем попадает в канализацию, либо в какой-либо резервуар.

2. Для тротуаров, в целях удаления влаги, используется та же система. Но при этом, стоит учитывать, что часть талой воды будет уходить в грунт, или испаряться.

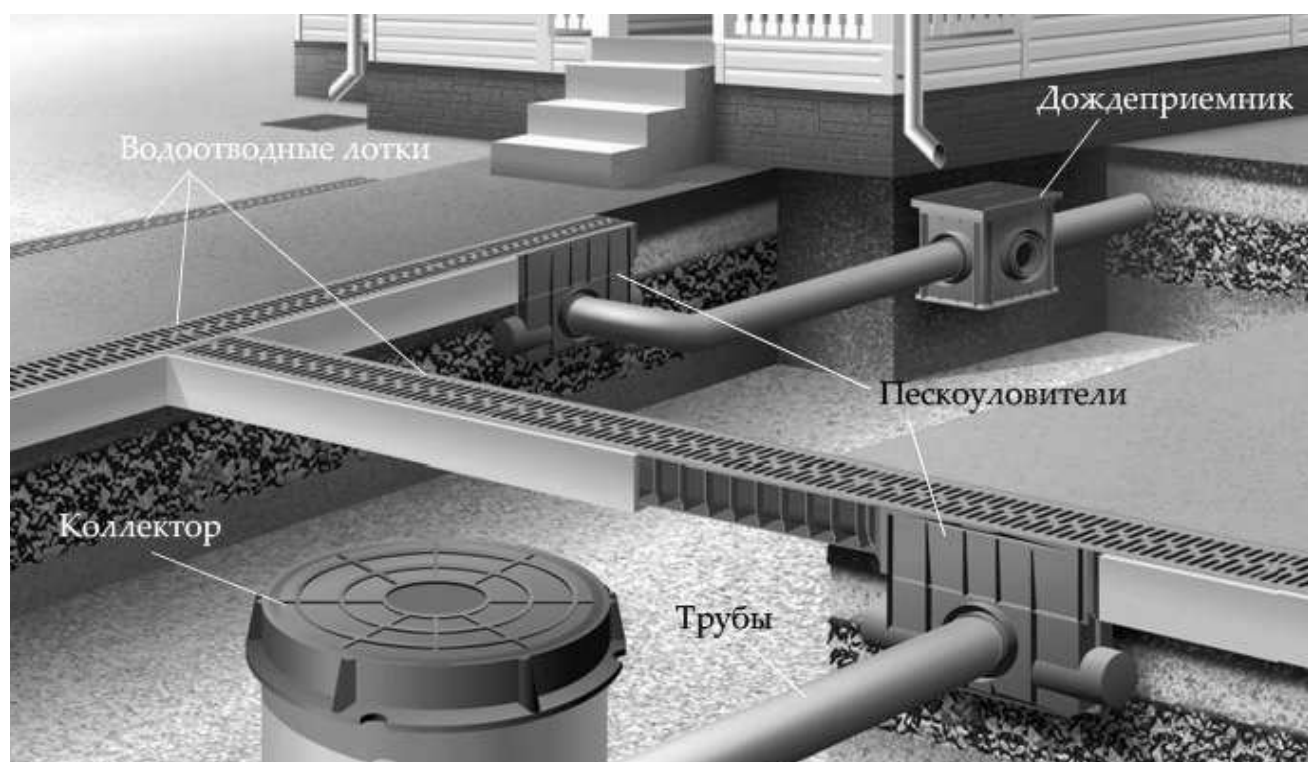


Рисунок 2 – Система линейного поверхностного водоотвода

Процесс испарения продолжается непрерывно. Точно также происходит и процесс возвращения молекул обратно в жидкость. На открытых пространствах, ветры уносят молекулы, которые перешли из жидкости в газ, тем самым не давая им вернуться обратно в жидкость.



### 3 РЕАЛИЗАЦИЯ СИСТЕМЫ ОБОГРЕВА

#### 3.1 Выбор греющего кабеля

Для того чтобы система стала антиобледенительной, необходимо использовать греющие кабели. Изучив их виды, можно сказать, что делятся они на две группы:

1) Резистивные кабели. Это кабели с постоянным сопротивлением, представляющие собой проволоку в изоляции. Проволока имеет внутреннее сопротивление и при подключении к электрической сети равномерно нагревается по всей длине. Эти кабели также делятся на два типа конструкции: одножильные и двухжильные.

Одножильный резистивный кабель имеет одну нагревательную жилу по всей длине. Он имеет нюанс, который следует учитывать при укладке, подключать питание необходимо с обоих концов.

Двужильный резистивный кабель состоит из двух проводов. Также, в нем присутствует еще и дополнительная токопроводящая жила. Питание подается с одной стороны, с другой крепится герметичная муфта, отсюда следует, что подключать можно с одной стороны, что упрощает монтаж.

Если выбирать между двумя данными группами кабелей, то у каждого есть свои положительные стороны. Так, одножильный кабель стоит гораздо дешевле, но двужильный легко устанавливается.

Но в свою очередь, все зависит от удельной тепловой емкости и параметров обогреваемой зоны. Для больших площадей лучше всего устанавливать двужильный кабель. А в системах с небольшим энергопотреблением могут быть использованы одножильные греющие кабели.

На рисунке 3 представлено строение данных типов кабелей.



Рисунок 3– Строение одножильного и двухжильного кабеля

2) Саморегулирующиеся кабели. Они изменяют отдаваемую ими мощность в зависимости от температуры окружающей среды. При повышении температуры среды мощность, которая выделяется кабелем, уменьшается, а при понижении температуры – увеличивается. Это происходит из-за конструкции саморегулирующегося кабеля: между двумя параллельными токонесущими жилами находится композитный материал, состоящий из полимера с вкраплениями токопроводящего материала. При повышении температуры полимер расширяется, отсюда, связи между этими вкраплениями нарушаются, что эквивалентно увеличению омического сопротивления. Ток падает и мощность, которая выделяется кабелем, уменьшается. Обратный процесс происходит при понижении температуры окружающей среды. Стоит отметить, что изменение мощности происходит на каждом сантиметре кабеля, а не на всей длине. Строение саморегулирующегося кабеля представлено на рисунке 4.



Рисунок 4 – Строение саморегулирующегося кабеля

Достоинства и недостатки приведенных выше типов нагревательных кабелей представлены в таблице 2.

Таблица 2 – Достоинства и недостатки греющих кабелей

	Резистивный кабель	Саморегулирующийся кабель
Стоимость	Стоимость ниже, чем у саморегулирующегося	Высокая стоимость кабеля
Экономия при эксплуатации	Греет по всей длине с одинаковой мощностью, отсюда следует, что высокое энергопотребление	Потребление электроэнергии на 10-15 % ниже, чем у резистивного кабеля
Длина кабеля	Обладают фиксированной длиной. Максимальная длина, обычно, 100-150 м	Длину можно изменить, нарезать на куски и т.д
Стойкость к скачкам напряжения	Могут перегореть в случае повышения напряжения в течение длительного периода	Более стойкие, нежели резистивные кабели
Срок службы	15 лет и более, но при хороших условиях	Длительный (срок службы матрицы 10-15 лет)
Безопасность	При скоплении грязи на участке или перекрещивании двух проводов происходит перегрев	Безопаснее, нежели резистивный кабель, защищен от перегрева

Выбор греющего кабеля зависит от заказчика, а также от того, какие цели он преследует. В таблице 3 приведем пример двух типов кабелей.

Таблица 3 – Сравнение двух типов кабелей

	SPYHEAT CD-20	SLL20
Тип кабеля	Резистивный	Саморегулирующийся
Количество жил	2	2
Удельное тепловыделение	20 Вт/п.м	20 Вт/п.м
Номинальное напряжение	220 В	220 В
Стоимость	8284 рублей за 100 метров	28119 рублей за 100 метров

Из таблиц 2 и 3 можно сделать вывод, что пользователь либо экономит на стоимости самого кабеля, либо на затратах на электроэнергию.

### 3.2 Способы обогрева поверхностей

Для оптимизации энергопотребления предлагается использовать следующие методы:

- 1) зональный обогрев,
- 2) учет текущего времени,
- 3) учет энергопотребления,
- 4) дистанционное управление системой.

Итак, рассмотрим каждый пункт отдельно. Особенностью системы является разбиение греющих элементов на зоны. Причем, греющие кабели можно расположить, как в шахматном порядке, так и в виде синусоиды и косинусоиды. Вследствие этого, нагревать можно как всю поверхность, так и отдельно по зонам. Таким образом, оптимизируются расходы электроэнергии, и повышается срок эксплуатации греющих кабелей. На рисунке 5 представлены зоны для тротуара с расположением греющих кабелей в шахматном порядке для

саморегулирующегося или двухжильного резистивного кабеля и одножильного греющего элемента соответственно.

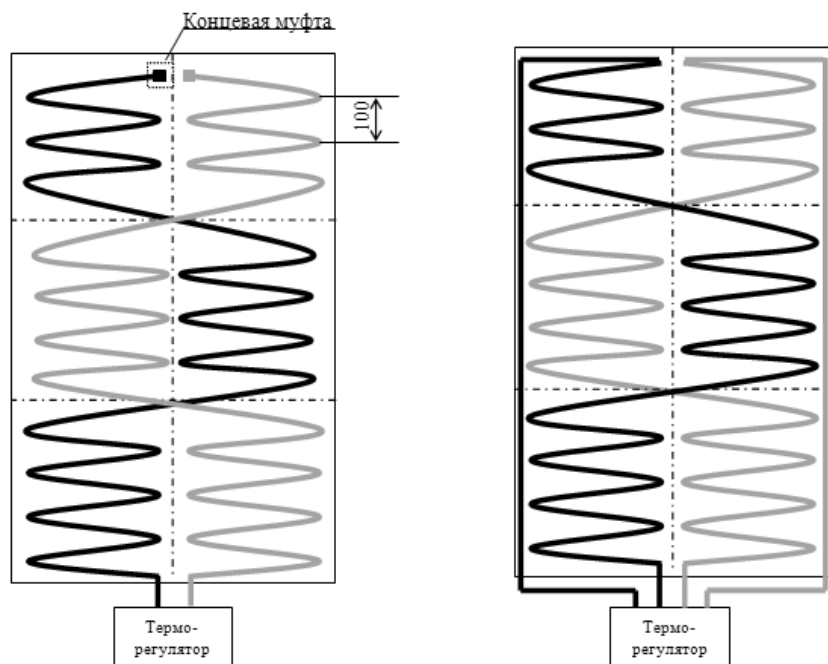


Рисунок 5 – Шахматный порядок укладки кабелей

На рисунке 6 показано схематичное расположение кабелей при укладке в виде синусоиды и косинусоиды для саморегулирующегося или двухжильного резистивного кабеля и одножильного греющего элемента.

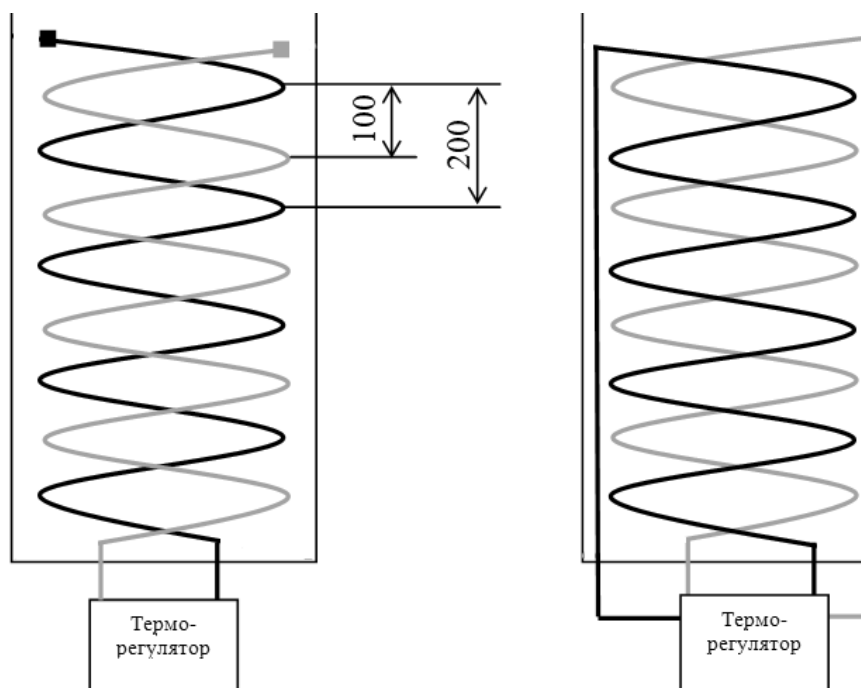


Рисунок 6 – Укладка кабеля в виде синусоиды и косинусоиды

На рисунках 7-9 изображено расположение греющих кабелей на крышах.

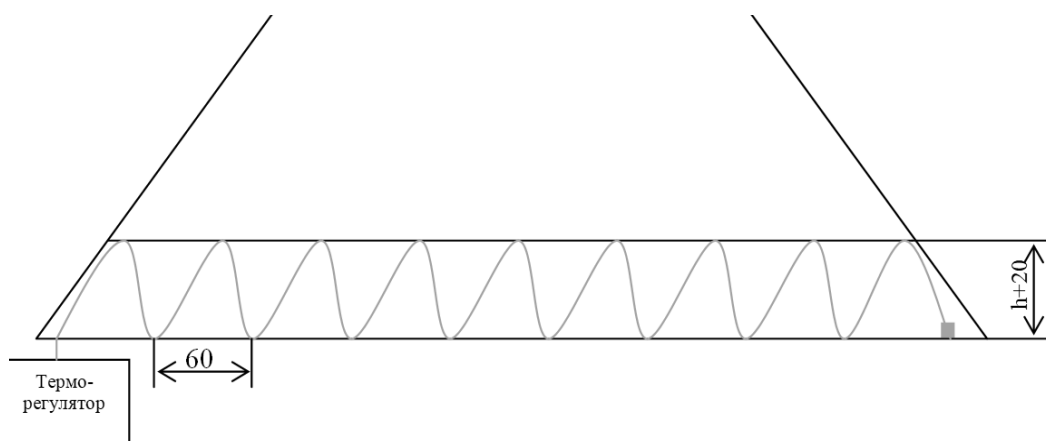


Рисунок 7 – Расположение двухжильного кабеля на крыше

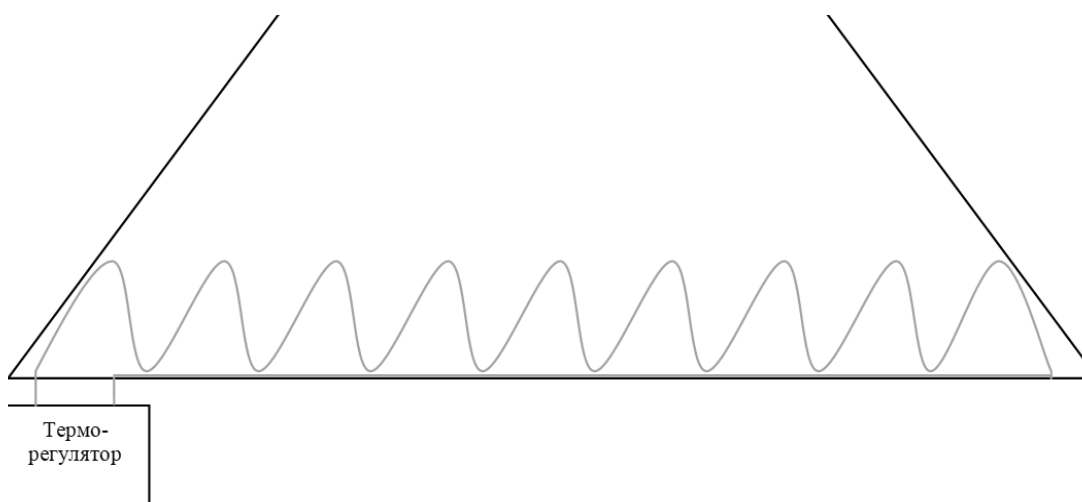


Рисунок 8 – Расположение одножильного кабеля на крыше



Рисунок 9 – Расположение кабеля на крыше. Вид с торца

Ориентировочная мощность нагревательного кабеля для обогрева открытых площадок, пешеходных переходов и тротуаров составляет 200 Вт/м<sup>2</sup>. Шаг укладки одного нагревательного кабеля для тротуара примем 10 см. Шаг укладки между первым и вторым кабелем также, 10 см. Для уменьшения затрат на потребление электроэнергии рекомендуется уменьшить зону, которая подлежит обогреву, вместо всей поверхности обогревать только колею.

Для крыши, в свою очередь, зонального обогрева нет, т.к. обогревать нужно только нижнюю часть крыши, где максимально сконцентрировано скопление снега. А также, невозможно на крыше уложить греющие кабели таким образом, чтобы они не соприкасались (при соприкосновении между собой, может возникнуть короткое замыкание и система выйдет из строя). Отсюда вывод, что кабель для крыш следует укладывать в один ряд по всей длине крыши. Высота укладки кабеля (h) – величина, которая равна длине ската крыши от стены до края по плоскости кровли (область, где наиболее вероятно образование наледи и скопление снега плюс 20 см). Шаг укладки нагревательного кабеля - величина для большинства типов кровли равна 60 см.

Учет текущего времени позволит нам рассчитать, когда включать зоны по одной, а когда все обе сразу. Ниже, в таблице 4 приведен тариф на электроэнергию по Челябинской области с 1 января 2018г., дифференцированный по трем зонам суток для проживающих в домах с газовыми плитами:

Таблица 4 – Тариф на электроэнергию

Пиковая зона (с 7 до 10 и с 17 до 21 часов)	3,94 руб за кВт.ч
Полупиковая зона (с 10 до 17 и с 21 до 23 часов)	3,03 руб за кВт.ч
Ночная зона (с 23 до 7 часов)	1,82 руб за кВт.ч

Из таблицы 4 можно сделать вывод, что в ночное время самый минимальный стоимостной тариф, а, следовательно, с 23 до 7 часов можно включать обе зоны для обогрева поверхности. В последующее же время, зоны будут меняться, т.е с 7

до 14 часов – нагрев зоны №1, с 14 до 23 часов – нагрев зоны №2. Также, можно заметить, что среднесуточная цена электроэнергии равна 2,93 руб за кВт.ч

С помощью датчика тока система будет фиксировать проходящий ток через цепь, рассчитывать мощность и отправлять ее на веб-сервис, где в свою очередь, будет уже подсчитана стоимость потраченной электроэнергии. Так же, пользователь сможет управлять системой в принудительно-ручном режиме, если он не хочет переплачивать деньги за то время, когда он будет отсутствовать дома, и не будет пользоваться результатом работы системы.

### 3.3 Алгоритм работы системы

Для работы системы в автоматическом, или как еще можно сказать, в интеллектуальном режиме, необходимо разработать такой алгоритм, чтобы на определенные входные параметры был соответствующий и правильный отклик.

Алгоритм работы системы представлен на рисунке 10.

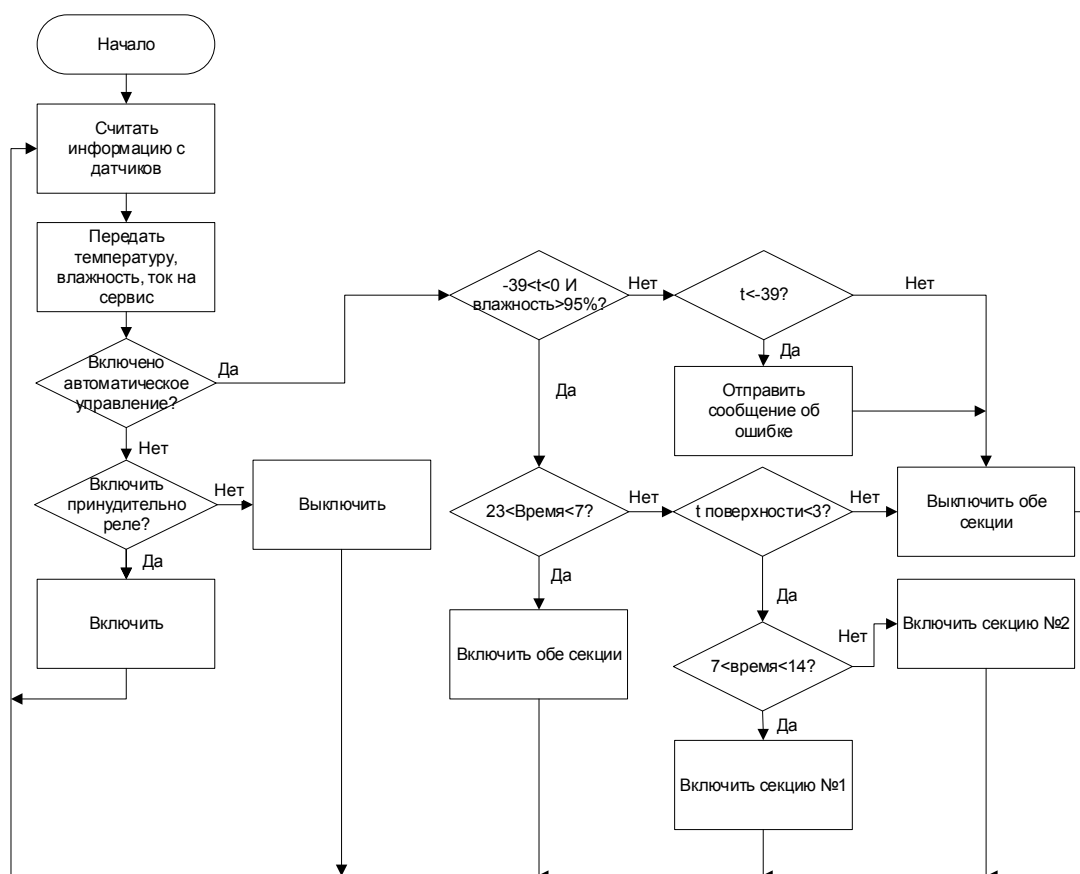


Рисунок 10 – Алгоритм работы системы без исключительных ситуаций



Код программы системы представлен в приложении А.

Алгоритм работы в общем виде выглядит следующим образом: сначала система будет «пробегаться» по штатному алгоритму, а затем, проверять исключительную ситуацию с помощью искусственного интеллекта и формировать выходное воздействие. В исключительных ситуациях упор делается на оттепель, когда в дневные часы температура воздуха повышается до положительных значений, а зимой опускается до отрицательных.

### 3.4 Управление системой с помощью искусственного интеллекта

Один из способов реализации искусственного интеллекта – это нейронная сеть. Она моделирует реакцию человеческого мозга, ее особенностью является способность к самообучению основываясь на предыдущем опыте [4]. Таким образом, система с каждым разом совершает все меньше и меньше ошибок.

В данной работе, нейросеть применяется для «предсказания», или, как еще можно сказать, для прогнозирования будущего поведения [5]. Суть предсказания заключается в следующем: если в утренние/вечерние часы температура воздуха варьируется от 0 до 4 градусов, то есть вероятность, что она резко опустится еще ниже, а, следовательно, чтобы не застыло то, что, возможно, таяло в течение дня, нужно убрать, а для этого нужно заранее включить нагревательные элементы.

Заложен следующий принцип работы:

- 1) Создается обобщающая выборка в txt документе.
- 2) Сеть обучается.
- 3) После обучения создается файл для контроллера.
- 4) Файл записывается на SD карту, которая устанавливается в контроллер и загружает параметры сети.
- 5) Формируется «вектор данных» по показаниям с датчиков, который далее подается на вход сети. Вектор данных – одномерный массив определенного размера, в котором важен порядок расположения элементов. Элементы этого

массива – числа. На выходе также, одномерный массив, где нулевой элемент – включение, первый элемент – выключение.

б) После подсчета и преобразования определяется выходной вектор сети и в зависимости от результата определяются и выполняются команды (вкл/ выкл).

Нейронная сеть состоит из множества нейронов, а он, в свою очередь, из значений, поступающих на вход и весов, которые определяют, насколько соответствующий вход нейрона влияет на его состояние. Поступившие на входы сигналы умножаются на свои веса, а далее, все произведения подаются на сумматор. Результатом работы сумматора является число, или как по-другому его называют, взвешенная сумма. Для формирования адекватного выходного сигнала, нейрон обрабатывает взвешенную сумму, используя функцию активации. Функция активации преобразует взвешенную сумму в какое-либо число, которое и будет являться выходом нейрона [6, 7]. В качестве активационной функции, как правило, используется сигмоида. Ее вид выражается формулой (3.1):

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.1)$$

Производная:

$$f(x)' = f(x) \cdot (1 - f(x)) \quad (3.2)$$

Таким образом, неактивированное состояние нейрона  $j$  для образа  $p$  записывается в виде взвешенной суммы:

$$s_{pj} = \sum_j w_{ij} y_{pi}, \quad (3.3)$$

где  $s_{pj}$  – взвешенная сумма выходов связанных нейронов предыдущего слоя на вес связи, или, неактивное состояние нейрона  $j$  для образа  $p$ ;

$w_{ij}$  – вес связи между  $i$  и  $j$  нейронами;

$y_{pi}$  – активированное состояние нейрона  $i$  предыдущего слоя для образа  $p$ .

Активационное состояние нейрона вычисляется по формуле (3.4).

$$y_{pj} = f_j(s_{pj}), \quad (3.4)$$

где  $y_{pj}$  – активационное состояние нейрона  $j$  для образа  $p$ ;

$f_j$  – функция активации;

$s_{pj}$  – неактивированное состояние нейрона  $j$  для образа  $p$ .

На рисунке 11 представлена структура искусственной нейронной сети (далее ИНС). Данная сеть относится к многослойным нейросетям, т.к имеет скрытые слои, а конкретнее, к двухслойным [8].

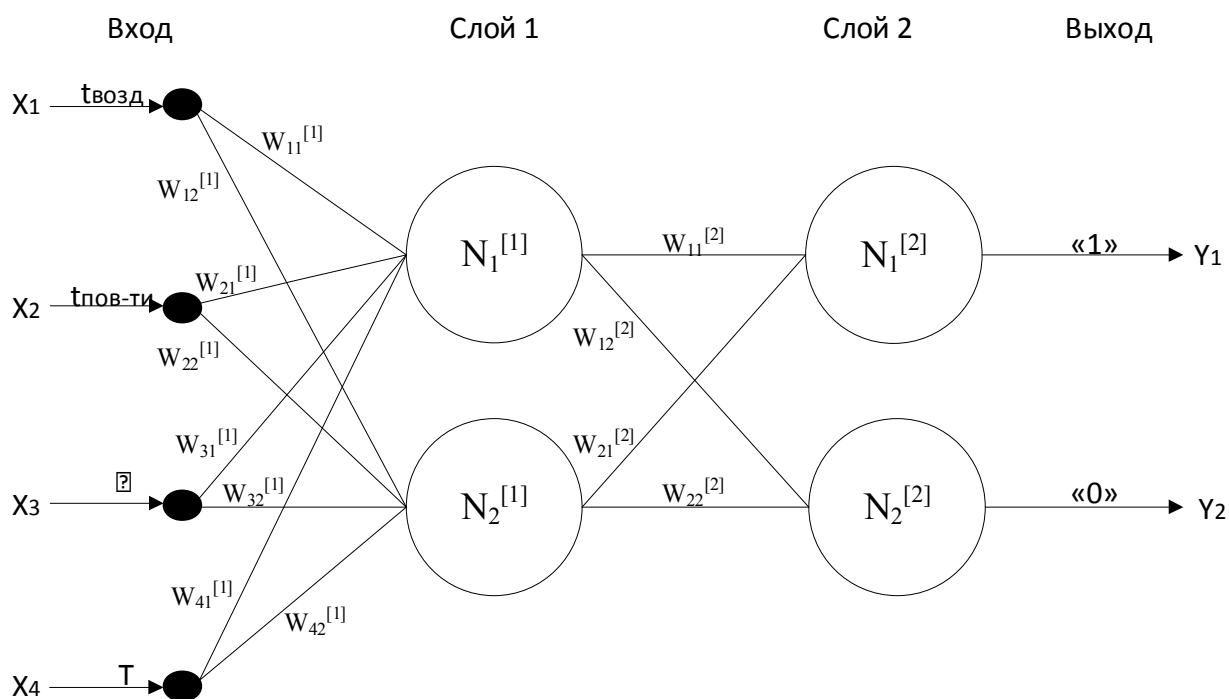


Рисунок 11 – Структура ИНС

Обозначения на рисунке \_\_:

$W_{ij}^{[i]}$  – массивы весовых коэффициентов  $i$ -го слоя;

$N_1^{[i]}$ - $N_2^{[i]}$  – нейроны  $i$ -го слоя.

Входными параметрами являются:

$x_1 - t_{\text{возд}}$  – температура воздуха;

$x_2 - t_{\text{пов-ти}}$  – температура поверхности;

$x_3 - \varphi$  – влажность воздуха;

$x_4 - T$  – текущее время измерения параметров.

Выходные сигналы:

$Y_1$  – «1» – включение нагревательного элемента.

$Y_2$  – «0» – выключение нагревательного элемента,

Для того чтобы нейронная сеть обладала способностью выдавать правильные результаты в ответ на входные данные, необходимо ее обучить [9]. Под обучением подразумевается поиск такого набора весовых коэффициентов, при котором входной сигнал после прохода по сети преобразуется в желательный нам выходной.

В качестве алгоритма обучения сети выбран метод обратного распространения ошибки. Он считается основой основ, хоть и имеет недостаток – неопределенно долгий процесс обучения [10]. Но, т.к в данном случае задача сети не сложная, следовательно, данный недостаток не является критическим. Этот метод определяет два «потока» в сети. Входные сигналы двигаются в прямом направлении, в результате чего получается выходной сигнал, из которого затем мы приобретаем значение ошибки. Величина ошибки двигается в обратном направлении, в результате происходит корректировка весовых коэффициентов связей сети [11, 12]. Данный метод относится к градиентным алгоритмам. Идея этих алгоритмов состоит в уменьшении предыдущего значения веса на величину производной от меры погрешности, умноженную на некоторый коэффициент. Процесс длится до тех пор, пока погрешность на выходе не будет установленной минимальной величины. Алгоритм обучения сети представлен на рисунке 12 в виде блок-схемы [13].

Величина ошибки определяется по формуле:

$$E_p = \frac{1}{2} \sum_j (t_{pj} - y_{pj})^2, \quad (3.5)$$

где  $E_p$  – величина функции ошибки для образа  $p$ ;

$t_{pj}$  – желаемый выход нейрона  $j$  для образа  $p$ ;

$y_{pj}$  – активированный выход нейрона  $j$  для образа  $p$ .

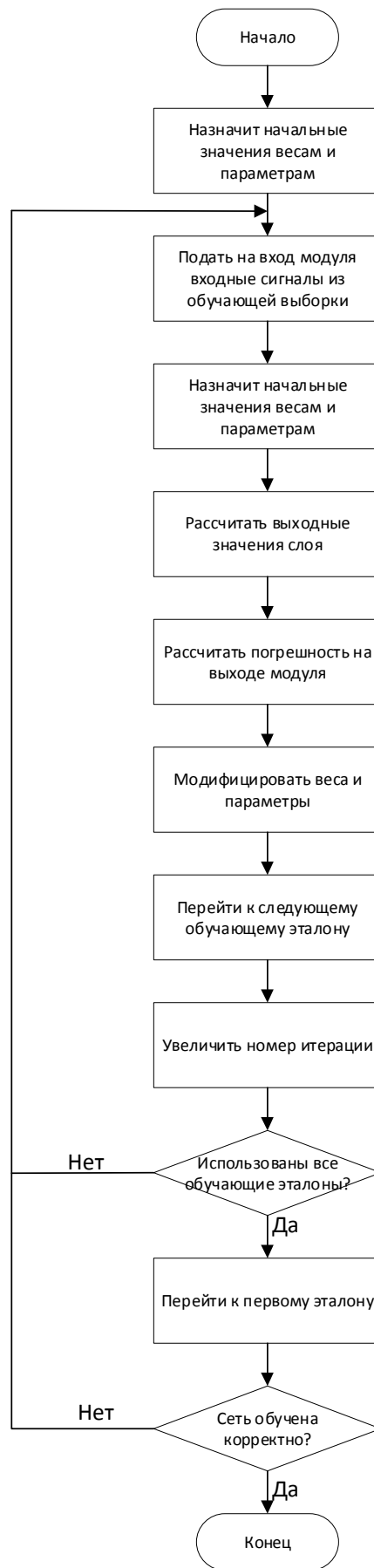


Рисунок 12 – Алгоритм обучения сети

Градиент функции потерь представляет собой вектор частных производных, который вычисляется по формуле:

$$\nabla E(w) = \left[ \frac{dE}{dw_1}, \dots, \frac{dE}{dw_n} \right], \quad (3.6)$$

где  $\nabla E(w)$  – градиент функции потерь от матрицы весов;

$\frac{dE}{dw_n}$  – частная производная функции ошибки по весу нейрона;

$n$  – общее количество весов сети.

Производная функции по конкретному образу рассчитывается по следующей формуле:

$$\frac{dE}{dw_{ij}} = \frac{dE}{dy_j} \times \frac{dy_j}{ds_j} \times \frac{ds_j}{dw_{ij}}, \quad (3.7)$$

где  $\frac{dE}{dw_{ij}}$  – значение производной функции ошибки по весу  $w_{ij}$ , между  $i$  и  $j$

нейронами;

$\frac{dE}{dy_j}$  – ошибка нейрона  $j$ ;

$\frac{dy_j}{ds_j}$  – значение производной функции активации по ее аргументу для нейрона

$j$ ;

$\frac{ds_j}{dw_{ij}}$  – выход  $i$  нейрона предыдущего слоя (относительно к нейрону  $j$ ).

Для выходного слоя ошибка рассчитывается по формуле:

$$\delta = t - y, \quad (3.8)$$

где  $\delta$  – ошибка выходного слоя;

$t$  – желаемое значение выхода;

$y$  – полученное значение на выходе.

Для скрытого слоя ошибка рассчитывается по формуле:

$$\delta_i = \frac{dy_j}{ds_j} \times \sum_j \delta_j \times w_{ij}, \quad (3.9)$$

$\delta_i, \delta_j$  – ошибка нейрона  $i$ -го скрытого и  $j$ -го следующего слоя соответственно;  
 $w_{ij}$  – вес связи между нейроном  $i$ -го текущего (скрытого) слоя и нейроном  $j$ -го выходного или тоже скрытого слоя.

Программный код обучения нейронной сети написан на языке C# и представлен в приложении Б.

### 3.5 Моделирование нейронной сети

Т.к области применения нейронных сетей в последнее время стремительно расширяются, в связи с этим появилась потребность в программных и аппаратных продуктах для реализации вычислений, специализирующихся на нейронных вычислениях. Стоит отметить, что большинство реализаций нейронных сетей представляет собой программные решения. Самым распространенным программным пакетом для изучения и моделирования нейронной сети является Neural Network Toolbox в интерактивной среде разработки алгоритмов и современного инструмента анализа данных MATLAB фирмы Mathworks. Данный пакет обеспечивает поддержку нейросетевых парадигм, а также, имеет открытую модельную архитектуру.

Для обучения составлена выборка из 1300 различных, неповторяющихся показателей для каждого датчика. Результат обучения представлен в окне Neural Network Training.

Алгоритм работы нейронной сети является итеративным, его шаги называют эпохами или циклами. Эпоха – это одна итерация в процессе обучения, которая включает предъявление всех примеров из обучающего множества и, возможно, проверку качества обучения на контрольном множестве [14].

На рисунке 13 можно заметить, что просмотр всех примеров обучающей выборки с одновременной коррекцией весов сети происходит 379 итераций

(эпох). Также, значение Mean Squared Error (MSE) – среднеквадратичной ошибки, которая используется для оценки функционирования сети, минимально.

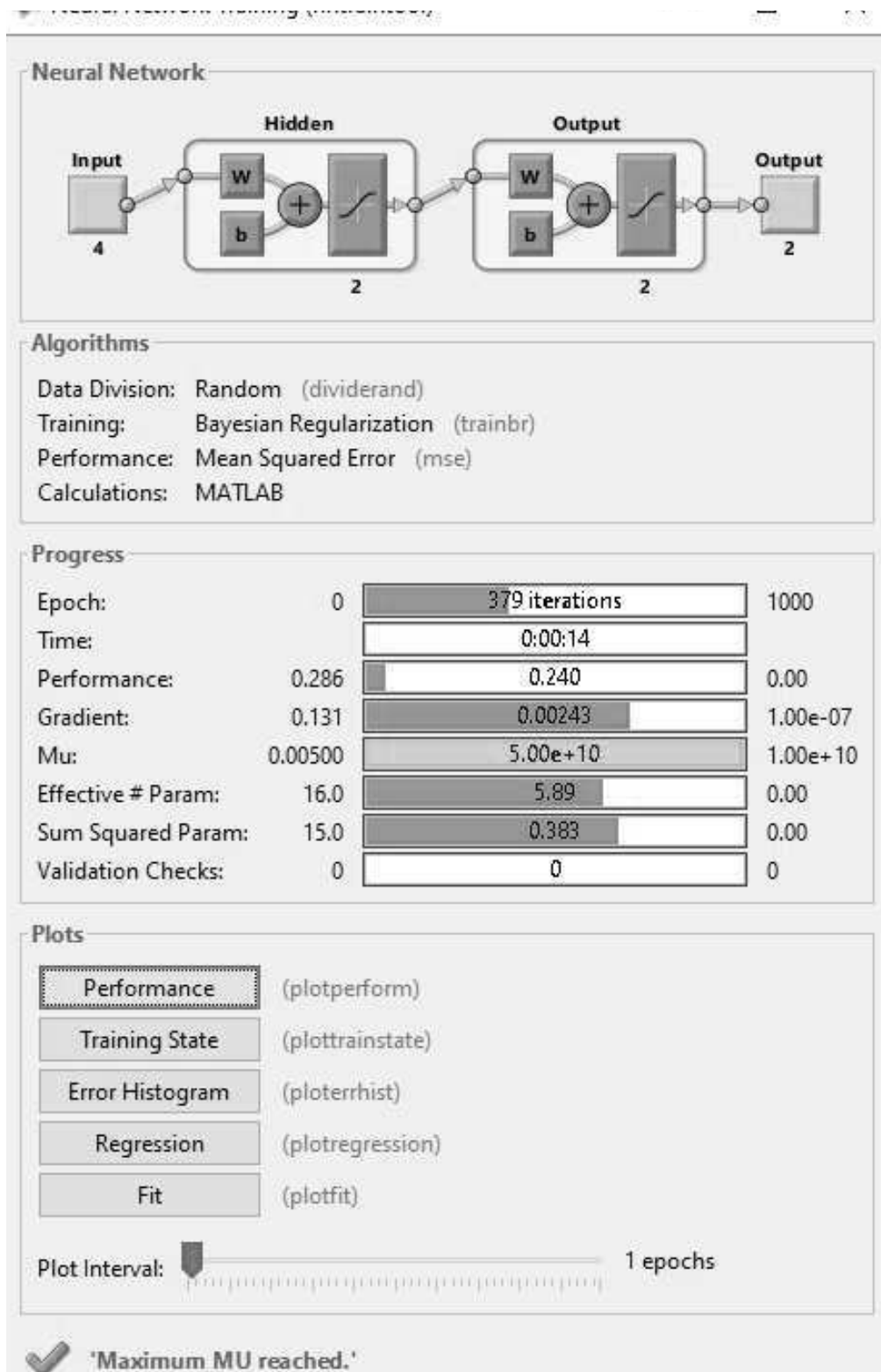


Рисунок 13 – Обучение нейронной сети



Качество обучения сети отображается графиком. Из рисунка 14 видно, что к концу процесса обучения ошибка становится очень малой.

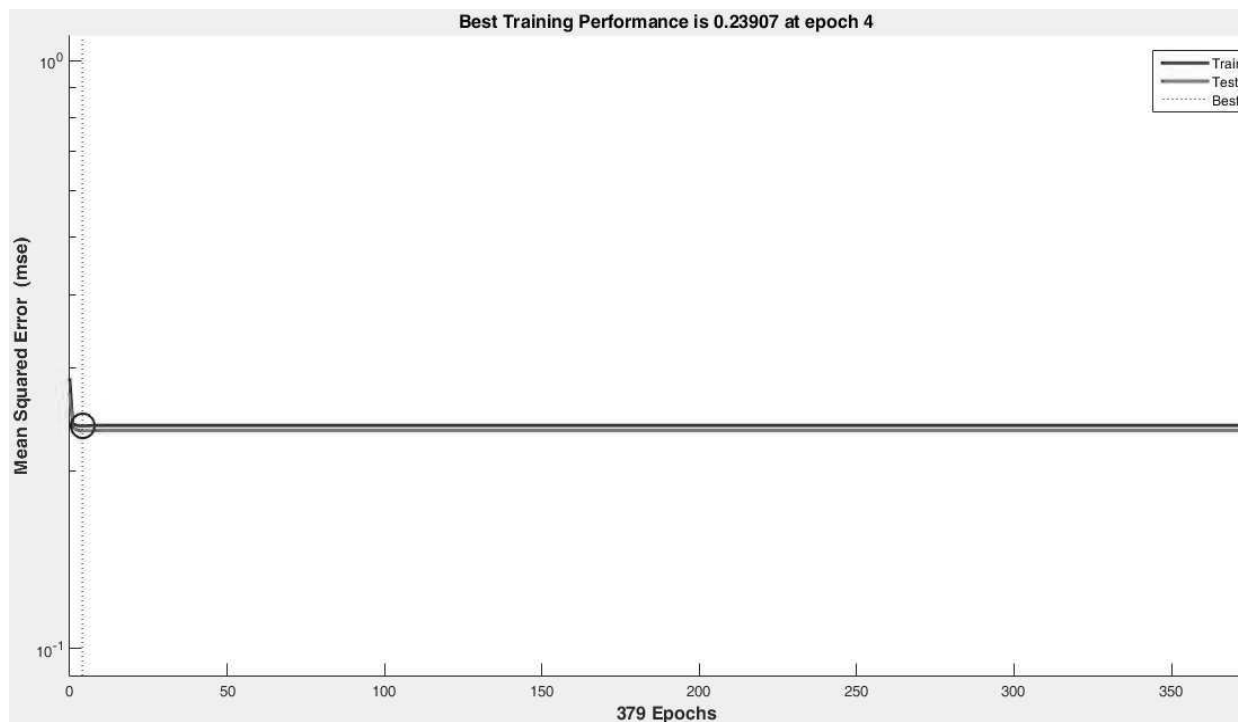


Рисунок 14 – Изменение ошибки в процессе обучения

Установка значений весовых коэффициентов происходит автоматически и случайным образом, но при этом, для усовершенствования работы их в любой момент можно изменить. Во время исследования было установлено, что нейросеть желательно обучать не более трех раз. Т.к слишком частое обучение приведет к переобучению сети и, следовательно, к неверным результатам при проверке различных тестовых выборок.

После завершения обучения сеть подвергается тестированию. Для проверки эффективности сети было выбрано 194 признаков вектора для проверки. В идеале, на выходе должен появляться числовой результат, схожий с целевым вектором. Вектор цели представляет собой ряд чисел, которые принимают два значения: 1 – включение, 0 – выключение.

## 4 КОНСТРУИРОВАНИЕ СИСТЕМЫ ОБОГРЕВА

### 4.1 Разработка системы

Система профилактики образования наледи – это антиобледенительная система, которая включает в себя: терморегулятор, датчики параметров окружающей среды, сервис с веб-интерфейсом и нагревательные элементы, которые будут препятствовать образованию наледи, и вследствие этого, продлевать эксплуатационные характеристики поверхности, а также, обеспечивать безопасность гражданам. Терморегулятор используется для управления нагревом и состоит из часов реального времени, датчика тока, модуля подключения к сети Интернет для взаимодействия с сервисом, реле и управляющего контроллера, в который заложен разработанный алгоритм управления.

Так же, на сервисе с веб-интерфейсом, можно контролировать состояние погоды, потребление энергии, и управлять системой, выбрав ручное или автоматизированное управление. Если в атмосфере создаются условия, которые могут стать причиной образования льда, программируемый терморегулятор «разрешает» подачу электроэнергии, и от греющего кабеля начинает выделяться тепло. Либо, если пользователь вручную включит/отключит нагревательные элементы с помощью веб-сервиса. Система должна работать до тех пор, пока существует вероятность образования льда, сосулек и т.д.

Построим объектную модель системы, по-другому ее еще называют диаграммой вариантов использования (use case diagram), в которой отражается прагматика разрабатываемой системы, описана структура объектов, из которых она состоит, их взаимосвязи и атрибуты [15].

Схема объектной модели системы представлена на рисунке 15. Соответственно, обозначения в ней: include – разновидность отношения базового варианта использования с другим вариантом использования, extended – определяет взаимосвязь между базовым вариантом использования и его

специальным случаем. Роли распределены между погодой и сервисом, в овалы, в свою очередь, заключены действия [16].



Рисунок 15 – Объектная модель системы

Разработка системы – трудный процесс, требующий представления последовательных действий. Понимая ее логическую структуру можно перейти к реализации ее физической части. Диаграмма деятельности – это логическая модель. Она дает нам представление, что должна делать система, и показывает жизненный цикл, поэтапность действий, которые необходимы для достижения цели с переходами из одного состояния в другое.

Обозначения на диаграмме:

● - Начальное состояние.

⦿ - Конечное состояние.

▭ - Действие.

◇ - Символ принятия решения с обозначениями условий возле переходов.

Для обозначения связи между операциями используется термин ребро (обычная стрелка).

Диаграмма деятельности отличается от традиционной блок-схемы более высоким уровнем абстракции и возможностью представления с помощью диаграмм деятельности управления параллельными потоками наряду с последовательным управлением [17].

На рисунке 16 представлена укрупненная схема диаграммы деятельности. Где  $t$  – температура воздуха,  $\phi$  – влажность воздуха,  $T$  – время на данный момент,  $t_{п}$  – температура поверхности.

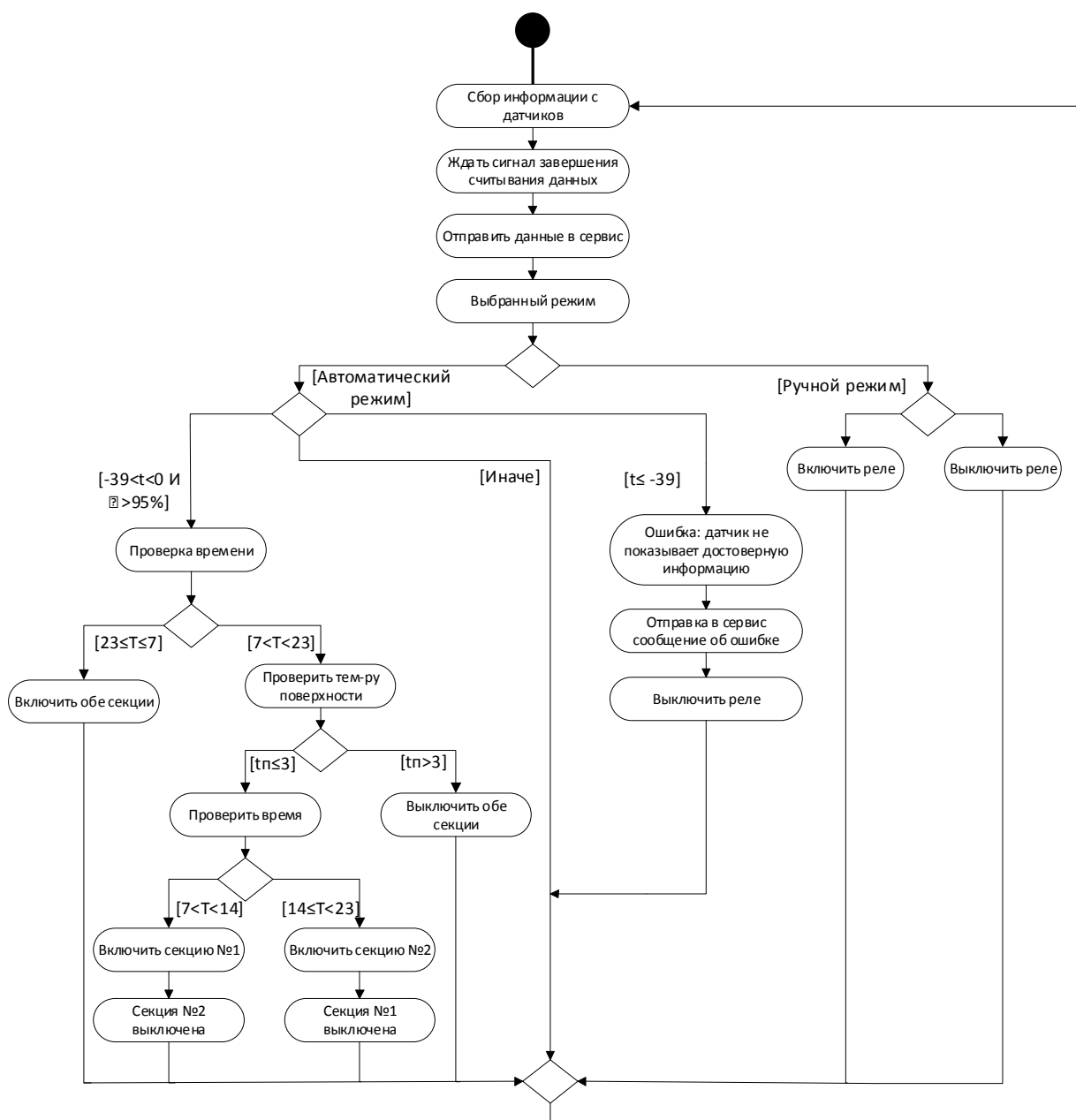


Рисунок 16 – Укрупненная диаграмма деятельности

Некоторые действия в диаграмме деятельности могут представлять собой отдельный, более раскрытый алгоритм.

На рисунках 17-18 представлены диаграммы деятельности для инициализации системы, и для отправки данных в сервис, соответственно.

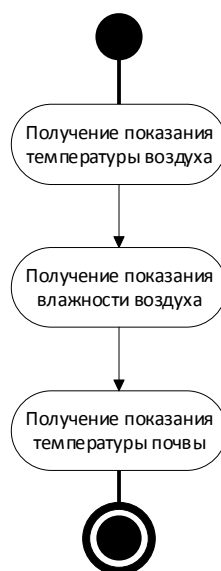


Рисунок 17 – Инициализация системы

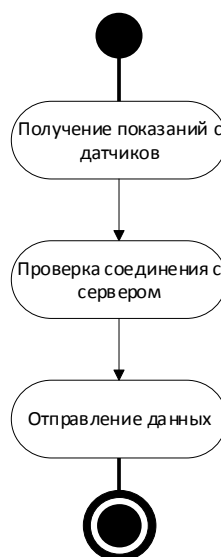


Рисунок 18 – Отправка данных в сервис

Система будет работать следующим образом: сначала входные данные проходят основной алгоритм работы системы, с помощью которого будет определено выходное воздействие, а затем по нейронной сети выявляются

исключительные ситуации и выбирается выход. Передача данных на сервис с веб-интерфейсом будет происходить каждую минуту, вследствие формирования непродолжительного ожидания отклика от самого сервиса.

Итак, изобразив процесс и действия, которые будут выполняться проектируемой системой, далее следует определиться с ее архитектурой и компонентами.

## 4.2 Выбор объекта управления

Контроллер – устройство управления электроникой с набором входов и выходов, который работает по написанной и загруженной в него программе. Контроллер обязательно присутствует в таких предметах, как в компьютерной мыши, телефоне, пульте, можно сказать, что практически в любом современном электронном устройстве и является его «мозгом». Контроллер - вещь сама по себе универсальная. Ко входам можно подключить как обычные кнопки (пульт), так и температурные датчики (кондиционер), модули беспроводной связи (телефон, интернет) и др.. Выходы могут управлять чем угодно, если на него подать соответствующий сигнал. Задача контроллера – измерять электрическое напряжение на входах и при изменении параметров, заданных в программе, подавать соответствующее напряжение на выходы. Для реализации данного проекта будем выбирать объект управления исполнительными механизмами между одноплатным компьютером Raspberry Pi и платой на микроконтроллере (далее МК) – Arduino. Рассмотрим каждое из устройств в деталях и выделим свои плюсы и минусы при использовании в данном проекте.

### 4.2.1 Raspberry pi

Raspberry Pi – одноплатный компьютер, то есть компьютер, который собран на одной печатной плате, на которой установлены микропроцессор, пины ввода и вывода информации и другие модули. Также, эта плата очень малого размера -

примерно 8,5\*5,5 см. Raspberry Pi имеет невысокую стоимость для своей комплектации, напоминающего мини-компьютер – средняя цена за одну плату составляет около \$25-30. Самая дешевая версия – Zero, около \$5. Продажа платформ началась в начале 2012 г., на данный момент, продажа мини-компьютеров превысила более 13млн. экземпляров Raspberry Pi.

Raspberry Pi выпускается в двух следующих версиях: A, A+, B, B+, 2B, Zero, 3B, Zero W. Сравним наиболее популярные версии и основные характеристики Raspberry Pi – A, B+, 3B, Zero, :

- 1) Процессор ARM11 – у A, B, Zero, а у 3B - ARM Cortex-A53.
- 2) Частота - 700 МГц –A, B+; 1,1 ГГц - Zero; 1,2 ГГц - 3B .
- 3) Оперативная память - 256 Мб у A, 512 Мб у B+ и Zero, 1 Гб у 3B.
- 4) USB входы/выходы – 1 порт у A и Zero, 4 порта у B+ и 3B.
- 5) Ethernet вход/выход – есть только у B+ и 3B.
- 6) Интерфейсы ввода/вывода: контакты 26 пинов только у A, в остальных версиях по 40.
- 7) Wi-Fi только у версии 3B, стандарта 802.11n.

Безусловно, Raspberry Pi является полнофункциональным компьютером т.к. обладает атрибутами настоящего компьютера: выделенным процессором, памятью и графическим драйвером для вывода через HDMI. Также, на него можно установить специальную версию операционной системы (далее ОС) Linux, следовательно, можно использовать программы для этой ОС.

Raspberry Pi для работы требуется постоянное напряжение 5V, а также, бесперебойное питание и обеспечение дополнительного оборудования, которое будет гарантировать подачу постоянного тока. Это значит, что идея работы системы, основанной на этой микроЭВМ, которая питается от батарейки – не удачная [18].

У Raspberry Pi есть огромный плюс – это встроенный Ethernet-порт, который почти не требует настройки, обеспечивает легкий доступ к сети, достаточно установить USB-адаптер для Wi-Fi и установить драйвер. После этой процедуры

можно использовать ОС для подключения к веб-сервисам, поиску в Интернете и прочее. Схема Raspberry Pi B+ представлена на рисунке 19.

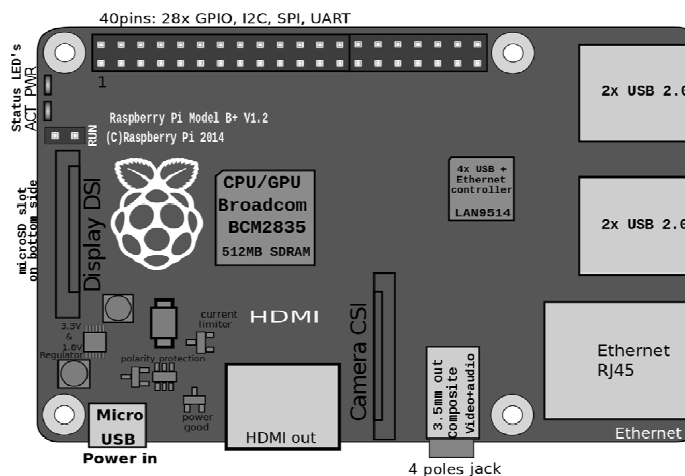


Рисунок 19 – Стандартный представитель плат Raspberry Pi B+

#### 4.2.2 Контроллер Arduino

В отличие от Raspberry Pi, платы Arduino – это микроконтроллер, а не полноценный компьютер. На них нет ОС, Arduino просто выполняет код, который был загружен программистом. Отсутствие ОС не делает этот контроллер плохим, а, наоборот, при работе нет задержек, выполнение кода протекает проще и легче, нет ошибок, которые могут быть связаны с ОС. Также, микроконтроллеры потребляют единицы-десятки мА, и скорость реакции значительно лучше благодаря имеющейся однопоточности.

Главный плюс плат Arduino – простое и легкое взаимодействие с датчиками и устройствами, поэтому Arduino отлично подходит для аппаратных проектов, где необходимо получать сигналы с датчиков и реагировать на них, либо, где требуется ручной ввод, а также, для организации взаимодействия других устройств и исполнительных механизмов, где ОС вовсе не нужна.

В качестве примера приведем основные характеристики популярной и универсальной серии контроллеров Arduino – Uno:

- 1) Микроконтроллер ATmega328.
- 2) Рабочее напряжение 5В.



- 3) Входное напряжение (рекомендуемое) 7-12 В.
- 4) Количество цифровых входов/выходов – 14 (6 из которых могут использоваться как выходы широтно-импульсной модуляции (далее ШИМ)).
- 5) Количество аналоговых входов – 6.
- 6) Постоянный ток через вход/выход – 40 мА.
- 7) Flash-память – 32 Кб, где 0,5 используется для загрузчика.
- 8) Оперативное запоминающее устройство (ОЗУ) – 2 Кб.
- 9) Electrically Erasable Programmable Read-Only Memory (EEPROM) – 1 Кб.
- 10) Тактовая частота – 16 МГц.

Питание Arduino Uno может получать как от USB подключения, так и от батарейки или обычной электрической сети. Источник определяется автоматически. Платформа работает при напряжении от 5 до 20 В. Но, при напряжении менее 7 В работа может быть неустойчивой, а напряжение более 12 В может привести к перегреву и повреждению. Отсюда следует, что рекомендуемый диапазон: 7–12 В.

Контакты для доступа к питанию:

VIN: Напряжение от внешнего источника питания. Через этот вывод можно подавать внешнее питание и потреблять ток, если к устройству подключён внешний адаптер. Если подключение через универсальную последовательную шину (далее USB), то напряжение будет 5 В.

5V: Предоставляет 5 В вне зависимости от поданного напряжения. Работает процессор на этом напряжении, а максимально допустимый ток, получаемый с этого контакта – 800 мА.

3.3V: Предоставляет 3,3 В от стабилизатора платы. Максимальный ток вывода — 50 мА.

GND: Вывод земли.

IOREF: Вывод предоставляет платам расширения информацию о рабочем напряжении микроконтроллера.

Выполнять код Arduino начинает сразу, после включения и завершает работу мгновенно, если плату отключить от источника питания. Для расширения

функциональных возможностей устройства, нужно подключить периферию к контактам самой платы, либо же к платам расширения. Для Arduino существует множество модулей, которые предназначены для различных специфических функций и задач. Она легко взаимодействует с сенсорами, датчиками, модулями и шилдами. Например, Arduino, с помощью реле, может управлять нагрузками, которые питаются от 220 В, или же, с помощью герметичных датчиков измерять уровень воды.

Огромный минус у Arduino, в отличие от Raspberry Pi, в том, что она без дополнительных модификаций не способна работать по сети интернет. Чтобы создать надежное соединение, необходимо подключить соответствующие модули к плате. Как было сказано выше, Arduino совместима с множеством разных совместимых модулей. Таким образом, обеспечить систему интернетом будет не трудной задачей.

#### 4.2.3 Обоснование использования контроллера

Так как задача проектируемой системы четко поставлена и определена, следовательно, проще выбрать контроллер и периферийные устройства. Платформа Arduino проста в обслуживании, программировании, не требовательна в питании, а также, не капризна к подключению внешних устройств. На рынке плат, Arduino огромное множество и они различных размеров, поэтому проектировать можно как масштабные системы, где требуется подключать много периферии, так и в мини-проектах. В понятной и простой среде разработки Arduino Integrated Development Environment (далее IDE) можно написать код, который позволит использовать устройство, не вмешиваясь в его работу, а также, работая автономно.

Теперь приведем основные аргументы в пользу Arduino по сравнению с Raspberry Pi. Основные плюсы одноплатного контроллера:

- 1) высокая производительность,
- 2) удобство работы с интернетом, но сложность подключения модуля,

- 3) простой и понятный язык программирования,
- 4) доступность периферийных устройств,
- 5) дешевизна.

Недостатком является скорость реакции в быстродействующих проектах. Но нас этот недостаток не смущает, потому что быстродействие в данном проекте не требуется.

Из преимуществ данного контроллера потребуется удобство работы с данными с интернет протоколами, и легкость работы с датчиками и внешними устройствами. К тому же, как говорилось ранее, сотни мА, которые Raspberry Pi потребляет при своей работе на простое функционирование, использование микроЭВМ сходит на нет. Кроме того, для взаимодействия Raspberry Pi с датчиками и модулями, нужно использовать специальное программное обеспечение.

В отличие от Raspberry Pi, Arduino не подвержен хакерским атакам, т.к. выполняется только заложенная в него программа, и, принимаемая информация, содержащая тело вируса никак не сможет его перепрограммировать. Из-за использования в проекте интернет технологии, этот критерий является очень важным.

Опираясь на документацию, было рассмотрено расположение элементов на плате arduino mega (рисунок 20).

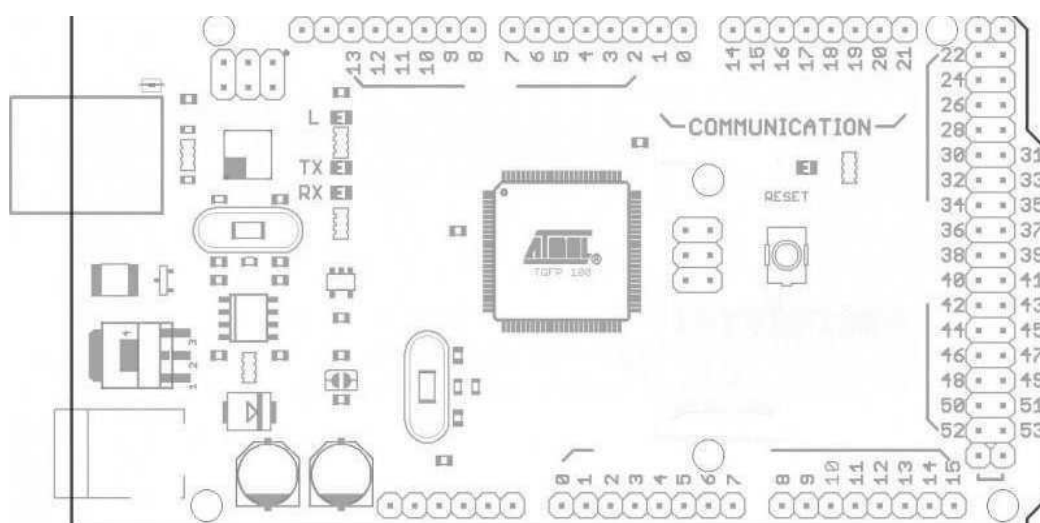


Рисунок 20 – Схема расположения элементов на плате Arduino Mega

#### 4.2.4 Выбор модификации Arduino

Так как в предыдущем пункте мы сделали акцент на выборе Arduino, как на главном элементе разрабатываемой системы, далее следует выбрать модификацию Arduino. Сделать выбор придется между Arduino Uno и Arduino Mega. Сравним их.

Различие заключается не только в размере плат, но и в характеристиках. В п.4.2.2 приведены основные характеристики Arduino Uno, теперь приведем характеристики для Arduino Mega. У него более мощный микроконтроллер - ATmega2560. У Arduino Mega 256КБ постоянной, 4 Кб EEPROM-памяти и 8 Кб оперативной, намного больше портов, их всего 60, из них 16 аналоговых и 15 с ШИМ. Размер платы: 101x53 мм [19]. Итак, отсюда можно сделать вывод, что этот контроллер отлично подходит для систем, где требуется огромное количество подключений различных датчиков и модулей, а также, для проектов, в которых потребуется «продвинутый» чип, выполняющий мощные инструкции одного цикла.

Питание, распределение напряжений и принципы взаимодействия аналогичны базовой модели – Arduino Uno.

#### 4.3 Функциональный состав

Рассмотрим характеристики и возможности датчиков и модулей, которые используются для проектирования терморегулятора системы антиобледенения.

Функциональная схема взаимодействия элементов системы представлена на рисунке 21.

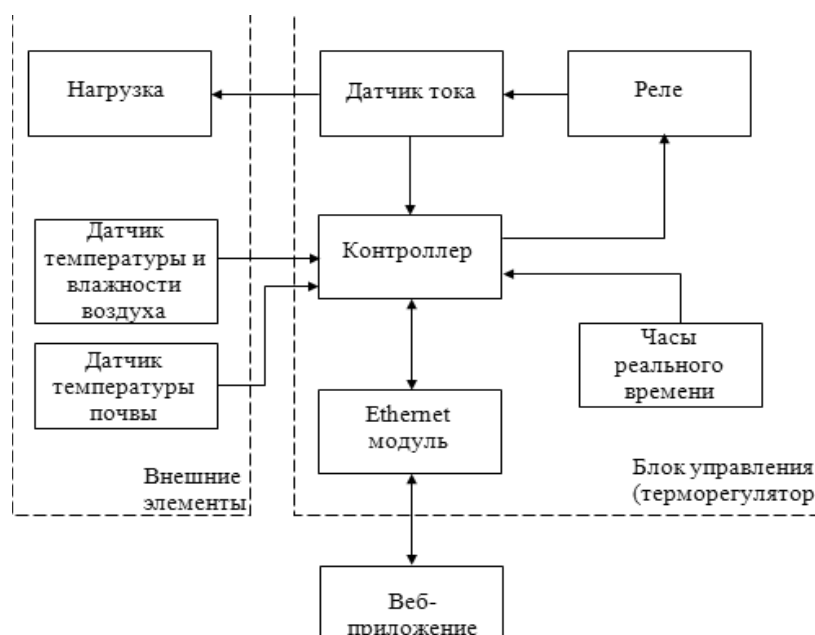


Рисунок 21 – Функциональная схема

На контроллер поступают данные с часов реального времени, датчика температуры и влажности воздуха, датчика температуры почвы. Посредством Ethernet модуля, контроллер взаимодействует с веб приложением, как на отправку данных, так и на прием. С помощью алгоритма, контроллер вычисляет ситуацию ледообразования, если условия неблагоприятные, то на силовое реле подается сигнал о его включении, и по цепи к нагрузке (т.е греющим кабелям) протекает ток, который, в свою очередь, фиксируется датчиком тока. Датчик тока передает значение контроллеру, где далее, от показателя тока, высчитывается количество потребленной мощности.

#### 4.3.1 Датчик температуры и влажности DHT22

Датчик DHT22 – это простой и не дорогой модуль для измерения температуры воздуха и относительной влажности. Имеет более точный диапазон измерений, нежели DHT11. Датчик состоит из чувствительного емкостного датчика и термистора, а также 8-ми битного чипа, который преобразует аналоговый сигнал на входе к датчику, в цифровой на его выходе.

Преимущества данного датчика, это небольшой размер, высокая дальность передачи данных, низкое энергопотребление, и, высокая точность измерений, а в проектируемой системе – широкий диапазон измерения температуры, вплоть до минус 40 градусов. Недостаток же, в свою очередь – высокая задержка 2 секунды, которая используется для конвертации [20].

Все элементы заключены в пластиковый корпус, где спереди расположены отверстия, а сзади – гладкая поверхность. Для крепления датчика предусмотрено отверстие, размером 3мм. Габариты датчика представлены на рисунке 22.

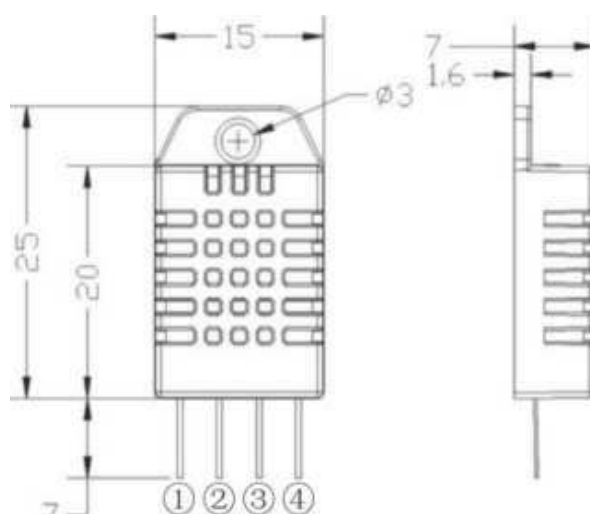


Рисунок 22 – Датчик DHT22

Для подключения снизу существует 4 коннектора, с шагом 2,54 мм:

- 1) питание ( от 2 В до 5 В),
- 2) вывод данных,
- 3) данный вывод не используется,
- 4) земля.

Основные параметры:

- 1) Напряжение питания: от 3,3 В до 6 В.
- 2) Цифровой выходной сигнал.
- 3) Диапазон измерения относительно влажности от 0% до 100% с точностью в 2%.

4) Диапазон измерения температур от минус 40 до плюс 125 с погрешностью в 0,5 .

5) Время опроса датчика 2 секунды.

6) Максимально потребляемый ток 2,5 мА (при запросе данных), 100 мкА (в режиме ожидания).

При передаче данных датчик выступает в роли ведомого устройства, а МК, в свою очередь, ведущим и отвечает за инициирование связи.

Процесс коммуникации следующий: МК подаёт стартовый сигнал, прижимая шину к земле на 18-20 мс, затем отпускает линию и следит за уровнем на ней, примерно 20-40 мкс. Датчик, обнаружив сигнал и подождав пока уровень снова станет высоким, сам прижимает шину к земле на 80 мкс, за это время делаются измерения и преобразование результатов, далее он отпускает линию на 80 мкс, что указывает на то, что он готов отправить данные. Затем он отправляет 40 бит данных. Перед отправкой каждого бита датчик прижимает шину к земле на 50 мкс, за которым следует 26-28 мкс для «0» или 70 мкс для «1». По завершении связи линия вытягивается подтягивающим резистором и переходит в состояние ожидания. Временная диаграмма процесса коммуникации представлена на рисунке 23.

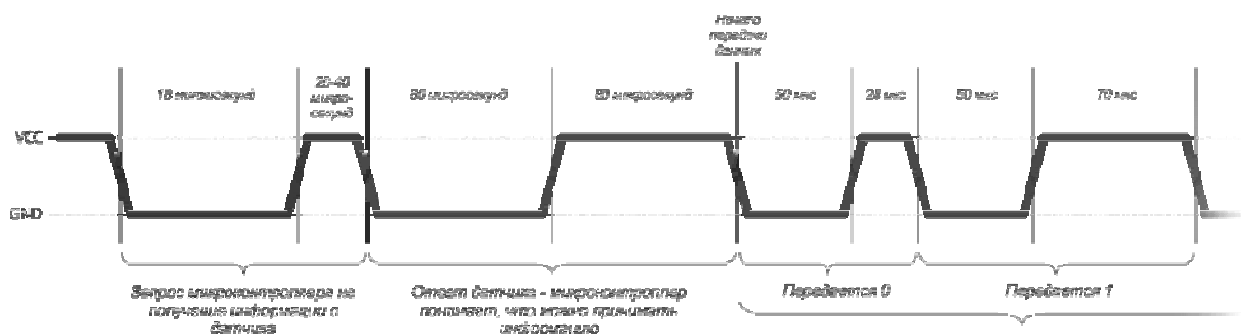


Рисунок 23– Процесс коммуникации

При подключении данного датчика, желательно подключить подтягивающий резистор на 10 кОм между питанием и выводом данных, дабы избежать дребезга контактов. Для работы с ним в IDE, необходимо установить дополнительно библиотеку DHT22.

### 4.3.2 Датчик тока

Для измерения и контроля проходящего тока по цепи применяется датчик тока. Проводя анализ, был выбран экономичный, прецизионный интегральный датчик тока ACS712 компании Allegro. Он основан на принципе, который был открыт в 1879г. Эдвином Холлом, и названным его именем.

Суть эффекта Холла: если проводник с током помещен в магнитное поле, то на краях его возникает электродвижущая сила (ЭДС), которая направлена перпендикулярно к направлению тока и к направлению магнитного поля. Микросхема ACS712 выпускается в миниатюрном 8-ми выводном корпусе. Она состоит из высокоточного линейного датчика Холла с малым напряжением смещения и медного проводника, проходящего у поверхности чипа и выполняющего роль сигнального пути для тока (рисунок 24). Проходящий через проводник ток, создает магнитное поле, воспринимаемое встроенным в кристалл элементом Холла [21]. Создается линейная зависимость силы магнитного поля от проходящего тока. Встроенный формирователь сигнала фильтрует создаваемое чувствительным элементом напряжение и усиливает его до уровня, который может быть измерен с помощью аналого-цифрового преобразователя (далее АЦП) микроконтроллера.

Опираясь на документацию, было рассмотрено расположение выводов и типовая схема его включения (рисунки 24-25).

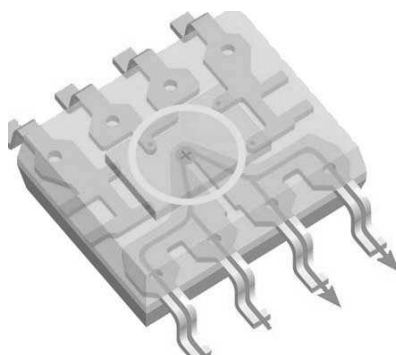


Рисунок 24 – Внутренняя конструкция датчика ACS712



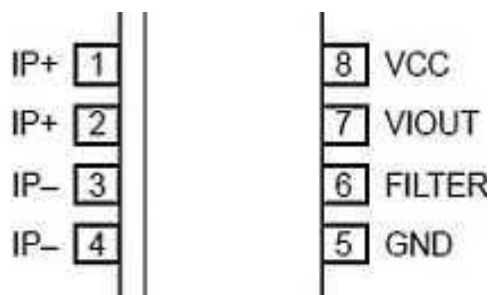


Рисунок 25– Расположение выводов датчика тока ACS712

Выводы 1-2 и 3-4 образуют проводящий путь для измеряемого тока с внутренним сопротивлением порядка 1,2 мОм, что определяет очень малые потери мощности. Выводы датчика 5-8 электрически изолированы от контактов силового проводника.

Выходное напряжение датчика пропорционально току, который протекает через проводящий путь. Выпускается три варианта токового датчика для разных диапазонов измерения:  $\pm 5$  А,  $\pm 20$  А и  $\pm 30$  А. Соответствующие уровни чувствительности по datasheet составляют: 185 мВ/А, 100 мВ/А и 66 мВ/А.

На рисунке 26 изображен график зависимости выходного напряжения от измеряемого тока при напряжении питания 5 В для датчика на 20А.

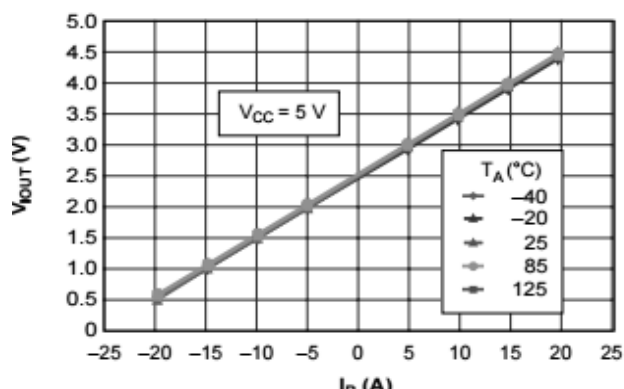


Рисунок 26 – Зависимость выходного напряжения от измеряемого тока

Представим математические выражения, которые используются в преобразовании для датчика на 20А, с чувствительностью 100мВ/В и при входном напряжении, которое равно 5 В.

Выходной код АЦП:

$$\text{Count} = \left( \frac{1023}{V_{CC}} \right) \times V_{in}, \quad (4.1)$$

где выходное напряжение  $V_{in}$  рассчитывается по формуле (4.2)

$$V_{in} = \left( \frac{V_{CC}}{2} \right) + 0,1 \times I \quad (4.2)$$

Отсюда следует, что формула (4.1) приобретает следующий вид:

$$\text{Count} = \left( \frac{1023}{V_{CC}} \right) \times \left( \frac{V_{CC}}{2} + 0,1 \times I \right) \quad (4.3)$$

где  $V_{CC} = 5 \text{ В}$ ,

В итоге, формула для вычисления тока будет следующей:

$$I = 0,0488 \times (\text{Count} - 512), \quad (4.4)$$

Датчик имеет аналоговый выход. На практике же, сначала считывается аналоговое показание напряжения, затем, с помощью преобразований, высчитывается потребленное напряжение в вольтах, и, наконец, потребленный ток в амперах.

Основные характеристики для датчика на 20 А:

- 1) Рабочее напряжение: 5 В.
- 2) Чувствительность: 100 мА/В.
- 3) Рабочая температура: от минус 40 до плюс 125 °С.
- 4) размер: 35×18 мм.

#### 4.3.3 Модуль часов реального времени

Модуль DS3231 – это недорогая плата с точными часами реального времени и температурной компенсацией кварцевого генератора, то есть, подстраиваясь под изменения температуры, делает время более точным. Из-за того, что плата включает в себя получение питания от литиевой стандартной батареи CR2032, он может поддерживать бесперебойную работу при отключении модуля от источника питания. Содержит микросхему памяти AT24C32 емкостью в 4 Кб [22].

В качестве работы со временем, можно использовать функцию `millis()`, но, эта функция сбрасывается до нуля при отключении контроллера из питания, также, она переполняется примерно через 50 дней, следовательно, точность времени будет значительно сбиваться.

Основные характеристики:

- 1) Подсчет часов, минут, секунд, дат, месяцев, лет (високосные года учитываются до 2100 года).
- 2) Рабочая температура от минус 40°C до плюс 85°C.
- 3) Напряжение питания (в том числе и от батареи): 3,3 В – 5,5 В.
- 4) Максимальное потребление тока 650нА.
- 5) Отклонение не более 2 минут в год.

Подключение по I<sup>2</sup>C интерфейсу, передача данных происходит с частотой 400 кГц при рабочем напряжении питания 5 В. I<sup>2</sup>C – это двухпроводной последовательный интерфейс. Протокол разработан таким образом, чтобы гарантировать надежный прием передаваемых данных. В данном случае, МК инициирует передачу данных и формирует сигналы синхронизации, датчик начинает передачу только по команде МК. Используются две двунаправленные линии связи: SCL – сигнал синхронизации и SDA – сигнал данных. Если передача данных на шине отсутствует, то сигналы SCL и SDA имеют высокий уровень. Существуют сигналы `start` и `stop`, которые формируются МК и требуются для определения начала и конца передачи данных. `Start` формируется переходом сигнала SDA из высокого уровня в низкий при высоком уровне сигнала SCL. `Stop` определяется как переход SDA из низкого уровня в высокий при высоком уровне SCL. Отсюда следует, что при передаче данных сигнал SDA может изменяться только при низком уровне сигнала SCL. Временная диаграмма «start/stop» представлена на рисунке 27.

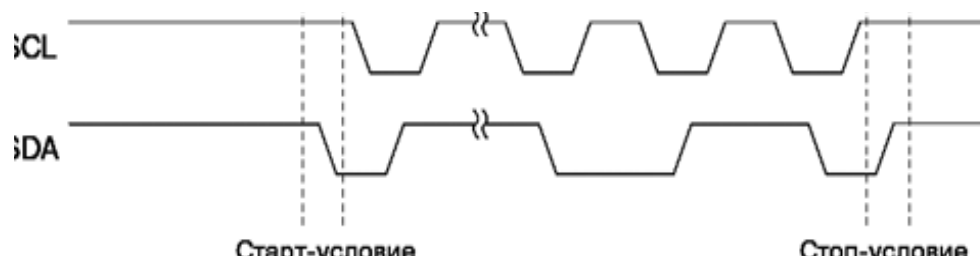


Рисунок 27 – Временная диаграмма «start/stop» условия шины I<sup>2</sup>C

Каждый информационный байт (8 битов) содержит 9 тактовых периодов линии SCL. В девятом такте устройство-получатель формирует сигнал подтверждения приема данных – ACK, подав отрицательный импульс. Диаграмма подтверждения приема байта по шине I<sup>2</sup>C показана на рисунке 28.

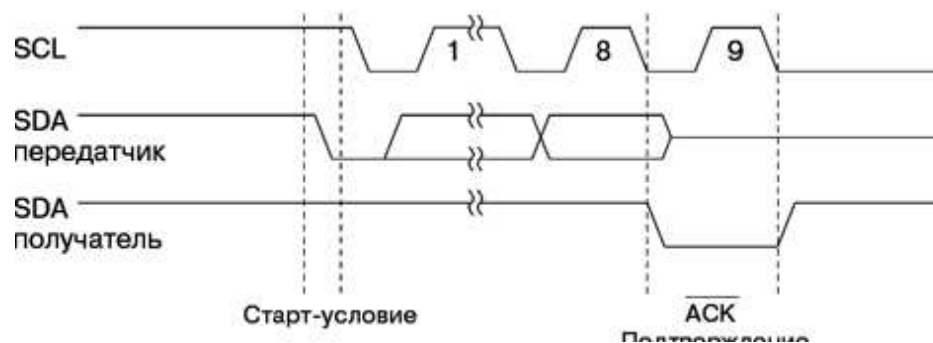


Рисунок 28 – Диаграмма подтверждения приема байта

Имеется 6 выводов на модуле (рисунок 29):

- 1) 32K – Выход с частотой 32кГц.
- 2) SQW – Программируемый выход Square-Wave сигнала.
- 3) SCL – Обмен данными с часами по интерфейсу I<sup>2</sup>C .
- 4) SDA – Передача данных с часов.
- 5) VCC – Питание от 5 В.
- 6) GND – Земля.

Согласно документации, на чипе DS3231 имеется дополнительно еще два вывода:

V<sub>bat</sub> – питание от батареи.

RST – вывод внешнего сброса.

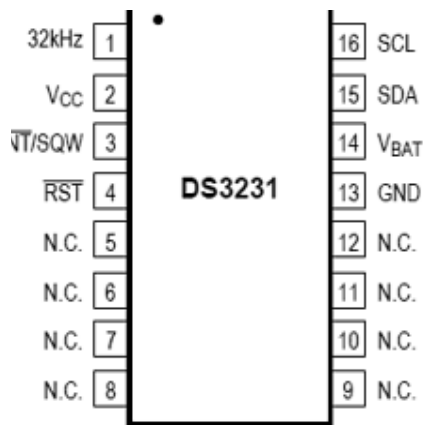


Рисунок 29 – Выводы чипа DS3231

Для работы требуется установка библиотеки `iarduino_RTC.h`. В примерах к данной библиотеке можно воспользоваться функцией `settime()` для установки времени.

#### 4.3.4 Датчик для измерения температуры почвы

Для измерения температуры почвы понадобится датчик во влагозащитном корпусе, который будет внедрен в обогреваемую поверхность. Данный датчик есть на рынке периферийных устройств, он основан на микросхеме DS18B20.

DS18B20 – это цифровой измеритель температуры. Датчик очень прост в использовании, т.к. он, как уже говорилось, цифровой, а также, у него один контакт, с которого идет получение требуемых данных. То есть, существует возможность подключения к контроллеру множество таких сенсоров, вплоть до 127 датчиков параллельно, и считывания температуры с каждого отдельно. Однако, ведущее устройство может быть только одно. Параметры контроля могут быть заданы пользователем и сохранены в энергонезависимой памяти датчика.

Обмен данными с контроллером происходит по однопроводной линии связи, используя протокол интерфейса 1-Wire [23]. Данный интерфейс является двунаправленным полудуплексным последовательным протоколом обмена данными через один сигнальный провод. Имеется заранее определенные типы сигналов – импульс сброса, импульс присутствия, запись 0, запись 1, чтение 0 и

чение 1. Все эти сигналы, кроме импульса присутствия, формируются на шине главным устройством, в данном случае это МК.

Принцип формирования сигналов во всех случаях одинаковый. В начальном состоянии 1-Wire шина с помощью резистора подтянута к плюсу питания. Главное устройство «проваливает» на определенное время 1-Wire шину в ноль, затем «отпускает» ее и, если нужно, «слушает» ответ подчиненного устройства, т.е. термометра DS18B20. Пример инициализации представлен на рисунке 30. Здесь, МК на 480 мкс «опускает» 1-Wire шину в ноль, а затем «отпускает» ее. Если к шине подключен термометр DS18B20, то он обнаруживает положительный перепад и после паузы в 15-60 мкс отвечает микроконтроллеру импульсом присутствия – «проваливает» шину в ноль на время от 60 до 240 мкс.



Рисунок 30 – Инициализация

У каждой микросхемы DS18B20 есть уникальный серийный код, прошитый на производстве, длиной в 64 разряда, который хранится в постоянном запоминающем устройстве (далее ПЗУ). Еще одна особенность – работа без внешнего источника питания. Такая возможность предоставляется через подтягивающий резистор. Высокий сигнал заряжает внутренний конденсатор, который питает прибор, в тот момент, когда на шине низкий уровень. Такое явление называется «паразитное питание». Таким образом, запитывать данный датчик можно через контакт данных, и, следовательно, использовать только два контакта для подключения. Однако, максимально измеряемая температура при этом +100 °C. Если необходимо расширить диапазон, то необходимо применять внешнее питание.

Основные характеристики:

- 1) Питание от 3,3 В до 5 В.
- 2) Диапазон измеряемых температур от минус 55°C до плюс 125°C.
- 3) Ток потребления при бездействии и при опросе: 750 нА и 1 мА соответственно.
- 4) Погрешность не превышает 0,5 °С.

Для подключения используются 3 вывода: GND – земля, VDD – питание 5 В, DQ – вывод данных, подключаемый к любому цифровому пину.

Схема датчика представлена на рисунке 31.

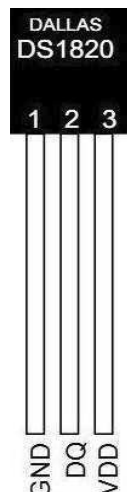


Рисунок 31 – Датчик DS18B20

В таблице команд, находящейся в справочной литературе на датчик DS18B20 можно найти зарезервированные байты данных. Так, отправив на шину данных байт 0x44, термометр начнет измерять и конвертировать температуру. 0xBE – чтение содержимого (температуры в сконвертированном виде).

Для работы в IDE требуется подключение библиотеки OneWire.h, которую можно скачать с официального сайта Arduino. Также, из внешней дополнительной обвязки, необходимо поставить подтягивающий резистор номиналом 4,7 кОм между выводом данных и питанием.

### 4.3.5 Ethernet shield

Ethernet shield – это плата расширения, которая позволяет контроллеру Arduino работать в сети Интернет, а также, в локальной сети для приема и передачи информации. Данное периферийное устройство позволяет управлять объектами удаленно, через веб-сайты, а также, отправлять и получать информацию. Он основан на чипе W5100, имеющий буфер на 16Кб. Скорость подключения – 10 и 100 Мбит/с. Подключается непосредственно на саму плату Arduino сверху – «этажеркой», с помощью штыревых разъемов, через serial peripheral interface (далее SPI) порт, рабочее питание 5В. SPI интерфейс – это последовательный синхронный протокол передачи, который используется для обмена данными с одним или несколькими устройствами на небольших расстояниях. Является одним из популярных интерфейсов для соединения схем между собой. Соединение с интерфейсом SPI происходит благодаря розетке (рисунок 32), которая расположена на нижней стороне платы.



Рисунок 32 – Входы SPI разъема

На данной плате есть слот для SD-карты, на которой можно хранить большие массивы информации, связь с ней осуществляется также, посредством шины SPI. Для полноценной работы требуется подключить Ethernet shield к роутеру с помощью кабеля – TCP/IP (витой пары). Соединяется с сетью с помощью разъема RJ45, которая подключается к Ethernet-контроллеру ко входам 5-6, 8-9. Микросхема W5100 фирмы Wiznet, обеспечивает сетевой стек IP, аппаратно реализует протоколы транспортного, сетевого и канального уровней: TCP, UDP, IPv4, ICMP, ARP, IGMP и MAC [24].

Модуль совместим со всеми версиями главного модуля Arduino. Контактными 10 и 4 Ethernet модуль владеет по умолчанию. Причем, 4-й pin используется для



подключения SD-карты, а 10-й – для обращения к Ethernet контроллеру, следовательно, данные контакты не должны быть заняты другими устройствами на Arduino Mega. Также, возможно подключение, как в роли клиента, так и в роли сервера.

При подключении к Arduino Mega, Ethernet shield занимает следующие цифровые контакты для реализации протокола SPI:

50 – MISO – линия для передачи данных от ведомого устройства к ведущему.

51 – MOSI – линия для передачи данных от ведущего устройства к ведомым.

52 – SCK – тактовые импульсы, которые генерируются ведущим устройством для синхронизации процесса передачи данных.

53 – SS – вход на ведомых устройствах, с помощью которого ведущий может инициировать обмен данными с периферийным устройством. Используется для активизации SD-карты

В работу SPI интерфейса заложен следующий принцип: ведущий передает данные по линии MOSI синхронно со сгенерированным им же сигналом SCLK, а подчиненный захватывает переданные биты данных по определенным фронтам принятого сигнала синхронизации. Одновременно с этим, подчиненный тоже отправляет свою посылку данных по линии MISO. Стоит отметить, что взаимодействие устройств начинается, когда на выход SS подается низкий уровень сигнала. На рисунке 33 представлено подключение к шине SPI.

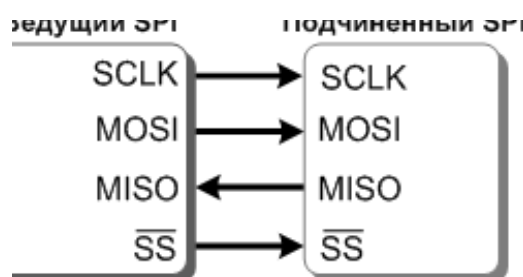


Рисунок 33– Подключение устройств по шине SPI

В роли ведущего шины выступает микроконтроллер, а внешние устройства, подключенные к ведущему, образуют подчиненные шины.

Данные линии на чипе W5100 находятся под номерами: MISO – 27, MOSI – 28, SCLK – 30, SCS – 29.

Представленные ниже контакты чипа W5100 используются для объединения в системную магистраль:

- 1) Шина адреса (Address Bus) – 38-42, 45-54.
- 2) Шина данных (Data Bus) – 19-26.
- 3) Шина управления (Control Bus) – 55-58.

Плата поддерживает технологию Power over Ethernet, т.е. позволяет передавать удаленному устройству электрическую энергию вместе с данными через стандартную витую пару в сети Ethernet [25]. Также, для удобства использования, на модуле расположено несколько светодиодов-индикаторов:

PWR: показывает наличие питания основного устройства и платы расширения.

LINK: светится, если есть соединение с сетью; мигает во время передачи или получения данных.

FULLD: светится, если сетевое соединение поддерживает полнодуплексный режим работы.

100M: светится, если сетевое соединение относится к классу 100 Мбит/с.

RX: мигает в процессе получения данных.

TX: мигает в процессе отправки данных.

COLL: информирует об обнаружении сетевых коллизий.

При разработке проектов в IDE, где задействовано подключение к интернету, необходимо подключение библиотеки Ethernet.h, для работы с SD-картой, библиотека SPI.h.

#### 4.3.6 Силовой двухканальный релейный модуль

Реле — электрическое устройство, которое предназначено для замыкания и размыкания различных участков электрических цепей при заданных изменениях входных величин. Типичное использование его заключается в том, чтобы

позволить схеме низкого напряжения постоянного тока управлять высоковольтной цепью. Реле состоит из следующих компонентов:

1) электромагнита – представляет собой электрический провод, намотанный на катушку с сердечником из магнитного материала.

2) Якоря – пластина из магнитного материала, через толкатель управляющая контактами.

3) Переключателя – могут быть замыкающими, размыкающими, переключающими.

При пропускании электрического тока через обмотку электромагнита возникающее магнитное поле притягивает к сердечнику якорь, который через толкатель смещает и тем самым замыкает или размыкает цепь питания нагрузки. В случае с модулем, помимо реле в нем еще содержится оптоэлектронная развязка с транзистором, которая защищает выводы Arduino от скачков напряжения на катушке. Модуль изготовлен специально для управления слаботочных плат, которые на выходе могут выдать ток не более 40 мА.

Рассмотрим релейный модуль фирмы SONGLE, модель SRD-05VDC-SL-C. Его основные технические характеристики:

1) Потребляемый ток: от 30 мА до 40 мА.

2) Нагрузка: 10 А при 250 В / 10 А при 125 В при постоянном токе; 10 А при 30 В / 10 А при 28 В при переменном токе.

3) Рабочая температура: от минус 25°C до плюс 70°C.

У двухканального модуля существует 2 входных разъема. Первый используется для управления модулем реле, и использует:

VCC – питание 5 В, как в режиме питания, так и в режиме переключения.

GND – земля.

IN1, IN2 – контакт управления, который подключается к порту контроллера. Вывод 1 и 2, соответственно.

Второй разъем нужен для подключения внешнего источника питания, по умолчанию между контактами JD-VCC и VCC установлена перемычка.

А также, для каждого канала существует 3 клеммы на выходе:

NO — нормально разомкнутый (Normally Open).

NC — нормально замкнутый (Normally Closed).

COM — общий (Common).

К контактам NO или NC подключается нагрузка, которой надо управлять, COM подключается к питанию (220 В, например). Для работы с релейным модулем в среде IDE подключение библиотек не требуется.

Принцип работы двухканального реле: такой тип реле состоит из двух независимых частей, где общие только питание VCC и GND. При подаче питания, вывод IN1 находится в высоком состоянии, для переключения первого канала реле, необходимо вывод IN1 перевести в отрицательное состояние, то есть замкнуть цепь на землю. Зажигается светодиод, который показывает пользователю, что цепь замкнулась. Через светодиод, который находится в оптопаре, начинает протекать ток. Следом откроется фототранзистор, через который ток течет на базу транзистора Q1, который откроется и реле сработает. Второй канал реле работает аналогично, только сигнал подавать на вывод IN2. Для работы от отдельного источника питания, необходимо убрать перемычку и подключить питание к JD-VCC и GND.

#### 4.4 Подключение элементов

Подключение датчиков выполняется последовательно, по типу клеммной колодки. Клеммная колодка – это изделие, которое предназначено для соединения проводов.

Схематичные расположения элементов в макете системы представлены на рисунках 34-35.

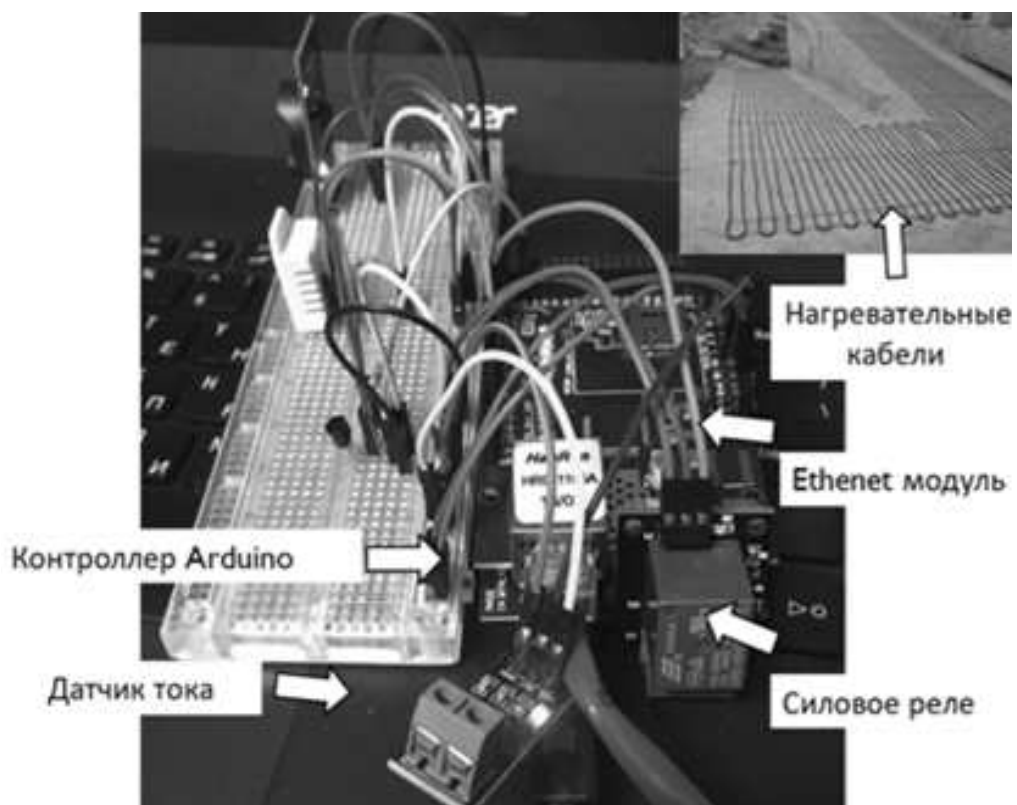


Рисунок 34 – Схематичное расположение элементов системы



Рисунок 35 – Схематичное расположение остальных элементов

Для реализации устройства была составлена электрическая принципиальная схема, в которой показывается принцип соединения элементов между собой. Она изображена в приложении В. Разработка схемы производилась в программе EasyEDA – это бесплатное облачное средство, предназначенное для разработки электрических принципиальных схем и разводки печатных плат. Перечень используемых компонентов представлен в приложении Г.

Для разводки печатной платы была использована та же программа. Она обладает всем необходимым инструментарием, а также, на сайте [easyeda.com](http://easyeda.com) можно отправить заказ на изготовление платы. Опираясь на принципиальную схему и выбранные модели компонентов, разведена печатная плата, представленная в приложении Д.

В результате она получилась двухсторонней, размером 162 мм × 128 мм. Данную плату можно отнести к четвертому классу точности, т.к. минимальное расстояние между проводниками составляет 0.2 мм. Утолщенные силовые дорожки на двух сторонах платы предназначены для проведения уровня напряжения питания. Их ширина обуславливается тем, что им придется пропускать большой ток. Клеммы для подключения нагревательного элемента и силовые элементы, а именно, датчик тока и силовое реле, расположены как можно дальше от микросхем и в левой части платы, для защиты от пробоя микросхем. Основные элементы подписаны согласно принципиальной схеме.

На фронтальной стороне расположены практически все элементы, резисторы, конденсаторы, светодиоды и прочее, кроме микросхемы ACS712, она находится в нижней части.

Припой оловянно-свинцовый ПОС-61 ГОСТ 21931-76 с концентрацией сплавов 61% олова и 39% свинца используется для пайки элементов. Он по своим характеристикам должен соответствовать ГОСТу. Паять необходимо по отраслевому стандарту, чтобы исключить изготовление бракованных изделий. Плата покрывается полиуретановым лаком УР-231 для защиты изделия от пыли и влаги.

## 4.5 Программирование в Arduino IDE

Arduino IDE – это среда программирования, позволяющая писать код программы в удобном текстовом редакторе, компилировать их в машинный код и загружать непосредственно на все версии Arduino. Приложение является официальным и полностью бесплатным. Программы пишутся на языке C++, который дополнен функциями ввода/вывода на контактах.

Для того чтобы контроллер и датчики, либо подключенные устройства могли контактировать друг с другом, нужно скачать и подключить необходимые библиотеки. Они необходимы для установления взаимодействия [26].

Большинство микроконтроллеров имеют прерывания, и ATmega 2560 в том числе. Прерывание позволяет приостановить выполнение основного кода для выполнения какого-либо действия. Оно экономит процессорное время, которое без прерываний тратилось бы на подсчет миллисекунд в случае использования функции `delay`.

Так как, для данного проекта, время считывания показателей с датчика и отправка данных на сервис, занимает минуту, отсюда следует, что при дальнейшем расширении каких-либо функций системы эта минута может использоваться на другие события.

Для оптимизации работы кода и микроконтроллера нужно полностью отказаться от функции `loop()` – стандартной функции бесконечного цикла в Arduino, вместо нее используем прерывание таймер-счетчика 0. Для этого необходимо определение по вектору обработчика прерывания посредством функции `TIMER0_COMPA_vect`. С помощью нее обработчик будет выполнять все те операции, что раньше выполнялись в `loop()`.

Таймер-счётчик микроконтроллера `arduino` - это автономное периферийное устройство, на вход которого, в данном случае, подаются импульсы с тактового генератора. Arduino имеет три таймера: `Timer0`, `Timer1` и `Timer2`. Используемый в данном проекте `Timer0` уже настроен для генерации миллисекундных прерываний, обновляя счетчик миллисекунд, передаваемый в `millis()`. Функция

millis() возвращает количество миллисекунд с начала выполнения текущей программы.

Timer 0 является 8-битным, считает значения от 0 до 255 и генерирует прерывание. Его характеристики:

- 1) таймер сброса при совпадении;
- 2) два независимых выхода по совпадению;
- 3) изменяемый период ШИМ сигнала;
- 4) фазовый корректор ШИМ сигнала;
- 5) тактовый генератор;
- 6) три независимых источника прерывания.

Источником тактового сигнала является тактовый сигнал, используемый для всего микроконтроллера с использованием делителя. Таймер-счетчика 0 имеет в своем составе следующие регистры:

- а) регистр TCCR0A – регистр управления А.

7	6	5	4	3	2	1	1
COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00

Биты COM0A1 и COM0A0 – определяют состояние вывода ОС0А при наступлении события «Совпадение».

Биты COM0B1 и COM0B0 – аналогично, но для вывода ОС0В.

Биты WGM01 и WGM00 – определяют режим работы таймера-счетчика 0.

- б) Регистр TCCR0B – регистр управления В.

7	6	5	4	3	2	1	0
FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00

Биты FOC0A и FOC0B – принудительное изменение состояния вывода ОС0А и ОС0В соответственно.

Бит WGM02 – вместе с WGM01 и WGM00 из регистра TCCR0A определяет режим работы таймера/счётчика T0.



Биты CS02, CS01, CS00 – определяют источник тактового сигнала для таймера-счетчика 0.

в) Регистр TIMSK – регистр масок прерываний таймер-счетчика 0.

7	6	5	4	3	2	1	0
-	-	-	-	-	OCIE0B	TOIE0	OCIE0A

Биты OCIE0B и OCIE0A разрешают прерывания при совпадении с А и В.

Бит TOIE0 разрешает прерывание по переполнению при установке 1.

Записав в эти биты 0, прерывания от таймера-счетчика будут запрещены.

г) Регистр TIFR0 – регистр флагов прерываний для таймер-счетчика 0.

7	6	5	4	3	2	1	0
	-	-	-	-	OCF0B	OCF0A	TOV0

Данные биты устанавливаются в 1 в зависимости от того, какое прерывание поступило - совпадение с А, В, либо переполнение.

д) OCR0A и OCR0B – регистры сравнения А и В соответственно.

е) TCNT0 – счетный регистр таймер-счетчика 0.

Настройка таймера-счетчика 0 представлена на рисунке 36. Стоит отметить, что функция sei() устанавливает флаг глобальных прерываний в регистре статуса SREG.

```

TCCR0A |= (1 << WGM01); //сброс при совпадении
OCR0A = 0xF9; //начало отсчета до переполнения 249 (1мс)
TIMSK0 |= (1 << OCIE0A); //разрешаем прерывание при совпадении
TCCR0B |= (1 << CS01) | (1 << CS00); //делитель частоты 64
sei(); // разрешаем глобальные прерывания

```

Рисунок 36 – Настройка таймер-счетчика 0.

## 4.6 Взаимодействие с сервисом

Разрабатываемое устройство в сети выступает в качестве клиента. Взаимодействие между сервером и клиентом осуществляется с помощью HTTP – запросов. Формируется запрос, который в дальнейшем отправляется на сервер, после чего серверное программное обеспечение обрабатывает данный запрос, формирует ответ о выбранном способе управления (автоматическое/ручное), и передает обратно клиенту. В режиме клиента, устройство организует соединение с удаленным сервером, на который посылаются пакеты с данными с заданной периодичностью. Выход в интернет обеспечивает плата расширения Arduino Ethernet и стандартная библиотека Ethernet, которая поставляется в комплекте со средой разработки [27]. На рисунке 37 можно увидеть схему взаимодействия клиент-сервер.

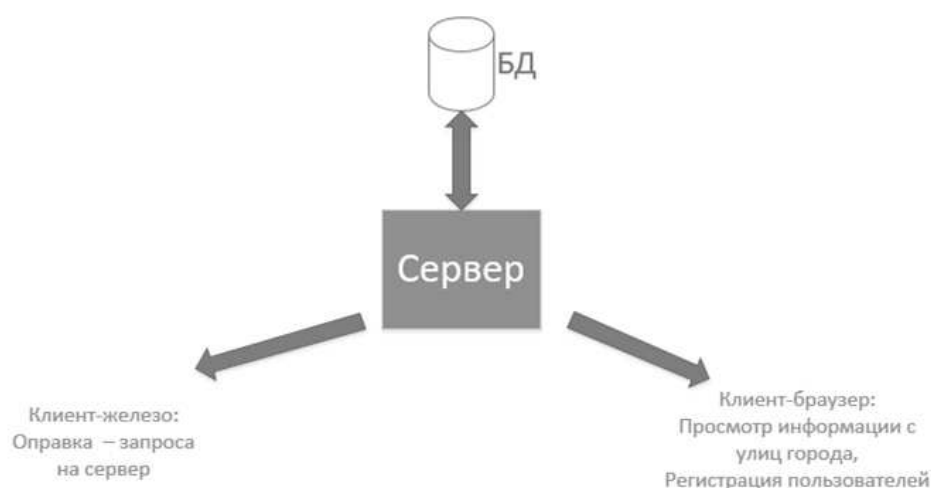
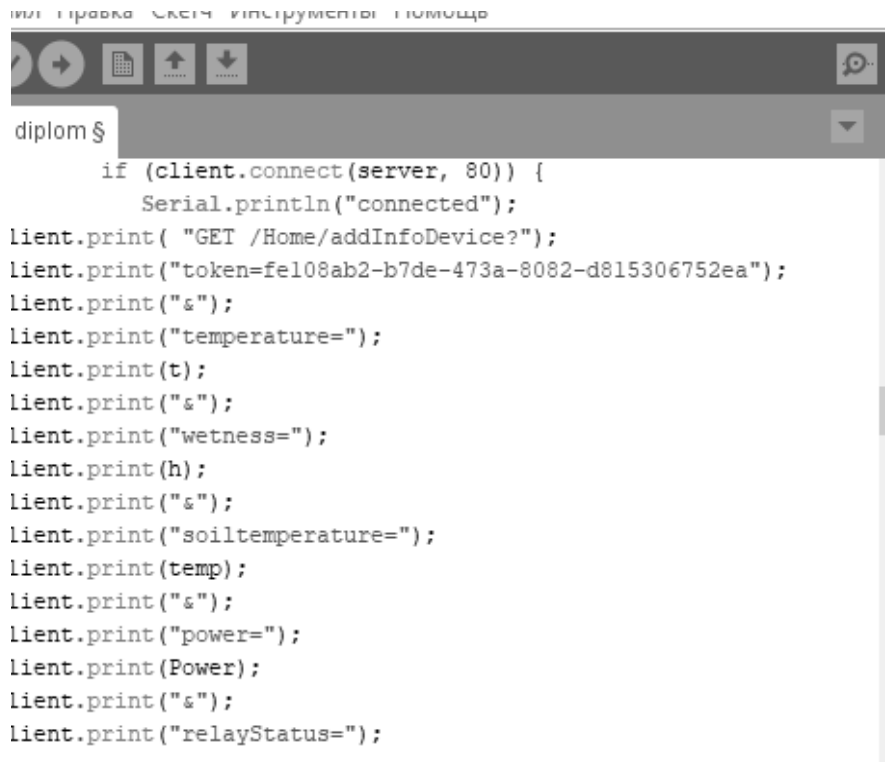


Рисунок 37– Взаимодействие устройства с сервером

В качестве протокола передачи данных (HTTP) запроса используется GET метод. Данный метод является безопасным, т.к он может лишь запросить информацию, но не может изменить запрашиваемый ресурс. Таким образом, это не приведет к нежелательным результатам для пользователя, других лиц и сервера. Также, на стороне сервера он обрабатывается меньшим потреблением ресурсов.

Фрагмент кода запроса представлен на рисунке 38.

A screenshot of an IDE window showing a code snippet. The window title is "Имя файла: sketch.instruments.instrum...". The code is as follows:

```
diplom $  
    if (client.connect(server, 80)) {  
        Serial.println("connected");  
lient.print( "GET /Home/addInfoDevice?");  
lient.print("token=fel108ab2-b7de-473a-8082-d815306752ea");  
lient.print("&");  
lient.print("temperature=");  
lient.print(t);  
lient.print("&");  
lient.print("wetness=");  
lient.print(h);  
lient.print("&");  
lient.print("soiltemperature=");  
lient.print(temp);  
lient.print("&");  
lient.print("power=");  
lient.print(Power);  
lient.print("&");  
lient.print("relayStatus=");
```

Рисунок 38 – Реализация передачи данных с помощью GET метода

Всю работу в связи с сервером на себя берет контроллер. Он отправляет данные каждую минуту и ждет от сервера ответа на запрос о режиме работы, в случае ручного режима еще и значение о выходе. Контроллер принимает ответ от сервера и согласно ему устанавливает состояние своего выхода.

## 5 РАСЧЕТ МАТЕРИАЛЬНЫХ ЗАТРАТ

Энергопотребление является существенным фактором использования системы антиобледенения. Зачастую, именно наличие свободной электроэнергии ограничивает Заказчика, чтобы уложить кабели на всех участках. Нецелесообразно использовать данную систему антиобледенения на больших площадях, так как положительные моменты от использования системы перекрываются высокими финансовыми затратами на оплату электроэнергии. Отсюда следует, что обогревать необходимо только рабочую зону.

Комитет по архитектуре и градостроительству города Москвы (МосКомАрхитектура) рекомендует производить следующий расчет для эксплуатационных затрат при работе систем отопления электроэнергией:

$$C_{\text{год}} = P_{\text{н}} \times h \times s, \quad (5.1)$$

где  $C_{\text{год}}$  – стоимость работы системы в течение года, руб;

$P_{\text{н}}$  – номинальная мощность системы, кВт;

$h$  – количество часов работы системы в год;

$s$  – стоимость 1кВт/час электроэнергии, руб.

Чтобы определить эксплуатационные затраты данной системой, предлагается количество ее работы в год определять следующим образом: считать, из примера на погоде Южного Урала, что система включается в середине ноября, а выключается в начале апреля. Отсюда следует, что система эксплуатируется около 5 месяцев, т.е, примерно 135 календарных дней по 24 часа, итого, система работает 3240 часа. Принять, что 25% времени система не работает, т.е отключена автоматикой из-за выхода показателей за пределы рабочих показателей или отсутствия осадков, или принудительное отключение пользователем. Таким образом, рабочих часов системы принять  $3240 \times 0,75 = 2430$  рабочих часов системы.

Для примера производится расчет ориентировочный годовой стоимости эксплуатации системы антиобледенения в домах с газовыми плитами. Если принять длину греющего кабеля 100 м, номинальной мощностью в 2кВт,

рассчитав по формуле (5.1) получим:  $C_{\text{год,газ}} = 2 \text{ кВт} \times 2430 \times 2,93 = 14240$  руб/год. Для домов, оборудованных электрическими плитами или электроотопительными приборами, стоимость затрат при работе системы будет следующей:  $C_{\text{год,эл}} = 2 \text{ кВт} \times 2430 \times 1,85 = 9015$  руб/год.

Если использовать в системе саморегулирующийся греющий кабель, то из-за автоматического регулирования тепловыделения в ответ на изменения внешней температуры, расход электроэнергии снизится на 10-15 %. Итого, затраты на год составят около 12104 руб/год, для домов с газовыми плитами, или 7662 руб/год для домов с электрическими плитами.

## ЗАКЛЮЧЕНИЕ

При разработке выпускной квалификационной работы был проведен анализ существующих систем профилактики наледи. Были выявлены недостатки и принято решение о разработке нового устройства. Кроме этого, были исследованы принципы ледообразования и обледенения поверхностей.

Результатом выполненной работы является спроектированная система антиобледенения, позволяющая работать как в автоматическом режиме, так и в ручном с удаленным управлением. Система взаимодействует с сервисом на отправку данных для on-line мониторинга погоды и прием для определения режима работы и управления кабелями, в случае ручного режима. Помимо этого, внедрен искусственный интеллект, который работает в исключительных ситуациях, и направлен в большей степени на оттепель.

Предложены различные конструкции укладки нагревательных кабелей, которые позволят уменьшить затраты на потребление электрической энергии, а также, рассмотрены их типы, выявлены достоинства и недостатки.

В пояснительной записке к выпускной квалификационной работе отображены все основные этапы проектирования системы и ее разработка.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Любушкина, С.Г. Землевладение: учебное пособие для вузов / С.Г.Любушкина, В.А.Кошевой. – М.: Владос, 2014. – 176 с.
- 2 Литвинов, И.В. Структура атмосферных осадков. / И.В. Литвинов. – Л.: Изд-во «Гидрометеиздат», 1974. – 154 с.
- 3 Пермьяков, Г.Н. Атмосферные явления природы и их регулирование. / Г.Н. Пермьяков. – СПб.: Изд-во «Нестор-История», 2012. – 100 с.
- 4 Вьюгин, В.В. Математические основы машинного обучения и прогнозирования. / В.В.Вьюгин. – М.: МЦНМО, 2013. – 390 с.
- 5 Хайкин, С. Нейронные сети: полный курс. / С. Хайкин. – М.: Издательский дом «Вильямс», 2008. – 1103 с.
- 6 Николенко, С.И. Глубокое обучение: погружение в мир нейронных сетей. / С.И. Николенко, А.В. Кудрин, Е.А. Архангельская. – СПб.: Изд-во «Питер», 2017. – 480 с.
- 7 Рашид, Т. Создаем нейронную сеть. / Т.Рашид. – М.:Изд-во «Вильямс», 2017. – 272 с.
- 8 Галушкин, А.И. Нейронные сети: основы теории. / А.И. Галушкин. – М.: «Горячая линия-Телеком», 2017. – 496 с.
- 9 Головкин, В.А. Нейронные сети. Обучение, организация и применение. /В.А. Головкин. – М.: ИПРЖР, 2001. – 256 с.
- 10 Николенко, С.И. Самообучающиеся системы. / С.И.Николенко, А.Л.Тулупьев. – М.: МЦНМО, 2009. – 288 с.
- 11 Барский, А.Б. Нейронные сети: распознавание, управление, принятие решений. /А.Б. Барский. – М.: Финансы и статистика, 2004. – 176 с.
- 12 Боровиков, В.П. Нейронные сети. Statistica Neural Networks. Методология и технологии современного анализа данных. /В.П. Боровиков. – М.: Горячая линия – Телеком, 2015. – 392 с.

- 13 Осовский, С.А. Нейронные сети для обработки информации: Пер с польского И.Д. Рудинского. / С. Осовский. – М.: Финансы и статистика, 2002. – 344 с.
- 14 Медведев, В.С. Нейронные сети. MATLAB 6. / В.С. Медведев, В.Г. Потемкин. – М.: Изд-во «Диалог-МИФИ», 2012. – 496 с.
- 15 Новиков, Ф.А. Основы моделирования на UML: учебное пособие. / Ф.А. Новиков, Д.Ю. Иванов. – СПб.: СПбГУ ИТМО, 2010. – 200 с.
- 16 Леоненков, А.В. Самоучитель UML 2. / А.В. Леоненков. – СПб.: Изд-во «БХВ-Петербург», 2007. – 576 с.
- 17 Хетагуров, Я.А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ): учебное пособие для студентов высших учебных заведений. / Я.А. Хетагуров – М.: Изд-во «БИНОМ», 2015. – 243 с.
- 18 Петин, В.А. Микрокомпьютеры Raspberry Pi. Практическое руководство. / В.А.Петин. – СПб.: Изд-во «БВХ-Петербург», 2015. –240 с.
- 19 Соммер, У. Программирование микроконтроллерных плат Arduino/Freeduino: Пер с нем. – 2-е изд. / У. Соммер. – СПб.: Изд-во «БВХ-Петербург», 2016, – 256 с.
- 20 DHT22 (Datasheet). [Электронный ресурс]. URL: file:///C:/Olesya/01/Downloads/DHT22-Datasheet.pdf (дата обращения 03.04.2018).
- 21 ACS712 (Datasheet). [Электронный ресурс]. URL: file:///C:/Olesya/01/Downloads/ACS712-Datasheet.pdf (дата обращения 05.04.2018).
- 22 DS3231 (Datasheet). [Электронный ресурс]. URL: file:///C:/Olesya/01/Downloads/DS3231.pdf (дата обращения 20.03.2018).
- 23 DS18B20 (Datasheet). [Электронный ресурс]. URL: file:///C:/Olesya/01/Downloads/DS18B20.pdf (дата обращения 22.03.2018).
- 24 Петин, В.А. Arduino и Raspberry Pi в проектах Internet of Things. / В.А. Петин. – СПб.: Изд-во «БВХ-Петербург», 2016. –320 с.



- 25 Arduino Ethernet Shield W5100 (Datasheet). [Электронный ресурс]. URL: <file:///C:/Olesya/01/Downloads/AESW5100.pdf> (дата обращения 22.03.2018).
- 26 Блум, Д. Изучаем Arduino: Инструменты и методы технического волшебства. / Д. Блум. – СПб.: Изд-во «БХВ-Петербург», 2015. – 336 с.
- 27 Иго, Т. Arduino, датчики и сети для связи устройств: Пер с англ. – 2-е изд. / Т. Иго. – СПб.: Изд-во «БВХ-Петербург», 2015. – 554 с.
- 28 Лещинская, Л.В. Отопление загородного дома. / Л.В. Лещинская, А.А. Малышев. – М.: Изд-во «АН-СССР», 1955. – 384 с.

## ПРИЛОЖЕНИЯ

### Приложение А

#### Исходный код аппаратной части системы

```
#include <SD.h>
#include <OneWire.h>
#include <DHT.h>
#include <SPI.h>
#include <Ethernet.h>
#include <iarduino_RTC.h>

#define temLowTrigger 30
#define humLowTrigger 95
#define DHTPIN 7
#define DHTTYPE DHT22
#define DS18B20PIN 24

#define INPUT_W_COUNT 2
#define INPUT_N_COUNT 2
#define OUTPUT_W_COUNT 2
#define OUTPUT_N_COUNT 2

#define INPUT_VECTOR_SIZE 4
#define TMP_VECTOR_SIZE 50
File myFile;
float input[INPUT_VECTOR_SIZE];
float Wtmp[TMP_VECTOR_SIZE];
float INPUT_W[INPUT_N_COUNT][INPUT_W_COUNT];
float OUTPUT_W[OUTPUT_N_COUNT][OUTPUT_W_COUNT];
float INPUT_OUT[INPUT_N_COUNT];
float NET_OUT[OUTPUT_N_COUNT];
int FileCount = 0, CountW = 0;

byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02 };
EthernetClient client;

DHT dht(DHTPIN, DHTTYPE);
int relay1 = 2;
int relay2 = 3;
iarduino_RTC time(RTC_DS3231);
OneWire ds(DS18B20PIN);

const int analogIn = A0;
int mVperAmp = 100;
```

```
int RawValue= 0;
float ACSoffset = 2.5;
double Voltage = 0;
double Amps = 0;
float Power = 0;
float temp = 0;

const int analogIn2 = A1;
int mVperAmp2 = 100;
int RawValue2 = 0;
float ACSoffset2 = 2.5;
double Voltage2 = 0;
double Amps2 = 0;
float Power2 = 0;
float temp2 = 0;

float t =0;
float h =0;
char server[] = "anti-ice.azurewebsites.net";

int a=0;
int buff=0;

int i = 0;
ISR(TIMER0_COMPA_vect) //таймер счетчик0 при совпадении
{
  i++;
  if(i == 60000)
  {
    RawValue = analogRead(analogIn);
    Voltage = (RawValue / 1023.0) * 5;
    Amps = ((Voltage - ACSoffset) / mVperAmp);
    Power = 220 * Amps;

    RawValue2 = analogRead(analogIn2);
    Voltage2 = (RawValue2 / 1023.0) * 5;
    Amps2 = ((Voltage2 - ACSoffset2) / mVperAmp2);
    Power2 = 220 * Amps2;

    h = (float) dht.readHumidity();
    t = (float) dht.readTemperature();
    temp = get_data_ds18b20();
```

```
if (client.connect(server, 80)) {
    Serial.println("connected");
    client.print( "GET /Home/addInfoDevice?");
    client.print("token=fe108ab2-b7de-473a-8082-d815306752ea");
    client.print("&");
    client.print("temperature=");
    client.print(t);
    client.print("&");
    client.print("wetness=");
    client.print(h);
    client.print("&");
    client.print("soiltemperature=");
    client.print(temp);
    client.print("&");
    client.print("power=");
    client.print(Power);
    client.print("&");
    client.print("relayStatus=");
    client.print(a);

    client.println(" HTTP/1.1");
    client.println("Host: anti-ice.azurewebsites.net");
    client.println("Connection: close");
    client.println();
    client.println();

    while (client.available())
    {
        char c = client.read();
        if ( c =='2')
        {
            buff=2;
        }
        if ( c =='1')
        {
            buff=1;
        }
        if ( c =='0')
        {
            buff=0;
        }
    }
}
```

```

if (buff=0){
    if ( t<= -39){
        Serial.println(" The sensor does not show reliable information");
        digitalWrite(relay1, LOW);
        a = 0;
        client.print(a);
        digitalWrite(relay2, LOW);
    }

    if((h > humLowTrigger)&& (-39<t < temLowTrigger)) {
        if ((21<=time.Hours)&&(time.Hours<=6)&&(time.minutes>=0)){
            digitalWrite(relay1, HIGH);
            a = 1;
            client.print(a);
            digitalWrite(relay2, HIGH);
        }
        else{
            if (temp < 35){
                if((7<=time.Hours)&&(time.Hours<=13)&&(time.minutes>=0)){
                    digitalWrite(relay1, HIGH);
                    a = 1;
                    client.print(a);
                    digitalWrite(relay2,LOW);
                }
            }
        }
        else {
            digitalWrite(relay1, LOW);
            a = 0;
            client.print(a);
            digitalWrite(relay2, HIGH);
        }
    }
}
else {digitalWrite(relay1, LOW);
a = 0;
client.print(a);
digitalWrite(relay2, LOW);
}

}

client.stop();
client.flush();
}

```

```

        if ( buff=1)
        {
digitalWrite (relay1, HIGH);
        }
else
        {
digitalWrite(relay1, LOW);
        }
        CreateVector();//данные с датчиков
        NeuralNetUpdate();//расчет вектора сети

        i = 0;
        }

}

void setup() {
  Serial.begin(9600);
  Ethernet.begin(mac);
  dht.begin();
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  time.begin();
  time.settime(30,59,13,18,10,17,3);
  ReadDataFromSD();

  TCCR0A |= (1 << WGM01); //сброс при совпадении
  OCR0A = 0xF9; //начало отсчета до переполнения 249 (1мс)
  TIMSK0 |= (1 << OCIE0A); //разрешаем прерывание при совпадении
  TCCR0B |= (1 << CS01) | (1 << CS00); //делитель частоты 64
  sei(); // разрешаем глобальные прерывания
}

void loop()
{

}

float get_data_ds18b20() {
  byte i;
  byte present = 0;
  byte data[12];
  byte addr[8];

```

```

int Temp;
float fTemp=0.0;

if( !ds.search(addr)) {
  ds.reset_search();
  return;
}
if( OneWire::crc8( addr, 7) != addr[7]) {
  Serial.print("CRC is not valid!\n");
  return 999;
}
if( addr[0] != 0x28) {
  Serial.print("Device is not a DS18S20 family device.\n");
  return 999;
}
ds.reset();
ds.select(addr); // запустить конвертацию температуры датчиком
ds.write(0x44,1); //измерить температуру и записать в sram
present = ds.reset();
ds.select(addr);
ds.write(0xBE); //последовательно считать 9 байтов SRAM
for ( i = 0; i < 9; i++) {
  data[i] = ds.read();
}
Temp=((data[1]<<8)+data[0]);
fTemp=1.0*Temp/16+(float(Temp%16))*1.0/16; // перевод в значение float

return fTemp;
}

void NeuralNetUpdate(){

for(int N = 0; N < INPUT_N_COUNT; N++){ //Входной слой
  INPUT_OUT[N] = 0.0;
  for(int W = 0; W < INPUT_W_COUNT; W++){
    INPUT_OUT[N] += input[W] * INPUT_W[N][W];
  }
  INPUT_OUT[N] = Sigmoid(INPUT_OUT[N]);
}

for(int N = 0; N < OUTPUT_N_COUNT; N++){ //Выходной слой
  NET_OUT[N] = 0.0;
  for(int W = 0; W < OUTPUT_W_COUNT; W++){

```

```

    NET_OUT[N] += INPUT_OUT[W] * OUTPUT_W[N][W];
  }
  NET_OUT[N] = Sigmoid(NET_OUT[N]);
  if(NET_OUT[N] == 1.0){ command(N); }
}
}

```

```

void command(int value){
  if(value == 0){
    digitalWrite (relay1, LOW);
    return;
  }
  if(value == 1){
    digitalWrite (relay1, HIGH);
    return;
  }
}

```

```

float Sigmoid(float value){
  if(value > 0.50f){return 1.0;}
  return 0.0f;
}

```

```

void CreateVector(){
  input[1] = (float) dht.readTemperature();
  input[2] = get_data_ds18b20();
  input[3] = atof (time.gettime("H.i"));
  input[4] = (float) dht.readHumidity();
}

```

```

void LoadInputW(){
  for(int N = 0; N < INPUT_N_COUNT; N++){
    for(int W = 0; W < INPUT_W_COUNT; W++){
      INPUT_W[N][W] = Wtmp[CountW];
      CountW++;
    }
  }
}

```

```

void LoadOutputW(){
  for(int N = 0; N < OUTPUT_N_COUNT; N++){
    for(int W = 0; W < OUTPUT_W_COUNT; W++){
      OUTPUT_W[N][W] = Wtmp[CountW];
    }
  }
}

```



```

CountW++;
    }
}

void ReadDataFromSD(){
    pinMode(10, OUTPUT);
    if (!SD.begin(10)) {
        return;
    }

    myFile = SD.open("test.txt");
    if (myFile) {
        int i = 0;
        char tmp[32];
        while (myFile.available())//считываем все данные из файла
        {
            tmp[i] = myFile.read();
            if(tmp[i] == ';'){
                tmp[i] = '\0';
                Wtmp[FileCount] = atof(tmp);
                FileCount++;
                i = 0;
            }
            else
            {
                i++;
            }
        }
        myFile.close();
    }
    LoadInputW(); //Загружаем первый слой
    LoadOutputW(); //Загружаем последний слой
}

```

## Приложение Б

### Исходный код обучения нейронной сети

```
using System;
using System.Collections.Generic;
using System.IO;
class Test
{
    static int InputN = 2, OutN = 2;
    static int InputW = 2, OutW = 2;

    static Random run = new Random();

    static float[,] INPUT_W = new float[InputN, InputW];
    static float[,] OUTPUT_W = new float[OutN, OutW];

    static float[] INPUT_RESULT = new float[InputN];
    static float[] OUTPUT_RESULT = new float[OutN];

    static float[] OUTPUT_ERROR = new float[OutN];
    static float[] INPUT_ERROR = new float[InputN];

    static List<float[]> Input_patern = new List<float[]>();
    static List<float[]> Out_patern = new List<float[]>();

    static float Sig = 0.5f;
    static float etta = 0.001f;

    static void init_net()
    {
        for (int i = 0; i < InputN; i++)
        {
            for (int j = 0; j < InputW; j++)
            {
                INPUT_W[i, j] = (float)run.NextDouble() - 0.5f;
            }
        }
        for (int i = 0; i < OutN; i++)
        {
            for (int j = 0; j < OutW; j++)
            {
                OUTPUT_W[i, j] = (float)run.NextDouble() - 0.5f;
            }
        }
    }
}
```

```

}

static void DirectMotion(float[] vector)
{
    if (vector.Length > InputW) {
        return;
    }

    for (int i = 0; i < InputN; i++)
    {
        INPUT_RESULT[i] = 0.0f;
        for (int j = 0; j < InputW; j++)
        {
            INPUT_RESULT[i] += INPUT_W[i, j] * vector[j];
        }
        INPUT_RESULT[i] = Sigma(INPUT_RESULT[i]);
    }

    for (int i = 0; i < OutN; i++)
    {
        OUTPUT_RESULT[i] = 0.0f;
        for (int j = 0; j < OutW; j++)
        {
            OUTPUT_RESULT[i] += OUTPUT_W[i, j] * INPUT_RESULT[j];
        }
        OUTPUT_RESULT[i] = Sigma(OUTPUT_RESULT[i]);
    }
}

static float calculate_global_error()
{
    float error = 0.0f;
    for (int i = 0; i < OUTPUT_ERROR.Length; i++)
    {
        error += Math.Abs(OUTPUT_ERROR[i] * OUTPUT_ERROR[i]);
    }
    return error * 0.5f;
}

static float BackPropagation(float[] vector, float[] target)
{

```

```

DirectMotion(vector);

    for (int i = 0; i < OUTPUT_ERROR.Length; i++)
    {
        OUTPUT_ERROR[i] = 0.0f;
        OUTPUT_ERROR[i] = target[i] - OUTPUT_RESULT[i];
    }
    for (int neurons = 0; neurons < INPUT_ERROR.Length; neurons++)
    {
        INPUT_ERROR[neurons] = 0.0f;
        for (int out_n = 0; out_n < OutN; out_n++)
        {
            INPUT_ERROR[neurons] += OUTPUT_ERROR[out_n] *
OUTPUT_W[out_n, neurons];
        }
    }

    for (int n = 0; n < InputN; n++)
    {
        for (int w = 0; w < InputW; w++)
        {
            INPUT_W[n, w] += (etta * INPUT_ERROR[n] * vector[w]);
        }
    }
    for (int n = 0; n < OutN; n++)
    {
        for (int w = 0; w < OutW; w++)
        {
            OUTPUT_W[n, w] += (etta * OUTPUT_ERROR[n] * INPUT_RESULT[w]);
        }
    }
    return calculate_global_error();
}

static void Learn()
{
    int Iteration = 0;
    float Error = 0.0f;
    while (Iteration < 250000)
    {
        Error = 0.0f;
        for (int paterns = 0; paterns < Input_patern.Count; paterns++)
    {

```

```

Error += BackPropagation(Input_pattern[paterns], Out_pattern[paterns]);
    }
    Console.WriteLine(Error);
    Iteration++;
}
}

static float Sigma(float value) {
    if (value >= Sig) {
        return 1.0f;
    }
    return 0.0f;
}

public static void Main()
{
    string path = @"C:\Users\User\Desktop\MyText.txt";
    {
        List<string> Input_pattern = new List<string>();
        List<string> Out_pattern = new List<string>();
        Console.WriteLine();
        Console.WriteLine("*****считываем построчно*****");
        using (StreamReader sr = new StreamReader(path,
System.Text.Encoding.Default))
        {
            string line;
            while ((line = sr.ReadLine()) != null)
            {
                if (line != "")
                {
                    Input_pattern.Add(line.Substring(0, line.IndexOf(':')));
                    Out_pattern.Add(line.Substring(line.IndexOf(':') + 1));
                }
            }
            Console.ReadKey();
        }
        using (StreamWriter sr = new StreamWriter("Text.txt"))
        {
            foreach (var line in Out_pattern)
                sr.WriteLine(line);
        }
    }
}
}

```

Приложение В  
Схема электрическая принципиальная



## Приложение Г

Перечень элементов к схеме электрической принципиальной





Приложение Д

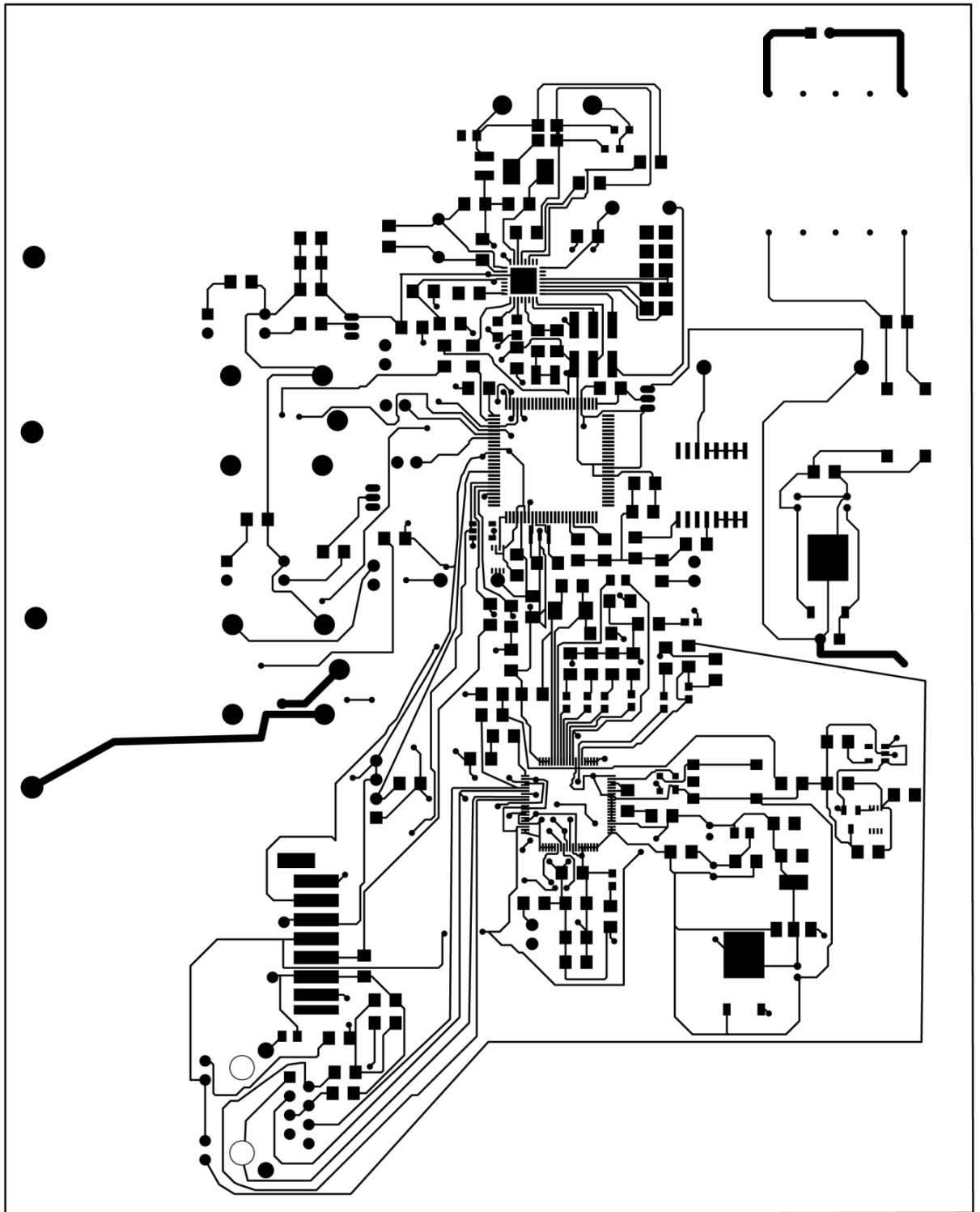


Рисунок Д.1 – Верхняя часть платы

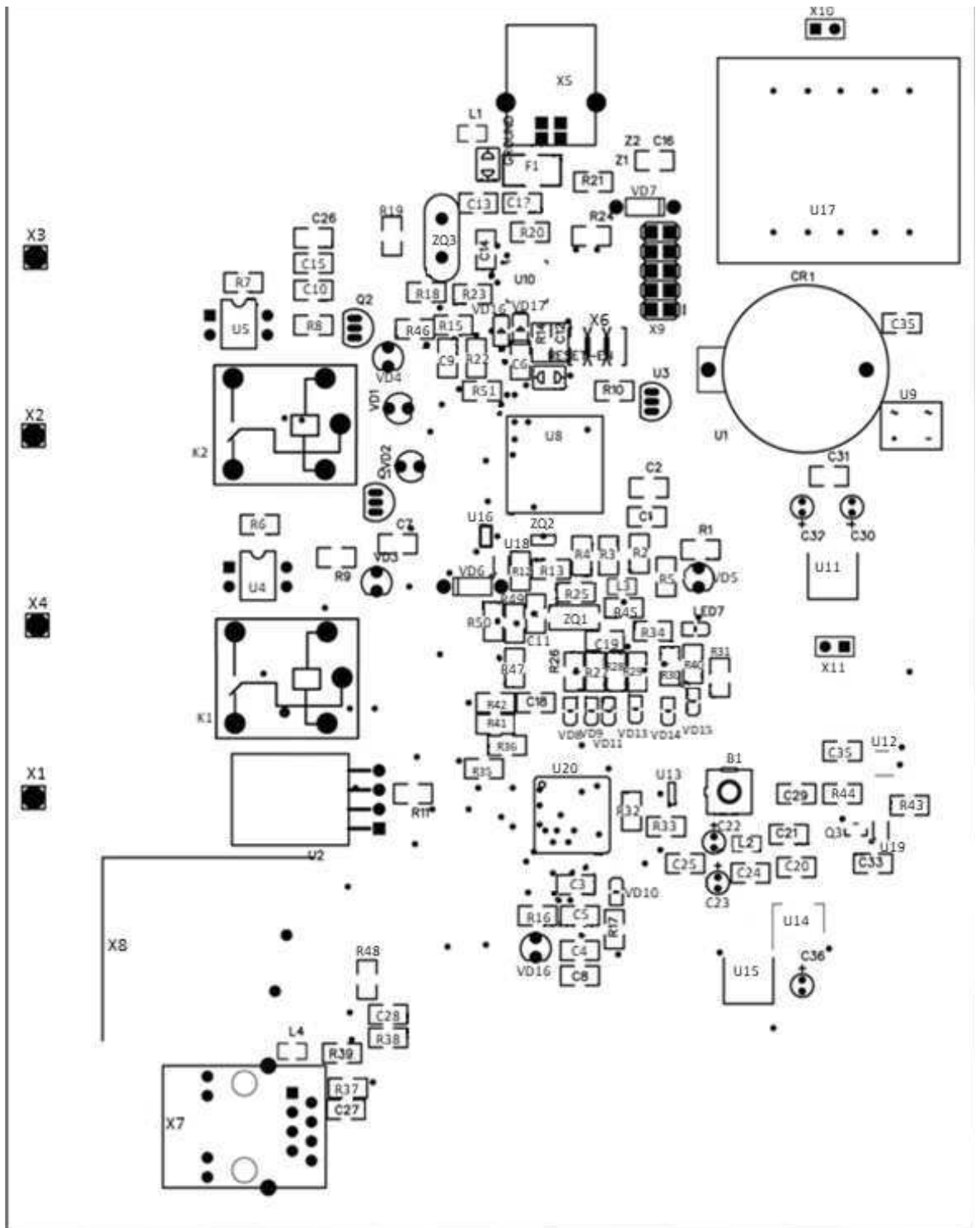


Рисунок Д.2 – Расположение элементов на верхней части печатной платы

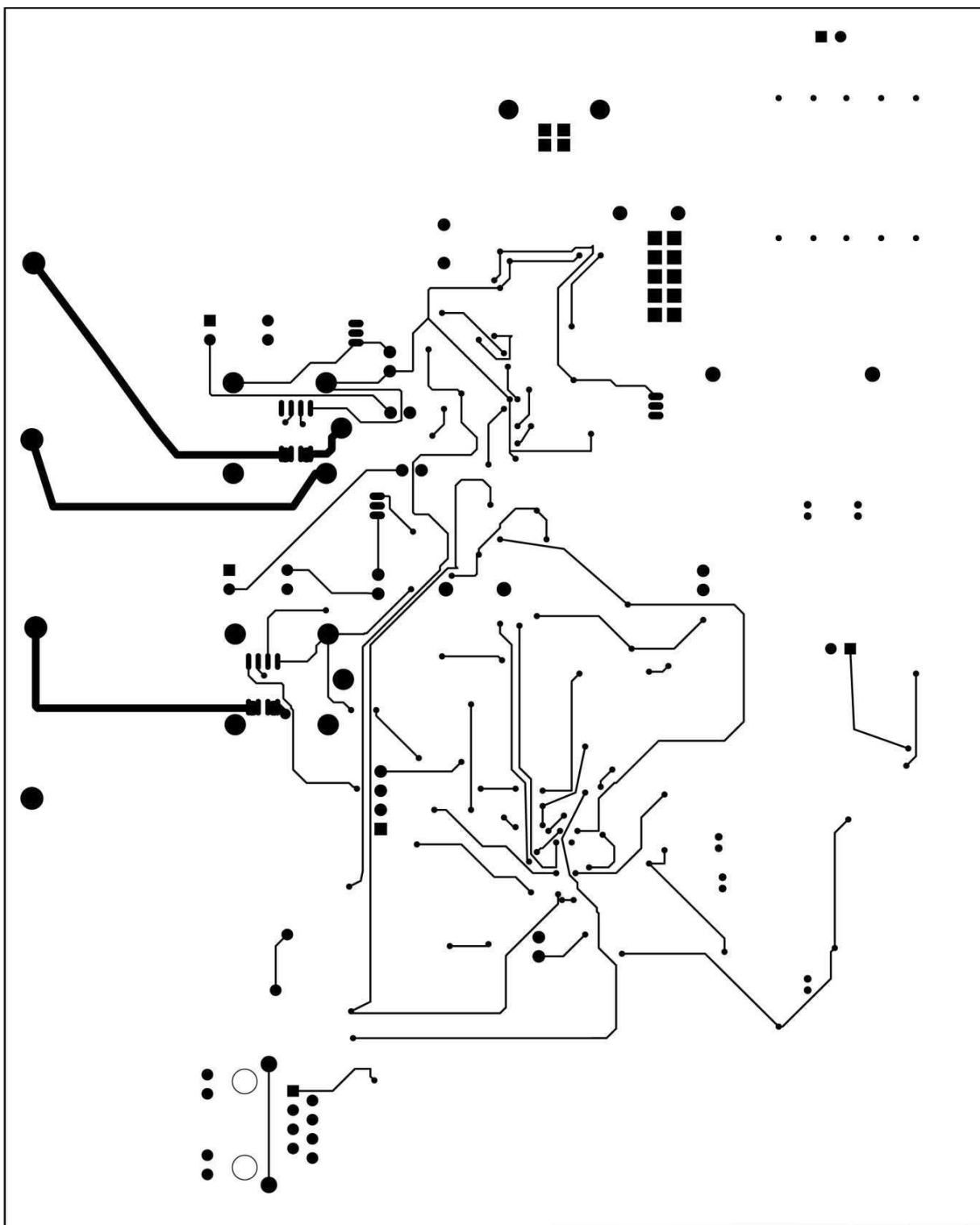


Рисунок Д.3 – Нижняя часть платы

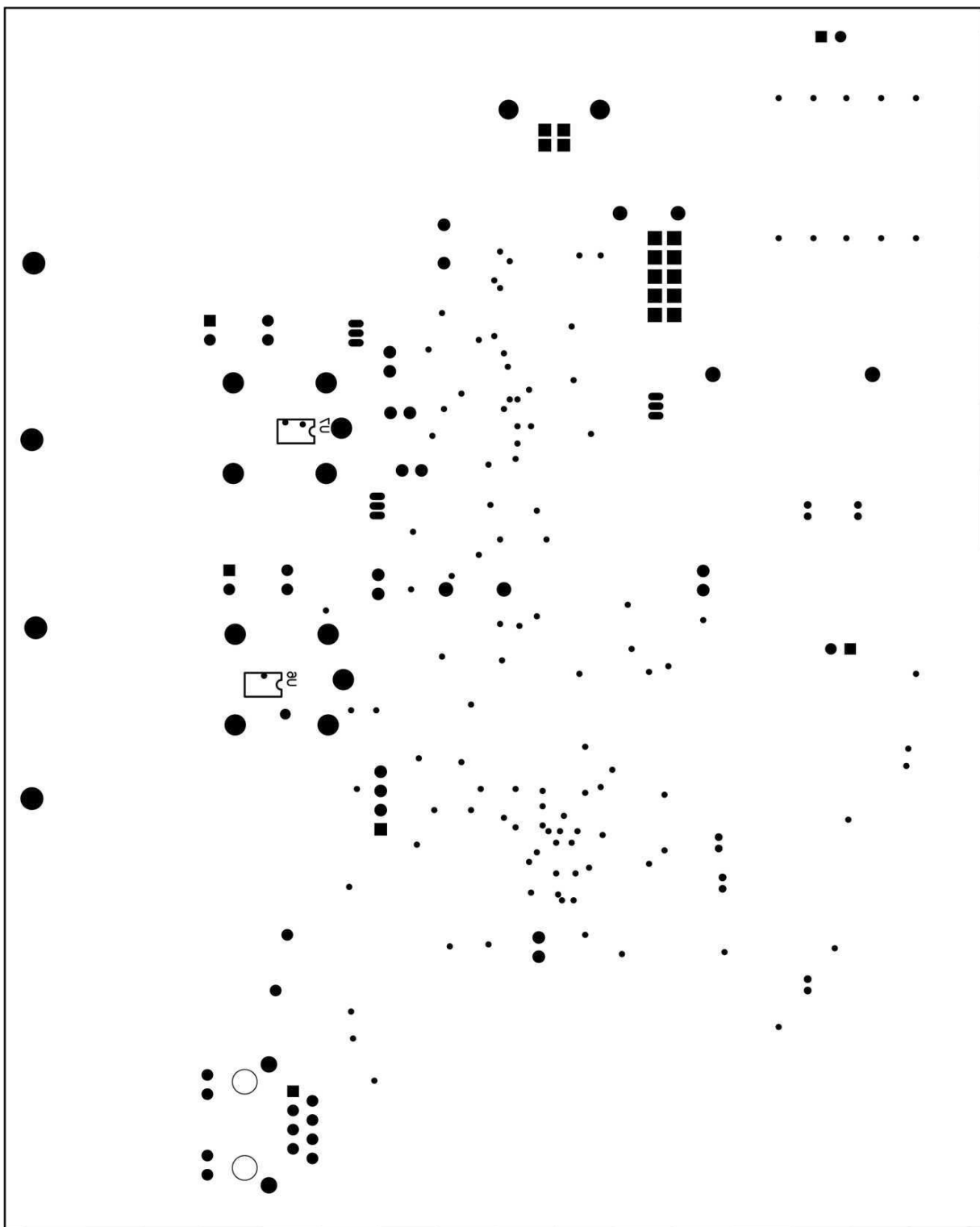


Рисунок Д.4 – Расположение элементов на нижней части печатной платы