

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

---

(наименование института полностью)

Кафедра «Промышленная электроника»  
(наименование)

11.04.04 Электроника и нанoeлектроника  

---

(код и наименование направления подготовки)

Электронные приборы и устройства  

---

(направленность (профиль))

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)**

на тему «Разработка и исследование системы автоматического полива с автономным водозабором»

Студент

И.В. Певнев

(И.О. Фамилия)

---

(личная подпись)

Научный  
руководитель

к.т.н., доцент, М.В. Позднов

(И.О. Фамилия)

Тольятти 2020

## Содержание

Введение.....	4
1 Анализ актуальности темы и выбор принципа автоматического полива.....	6
2 Обзор имеющихся на рынке систем автоматического полива растений.....	12
2.1 Прибор для полива компании Aquarius.....	13
2.2 Прибор для капельного орошения Aqua Planet-60.....	15
2.3 Устройство автоматического орошения ОГО-Родник – 1.....	17
2.4 Устройство для полива растений GARDENA.....	19
2.5 Устройство для полива растений Rachio 3.....	21
2.6 Таблица сравнения систем автоматического полива растений.....	24
2.7 Способы и устройства, которые будут применяться в системе автоматического орошения растений.....	25
3 Задача и цель диссертационной работы.....	26
4 Разработка структурной схемы устройства.....	27
5 Раздел описания технической части.....	30
5.1 Применение комплектующих для проекта системы автоматического орошения.....	30
5.1.1 Выбор подходящего микроконтроллера.....	30
5.1.2 Плата управления двигателями.....	41
5.1.3 Двигатель для полива и забора воды.....	44
5.1.4 Датчик для определения влажности почвы.....	45
5.1.5 Устройство для определения влажности воздуха и температуры DHT11.....	46
5.1.6 Датчики для создания удалённого доступа к система автономного орошения.....	49
5.1.7 Датчик температуры DS18B20.....	52
5.1.8 Кулер для контролирования климата в теплице.....	53

5.1.9 Виды электропитания к подключению системы автоматического полива растений .....	54
5.1.10 Модуль реального времени DS3231.....	62
5.1.11 Подогрев почвы.....	64
5.2 Программное обеспечение для проекта автоматического орошения растений .....	66
5.2.1 Программное обеспечение Arduino IDE.....	66
5.2.2 Программа для моделирования Tinkercad.....	68
5.3 Разработка принципиальной схемы устройства .....	69
6 Разработка проекта системы автоматического орошения.....	71
6.1 Используемые комплектующие для проекта .....	71
6.2 Создание макета системы автоматического орошения.....	72
6.2.1 Написание программы для отдельных видов культурных растений .....	73
6.2.2 Подключение каждого датчиков в отдельности к контроллеру Arduino UNO.....	74
6.3 Моделирование системы автоматического орошения в программе Tinkercad.....	83
Заключение .....	93
Список используемой литературы .....	95
Приложение А «Программа системы автоматического полива растени»	99

## Введение

В современном развивающемся мире, где каждый день происходят новые открытия в науке и технике для улучшения уровня жизни человека и автоматизации процессов в различных сферах деятельности человека. В нынешнее время имеются немало устройств и приборов, которые дают возможность сделать жизнь человека проще на работе или дома, а также экономить время и деньги. Данные приборы человек использует как дома такие как: автоматические пылесосы, мультиварка, климатические системы, автопилоты на автомобилях, так и на работе, где автоматизация процессов один из главных критериев повышения эффективности работы человека и производства.

Экономика многих стран зависит от сельского хозяйства. В этом исследовании важно сосредоточиться на некоторых жизненно важных характеристиках, таких как соответствующее количество электроэнергии, а также водоснабжения и подходящий график для полива сельскохозяйственных культур. Фермеры сталкиваются с проблемами в достижении этих стандартов, особенно тех, кто живет в нищете. В нынешнее время, когда у всех в кармане есть портативное устройства в виде телефона, с помощью которого можно осуществлять удаленную связь и вносить изменение в работу системы автоматического орошения растений. Это необходимо для уменьшения количества работников обслуживающие промышленные теплицы, повысит контроль качества, уменьшит количество используемой воды для полива что соответственно уменьшит количество потребляемого электричества, также эта система способна увеличить сельскохозяйственное производство, а также свести к минимуму ручное вмешательство человека в автоматическую систему полива растений. Все эти особенности делают эти исследования необходимыми для улучшения эффективность сельского хозяйства, и сокращения человеческого труда.

С каждым днём количество населения земли растёт и каждому человеку необходимо питания. Для этого существуют фермы для выращивания животных, а также промышленные теплицы для получения сельскохозяйственных растений. В сфера выращивания растений занимаются 9 % населения России, а цена валовых товаров сельского хозяйства превысила 5 триллионов рублей в 2018. Поэтому было принято решение провести анализ существующего рынка систем полива, собрать информацию по данной тематике, и осуществить создание новой системы автоматического полива с множеством дополнительных функций, которая сможет полностью удовлетворить пользователей. Поэтому необходимо развивать данное направление с помощью систем автоматизации и осуществлять полный уход за растением, это и стало актуальностью целью данной магистерской диссертации.

Данная магистерская работа посвящена разработке автоматического полива растений, позволяющий анализировать и своевременно производить полив растений, выполнять функцию забора воды в ёмкость для подогрева жидкости, функция подогрева земли, автоматически подстраивается под разные виды сельскохозяйственных культур, отопление помещения, возможность дополнительного освещения растений, а также выполнять проветривание помещения и быть полностью автономным устройством.

Во время выполнения данной выпускной диссертационной работы было выполнено разработка принципиальной схемы устройства, написание программного алгоритма работы в программе Arduino IDE, моделирование проекта в программе Tinkercad, изготовление прототипа системы автоматического полива на базе использования контроллера Arduino UNO, с возможностью удалённого доступа с помощью сетей Internet.

## **1 Анализ актуальности темы и выбор принципа автоматического полива.**

В нынешнее время, когда любой процесс работы человека хотят уменьшить или полностью заменить и сделать работу автоматической. Автоматизация используется в различных видах деятельности человек для получения более качественного продукта за меньший промежуток времени. К примеру, во время производства автомобилей большую часть работу выполняют роботы, которые отвечают за сварку автомобилей, покраску, сборку как это показано на рисунке 1.1. Применение автоматизации возможно в любых сферах жизни человека, как в медицине, строительстве, в военной технике так и в сельском хозяйстве.



Рисунок 1.1 – Применения роботов при изготовлении автомобиля

Применение автоматизации в сельском хозяйстве возможно при выращивании растений. Такие процессы как полив растений, проветривание помещения, подогрев грунта, контроль влажности воздуха, все это может быть автоматизировано что в свою очередь уменьшает нагрузку на человека. Что бы растение быстро и комфортно росло для этого нужно сделать благоприятные условия. Для этого и необходима система автоматического полива растений она самостоятельно справится с обслуживанием растений без участия человека.

В последние десятилетия в интеллектуальных сельскохозяйственных системах наблюдается быстрый прогресс. Результаты сельское хозяйство

имеет большое значение во всем мире. Действительно, в Индии, например, около 70% людей работают на жизненно важный сектор сельского хозяйства. В прошлом ирригационные системы были зависимы на мельницах для орошения фермы обычными методами. Старые системы являются основной причиной потери большого количества воды и, следовательно, могут уничтожить некоторые культуры из-за отсутствия достаточного количества воды.

Однако, с последними технологическими разработками появились инновационные системы для ирригации без вмешательства фермера в процесс полива. Поскольку зачастую для полива растений не хватает воды, от недостатка дождя на протяжении года и нехватка подземных вод, современные системы автоматического полива уменьшат эту проблему нехватки воды.

Действительно, интеллектуальные системы доказали свою способность регулировать орошение сельскохозяйственных культур. Это должно остановить трату воды при орошении. Кроме того, это способствует минимизировать количество сотрудников что приводит к экономии денег.

Сельское хозяйство развивается от механизированных методов в двадцатом веке до автоматизации процесса в 21 веке. В области сельского хозяйства развиваются полевые работы, которые требуют высокой точности в процессах для оптимизации производства и урожая, кроме того, ограничение стоимости производства. Чтобы достичь этих показателей, должны быть внедрены системы автоматизации. Важно, чтобы производитель учитывал ранние рамки механики, так можно достичь повышенного уровня автоматизации.

В этом проекте мы пытаемся решить проблемы орошения, такие как потребление большого количества воды. Эти ошибки влияют на деревья, так как их корни также могут влияет на общий запас воды. Необходимо приложить усилия и внести вклад в достижение желаемых целей. Поэтому усилия не должны ограничиваться индивидуальными усилиями. С

увеличением мирового населения растет потребность в урожаях сельскохозяйственных культур. Кроме того, потенциал и возможности фермера в области сельского хозяйства снижаются, это касается различных предприятий, которые привлекают работников.

Прогноз в отношении численности населения мира увеличится с 7,6 млрд в 2018 году до более 10 миллиардов к 2050 году. Эффективность систем автоматического полива становится основным спросом к сокращению фермеров. Ожидаемые цели этого проекта облегчаются и упрощают систему орошения путем установки и проектирование всей системы автоматического полива, повышение производительности урожая за счет уменьшение чрезмерного увлажнения почвы.

Система автоматического полива может предотвратить орошение, если почва уже влажная, либо если прошел дождь, так как через мерный полив вредит растениям. Система будет контролировать двигатели подачи воды что не даст в свою очередь осуществлять полив. Так как система полива является автономной то для полива человеческие силы не будут задействованы, что приводит к экономии для фермеров, а также исключается человеческие ошибки. Система автоматического полива экономит так же и водные ресурсы.

Во время анализа автоматических систем полива было проанализировано множество готовых систем полива, но не одна из существующих систем не выполняла всех необходимых задач такие как автоматическое проветривание помещения, забор воды из скважины или водоёма, подогрев грунта, удаленный доступ к системе полива, и программное регулирование полива для различных видов растений. Поэтому было принято решение о разработке и создании собственной система автоматического полива растений.

В настоящее время известно три вида полива растений.

- Орошение при помощи капель





Рисунок 1.2 – Капельный полив

Данный вид полива считается наиболее распространённым. Система капельного орошения устроена следующим образом – на необходимом участке полива прокладывается трубы, в которых находится вода, необходимо брать металлические или пластиковые трубы что бы в зимний период времени они не испортились, а в месте для полива устанавливается капельный клапан.

Недостатки капельного орошения в том, что для полива необходима чистая без загрязнений вода так как отверстия для полива очень узкое и оно со временем будет неисправно. Поэтому в начале подачи воды необходимо устанавливать фильтр.

- Дождевой полив растений



Рисунок 1.3 – Дождевой полив

Принцип данного полива заимствован у природы. Данная система полива растений копирует природный дождь. Обычно эту систему устанавливают под крышу всей теплицы, в этом случае капли воды попадают на листья и овощи, и растение получает дополнительное питание из воздуха и почвы. Дождевой полив очень прост в установке для этого необходимо просто установить разбрызгиватели под крышей теплицы.

Из недостатков данной системы можно так же отнести необходимость чистой воды, для этого нужно установить фильтр в начале подачи воды, а также вода должна подаваться под большим давлением. Данный вид орошения отлично подойдет для сельскохозяйственных растений, которые любят влажный воздух на пример огурцы.

- Внутрипочвенное орошение.



Рисунок 1.4 – Внутрипочвенный полив

Внутрипочвенный полив растений заключается в том, что под землю закапывается труба с отверстиями через которые влага и попадает в почву. При помощи ровного распространения воды грунт всегда влажный, а растение получает нужное питание. В при выборе данного метода влага быстро попадает к корням растения и обогащает их влагой, и лишняя вода не используется. При внутрипочвенном поливе подача воды должна быть подана под низким давлением.

После того как мы изучили основные виды полива растений, которые можно использовать в системе автоматического полива растений мы пришли к выводу что наиболее подходящий вариант — это капельное орошение.

## **2 Обзор имеющихся на рынке систем автоматического полива растений.**

В наше время имеются много устройств, приборов, элементов техники, которые помогают улучшить жизнь человека как дома, так и на работе, одна из таких устройств — это автоматический полив. В нынешнее время существует множество готовых систем автоматического полива, которые представлены на рынке.

Тем не менее состояние тепличных растений зависит от своевременного орошения и ухода, который трудно обеспечить современному человеку, если нет знаний по уходу за сельскохозяйственными растениями их с легкостью можно уничтожить, так же современному человеку приходится долго вне дома а растение за это время может засохнуть. Поэтому необходимо использовать устройства, которые в отсутствие человека смогут полностью ухаживать за растениями.

В процессе выполнения данной работы был проведен анализ существующих на рынке устройств системы автоматического полива растений.

Критерии, по которым был проведен анализ существующих устройств:

- удобство при эксплуатации
- качество
- экономичность
- стоимость
- возможность подключения через Bluetooth, Wi-Fi, Internet

Были выбраны устройства из разных категорий что бы объективно оценить каждый вид прибора.

## 2.1 Прибор для полива компании Aquarius

«Контроллер регулирует конструкцией орошения на садовых участках, в теплицах подавая воду через электромагнитные клапаны, которые работают от непостоянного тока напряжением 24V. Регулирование и программирование контроллера совершается по сети BLUETOOTH с обычного устройства ANDROID гаджета или планшета, но для этого необходима установка приложения Aquarius. Интерактивность мобильного устройства и контроллера Aquarius устанавливается в том, что программирование каналов совершается в приложении первоначально. После присоединения мобильного устройства и контроллера по BLUETOOTH, осуществляется перенос информации в память(кэш) контроллера. Если в приложении внести изменение схемы полива, то необходимо соединение непосредственно с контроллером, чтобы внести новые данные в контроллер. Уровень безопасности от влажности IP5. Память контроллера энергонезависима, в случае остановки энергоснабжения все программы сохраняются. Контроллер обеспечен датчиком температуры и если его запрограммировать автоматически изменять время полива в зависимости от перемены температуры воздуха, то контроль за ним не обязателен. Существует погодный датчик, он работает с прерыванием полива во время осадков, в тех местах, где осадки не попадают, полив продолжает свою работу. Данный датчик подходит от всех производителей, в комплектацию он не входит (Рисунок 1.1).» [1].

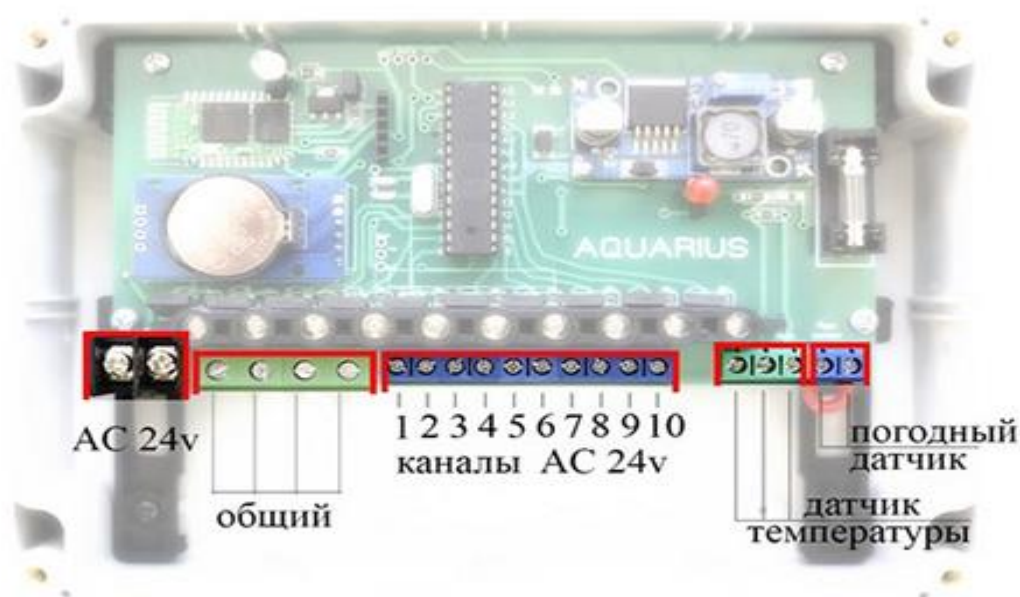


Рисунок 2.1 – Прибор AQUARIUS

«Так же к контроллеру AQUARIUS можно подключить датчик погоды он способствует прерыванию полива вовремя или после дождя. К контроллеру AQUARIUS можно подсоединить любой, погодный датчик. К важному свойству контроллера AQUARIUS можно отнести возможность присоединения датчика температуры окружающего воздушного пространства. Так же в приборе предусмотрена функция, которая определяет температуру воздуха и если температура опускается ниже положенной, то произойдет отключение системы полива. Контроллер обеспечен ПЗУ, поэтому в случае если отключится энергоснабжение, программа все данные сохранит. Возобновление полива произойдет сразу после того, как восстановится питание. К данному устройству существует приложение, благодаря которому пользователь может редактировать и устанавливать разные программы так называемое ручное управление. Так же важно знать, что так как устройство Android и контроллер связаны непосредственно через Bluetooth нужно помнить, что вовремя, когда передаются данные, устройства должны находиться рядом к друг другу, иначе связь будет

прервана». «Обычно это около несколько метров, но расстояние может и превышать заявленное количество. Все зависит от особенности используемого модуля Bluetooth вашего устройства, так же и проблема в преграде ограничивающих прямую видимость. Каждый канал программируется отдельно и для максимального времени полива для одной программы составляет 255 минут, если нужно будет поливать этот участок больше времени тогда используются сразу несколько программ, задав время начала следующего полива сразу после предыдущей программы. Рассмотрим 9-й канал, он может выступать в автоматической роли, парно включится и работать с любым с 1-го по 8-го включительно канал». «Подобным образом допускается управление подачей воды в конструкцию, включением общего клапана или включением поверхностного насоса. Непосредственно 10-й канал допускает включение клапана или иное устройство в режиме импульса. Это практично, в частности, для реализации тумана в теплице или на открытой веранде при высокой температуре. От короткого замыкания в контроллере существует защита для подключенных клапанов плавким предохранителем».

## **2.2 Прибор для капельного орошения Aqua Planet-60**

«Капельная система орошения применяется для полива овощных культур, плодовых деревьев, кустарников, даже комнатных растений и многих других зеленых насаждений во всем мире. Такой способ полива проявился и держится по сей день в таких странах, как Голландия, Турция, Германия, Израиль, Испания и многие другие страны пользуются капельным поливом в сезон огородничества».

«В этом уникальном построении системы продолжительность полива повышено до 8 часов. Коэффициент полезного действия данного рода полива овощных культур поднимает плодовой сбор в ориентировочно на 20%, если сравнивать с другими способами полива. Комплект капельного полива на 60

растений сочетает в себе все преимущества капельного полива и низкую цену. Он предназначен для установки в теплице или на грядке в огороде. Подключается к емкости от 100 литров и более. Оптимальная высота установки ёмкости 1 метр. Поливает теплой водой, нагретой на солнце. Капельный полив рассчитанный на 60 единиц растений комбинирует в себе многие преимущества полива за низкую цену. Комплект полива рассчитан для установок в огороде на открытом воздухе или в теплице. Подключение возможно к ёмкости составляющей не менее 100 литров и более. Приемлемая высота для установки ёмкости составляет один метр. Ёмкость с водой нагревается на солнце и полив происходит теплой водой. Подача воды совершается непосредственно в прикорневую область растений. Из отдельной капельницы сливается примерно 1,5-2 литра воды за 120 минут. Шланги, которые идут в комплекте капельного полива не беспричинно в черном цвете, дело в том, что в таких шлангах вода на солнце не зацветет и не будет выделяться слизь, которые засоряют капельницы. Размещение капельницы рекомендуется под растения индивидуально или ровно по грядке» [1].



2.2 Рисунок – Прибор для капельного орошения Aqua Planet-60



«Питание блока регулирования полива осуществляется через блок питания 12 В, так же от электросети 220В или от батареек АА, 1,5В. Во время первого включения система проходит тестирование, начинают загораться, чередуясь один за другим все светодиодные лампочки, срабатывает подающий насос, после его отключения через секунд 30 происходит включение реверсного насоса примерно на 5 секунд. После всего тестирования системы блок переключается в режим ожидания. Блок начнет работу в автоматическом режиме только после того, как закончится проверка. Светодиоды мигают от заданного действия. У каждого действия соответствующий светодиод, например у установленного времени полива (2 часа, 4 часа, 8 часов), в зависимости от периодичности полива (1 день, 2 дня, 3 дня, 7 дней). Если устройство находится в режиме автоматическом, то полив будет производиться в одно и тоже время при включенном блоке. Особенность устройства состоит в том, что при работе через шланги вода попадает в капельные дозаторы. Капельница представляет собой колышек с гребенкой. Результат капли образовывается при медленном просачивания воды с гребенки. Непосредственно потребление воды обуславливается от давления воды и в среднем расход воды составляет от 1,5 до 2 л примерно за 120 минут».

### **2.3 Устройство автоматического орошения ОГО-Родник – 1**

«Как написано на сайте производителя что компания была создана в 2012 году. Система полива «ОГО-Родник» определяет влажность почвы и при необходимости включает полив. Это допускает результативно потреблять воду и надлежащий полив растения, во время дождя полива не будет, а в жаркую погоду полив происходит чаще, при прохладной погоде полив будет происходить изредка. Особенность определение влажности почвы, изложенный в товарной линейке «ОГО-Родник», заключается в том что измеряется электрическое сопротивление почвы».

«При этом измерении решаются две проблемы, это проблема электрогальванического эффекта и стабильности измерения при различном коэффициенте в почве удобрения. При сборке системы автоматического полива «ОГО-Родник» не требует проведения большого количества коммуникаций и может быть установлен в любое готовое решение. Преимущество использования систему автоматического полива «ОГО-родник»: Во первое -ограниченное количество воды будет эффективно использовано особенно при долгосрочном отсутствии. Второе и основное отличие системы с датчиком влажности почвы от систем, построенных на таймерах. ОГО-Родник рассчитывает осадки в открытом грунте, температуру в теплице, при высоких и низких температурах влага из почвы испаряется по-разному, потребление воды отличается при росте растениями и плодоношении. Датчик влажности это отслеживает и не будет засухи или перелива, когда нужно будет осуществлять полив, таймеру эта функция не по силам, да и перелив растения ведёт к вымыванию удобрений и микроэлементов из почвы, что так же играет важную роль. Питание системы происходит от батарей типа АА позволяет не подключаться к сети 220в (рисунок 1.3). Не рекомендуется использовать автоматический полив ОГО-Родник-1, в водопроводе с высоким давлением, это приведёт к поломке устройства. Для водопровода с высоким давлением есть автоматический полив ОГО-Родник-3. Принцип работы автоматического полив «ОГО-Родник» состоит из блока контроллера, с измерительными щупами и электроклапана электроприводом. Контроллер благодаря щупам устанавливает влажность почвы и посылает сигнал для регулирования электроклапаном и индикатором. Индикатор дает информацию о работе устройства. Электроклапан отмыкает или замыкает воду для полива. Рационально подходит для водопроводов, где вода движется самотёком».



Рисунок 2.3 – Устройство системы автоматического орошения ОГ О-Родник

#### **2.4 Устройство для полива растений GARDENA**

Система автоматического полива GARDENA 1373 данный прибор будет эффективным для полива растений, которым нужен прикорневой полив. Система орошения GARGENA легко устанавливается в теплице и собирается. Теплицу можно использовать размером двадцать четыре квадратных метра, а количество растений может достигать до 40 штук. В данном устройстве есть возможность дополнительной установки приспособлений, которые полностью могут сделать полив автоматическим, данная возможность в комплект поставки не входит. На рисунке 1.4 показан прибор для полива растений компании GARDENA.



Рисунок 2.4 – устройство для полива растений GARDENA

Преимущество данного набора:

- нужное количество воды подается непосредственно близко к растению, что способствует экономичной трате воды.
- матер блок контролирует давление в системе до 1,5 бара, а также очищает воду от примесей.
- регулирование подачи воды происходит автоматически и достигает до 2 литров час.
- Устройство рассчитано на 40 растений.
- Простейшая сборка устройства
- В наборе предусмотрено колышки для комфортного закрепления шланга к земле.

В таблице 1 представлено комплектация для полива растений в теплице фирмы GARDENA.

Таблица 1 комплектация система полива GARDENA

№	Наименование комплектующих	Количество
1	Магистральный шланг	5 метров
2	Подающий шланг	25 метров
3	Блок мастер 1000	1 штук
4	T образные переходники для соединения	2 штук
5	Заглушка 13 мм	2 штук
6	Заглушка 4,6 мм	2 штук
7	Автоматическая капельница для полива с регулированием подачи воды	40 штук
8	Крепление шлангов к земле	20 штук
9	Деталь для прочистки капельниц	1 штук
10	T образный соединитель 13 мм	1 штук

Так же для полной комплектации и получения комфортной эксплуатации необходимо отдельно докупать:

- садовый шланг
- коннектор
- штуцер для соединения
- коннектор с автостопом
- таймер подачи воды

Данный список дополнительных комплектующий для системы полива растений GARDENA добавит к общей цене еще пять тысяч рублей.

### **2.5 Устройство для полива растений Rachio 3**

Rachio 3 — это флагманский интеллектуальный контроллер системы полива. Контроллер с самой высокой производительностью и самым высоким рейтингом, он предлагает лучший опыт в экономии воды и

удаленном поливе - все это объединено в одно простое в использовании приложение Rachio. Премиум Weather Intelligence™ Plus позволяет Rachio 3 точно прогнозировать изменения погоды и автоматически корректировать график полива в соответствии с нужным количеством воды каждый раз. Быстрая установка и встроенная справка в приложении позволят вам быстро запустить и запустить ваш контроллер. Внешнее изображение устройства для полива показано на рисунке 2.5



Рисунок 2.5 - Устройство для полива растений Rachio 3

Сократите свой экологический след с графиками полива, разработанными с учетом потребностей вашего двора, и автоматически корректируйте их в зависимости от погоды в вашем районе.

Weather Intelligence Plus использует расширенное прогнозирование погоды и сеть из более чем 300 000 метеостанций для обеспечения необходимого количества воды для ваших растений. Rachio 3 автоматически

пропускает полив, когда в прогнозе достаточно дождей, и меняется в зависимости от погоды, чтобы скорректировать ветер, минусовые температуры, фактические значения осадков и многое другое что создает для растения благоприятные условия для роста.

Система для полива растений Rachio 3 может делать соединения с телефонами на базе операционной оболочки iOS 10.3+ и Android 4.4+. Соединение прибора и телефона происходит через сеть Wi-Fi, на телефон необходимо установить дополнительное программное обеспечение для соединения с системой полива. Через данное приложение можно проводить анализ орошения растений и вносить свои пожелания для дальнейшего полива растений. Приложение Rachio работает как ваш личный помощник по поливу, чтобы гарантировать, что растения получают нужную им воду, когда им это нужно, без потерь. Приложение контролирует и оптимизирует ваш полив, так что вам не нужно. Просматривайте и управляйте зонами полива на ходу со смартфона, планшета или компьютера.

Прибор для полива Rachio 3 имеет разные модели, которые делятся на 8 зонный полив и на 16 зонный полив.

Устройство не может работать от постоянного тока. Для работы устройства необходимо блок питания, который находится в комплекте с устройством.

Комплектация устройства Rachio 3 Smart Sprinkler Controller состоит из следующих компонентов:

- Основной блок управления,
- Блок питания с 6- футовым шнуром,
- Монтажное оборудование,
- Краткое руководство пользователя.

В комплекте нет шлангов для полива, нет магистральных труб, нет капельниц для полива это все необходимо покупать отдельно.

## 2.6 Таблица сравнения систем автоматического полива растений

Проведя анализ рынка систем автоматического полива растений, можно сделать сравнение по заданным параметрам, которые представлены в нашем техническом задании. Параметра для сравнения будут:

- Удобство эксплуатации
- Качество прибора
- Стоимость
- Экономичность
- Возможность подключения к разным интерфейсам

В данной таблице 2.4 будет наглядно представлено приборы автоматического орошения. Оценивание будет проводиться по пяти бальной шкале.

Таблица 2.4 – Сравнение параметров

Наименование прибора	Удобство	Качество	Стоимость	Экономичность	Интерфейс
Прибор Aquarius	5	5	13000	4	5
Прибор Planet-60	4	3	4500	3	3
Прибор ОГО-РОДНИК	1	2	2700	2	1
Прибор GARDENA	3	4	3800	4	2
Прибор Rachio 3	4	5	25000	5	5

Сделав анализ рынка систем автоматического полива растений, можно сделать вывод что не одна из систем не может полностью удовлетворить поставленные задачи:

- обеспечить орошение растений в зависимости от влажности и температуры окружающей среды, и влажности почвы.
- делать удобное соединение с устройством по сети Wi-Fi, чтобы управлять системой полива с помощью телефона или персонального компьютера.
- делать проветривание помещение, повышенная температура может негативно сказываться на росте растений.



- обогрев почвы.
- различное программное обеспечение для различных сельскохозяйственных культур.

Проведя анализ имеющихся устройств для полива, можно сделать вывод об актуальности поставленной задачи моделирования и изготовления системы автоматического орошения растений.

## **2.7 Способы и устройства, которые будут применяться в системе автоматического орошения растений.**

После анализа готовых систем автоматического полива можно понять какие задачи будут решены в данном проекте.

Для системы орошения растений необходимо несколько вещей для составления графика полива:

- нынешняя влажность почвы;
- влажность и температура окружающей среды;
- сколько времени растение получало солнце;
- в зависимости от разновидностей растений необходим соответствующий режим полива;
- разделить полив растений в зависимости от их роста и плодоношения;

Можно использовать полив через таймер, зная производительность насосов мы можем проанализировать какое количество воды будет потрачено за один час полива, и исходя из полученных данных выстроить необходимый полив для растения. Но в данном случае мы не сможем контролировать средним объёмом воды так как это может не хватать растению влаги или наоборот быть излишним.

Полученные ранее данные из литературы по изготовлению систем автоматического орошения растений можно использовать следующий микроконтроллер Arduino UNO, а также необходимые датчики для анализа влажности почвы и температуры окружающей среды.

### 3 Задача и цель диссертационной работы

Проведя анализ систем автоматического орошения, который был выполнен выше, можно выразить цели данной исследовательской работы.

Целью данной исследовательской работы является изготовление системы автоматического орошения растений для промышленных и частных теплиц, который может выполнить необходимый полив растения в зависимости от поставленных параметров, а также производить самообучение для дальнейшей корректировки полива.

Для выполнения заданной цели нужно решить некоторые задачи:

- Обучиться возможности программированию на микроконтроллере Arduino UNO, сделать возможность синхронизации контроллера и датчиков для получения информации и управлению насосов;
- Необходимо произвести выбор необходимый датчиков для подключения к микроконтроллеру, научиться и понять возможности в области программирования в среде разработки Arduino IDE;
- Выполнить сборку и тестирование аппаратной части системы автоматического орошения растений;
- Написать программу для полива розничных видов сельскохозяйственных растений;
- Создать программу для соединения между микроконтроллером в Arduino UNO и Windows Forms, что бы можно было наглядно видеть, что происходит полив в теплице, при этом находится у себя дома;
- Моделирование системы автоматического полива в программе Tinkercad;
- Тестирование системы автоматического орошения и пуска наладка готового устройства;
- Провести и выполнить куда в дальнейшем будет развиваться системы автоматического орошения.

#### 4 Разработка структурной схемы устройства.

После составления цели и проставления задач необходимо выполнить структурную схему устройства. После анализа рынка систем автоматического полива необходимо определиться из каких комплектующих будет состоять устройство и какие задачи будет выполнять. На рисунке 4.1 представлена структурная схема прибора.



Рисунок 4.1 – Структурная схема системы автоматического орошения

Данная структурная схема делится на четыре блока:

1. Блок питания устройства

## 2. Блок датчиков состоящий из трёх модулей

- Измерение параметров почвы
- Измерение уровня воды
- Измерение температуры и влажности окружающей среды
- Измерение температуры почвы
- ESP8266
- Модуль реального времени

## 3. Блок управления представленный Arduino UNO

## 4. Блок управления двигателями состоящий из четырех двигателей:

- Двигатель отвечающий за полив растений
- Двигатель управления вентиляцией
- Двигатель забора воды
- Двигатель для подогрева почвы

«Рассмотри подробнее каждый из блоков структурной схемы, и проанализируем работу всего устройства» [1].

«Блок питания состоит из аккумуляторной батареи который подключается к блоку управления микросхемы. Благодаря которому и осуществляется питание всего устройства. Так же при желании можно установить солнечную батарею, которая будет заряжать аккумуляторную батарею, что в свою очередь позволит увеличить время автономной работы устройства. Так же можно использовать блок питания от 5В до 12В который в свою очередь будет включен в сеть 220В» [1].

«Блок датчиков состоит из трёх типов датчиков: датчики измерения влажности почвы, измерения уровня воды и измерения температуры и влажности воздуха. Благодаря данным датчикам осуществляется контроль оптимальной системы функционирования устройства для повышения выращивания растений» [1].

«Блок Arduino UNO представляет микроконтроллер, который получает сбор информации с датчиков и сохраняет её в память, и подает сигналы на

блок управление двигателями. Микроконтроллер – это специальная микросхема, используемая для контроля разными электронными устройствами. Микроконтроллер соединяет в себе сам микропроцессор, оперативное запоминающее устройство, постоянное запоминающее устройство и разную периферию внутри одного корпуса. Для разных моделей платформы Arduino используются разные микросхемы, для Arduino UNO используется микроконтроллер ATmega328. Благодаря данному блоку осуществляется работа всего устройства. Так же будет использоваться микроконтроллер ESP 8266 который необходим для создания удалённого доступа системы автоматического орошения» [1].

«Блок управления двигателями состоит из четырех двигателей, которые в свою очередь отвечают за разные действия (полив, забор воды, проветривание, подогрева почвы). Если питание от Arduino будет не хватать для питания всеми двигателями можно подключить дополнительное питание непосредственно к блоку управления двигателями» [1].

## **5 Раздел описания технической части**

В данном разделе будут описаны самые важные критерии создания системы автоматического орошения для теплиц. Описание комплектующих, микроконтроллеров, датчиков, насосов из которых будет состоять данное устройство для полива, соединение можно ими и краткое описание их работы.

### **5.1 Применение комплектующих для проекта системы автоматического орошения**

В данном разделе будет описаны какие микроконтроллеры, датчики, насосы, система питания прибора, а также из чего будет сделан удаленный доступ к устройству.

#### **5.1.1 Выбор подходящего микроконтроллера**

Микроконтроллер — это тип интегральной микросхемы (ИС) или, который часто используется в устройствах в вашем доме, которые включают в себя электронные схемы. Вы, наверное, слышали о микропроцессорах, например, Intel и они используются в качестве центрального процессора, фактически «мозга», в компьютере. Микропроцессоры представляют собой интегральные схемы, которые обрабатывают процессы в компьютерной программе, выполняют расчеты и отправляют данные в память на диск и обратно. Фото примера как выглядит микроконтроллер будет представлено на рисунке 5.1.

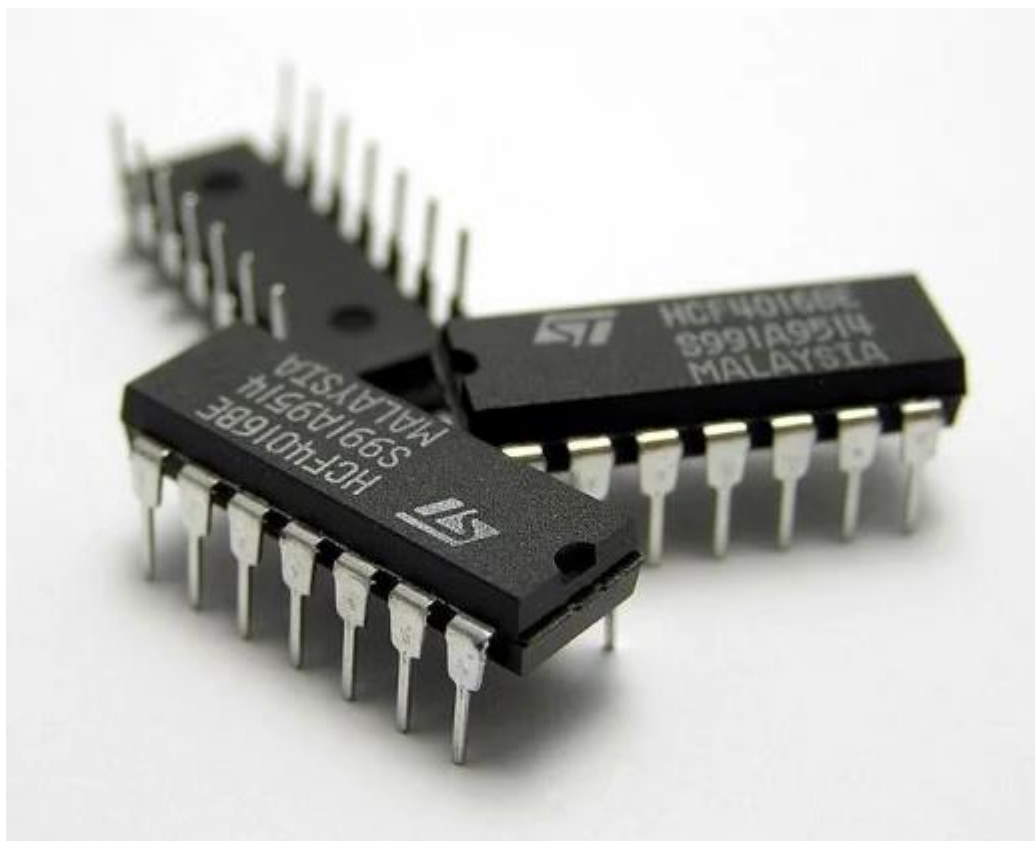


Рисунок 5.1 – Пример микроконтроллера

Микроконтроллер — это микропроцессор определенного типа. Он отличается от обычного микропроцессора следующими способами:

— Он имеет встроенные возможности ввода и вывода. Таким образом, он может читать и записывать цифровые и аналоговые значения, состояния и напрямую подключаться к «реальному миру». Микроконтроллер, в отличие от микропроцессора, может подключаться напрямую к переключателям, кнопкам, ЖК-дисплеям, светодиодам, реле и последовательным портам.

— Микроконтроллеры обычно используются для выполнения задач в оборудовании с низкой и средней сложностью. Это отличается от мощных числовых микропроцессоров, используемых в ПК, которые работают с различными программными приложениями.

— Микроконтроллеры часто используются в портативных устройствах, работающих от батарей, например, в цифровых камерах. Поэтому они часто имеют низкое энергопотребление при небольшом потреблении тока (в

отличие от теплоотвода с микропроцессором с вентиляторным охлаждением в настольном компьютере)

— По сравнению с микропроцессором в ПК, объем оперативной памяти в микроконтроллере обычно может составлять от 64 тыс. До 1 тыс.

— Программа в микроконтроллере обычно хранится в EPROM или EEPROM. Это тип энергонезависимой (программа не исчезает при выключении устройства) памяти, которую можно непрерывно стирать и перезаписывать.

Многие устройства используют микроконтроллеры. Вот пример некоторых из них:

— Охранная сигнализация включает микроконтроллер, который подключен к клавиатуре, дисплею и сенсорным, контактным входам. Микроконтроллеры, как правило, представляют собой автономные микросхемы с АЛУ (арифметическим логическим модулем), памятью и входами, выходами, содержащимися в одной интегральной схеме.

— Старые автоматические стиральные машины использовали кулачковый переключатель для последовательности операций во время цикла стирки. Это был довольно сложный переключатель, и он был установлен на конце вала ручки, которую вы использовали для выбора программы стирки. Новые машины используют микроконтроллер для последовательности операций. Другие приборы, такие как микроволновые печи и посудомоечные машины, могут включать микроконтроллер.

— В телевизорах используются микроконтроллеры для управления выбором каналов и считывания состояния кнопок на телевизоре.

— Микроконтроллеры используются для управления двигателем и отображения информации на приборной панели транспортных средств автомобиля.

— Цифровые камеры используют микроконтроллеры для обработки ввода с кнопок, управления захватом изображения и отображения.



Arduino — это аппаратно-программная платформа с открытым исходным кодом, основанная на микроконтроллерах Atmel. Открытый исходный код означает, что принципиальные схемы и исходный код программного обеспечения, используемого в проектах, находятся в свободном доступе и могут быть изменены энтузиастами. Платы для разработки Arduino с аналоговыми и цифровыми входами и выходами идеально подходят для художников, дизайнеров и любителей электроники, которые хотят собрать систему, не разбираясь в цифровых технологиях. Входные и выходные сигналы доступны на плате Arduino с помощью рядов гнездовых разъемов, в которые могут быть подключены отдельные выводы или разъемы SIL.

Совет по развитию Arduino будет иметь по крайней мере:

- цифровых выводов, которые могут быть входными / выходными каналами. Некоторые из них могут быть настроены как выходы с широтно-импульсной модуляцией. ШИМ-сигнал — это прямоугольная волна, длительность импульса которой можно варьировать. ШИМ используется для управления скоростью и положением двигателей и сервоприводов в робототехнике и приложениях дистанционного управления.

- аналоговых входных канала.

- По крайней мере, один последовательный порт, который также можно использовать для загрузки кода в Arduino

Некоторые платы также имеют ряд аналоговых выходных каналов.

Существует высокая степень гибкости в отношении функции выводов, некоторые из которых могут быть настроены как аналоговые или цифровые. Цифровые контакты могут быть настроены как вход или выход.

Платы Arduino воспринимают окружающую среду, поскольку они получают данные от датчиков, подключенных к этим аналоговым и цифровым входам. Они также могут управлять приводами, такими как двигатели, сигнальные устройства и электрические клапаны, или включать светодиоды, лампы или другие устройства визуальной

индикации. Возможности выходного привода выходов ограничены, поэтому обычно между транзистором вывода Arduino и управляемым устройством должны использоваться транзисторы, FETS или реле. Однако выход способен напрямую управлять светодиодами.

Разъемы на плате расположены стандартным образом так, что экраны могут быть подключены. Экраны — это модули со специальной функцией, такие как:

- Инфракрасный выход;
- Bluetooth;
- GSM;
- WIFI;
- Ethernet;
- управление включением / выключением двигателя с помощью реле;
- управление шаговым двигателем.

Поскольку код и оборудование Arduino имеют открытый исходный код, сторонние разработчики создали клоны официальных плат Arduino.

Платы Arduino имеют от 32 до 512 Кб флеш-памяти, которую можно использовать для хранения программ

Arduino может быть использована для создания множества интересных вещей. Вы можете легко добавить ЖК-дисплеи, клавиатуру или семь сегментов дисплея и быстро создать простую систему. Много информации и библиотек доступны на веб-сайте [Arduino.cc](http://Arduino.cc). Вы можете создать собственное дополнение к электронике или использовать щиты для построения модульной системы. Arduinos из-за их универсальности может использоваться для простых, обыденных задач или сложных приложений. Примером простого приложения может быть возможность включения устройства, когда температура в помещении падает ниже определенного уровня. Не будет слишком сложно расширить это приложение и добавить модуль GSM, который может отправлять текстовые

сообщения. Платы Arduino можно использовать для реализации сложной системы домашней автоматизации, которой можно управлять через смартфон.

На рынке микроконтроллеров Arduino существует большой выбор различных видов плат такие как:

- Arduino Micro
- Arduino Nano
- Arduino UNO
- Arduino Mega

И многие другие доработки с различными дополнительными встроенными функциями. Рассмотрим некоторые из них и сделаем вывод какой вариант Arduino нам подходящий для использования в проекте систем автоматического полива растений.

**Arduino Nano** — это небольшая, совместимая, гибкая и дружелюбная к макету плата микроконтроллера, разработанная итальянским Arduino.cc на основе ATmega328. Он поставляется с точно такой же функциональностью, как в Arduino UNO, но довольно небольшого размера. Он поставляется с рабочим напряжением 5В, однако входное напряжение может варьироваться от 7 до 12В. Arduino Nano содержит 14 цифровых выводов, 8 аналоговых выводов, 2 сброса и 6 выводов питания. Изображение Arduino Nano представлено на рисунке 5.2.



## Рисунок 5.2 – Arduino Nano

Каждый из этих цифровых и аналоговых выводов назначается с несколькими функциями, но их основная функция должна быть сконфигурирована как вход или выход. Они действуют как входные контакты, когда они соединены с датчиками, но, если вы управляете нагрузкой, используйте контакты как выходные. Arduino Nano поставляется с кварцевым генератором с частотой 16 МГц. Он используется для производства часов с точной частотой с использованием постоянного напряжения.

У Arduino Nano есть одно ограничение: он не поставляется с разъемом питания постоянного тока, что означает, что вы не можете подавать питание от внешнего источника питания. Эта плата не использует стандартный USB для соединения с компьютером, вместо этого она поставляется с поддержкой Mini USB. Крошечный размер и дружелюбный характер макета делают это устройство идеальным выбором для большинства областей применения, где размер электронных компонентов имеет большое значение.

Флэш-память составляет 16 КБ или 32 КБ, что зависит от платы Atmega, т.е. Atmega168 поставляется с 16 КБ флэш-памяти, в то время как Atmega328 поставляется с флэш-памятью 32 КБ. Флэш-память используется для хранения кода.

Эта плата очень похожа на другие платы Arduino, доступные на рынке, но небольшим размером выделяет эту плату среди других.

Далее рассмотрим самый популярный вид плат среди Arduino.

**Arduino Uno** — это плата микроконтроллера, разработанная Arduino.cc, которая представляет собой электронную платформу с открытым исходным кодом, в основном на основе микроконтроллера AVR Atmega328.

Текущая версия Arduino Uno поставляется с интерфейсом USB, 6 аналоговыми входами, 14 цифровыми портами ввода / вывода, которые используются для подключения к внешним электронным цепям. Из 14

портов ввода / вывода для вывода ШИМ можно использовать 6 контактов. На рисунке 5.3 представлен внешний вид платы Arduino UNO.



Рисунок 5.3 – Arduino UNO

Это позволяет разработчикам контролировать и воспринимать внешние электронные устройства в реальном мире.

Эта плата поставляется со всеми функциями, необходимыми для запуска контроллера, и может быть напрямую подключена к компьютеру через USB-кабель, который используется для передачи кода на контроллер с помощью программного обеспечения IDE (Integrated Development Environment), в основном разработанного для программирования Arduino. IDE одинаково совместима с системами Windows, MAC или Linux, однако Windows предпочтительнее использовать. Языки программирования, такие как C и C ++, используются в IDE.

Помимо USB, для питания платы можно использовать батарею или адаптер переменного тока в постоянный.

Платы Arduino Uno очень похожи на другие платы семейства Arduino с точки зрения использования и функциональности.

Существует множество версий плат Uno, однако Arduino Nano V3 и Arduino Uno являются наиболее официальными версиями, которые поставляются с 8-битным микроконтроллером Atmel от AVR Atmega328, где объем оперативной памяти составляет 32 КБ.

Когда характер и функциональность задачи становятся сложными, в платы можно добавить SD-карту Micro, чтобы они хранили больше информации.

Arduino Uno поставляется с интерфейсом USB, т.е. на плате добавлен порт USB для развития последовательной связи с компьютером.

Микроконтроллер Atmega328 размещен на плате, которая поставляется с рядом функций, таких как таймеры, счетчики, прерывания, ШИМ (Широтно-импульсная модуляция), ЦП (центральный процессор), данные ввода / вывода, все это работает на тактовой частоте 16 МГц, которая помогает производить больше частоты и количества команд за такт.

Это платформа с открытым исходным кодом, где каждый может модифицировать и оптимизировать плату в зависимости от количества инструкций и задач, которые он хочет достичь.

Эта плата оснащена встроенной функцией регулирования, которая контролирует напряжение при подключении устройства к внешнему устройству.

Кнопка сброса добавляется на плату, которая сбрасывает всю плату и запускает запущенную программу на начальном этапе. Этот вывод полезен, когда плата зависает в середине работающей программы; нажатие на этот штифт очистит все в программе и запустит программу с самого начала.

В плату встроено 14 цифровых входов / выходов и 6 аналоговых выводов, что позволяет осуществлять внешнее соединение с любой цепью

платы. Эти контакты обеспечивают гибкость и простоту использования для внешних устройств, которые могут быть подключены через эти контакты. Для подключения устройств к плате не требуется жесткий и быстрый интерфейс. Просто подключите внешнее устройство к контактам платы, которые расположены на плате в форме заголовка.

Флэш-память объемом 13 КБ используется для хранения количества инструкций в виде кода.

Для включения платы требуется всего 5 В, что может быть достигнуто непосредственно с помощью USB-порта или внешнего адаптера, однако она может поддерживать внешний источник питания до 12 В, который можно регулировать и ограничивать до 5 В или 3,3 В в зависимости от требования проекта.

Так же существует менее популярная версия, которая называется *Arduino Mega*. Эта версия необходима для более больших проектов.

**Arduino Mega** - это микроконтроллерная плата на базе Atmega2560. Он поставляется с большим объемом памяти и выводов ввода / вывода по сравнению с другими платами, доступными на рынке. На рисунке 5.4 будет представлен внешний вид *Arduino Mega*.



Рисунок 5.4 – Arduino Mega

«На плате имеется 54 цифровых входа / выхода и 16 аналоговых контактов, которые делают это устройство уникальным и выделяющимся среди других. Из 54 цифровых входов / выходов 15 используются для ШИМ (широотно-импульсная модуляция). На плате добавлен кварцевый генератор с частотой 16 МГц. Эта плата поставляется с портом USB-кабеля, который используется для подключения и передачи кода с компьютера на плату» [24].

«Разъем питания постоянного тока связан с платой, которая используется для питания платы. В некоторых версиях платы Arduino эта функция отсутствует, например, в Arduino Pro Mini отсутствует разъем питания постоянного тока. Эта плата поставляется с двумя регуляторами напряжения, то есть 5 В и 3,3 В, которые обеспечивают гибкость регулирования напряжения в соответствии с требованиями по сравнению с Arduino Pro Mini, который поставляется только с одним регулятором напряжения. Между Arduino Uno и Arduino Mega нет особой разницы, за исключением того, что у Arduino Mega больше места для памяти, больше размер и больше выводов ввода / вывода» [24].



«Программное обеспечение Arduino под названием Arduino IDE используется для программирования платы, которая является распространенным программным обеспечением, используемым для всех плат, принадлежащих семейству Arduino. Наличие Atmega16 на плате отличает его от Arduino Pro Mini, который использует USB-последовательный преобразователь для программирования платы» [24].

«Проведя анализ существующих на рынке плат с микроконтроллером, можно сделать вывод для проекта систем автоматического орошения более подходящий микроконтроллер на плате Arduino UNO. Так как нам будет хватать наличия доступную ввода / вывода, а объём готовой программы поместится в память EEPROM» [24].

### **5.1.2 Плата управления двигателями.**

Для системы автоматического полива растений необходимо расширительная плата для подключения к ней:

- Двигателя для полива растений;
- Двигателя для забора воды;
- Двигателя для проветривания помещения;
- Двигателя для прогрева почвы.

Поэтому необходимо найти плату для управления четырьмя двигателями и что бы она могла соединяться с Arduino UNO.

Для этого проекта отлично подойдет расширительная плата для Arduino, а называется она L293D Motor Shield.

Если вы планируете собрать новый проект, вам, в конечном счете, захочется научиться управлять различными двигателями, такими как двигатели постоянного тока, шаговые двигатели и сервоприводы. Один из самых простых и недорогих способов сделать это - соединить L293D Motor Driver Shield с Arduino. Это полнофункциональный моторный щит - идеально подходит для многих проектов роботов и ЧПУ. На фото 5.5 изображено фото L293D Motor Driver Shield.

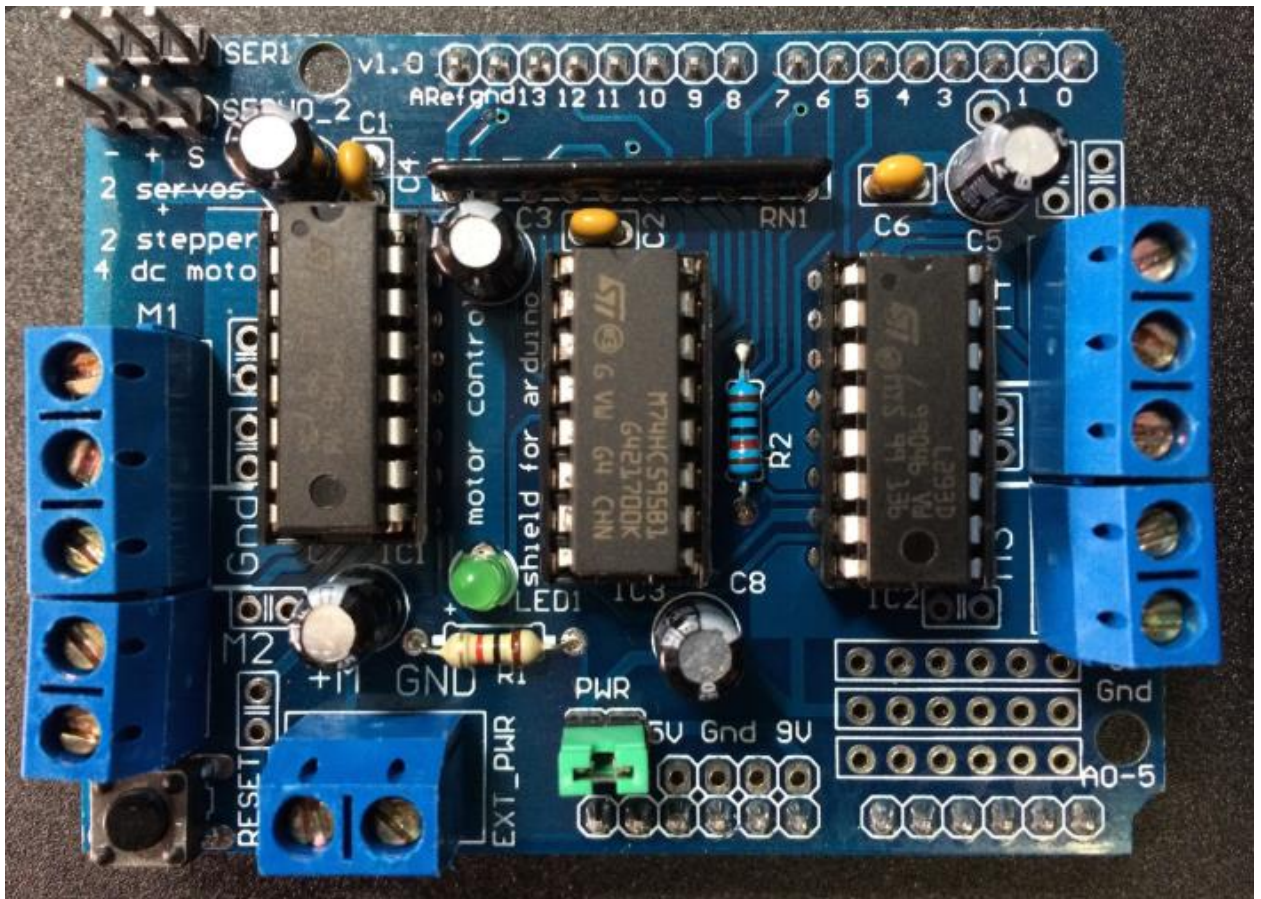


Рисунок 5.5 – Фото L293D Motor Driver Shield

Он может управлять:

- 4 двунаправленных двигателя постоянного тока с 8-битным выбором скорости (0-255)
- 2 шаговых двигателя (однополярный или биполярный) с одинарной, двойной катушкой, с чередованием или микрошагом.
- 2 серводвигателя

L293D — это двухканальный драйвер двигателя H-Bridge, способный управлять парой двигателей постоянного тока или одним шаговым двигателем.

Поскольку экран поставляется с двумя наборами микросхем драйверов двигателя L293D, это означает, что он может по отдельности приводить до четырех двигателей постоянного тока, что делает его идеальным для создания четырехколесных роботизированных платформ.

Существует три варианта подачи питания для двигателей через щит.

— Один источник питания постоянного тока для Arduino и двигателей: если вы хотите иметь один источник питания постоянного тока для Arduino и двигателей, просто подключите его к разъему постоянного тока на Arduino или к 2-контактному блоку EXT\_PWR на экране. Установите переключатель на щитке двигателя. Этот метод можно использовать только при напряжении питания двигателя менее 12 В.

— Arduino питается от USB, а двигатели - от источника постоянного тока. Если вы хотите, чтобы Arduino отключался от USB, а двигатели - от источника постоянного тока, подключите USB-кабель. Затем подключите питание двигателя к блоку EXT\_PWR. Не ставьте переключатель на защиту.

— Два отдельных источника питания постоянного тока для Arduino и двигателей: если вы хотите иметь 2 отдельных источника питания постоянного тока для Arduino и двигателей. Подключите источник питания для Arduino к разъему постоянного тока и подключите источник питания двигателя к блоку EXT\_PWR. Убедитесь, что переключатель снята с экрана двигателя.

В качестве бонуса щит предлагает следующие функции:

— Экран поставляется с массивом резисторов, обеспечивающих отключение двигателей при включении питания.

— Встроенный светодиодный индикатор показывает, что питание двигателя в порядке. Если он не горит, двигатели не будут работать.

— Сброс — это не что иное, как кнопка сброса Arduino. Данная кнопка установлена на корпусе шилда для удобства использования.

«Выходные каналы обеих микросхем L293D выведены на край экрана с помощью двух 5-контактных винтовых клемм, а именно. M1, M2, M3 и M4. К этим клеммам можно подключить четыре двигателя постоянного тока напряжением от 4,5 до 25 В. Каждый канал модуля может подавать до 600 мА на двигатель постоянного тока. Однако величина тока, подаваемого на двигатель, зависит от источника питания системы» [23].

Вы также можете подключить два шаговых двигателя к выходным клеммам. Один шаговый двигатель к порту двигателя M1-M2 и другой к M3-M4.

Клемма GND также предоставляется, если у вас есть однополярный шаговый двигатель. К этому терминалу можно подключить центральные отводы обоих шаговых двигателей.

L293D Motor Shield выводит 16-битные выходные линии ШИМ на два 3-контактных разъема, к которым можно подключить два серводвигателя.

### 5.1.3 Двигатель для полива и забора воды

Water sensor это датчик который используется для обнаружения влаги:

- Воды в ёмкости
- Дождь
- Утечка воды
- Наводнение

Если влага попадает на поверхности датчика, тогда на Arduino отправляются значения, которые символизируют о наличии влаги. Полученный данные можно использовать для различных проектов.

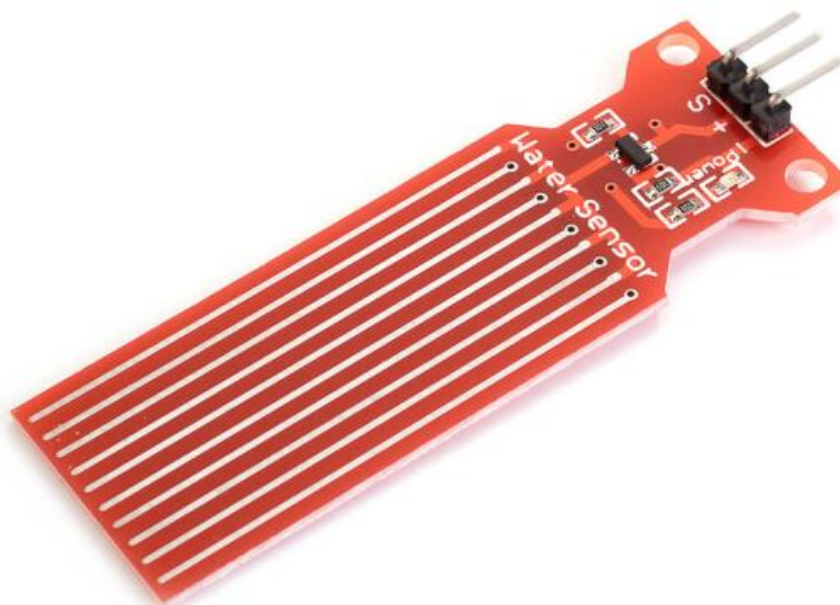


Рисунок 5.6 – Water sensor

«Данный датчик способен определить уровень воды в разных ёмкостях, где не возможен визуальный контроль, для того чтобы предотвратить переполнение емкости водой. В настоящий момент на рынке представлены различные датчики такие как поплавковый, врезные и погружные. Данный вид датчика является погружной. Чем больше он погружен в воду тем меньше будет его сопротивление между двумя соседними контактами» [24].

Характеристики датчика уровня воды:

- Напряжение – от 3.3 В до 5 В
- Выход – аналоговый
- Зона обнаружения - 16x30мм
- Рабочая температура датчика – от 10 до 30 °С

#### **5.1.4 Датчик для определения влажности почвы.**

Это простой датчик влажности почвы, предназначенный для определения влажности почвы. Если в почве не хватает воды, аналоговое значение, выводимое датчиком, уменьшится, если влажности много значения датчика будут увеличиваться. Если использовать этот датчик, чтобы сделать автоматическое устройство полива, он может определить, нуждается ли растение в поливе и не дать ему засохнуть, когда человека нет рядом. Использование датчика с контроллером Arduino делает земледелие удобнее и ваш сад умнее.

Модуль датчика влажности почвы не так сложен, как вы думаете, и если вам нужно определить влажность почву в данном проекте, это будет удобный выбор для использования.

Датчик устанавливается с двумя зондами, вставленными в почву, а потом ток проходит через почву, датчик получит значение сопротивления, считывая текущие изменения между двумя датчиками и преобразовать это значение сопротивления в количество влаги в почве. Чем выше влажность, тем более высокая проводимость у почвы.

Поверхность датчика подверглась процессу металлизации для продления срока его службы. Для его работы нужно установить датчик в

почву, а затем используйте конвертер AD, чтобы прочитать его. С помощью этого датчика Arduino может понимать, что влаги в почве мало и включит насос для полива растения. На рисунке 5.7 изображен датчик влажности почвы.



Рисунок 5.7 – Датчик влажность почвы

Общие параметры датчика влажности почвы:

- Тип датчика: аналоговый выход;
- Напряжение питания: 3,3 - 5 В;
- Выходное напряжение: 0 - 2,3 В;
- Рабочий ток: 20 мА и менее.

### **5.1.5 Устройство для определения влажности воздуха и температуры DHT11**

Влажности и температуры DHT11 датчик делает его очень легко добавить данные влажности и температуры для ваших проектов DIY электроники. Он идеально подходит для удаленных метеостанций, систем управления окружающей средой дома, а также систем наблюдения за фермой или садом.

В этом уроке я сначала немного расскажу о влажности, а затем объясню, как DHT11 измеряет влажность. После этого я покажу вам, как подключить DHT11 к Arduino, и дам пример кода, чтобы вы могли использовать DHT11 в своих собственных проектах.

Что такое относительная влажность.

DHT11 измеряет относительную влажность. Относительная влажность — это количество водяного пара в воздухе относительно точки насыщения водяного пара в воздухе. В точке насыщения водяной пар начинает конденсироваться и накапливаться на поверхностях, образующих росу.

Точка насыщения меняется в зависимости от температуры воздуха. Холодный воздух может содержать меньше водяного пара, прежде чем он станет насыщенным, а горячий воздух может содержать больше водяного пара, прежде чем он станет насыщенным.

Формула для расчета относительной влажности:

Относительная влажность выражается в процентах. При относительной влажности 100% происходит конденсация, а при относительной влажности 0% воздух полностью сухой.

«DHT11 обнаруживает водяной пар, измеряя электрическое сопротивление между двумя электродами. Датчик влажности представляет собой влагу удерживающую подложку с электродами, нанесенными на поверхность. Когда водяной пар поглощается подложкой, ионы выделяются подложкой, что увеличивает проводимость между электродами. Изменение сопротивления между двумя электродами пропорционально относительной влажности. Более высокая относительная влажность уменьшает сопротивление между электродами, в то время как более низкая относительная влажность увеличивает сопротивление между электродами» [17].

Сняв пластиковый корпус, вы можете увидеть электроды, нанесенные на основание:

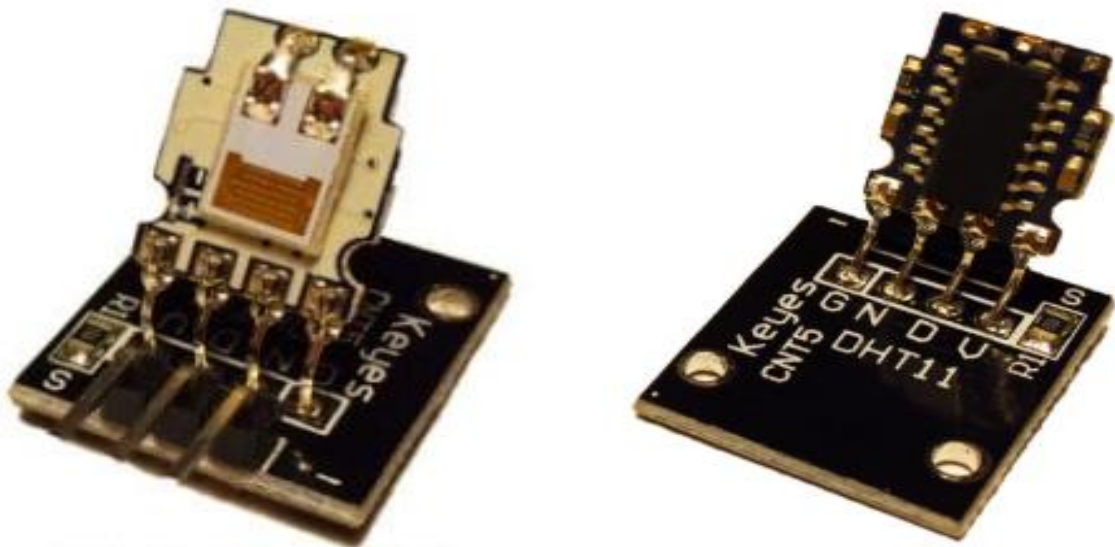


Рисунок 5.8 – Датчик относительной влажности DHT11 без пластикового корпуса

Микросхема, установленная на задней панели устройства, преобразует измерение сопротивления в относительную влажность. Он также хранит калибровочные коэффициенты и управляет передачей сигнала данных между DHT11 и Arduino.

DHT11 использует только один сигнальный провод для передачи данных в Arduino. Питание подается от отдельных проводов 5 В и заземления. Требуется подтягивающий резистор 10 кОм между сигнальной линией и линией 5 В, чтобы уровень сигнала по умолчанию оставался высоким, иногда данный резистор уже сразу установлен на корпусе датчика относительной влажности воздуха и температуры.

Есть две разные версии DHT11, с которыми вы можете столкнуться. Один тип имеет четыре контакта, а другой тип имеет три контакта и установлен на небольшой печатной плате. Версия, монтируемая на печатную плату, хороша тем, что она включает в себя поверхностный монтажный резистор 10 кОм для сигнальной линии. На рисунке 5.9 представлены два вида датчиков.





Рисунок 5.9 – Слева Датчик влажности воздуха и температуры, справа Датчик DHT11 который крепится на печатную плату

Вот диапазоны и точность DHT11:

- Диапазон влажности: 20-90% относительной влажности
- Точность влажности:  $\pm 5\%$  относительной влажности
- Диапазон температур: 0-50 ° C
- Точность температуры:  $\pm 2\%$  ° C
- Рабочее напряжение: от 3 В до 5,5 В

«Датчик необходим в автоматике управления вентиляции, то есть контролировать температуру воздуха и при необходимости открывать для проветривания помещение» [1]. Так же данный датчик будет использоваться в трубах для подогрева почвы что бы можно было определять температуру почвы.

#### **5.1.6 Датчики для создания удалённого доступа к система автономного орошения.**

ESP8266 - модуль с поддержкой Wi-Fi на кристалле, разработанный системой Espressif. Он в основном используется для разработки встроенных приложений IoT (Internet of Things).

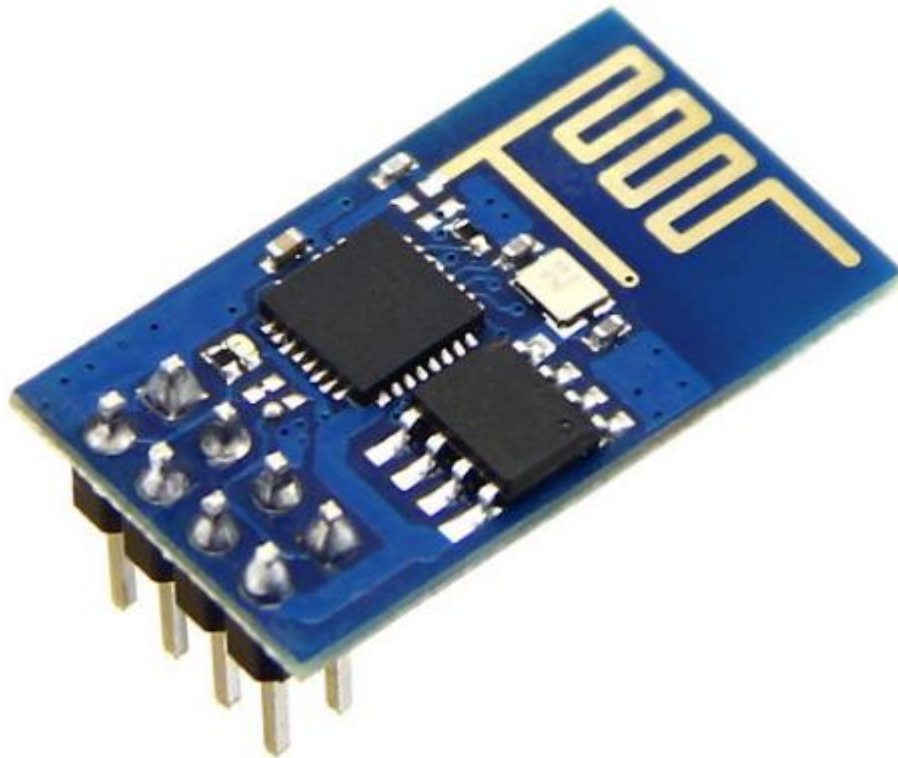


Рисунок 5.10 – Датчик ESP 8266

ESP8266 поставляется с возможностями

- 2,4 ГГц Wi-Fi,
- универсальный ввод / вывод,
- Протокол последовательной связи Inter-Integrated Circuit,
- аналого-цифровое преобразование
- Последовательный периферийный интерфейс протокол последовательной связи,
  - Интерфейс I<sup>2</sup>S (Inter-IC Sound) с DMA (прямой доступ к памяти) (совместное использование контактов с GPIO),
  - UART (на выделенных контактах плюс UART только для передачи можно включить на GPIO2), и
  - широтно-импульсная модуляция (ШИМ).

В нем используется 32-разрядный процессор RISC на базе Tensilica Xtensa L106, работающий на частоте 80 МГц. Он имеет загрузочное ПЗУ на

64 КБ, оперативную память на 64 КБ и оперативную память на 96 КБ. Доступ к внешней флэш-памяти возможен через SPI.

Модуль ESP8266 — это недорогой автономный беспроводной приемопередатчик, который можно использовать для разработки конечных точек IoT.

Для связи с модулем ESP8266 микроконтроллеру необходимо использовать набор AT-команд. Микроконтроллер связывается с модулем ESP8266-01 с помощью UART с заданной скоростью передачи.

Есть много сторонних производителей, которые производят разные модули на основе этого чипа. Таким образом, модуль поставляется с различными вариантами доступности выводов, такими как,

- ESP-01 поставляется с 8 выводами - антенной трассировки печатной платы.

- ESP-02 поставляется с 8 контактами (3 GPIO) - разъемом антенны U-FL.

- ESP-03 поставляется с 14 контактами, (7 контактов GPIO) - керамическая антенна.

- ESP-04 поставляется с 14 контактами (7 контактов GPIO).

На датчике есть следующие контакты:

- 3V3 - 3,3 В Pin питания.

- GND - Контакт заземления.

- RST - Активный низкий сброс сброса.

- EN - Активно высокий разрешающий контакт.

- TX - последовательный вывод передачи UART.

- RX - Последовательный ПИН-код UART.

Пины ввода / вывода общего назначения. Эти контакты определяют, в каком режиме модуль запускается. Он также определяет, используются ли контакты TX / RX для

программирования модуля или для целей последовательного ввода-вывода.

### 5.1.7 Датчик температуры DS18B20

Датчик температуры DS18B20 является цифровым датчиком температуры. Для связи с Arduino требуется всего одна линия данных, по которой поступают данные. На рисунке 5.11 показан внешний вид датчика DS18B20.

Он может получать питание от внешнего источника питания или получать энергию непосредственно от Arduino так называемый паразитный режим, что в свою очередь устраняет необходимость во внешнем источнике питания.

В таблице 5.1 показано, как подключить датчик DS18B20 к плате Arduino

Таблица 5.1 – подключение датчика DS18B20

№	Вывод	Применение
1	GND	Подключение минуса к датчику
2	DQ	Подключение любого цифрового вывода
3	VDD	Подключение питания 5В

Каждый датчик температуры DS18B20 имеет уникальный 64-битный серийный код. Это позволяет подключить несколько датчиков к одному проводу данных. Таким образом, вы можете получить температуру от нескольких датчиков, используя только один цифровой контакт Arduino.

краткий обзор наиболее важных характеристик температурного датчика DS18B20:

- Связь по однопроводной шине
- Диапазон электропитания: от 3,0 до 5,5 В
- Диапазон рабочих температур: от -55°C до + 125°C
- Точность +/- 0,5 °C (в диапазоне от -10°C до 85°C)

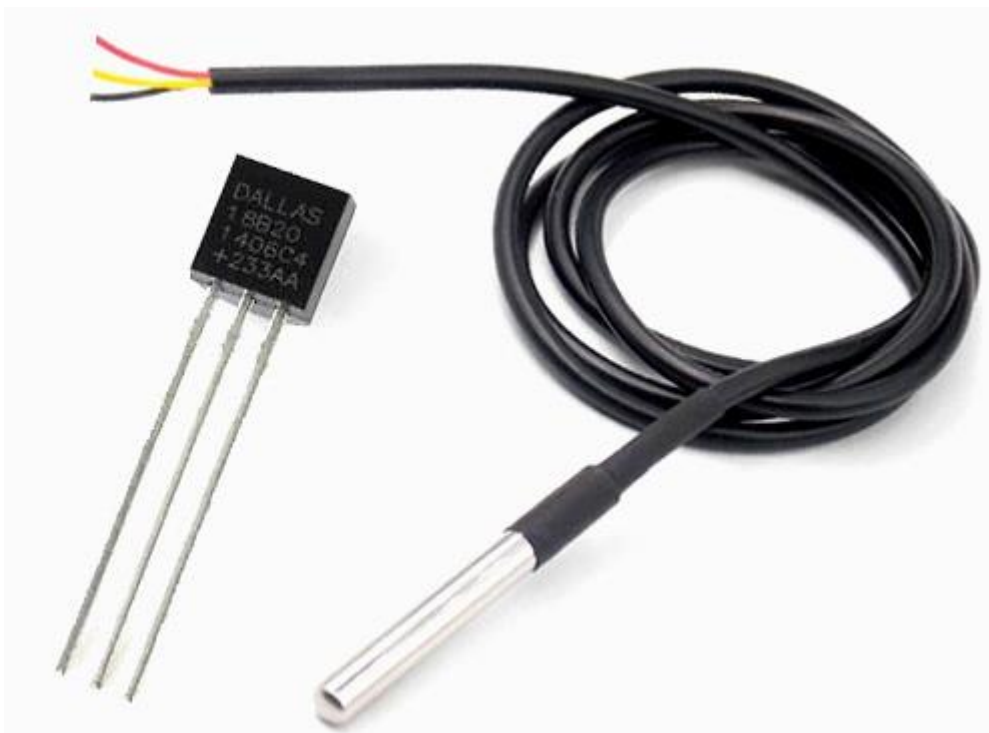


Рисунок 5.11 – Датчик температуры DS18B20

### **5.1.8 Кулер для контролирования климата в теплице**

В данном проекте будет реализована функция по проветриванию помещения, а также потребуется кулер для функции подогрева почвы. Для этого будет использоваться компьютерный кулер для охлаждения процессора.

Кулер применяется при необходимости обеспечения большего потока воздуха в контрольных точках. На рисунке 5.11 будет представлен внешний вид используемого кулера охлаждения.



Рисунок 5.11 – Кулер для проветривания

### **5.1.9 Виды электропитания к подключению системы автоматического полива растений**

Питание играет одну из важной роли в жизни электроприборов, в том числе и в автоматическом поливе. При помощи постоянного питания система полива сможет работать бесперебойно, для этого необходимо выяснить, какое питание подойдет лучше для данного устройства. Рассмотрим виды питаний, их плюсы и минусы в использовании.

Солнечная батарея- это группирование фотоэлектрических генераторов— полупроводниковых механизмов, непосредственно модифицирующих солнечную энергию в постоянный электрический ток, в отличие от солнечных коллекторов, вырабатывающих нагревание материала-теплоносителя.

Типичная система солнечной энергии включает в себя солнечные панели, инвертор, оборудование для крепления панелей на крыше и систему мониторинга производительности, которая отслеживает производство

электроэнергии. Солнечные панели собирают энергию от солнца и превращают ее в электричество, которое передается через инвертор и преобразуется в форму, которую можно использовать для питания устройства.

Солнечные батареи используют три электрода. Он имеет основание из литиевой пластины, два слоя электрода, разделенных тонким листом пористого углерода, и титановой сетки, в которой размещался чувствительный к красителю фотозлектрод. Ионы батареи окисляются в пероксид лития пористыми материалами, которые химически разлагаются на ионы лития и хранятся в виде металлического лития. В устройстве используется обычный жидкий электролит, состоящий из части соли и части растворителя.

Солнечные батареи не являются обычным типом батарей, это так называемые батареями глубокого цикла. Для фотоэлектрического (фотоэлектрического) солнечного энергетического оборудования требуется батарея для хранения энергии, накопленной в солнечные часы, для использования ночью. Эти солнечные батареи накапливают энергию от основного солнца, в то время как обычные батареи накапливают энергию, получаемую от электричества источника.



Рисунок 5.12 Солнечная батарея

Аккумулятор — это электрохимический накопитель энергии. Обычно он состоит из комбинации электрохимических элементов, так называемых гальванических элементов. Эти элементы содержат два электрода, разделенных ионопроводящим, жидким или твердым электролитом. Они состоят из разных материалов (например, литий, щелочной марганец, свинец). В зависимости от используемой химической системы аккумуляторные системы имеют разные уровни напряжения и плотности энергии. Материал, используемый для электродов, определяет, насколько высоко номинальное напряжение. Энергия, которая может быть сохранена, зависит от природы и количества материала, используемого в батарее.

Во время разряда энергия, запасенная в химической форме, преобразуется в электрическую энергию посредством электрохимической реакции и потоков электричества.

#### Литий-ионные

Литий-ионная батарея — это передовая технология, в которой ионы лития являются ключевым компонентом электрохимии. Во время цикла разряда атомы лития в аноде ионизируются и отделяются от своих электронов. Ионы лития движутся от анода и проходят через электролит, пока не достигнут катода, где они комбинируют со своими электронами и электрически нейтрализуются. Ионы лития достаточно малы, чтобы перемещаться через микро проницаемый сепаратор между анодом и катодом. Частично из-за небольшого размера лития (третье место после водорода и гелия) литий-ионные аккумуляторы способны иметь очень высокое напряжение и накопление заряда на единицу массы и единицу объема.



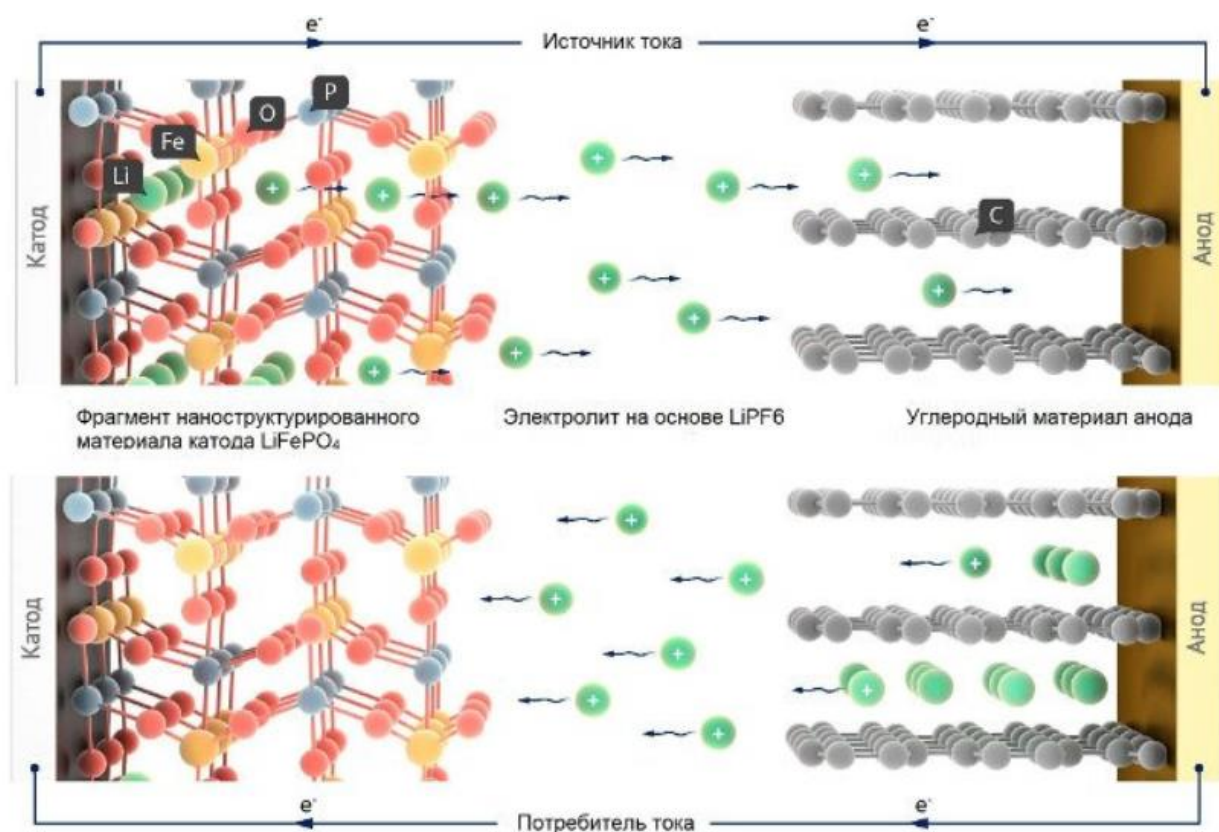


Рисунок 5.13 Электрохимическая схема разряда

Литий-ионные аккумуляторы могут использовать различные материалы в качестве электродов. Наиболее распространенной комбинацией является оксид лития-кобальта (катод) и графит (анод), который чаще всего встречается в портативных электронных устройствах, таких как мобильные телефоны и ноутбуки. Другие катодные материалы включают оксид лития-марганца (используется в гибридных электрических и электрических автомобилях) и литий-фосфат железа. Литий-ионные аккумуляторы обычно используют в качестве электролита эфир (класс органических соединений).

Преимущества литий-ионных батарей. По сравнению с другими технологиями аккумуляторных батарей высокого качества (никель-кадмиевые или никель-металлогидридные), литий-ионные аккумуляторы имеют ряд преимуществ. Они имеют одну из самых высоких плотностей энергии среди всех аккумуляторных технологий сегодня (100-265 Вт / кг или 250-670 Вт / л). Кроме того, элементы литий-ионного аккумулятора могут

выдавать до 3,6 В, что в 3 раза выше, чем у таких технологий, как Ni-Cd или Ni-MH. Это означает, что они могут подавать большие объемы тока для мощных применений, в которых литий-ионные аккумуляторы также требуют сравнительно небольших затрат на обслуживание и не требуют плановых циклов для поддержания срока их службы. Литий-ионные аккумуляторы не имеют эффекта памяти, вредный процесс, при котором повторяющиеся циклы частичной разрядки/зарядки могут заставить аккумулятор «запоминать» меньшую емкость. Литий-ионные аккумуляторы также имеют низкий уровень саморазряда, около 1,5-2% в месяц. Они не содержат токсичного кадмия, что облегчает их утилизацию, чем никель-кадмиевые батареи.

Недостатки литий-ионных аккумуляторов. Несмотря на свои технологические перспективы, литий-ионные аккумуляторы все еще имеют ряд недостатков, особенно в отношении безопасности. Литий-ионные аккумуляторы имеют тенденцию перегреваться и могут быть повреждены при высоком напряжении. В некоторых случаях это может привести к тепловому разгону и сгоранию.



Рисунок 5.14 - Литий-ионный аккумулятор

### Свинцово-кислотные

Несмотря на очень низкое отношение энергии к весу и низкое отношение энергии к объему, его способность подавать высокие импульсные токи означает, что элементы имеют относительно большое отношение мощности к весу. Эти функции, а также их низкая стоимость делают их привлекательными для использования.

Свинцово-кислотный не терпит глубоких циклов. Полный разряд вызывает дополнительную нагрузку, и каждый цикл лишает аккумулятор некоторого срока службы. Эта характеристика износа также в различной степени относится к другим химическим составам батарей. Во избежание перегрузки аккумулятора из-за повторяющейся глубокой разрядки рекомендуется использовать батарею большего размера.

Свинцово-кислотная батарея имеет одну из самых низких плотностей энергии, что делает ее непригодной для портативных устройств. Кроме того,

производительность при низких температурах является незначительной. Саморазряд составляет около 40% в год, один из лучших на аккумуляторных батареях. Для сравнения никель-кадмиевые саморазряды этого количества через три месяца. Высокое содержание свинца делает свинцовую кислоту вредной для окружающей среды.

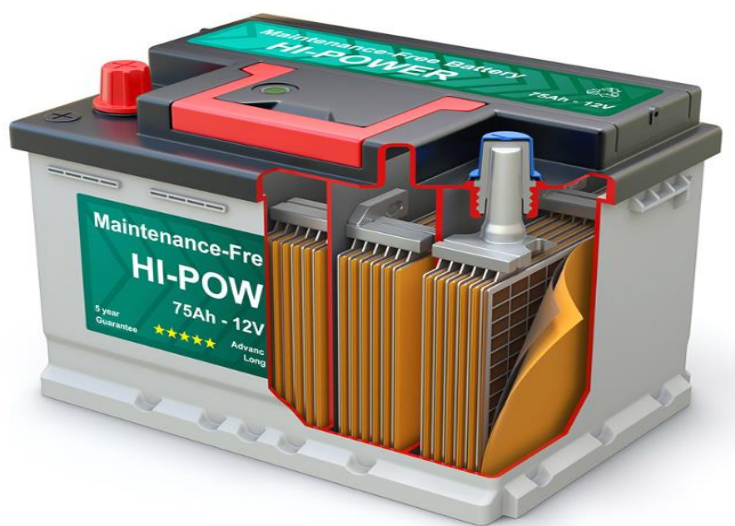


Рисунок 5.15 - Свинцово-кислотный аккумулятор

Батарея «Крона» — это батарея солевого вида, характерная ёмкость батареи- 625 мА ч, имеет напряжение 9V. Батарея «Крона» представлена как аккумуляторная, так и как обычная одноразовая батарея. Отличие между аккумуляторной и одноразовой батареи заключается в напряжении, как говорилось выше, одноразовая батарея имеет напряжение 9V, а аккумуляторная 8,4 V, а также в подзарядке и сроке службы батареи.

На рисунке ниже представлена батарея «Крона»



Рисунок 5.17 - Батарея «Крона»

Виды крон можно разделить на два типа, первый тип — это одноразовые (таблица 5.1), второй тип, это- многоразовые (таблица 5.2).

Таблица 5.2 – Одноразовые кроны

Вид	Элементов
Марганцево-цинковые солевые	6 элементов
Щелочные	6 элементов
Литий-железодисульфидные	6 элементов
Марганцево-литиевые	3 элемента
Воздушно-цинковые («Крона ВЦ»)	6 элементов

Таблица 5.3 – Многоразовые кроны

Вид	мА ч
Никель-кадмиевые	120 мА•ч (номинальное напряжение 8,4 В)
Никель-металлогидридные	170—300 мА•ч (номинальное напряжение 8,4 В)
Литий-ионные	350—700 мА•ч (номинальное напряжение 7,4 В)
Литий-железо-фосфатные	320•ч

Адаптер преобразует питание от сети переменного тока в 9-вольтовое постоянное напряжение для питания большинства инструментальных 9-вольтовых аккумуляторных устройств. Максимальный ток 500 мА - Отрицательный, положительно заряженный источник питания может подавать 500 мА для питания даже самых энергоемких устройств. Вход: 100-240 В переменного тока; Выход: 9 В постоянного тока, 500 мА.



Рисунок 5.18 - Блок питания 9 вольт

### 5.1.10 Модуль реального времени DS3231

DS3231 - это недорогой, чрезвычайно точные часы реального времени со встроенным кварцевым генератором с температурной компенсацией и кристаллом. Устройство имеет вход для батарейки и поддерживает точное хронометраж при включении питания устройства. Интеграция кристаллического резонатора повышает долговременную точность устройства, уменьшает количество деталей в производственной линии.

DS3231 доступен в коммерческих и промышленных диапазонах температур, и предлагается в 16-контактном выводе.

DS3231 поддерживает секунды, минуты, часы, день, дату, месяц и год. Дата в конце месяца корректируется на месяцы с меньшим количеством чем 31 день, включая исправления для високосного года. Часы работают в 24-часовом или 12-часовом формате с индикатором AM / PM. Два программируемых времени суток аварийные сигналы. Адрес и данные передаются последовательно через двунаправленную шину I2C.

Прецизионное эталонное напряжение с температурной компенсацией и схема компаратора контролирует состояние VCC для обнаруживать сбоя в питании, обеспечивает выход сброса и при необходимости автоматически переключаться на резервное питание.

Данный модуль отлично подойдет для нашего проекта. Так как с помощью данного модуля можно реализовать дополнительное количество воды во время набора зелени и вовремя, когда у растения созревают плоды.

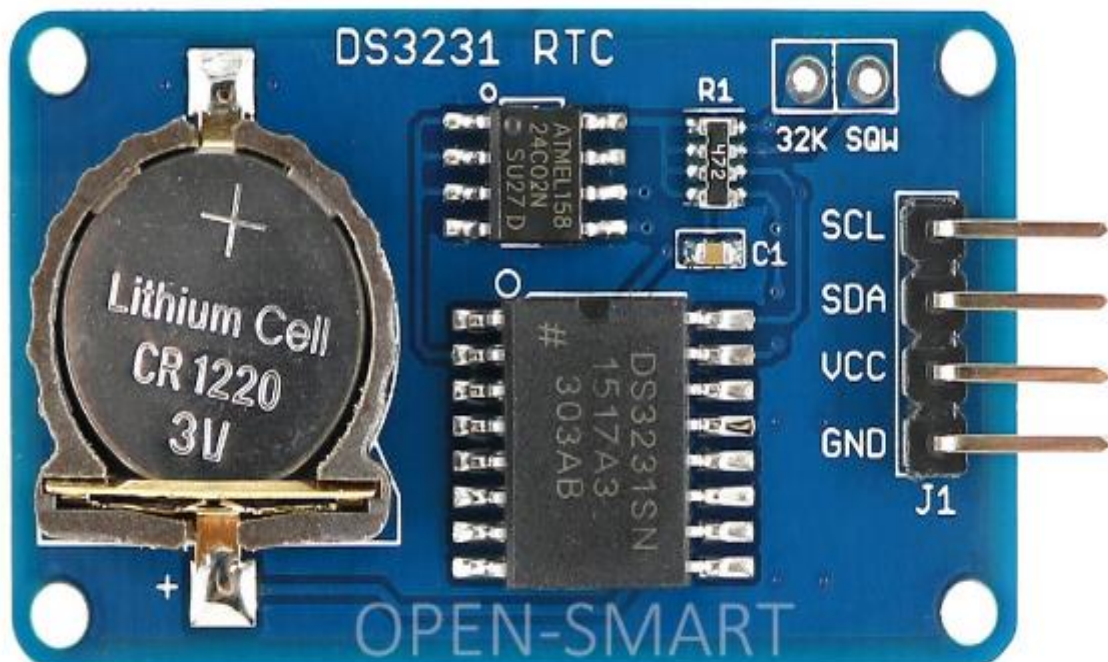


Рисунок 5.19 – Модуль реального времени DS3231

Для того что бы подключить датчик к Arduino UNO необходимо подключить к датчику питание, а также задействовать выводы SDA и SCL пример подключения датчика будет показан на рисунке 6.6.

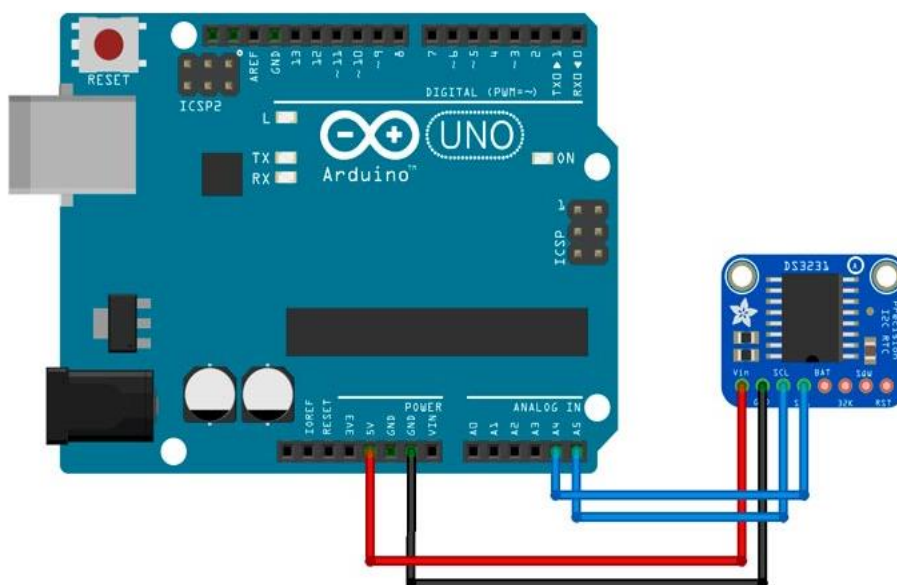


Рисунок 5.20 – Схема подключения датчика DS3231 к контроллеру Arduino UNO

Все вышеописанные электронные компоненты будут использоваться в проекте автоматического полива. Все они имеют низкую стоимость что даст возможность собрать качественный и недорогую систему автоматического орошения.

### 5.1.11 Подогрев почвы.

Важным критерием в выращивание растений является влажная и теплая почва. Как поливать и повысить влажность почвы было описано выше, а что бы подогреть ее можно использовать различные виды нагревателей.

Подогрев земли можно сделать с помощью электрический нагревательных элементов. Данный способ используется в домашних условиях при создании теплого пола.



Для этого используют специальный кабель, к которому подключается электричество и с помощью терморегулятора, температура почвы увеличиваем или уменьшаем. Что бы использовать данный метод необходимо большая мощность электричества, так как при работе нагревателя используется большое количество электричества. Так же не во всех сельскохозяйственных территориях могут быть необходимая мощность электросетей.

Другой способ подогрева почвы — это использовать газ для нагрева воды. Это более дешёвый способ подогрева почвы. Данный способ имеет большой недостаток так как природный газ очень дорогой для транспортировки по трубам, поэтому данный вариант хорошо подойдет в той местности, где уже проведен газ.

Во время разработки системы автоматического орошения при ознакомлении с данной темой подогрева почвы был разработан новый способ подогрева почвы. Для этого способа необходимо закапать в почву трубы диаметром пятьдесят миллиметров, а конец трубы вывести во внутреннюю часть крыши теплицы как это показано на рисунке 5.21.



Рисунок 5.20 – Способ подогрева теплицы с помощью теплого воздуха

В трубу необходимо установить датчик DHT11 который будет давать информацию о температуре воздуха в почве или использовать датчик температуры DS18B20 с помощью которого можно делать измерение температуры земли. При необходимости подогрева почвы кулер нагнетает теплый воздух из-под крыши в трубы который проходя по трубам подогревает почвы. Данный способ является самым экономичным так как он не использует электричество или газ во время работы, но менее эффективным, так как теплоемкость воздуха очень маленькая, но при использовании вентилятора и подогревателей можно повысить эффективность. Так же для повышения эффективности можно использовать тепловые пушки.

## **5.2 Программное обеспечение для проекта автоматического орошения растений**

В современном мире существует множество программ для написания программ для микроконтроллера Arduino UNO. Самая популярная программа имеет одноименное название Arduino IDE и является собственной разработкой компании Arduino, с помощью этой программы мы и напишем программу для проекта системы автоматического орошения. А что бы проверить правильность работы полученной программы, необходимо будет построить макет в программе Tinkercad.

### **5.2.1 Программное обеспечение Arduino IDE**

Arduino IDE (интегрированная среда разработки) отлично справляется с поставленной задачей: это простая среда разработки приложений с одним файлом. В нем достаточно встроенных инструментов, чтобы помочь достичь этой цели. Но для разработки более крупных приложений - будь то разработка библиотек Arduino или разработка новых ядер Arduino — это не сравнится с полнофункциональной средой C / C ++.

Когда мы начинаем разработку проекта Arduino, обычно мы склонны использовать Arduino IDE. Во всяком случае, есть несколько альтернатив. Как мы знаем, Arduino является одной из самых популярных плат для макетирования, для создания проекта.

Обычно Arduino IDE — это первая IDE, которую мы используем при первом приближении к Arduino. Это происходит по нескольким причинам: он прост в использовании, поддерживает все платы Arduino и имеет встроенный менеджер библиотек, который также прост в использовании. Более того, Arduino IDE очень удобен для пользователя, не имеет слишком много опций, меню и т. д., что может напугать неопытного пользователя. Это так просто, что нам не нужно беспокоиться о том, как это работает, мы можем сосредоточиться только на процессе разработки. Мы пишем код Arduino, и Arduino IDE компилирует его и загружает скомпилированный код в плату Arduino.

Более того, он имеет открытый исходный код и работает на нескольких ОС, таких как Windows, Mac OS X и Linux.

В любом случае, если вы не чувствуете себя комфортно с Arduino IDE, есть несколько альтернатив, которые вы можете использовать при разработке проекта Arduino. Но Arduino IDE является самым удобным и популярной программой для написания программы на Arduino UNO.

С момента запуска платформы с открытым исходным кодом Arduino, бренд зарекомендовал себя в центре обширного сообщества с открытым исходным кодом. Экосистема Arduino состоит из разнообразного сочетания аппаратного и программного обеспечения. Универсальность Arduino и его простой интерфейс делают его ведущим выбором для широкого круга пользователей по всему миру - от любителей, дизайнеров и художников до прототипов продуктов.

Плата Arduino подключена к компьютеру через USB, где она соединяется со средой разработки Arduino (IDE). Пользователь записывает код Arduino в IDE, а затем загружает его в микроконтроллер, который

выполняет код, взаимодействуя с входами и выходами, такими как датчики, двигатели и источники света.

И новички, и эксперты имеют доступ к огромному количеству бесплатных ресурсов и материалов для их поддержки. Пользователи могут искать информацию о том, как настроить свою доску или даже как запрограммировать на Arduino. Открытый исходный код Arduino сделал его особенно дружелюбным для новых и опытных пользователей. В сети доступны тысячи примеров кода Arduino. В этом посте мы познакомим вас с некоторыми основными принципами программирования для Arduino.

Как и в большинстве программ, существуют меню, которые позволяют выполнять различные действия, такие как создание новых файлов, их сохранение и многое другое в верхней части интерфейса программного обеспечения. Существуют значки кнопок, которые также позволяют быстро получить доступ к наиболее часто выполняемым действиям. Нажатие кнопки проверки, нет ли ошибок в вашем коде. Нажатие кнопки «Загрузить» переносит ваш код с вашего компьютера на Arduino, чтобы он мог работать на вашей плате Arduino. Есть окно, где набираете свою программу, и области сообщений, которые предоставляют информацию об этой программе.

### **5.2.2 Программа для моделирования Tinkercad**

В мире трехмерного моделирования Tinkercad зарекомендовал себя как достойное введение в автоматизированное проектирование. Это бесплатная и интуитивно понятная веб-программа САПР, которую может использовать каждый. На самом деле, если вы хотите начать работу с Tinkercad, у нас даже есть учебник для начинающих, который поможет вам в этом.

Недавно Tinkercad представил новые возможности: расширение для включения в свою конструкцию схем, называемое Tinkercad Circuits. Это приносит совершенно новую сторону в Tinkercad, вращаясь вокруг моделирования цепей с Arduino.

Arduino - платформа электронного прототипирования с открытым исходным кодом, которая также продает микроконтроллеры. Tinkercad

Circuits позволяет любому виртуально создавать и программировать проекты Arduino без физического оборудования.

Поначалу это может показаться большим и не удобной программой – но изучив ее более подробно можно прийти к выводу что данная программа удобна в использовании и информативная. Данная программа эта креативная платформа - отличный инструмент для создания прототипов.

Благодаря данной программе можно будет сделать моделирования системы автоматического орошения растения.

### **5.3 Разработка принципиальной схемы устройства**

После того как нам стало известны все компоненты, которые будут использоваться в системе автоматического полива растений можно приступить в разработке принципиальной схемы, которая наглядно покажет, как необходимо соединять датчики с платами и выводить питания на нужные двигатели. Для этого нам понадобится программа для создания схем Компас 3D. На рисунке 5.21 представлена принципиальная схема устройства.

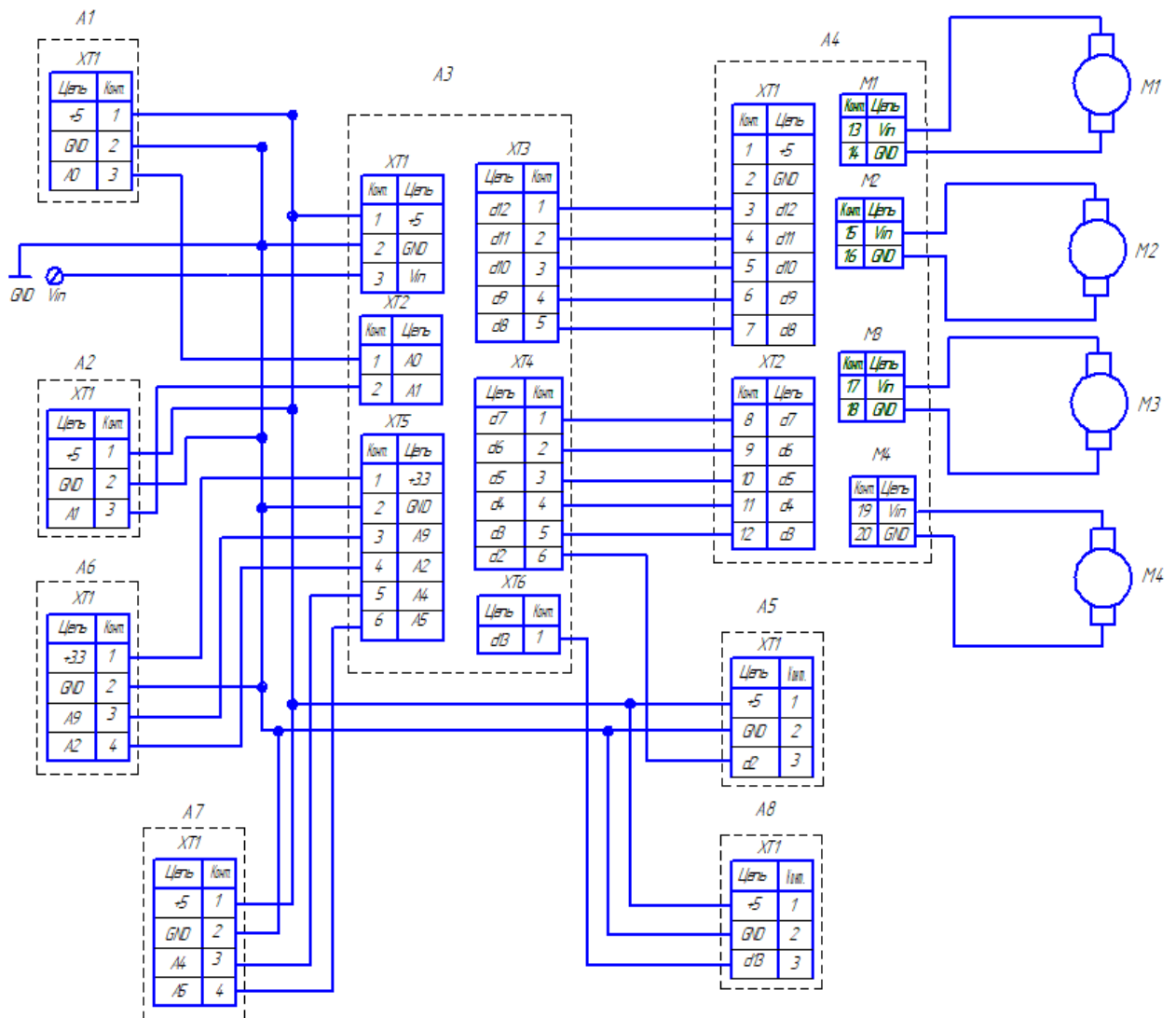


Рисунок 5.21 – Принципиальная схема проекта

Так же для понимания принципиальной схемы в таблице 5.4 представленной и описаны все комплектующие указанные на принципиальной схеме. В таблице 5.4 представлены список компонентов.

Таблица 5.4 – Список компонентов представленные в принципиальной схеме.

Обозна.	Наименование	Количество
	Блок управления	
A1	Датчик влажности почвы	1
A2	Датчик уровня воды	1
A3	Платформа Arduino UNO	1
A4	Расширительная плата Motor Shield	1

A5	Датчик температуры и относительной влажности	1
A6	Датчик удаленного доступа ESP8266	1
A7	Датчик реального времени	1
A8	Датчик температуры земли	1
	Двигатели	
M1	Мотор для полива	1
M2	Мотор для забора воды из скважины	1
M3	Мотор для проветривания теплицы	1
M4	Мотор для подогрева земли	1

## **6 Разработка проекта системы автоматического орошения**

В данном разделе будет описан процесс разработки проекта системы автоматического полива растений. Написание программного обеспечения для проекта в программе Arduino IDE. Модуляция системы в виртуальной программе Tinkercad Arduino, в которой можно будет проверить работоспособность готовой программы перед созданием проекта.

### **6.1 Используемые комплектующие для проекта**

В создании этого проекта использовались комплектующие, материалы и программное обеспечение, которое будет указано в таблице 6.1.

Таблица 6.1 – список используемых комплектующих и программного обеспечения.

№	Наименование	Количество
1	Контроллер Arduino UNO	1
2	Расширительная плата Motor Shield	1
3	Насос для забора воды и полива Submersible Motor Water	3
4	Датчик определения влажности почвы Keyestudio	1
5	Датчик уровня воды Water sensor	1
6	Датчик температуры и относительной влажности DHT-11	2
7	Кулер	2
8	Датчик температуры почвы	1
9	Датчик удаленного доступа ESP8266	1
10	Модуль реального времени	1
11	Среда разработки Arduino IDE	1
12	Программа Tinkercad	1
13	Расходные материалы	1

Описание процесса создания макета системы автоматического полива растений. Он будет проходить в пять этапов:

- Необходимость создания и тестирования отдельных частей проекта из который будет создан макет.
- Создание программы для каждого датчика
- Объединение в единую программы для проекта
- Написание программы для проекта и тестирование ее на макете

## **6.2 Создание макета системы автоматического орошения.**



В создании проекта системы автоматического орошения начинается с подключения датчиков к микроконтроллеру Arduino UNO и написание в Arduino IDE программы для анализа информации с датчиков и своевременное реагирование на полученные данные.

### **6.2.1 Написание программы для отдельных видов культурных растений**

Каждое сельскохозяйственное растение нуждается в индивидуальном уходе и поливе. Какое-то растение нравится более влажная почва и сухой воздух, а для кого-то на оборот. Данные параметры будут учитывать в системе автоматического орошение растений.

Для начала проведем анализ популярные сельскохозяйственные растения, которые выращивают в теплицах:

Огурцы – культура влаголюбивая. Данная культура любит чтобы почва и воздух были влажными. Особенно необходим полив во время плодоношения. Вода должна быть теплой, нельзя использовать воду из колодца так как она холодная. В зависимости от грунта и погодных условий норма полива составляет 2-5л на одно растение. Желательно использовать капельный полив. Влажность воздуха должна составлять 70%-80%.

Томаты – предпочтительней использовать влажный грунт и сухой воздух. Полив должен осуществляться один раз в три дня, но полив должен быть обильным. На одно растение уходит 7-10 литров воды, чтобы глубоко пролить грунт, почвы так же желательно замульчировать. Влажность воздуха должна быть 55%-60%.

Перец – культура, которая любит влагу. Полив ежедневный, но небольшими порциями по 2-4 литра на одно растение, и почвы желательно замульчировать. Влажность воздуха должна быть примерно 60%-70%.

Капуста – влаголюбивая культура полив должен быть обильным и частым. Количество воды на одно растение должно составлять 5л-7л, а полив осуществлять через 2-3 дня. Влажность воздуха должна составлять 70%. Температура воды для полива можно использовать прохладную.

Так же необходимо учитывать, что во время вегетации и плодоношения растению необходимо большее количество воды. Это объясняется ростом зеленой массы и корней. Так же вся полученная информация будет эффективнее расходовать воду для полива.

Все полученные параметры будут учитываться в системе автоматического полива. В таблице 6.2 будет составлен список растений и количество воды для их полива в зависимости от времени роста растения.

Таблица 6.2 – Необходимое количество воды для разных растений

№	Растение	Начало роста (количество литров для 1 растения)	Вегетация (количество литров для 1 растения)	Плодоношение (количество литров для 1 растения)
1	Огурцы	2 литра	3-4 литра	4-5 литров
2	Томаты	5 литров	7-8 литров	10 литров
3	Перец	2 литра	3 литра	4 литра
4	Капуста	3 литра	5 литров	7 литров

По полученной информации была написана программа полива для разных сельскохозяйственных растений, что благоприятно сказывается на росте растений.

### **6.2.2 Подключение каждого датчиков в отдельности к контроллеру Arduino UNO**

Для начала соединили Arduino Uno с расширительной платой Motor Shield, что позволяет управлять сразу двумя моторами, так же для открытия форточки, когда в теплице повышается температура с помощью шагового двигателя сможет её открыть. Приступаем к написанию скетча для управления 4 моторами (рисунок 6.1).

```

#include <AFMotor.h> // подключаем библиотеку для шилда
AF_DCMotor motor3(3); // подключаем мотор к клеммникам M3
AF_DCMotor motor4(4); // подключаем мотор к клеммникам M4

void setup() {

  motor3.setSpeed(255); // задаем максимальную скорость мотора
  motor3.run(RELEASE); // останавливаем мотор
  motor4.setSpeed(255); // задаем максимальную скорость мотора
  motor4.run(RELEASE); // останавливаем мотор

}

void loop() {

  motor3.run(FORWARD); // задаем движение вперед
  motor3.setSpeed(255); // задаем скорость движения
  delay(2000); // указываем время движения
  motor3.run(RELEASE); // останавливаем мотор M3
  delay(5000); // Указываем время задержки

  motor4.run(FORWARD); // задаем движение вперед
  motor4.setSpeed(255); // задаем скорость движения
  delay(2000); // указываем время движения
  motor4.run(RELEASE); // останавливаем мотор M4
  delay(5000); // Указываем время задержки
}

```

Рисунок 6.1 – Программа для управление моторами

После написание программы было проверено на работоспособность.

Подключаем датчик уровня воды Water sensor он имеет 3 контакта:

- +
- -
- S

Подключение к Arduino UNO происходит следующим образом:

- + - питание датчика +5В
- - - земля
- S – аналоговый вход подключенный к выводу A0

Далее пишем программу для данного датчика (рисунок 6.2 слева) и смотрим показание на мониторе порта (рисунок 6.2 справа)

```

int x;

void setup()
{
  pinMode(A0, INPUT); // к аналоговому входу A0 подключим датчик (англ. «input»)
  Serial.begin(9600); // подключаем монитор порт
}

void loop()
{
  x = analogRead(A0); // переменная "x" находится в интервале от 0 до 1023
  Serial.println(x); // выводим значение датчика на монитор
  delay(1000); // задержка в одну секунду
}

```

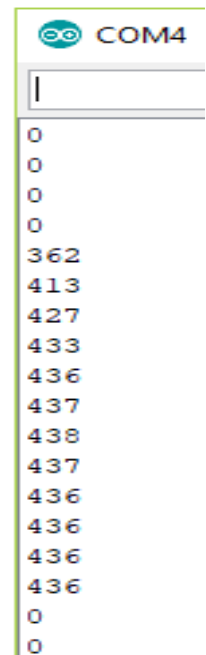


Рисунок 6.2 – Программа для датчика уровня воды

Подключение датчика влажности почвы Keyestudio к плате Arduino.

Данный аналоговый датчик служит для определения влажности почвы, в которую он помещен, с хорошей точностью. Этот датчик имеет три вывода:

- + – питание
- - – земля
- S – сигнал

Между контактами датчика проводится незначительное напряжение: если почва будет сухой тогда сопротивление почвы будет большим, получается, ток будет маленьким; если почва будет влажной – сопротивление почвы будет меньше, следовательно, ток будет большим. По показаниям датчика можно узнать о влажности почвы.

После подключения датчика к Arduino пишем программу для данного датчика влажности почвы (рисунок 6.3 слева), а также смотри показание датчика (рисунок 6.3 справа).

```

class Sensor // название класса
{
  public:
  int pin; // переменная с номером пина
  int data; // переменная с показаниями датчика
  void read() // добавляем функцию read
  {
    data=analogRead(pin); // считываем показания датчика в переменную data
  }
};
Sensor water; // создаем объект класса
void setup()
{
  Serial.begin(9600); // устанавливаем передачу по последовательному порту
  water.pin=A0; // задаем пин, к которому подключен датчик
  pinMode(water.pin, INPUT); // устанавливаем пин в качестве входа для считывания показаний
}
void loop()
{
  delay(500); // пауза
  water.read(); // считываем показания
  Serial.println(water.data); // посылаем показания в последовательный порт
}

```

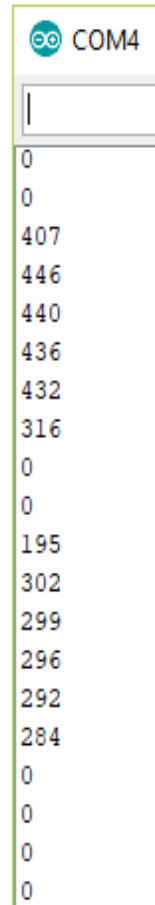


Рисунок 6.3 – программа работы датчика влажности почвы(слева), показание датчика влажности(справа)

Датчика температуры и относительной влажности DHT – 11 имеет три контакта, к которым и подключаются к Arduino.

- VCC – подключаем к 5В
- GND – подключаем к земле
- DATA – подключаем к pin 2

После подключения датчика начинаем писать программу (рисунок 6.4 слева) и смотрим результаты показания датчика на монитор порте (рисунок 6.4 справа). На основании этих данных мы сможем контролировать влажность воздуха с помощью проветривания помещения.

```

#include <dht11.h> // Добавляем библиотеку DHT11
dht11 DHT; // Объявление переменной класса dht11
#define DHT11_PIN 4 // Датчик DHT11 подключен к цифровому пину номер 4

void setup(){
  Serial.begin(9600); // Скорость работы порта
  Serial.println("DHT TEST PROGRAM "); // Выводим текст
  Serial.print("LIBRARY VERSION: "); // Выводим текст
  Serial.println(DHT11LIB_VERSION);
  Serial.println(); // Пустая строка
}

void loop(){
  int chk;
  ;
  // Мониторинг ошибок
  chk = DHT.read(DHT11_PIN); // Чтение данных
  switch (chk){
  case DHTLIB_OK:
    break;
  case DHTLIB_ERROR_CHECKSUM:
    Serial.println("Checksum error, \t");
    break;
  case DHTLIB_ERROR_TIMEOUT:
    Serial.println("Time out error, \t");
    break;
  default:
    Serial.println("Unknown error, \t");
    break;
  }
  // Выводим показания влажности и температуры
  Serial.print("Humidity = ");
  Serial.print(DHT.humidity, 1);
  Serial.print(", Temp = ");
  Serial.println(DHT.temperature, 1);
  delay(1000);
}

```

```

COM4
Humidity = 29, Temp = 30
Checksum error,
Humidity = 29, Temp = 30
Time out error,
Humidity = 29, Temp = 30
Checksum error,
Humidity = 29, Temp = 30
Time out error,
Humidity = 29, Temp = 30
Checksum error,
Humidity = 28, Temp = 30
Time out error,
Humidity = 28, Temp = 30
Checksum error,
Humidity = 29, Temp = 30
Time out error,
Humidity = 29, Temp = 30
Checksum error,
Humidity = 29, Temp = 30
Time out error,
Humidity = 29, Temp = 30
Checksum error,
Humidity = 32, Temp = 30
Time out error,
Humidity = 30, Temp = 30
Checksum error,
Humidity = 31, Temp = 30
Time out error,
Humidity = 31, Temp = 30
Checksum error,
Humidity = 31, Temp = 30

```

Рисунок 6.4 – программа и показания датчика DHT-11

Так же в данном проекте будет функция подогрева почву. Для этого используется дополнительный датчик DHT-11 благодаря которому будем получать информацию о температуры почвы. Для этого на глубину двадцать сантиметров закапываются трубы диаметром пятьдесят два миллиметра в которых и будет крепиться данный датчик. По этим трубам будет идти теплый воздух с верхней части теплице, так как воздух под крышей теплице очень теплый. С помощью кулера данный воздух будет через трубы проходить через почву и подогревать ее. Так же были варианты использования электрических и газовых нагревателей, но данный способ очень дорогостоящий, в отличии от того способа, который будем использовать в данном проекте.

Датчик ESP8266 это модуль Wi-Fi для подключения его к Arduino UNO и созданию удалённого доступа на проект систем автоматического орошения

растений. Нужно подключить USB-TTL программатор к ESP8266. Запустить программу Arduino IDE и в меню инструкций выбрать нужный пример для ESP8266, загрузить программу через USB-TTL.

В скетче так же нужно заполнить SSID и пароль от WIFI сети. Далее нажимаем компилировать и загружаем скетч. Далее открываем монитор последовательного порта и выбираем скорость 115200, в это время происходит подключение к роутеру, как произойдет подключение ESP 8266 с WiFi модемом на экране появится «WIFI connected» и «Server started». Ниже будет указан адрес IP модуля, копируем его и добавляем в браузере к примеру «<http://192.168.1.111/gpio/1>». Для проверки подключаем светодиод через резистор и есть в браузере отправить данную команду тогда светодиод загорится. На рисунке 6.5 показан пример программы для ESP8266.

```

1 #include <ESP8266WiFi.h>
2
3 const char* ssid = "YOUR_SSID"; // для SSID точки доступа
4 const char* password = "YOUR_PASSWORD"; // для пароля к точке доступа
5
6 int ledPin = 2; // контакт GPIO2 на ESP8266
7 WiFiServer server(80); // порт веб-сервера
8
9 void setup() {
10   Serial.begin(115200);
11   delay(10);
12
13   pinMode(ledPin, OUTPUT);
14   digitalWrite(ledPin, LOW);
15
16   // подключаемся к WiFi-сети:
17   Serial.println();
18   Serial.println();
19   Serial.print("Connecting to "); // "Подключение к "
20   Serial.println(ssid);
21
22   WiFi.begin(ssid, password);
23
24   while (WiFi.status() != WL_CONNECTED) {
25     delay(500);
26     Serial.print(".");
27   }
28   Serial.println("");
29   Serial.println("WiFi connected"); // "Подключение к WiFi
30                                     // выполнено"
31
32   // запускаем сервер:
33   server.begin();
34   Serial.println("Server started"); // "Сервер запущен"
35
36   // печатаем IP-адрес:
37   Serial.print("Use this URL to connect: "); // "Используем этот URL
38                                               // для подключения: "
39   Serial.print("http://");
40   Serial.print(WiFi.localIP());
41   Serial.println("/");
42 }
43
44 void loop() {
45   // проверяем, подключен ли клиент:
46   WiFiClient client = server.available();
47   if (!client) {
48     return;
49   }
50
51   // ждем, когда клиент отправит какие-нибудь данные:
52   Serial.println("new client"); // "новый клиент"
53   while(!client.available()){
54     delay(1);
55   }
56
57   // считываем первую строчку запроса:
58   String request = client.readStringUntil('\r');
59   Serial.println(request);

```



```

60 client.flush();
61
62 // обрабатываем запрос:
63 int value = LOW;
64 if (request.indexOf("/LED=ON") != -1) {
65     digitalWrite(ledPin, HIGH);
66     value = HIGH;
67 }
68 if (request.indexOf("/LED=OFF") != -1){
69     digitalWrite(ledPin, LOW);
70     value = LOW;
71 }
72
73 // выставляем значение на ledPin в соответствии с запросом:
74 //digitalWrite(ledPin, value);
75
76 // возвращаем ответ:
77 client.println("HTTP/1.1 200 OK");
78 client.println("Content-Type: text/html"); // "Тип контента:
79                                             // text/html "
80 client.println(""); // не забываем это...
81 client.println("<!DOCTYPE HTML>");
82 client.println("<html>");
83
84 client.print("Led pin is now: "); // "Контакт светодиода теперь
85                                 // в состоянии: "
86
87 if(value == HIGH) {
88     client.print("On"); // "Вкл"
89 } else {
90     client.print("Off"); // "Выкл"
91 }
92 client.println("<br><br>");
93 client.println("Click <a href=\"/LED=ON\">here</a> turn the LED on pin 2 ON<br>");
94             // на контакте 2"
95 client.println("Click <a href=\"/LED=OFF\">here turn the LED on pin 2 OFF<br>");
96             // на контакте 2"
97
98 client.println("</html>");
99
100 delay(1);
101 Serial.println("Client disconnected"); // "Клиент отключен"
102 Serial.println("");
103 }

```

Рисунок 6.5 – Программа для WiFi модуля ESP 8266

Так же необходимо написать программу для модуля реального времени DS 3231. Для этого необходимо подключить 5 В и землю к датчику, а также

задействовать выводы SDA и SCL их необходимо подключить к аналоговым входам на Arduino UNO.

После подключения модуль реального времени к Arduino необходимо написать программу. На рисунке 6.6 показана программа для модуля реального времени

```
#include <DS3231.h>           // Подключаем библиотеку Wire
DS3231 rtc(SDA, SCL);       // Инициализация DS3231

void setup()
{
  Serial.begin(115200);      // Установка последовательного соединения
  rtc.begin();              // Инициализировать rtc

  // Установка времени
  rtc.setDOW(FRIDAY);       // Установить день-недели
  rtc.setTime(16, 29, 0);    // Установить время 16:29:00 (формат 24 часа)
  rtc.setDate(31, 8, 2019); // Установить дату 31 августа 2019 года
}
void loop()
{
  Serial.print(rtc.getDOWStr()); // Отправляем день-неделя
  Serial.print(" ");

  Serial.print(rtc.getDateStr()); // Отправляем дату
  Serial.print(" -- ");

  Serial.println(rtc.getTimeStr()); // Отправляем время

  delay (1000);             // Задержка в одну секунду
}
```

Рисунок 6.6 – Программа для модуля реального времени

После того как были написаны отдельные программы для каждого используемого компонента в системе автоматического полива растений, необходимо соединить и написать готовую программу для орошения растений. Готовая программа для системы автоматического полива представлена в приложении А.

### 6.3 Моделирование системы автоматического орошения в программе Tinkercad

В программе Tinkercad было сделано моделирование система автоматического полива растения. Это необходимо для проверки написанной программы, так как если что-то написано неправильно можно повредить комплектующие, а в данная программа дает возможность проверить работоспособность программы. Последовательно будем подключать датчики. Схема подключения датчика температуры, представленная ниже на рисунке 6.7.

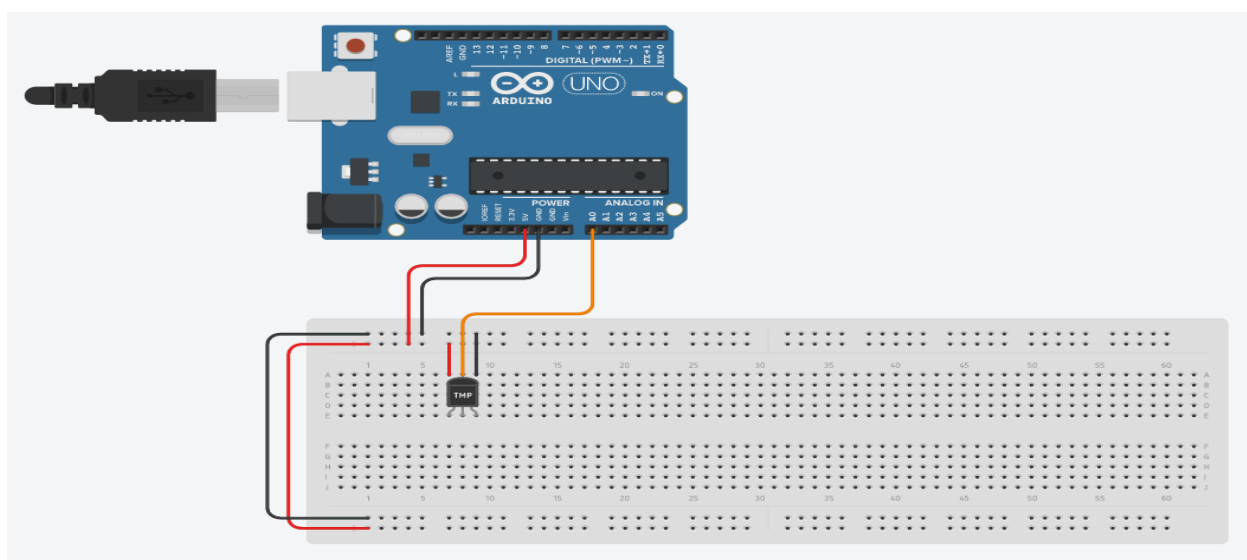


Рисунок 6.7 Подключение датчика температуры

Далее подключаем датчик влажности почвы в схеме он будет представлен как потенциометр и показан на рисунке 6.8.

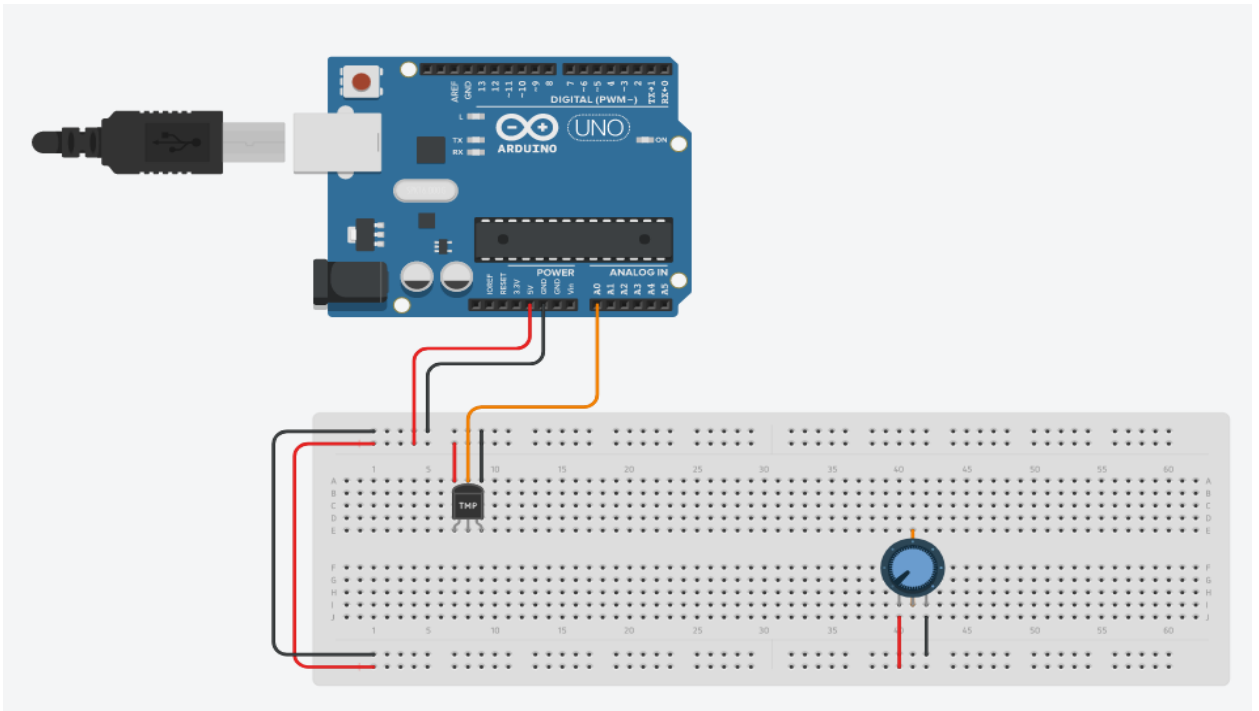


Рисунок 6.8 - подключение датчика влажности почвы

Далее подключаем двигатель для полива растений. Его необходимо подключить через транзистор. Так же подключаем светодиоды через резистор для наглядности работы. Представлено на рисунке 6.9.

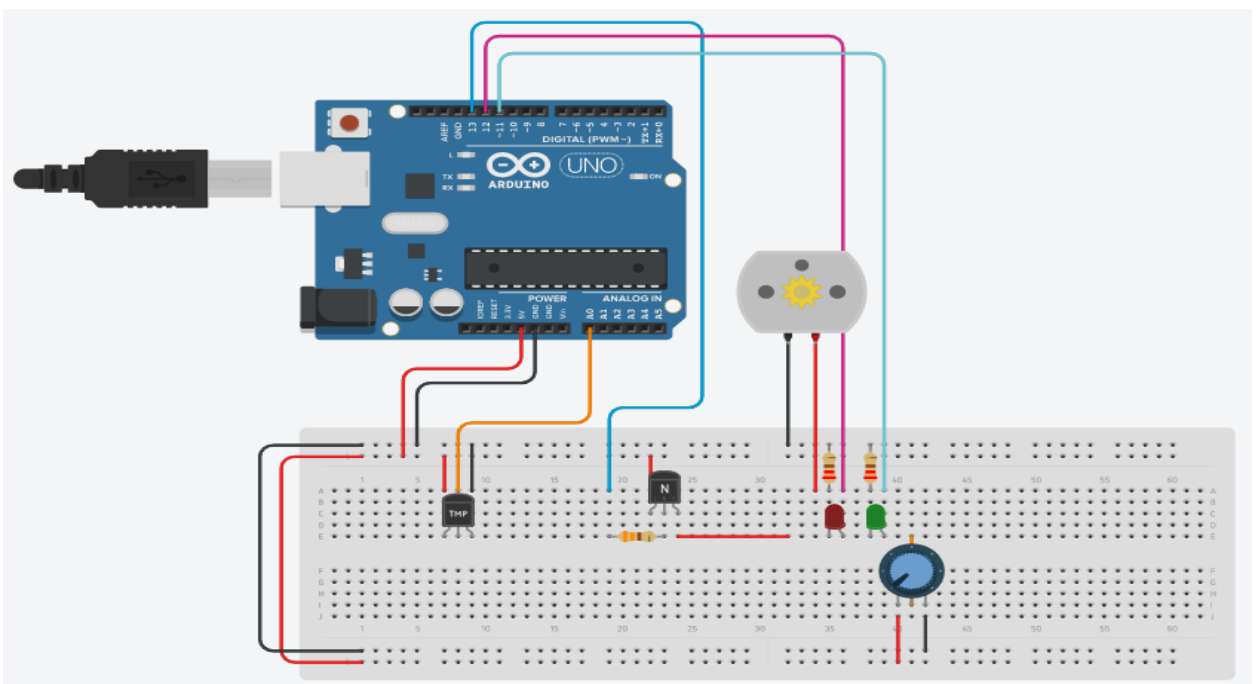


Рисунок 6.9 – Подключение двигателя и светодиодов

Далее подключаем экран как это показано ниже на рисунке для получения информации с датчика влажности и температуры. А также подключить питание для Arduino и датчиков. Можно использовать разные варианты сделать это через USB порт или через адаптер питания, но в данном проекте будет использоваться аккумулятор. Показано на рисунке 6.10.

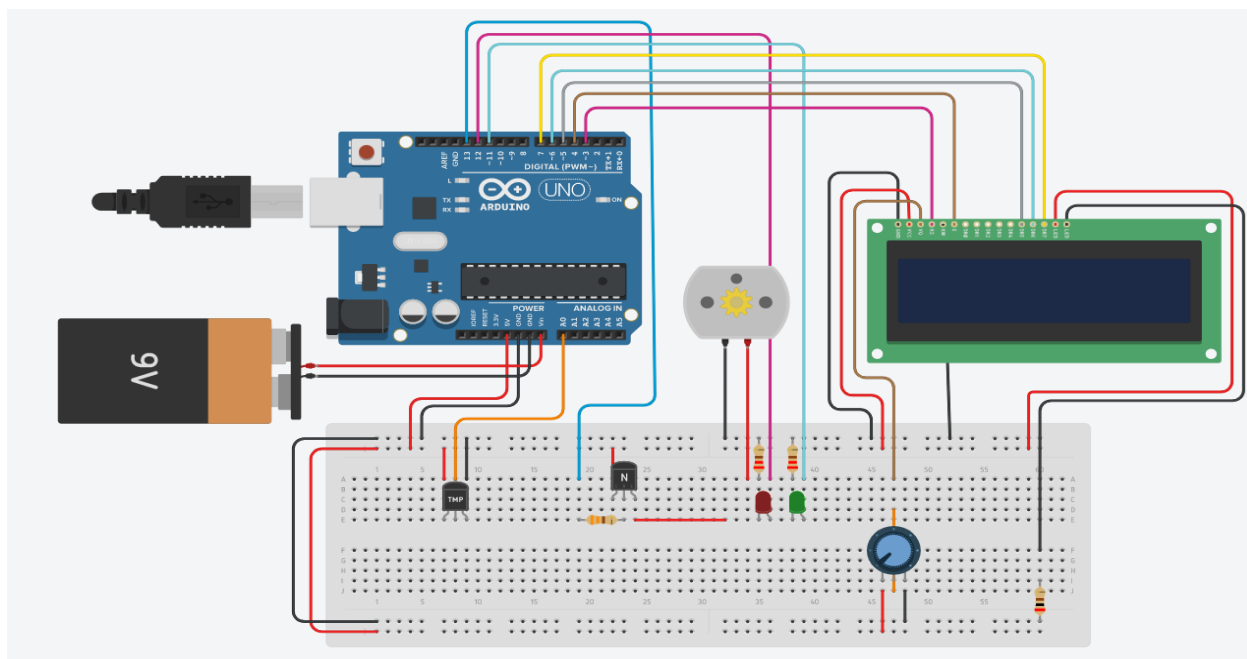


Рисунок 6.10 – Подключение экрана и питания

После этого подключаем WiFi модуль для создания удалённого доступа. Подключение модуля показано ниже на рисунке. Обратите внимание что питание для ESP8266 должен быть подключено питание 3.3В. Подключение беспроводного модуля ESP8266 показано на рисунке 6.11.

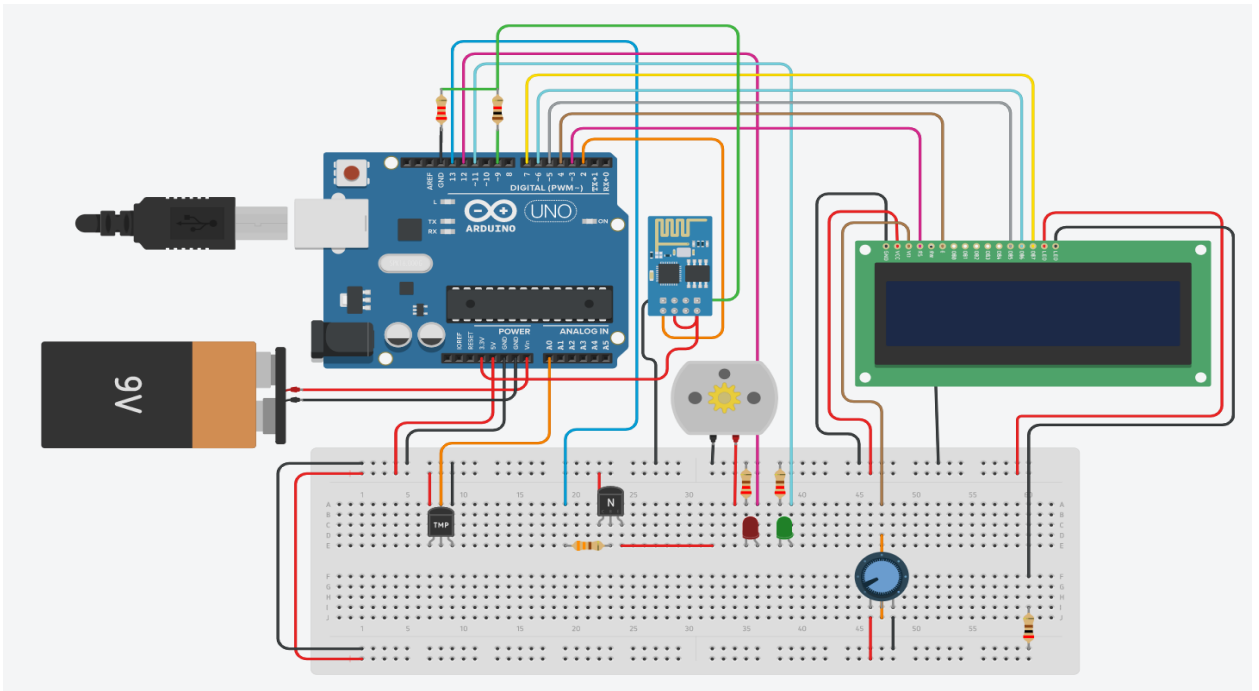


Рисунок 6.11 – Подключение модуля ESP8266

Так же подключаем датчик влажности окружающего воздуха и температуры благодаря которому будет осуществляться с помощью вентилятора проветривание помещения и подогрева почвы. Далее подключим датчик реального времени для определения нужной даты и регулирование количества подачи воды для разных типов растений.

Потом устанавливаем датчик уровня воды, который будет показывать сколько воды в ёмкости и, если воды в емкости недостаточно датчик уровня воды будет давать команду на включение насоса для забора воды из скважины или природного водоёма.

Датчик DHT11 будет использоваться так же для прогрева земли. В теплице горячий воздух скапливается под крышей поэтому его можно использовать для прогрева земли что в свою очередь положительно сказывается на растении. Данный способ очень экономичный, так как если использовать электрический подогрев или отопление с помощью газа, эти способы будут более дорогими при использовании.

После проверки готового скетча в программе Tinkercad мы убедились, что программа работоспособная, после этого можно приступать к сборке

макета и проверки программы на готовом устройстве системы автоматического орошения.

После написание программы и моделирование системы автоматического орошения необходимо построить блок схему по работе программы и наглядно показать процесс работы системы автоматического орошения растений. Блок схема программы будет представлена на рисунке 6.12

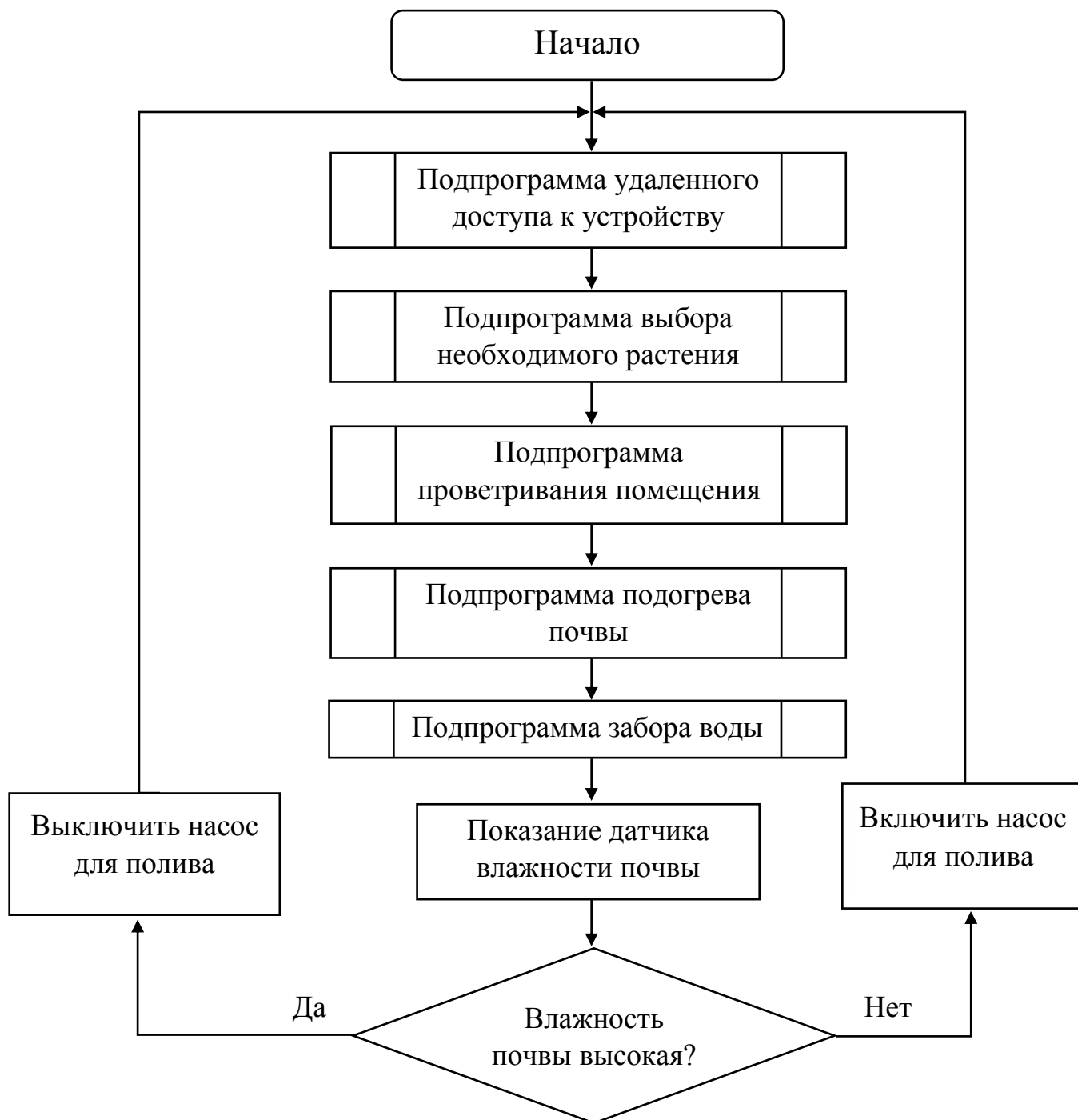


Рисунок 6.12 - Блок схема программы

В данной блок схеме имеются пять подпрограмм, которые отвечают:

- Подпрограмма для управления с помощью удаленного доступа;
- Подпрограмма для выбора нужного вида растения;
- Подпрограмма для подогрева почвы;
- Подпрограмма для проветривания помещения;
- Подпрограмма для забора воды.

Рассмотрим каждую подпрограмму в отдельности.

Подпрограмма для управления удаленным доступом представляет из-за датчик ESP3231 который может по сети WiFi подключится к серверу с помощью которого владелец с помощью телефона зайти на необходимый сервер и сможет наблюдать за работой системы авто полива а так же при необходимости делать ручные настройки системы автоматического полива растений.

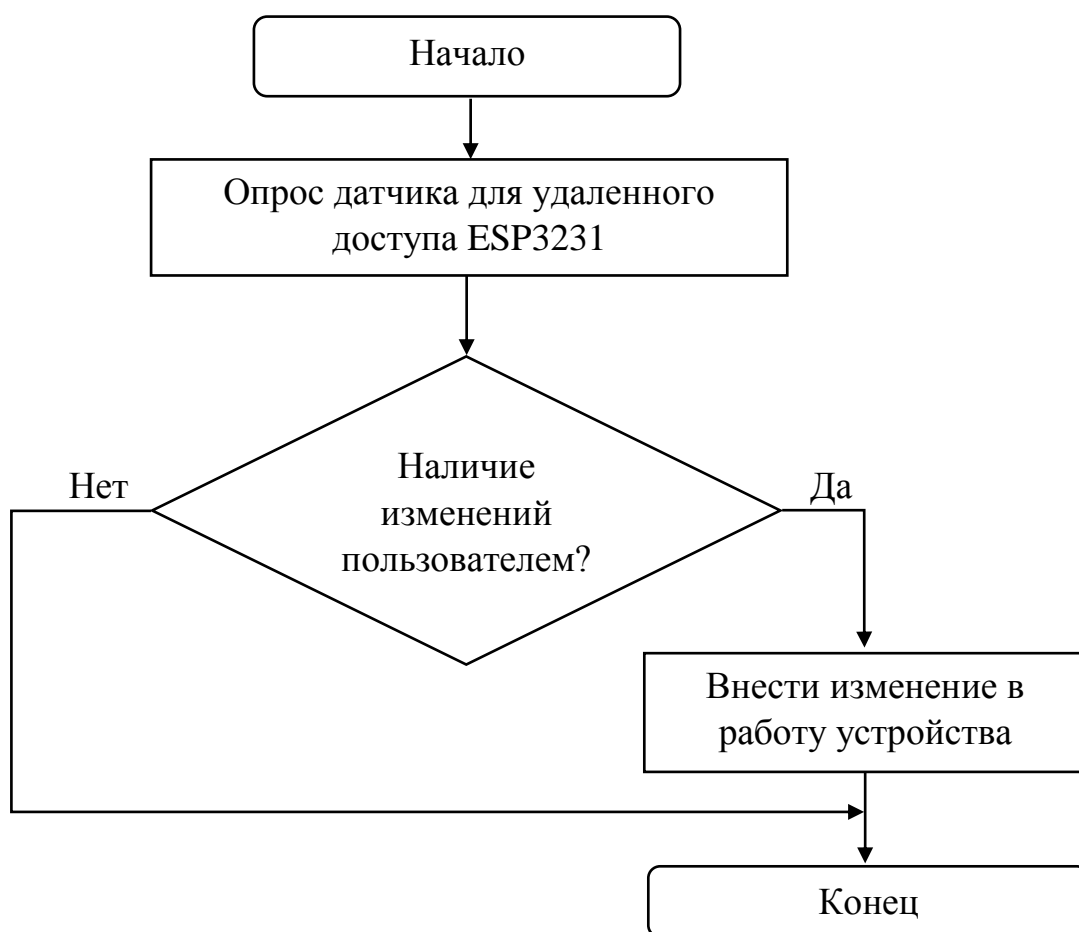


Рисунок 6.13 – Подпрограмма удаленного доступа



Подпрограмма для выбора необходимого растения. В алгоритме работы системы учитываются критерии разных видов растений такие как: помидор, огурец, перец, капуста. Это необходимо для того, что жизненный цикл этих растений отличаются друг от друга и потребление воды, и влажность воздуха нужно контролировать для каждого растения. Поэтому и была написана программа непосредственно для каждого растения что бы улучшить качество растения и повысить количество плодов. Подпрограмма удаленного доступа к устройству показана на рисунке 6.14

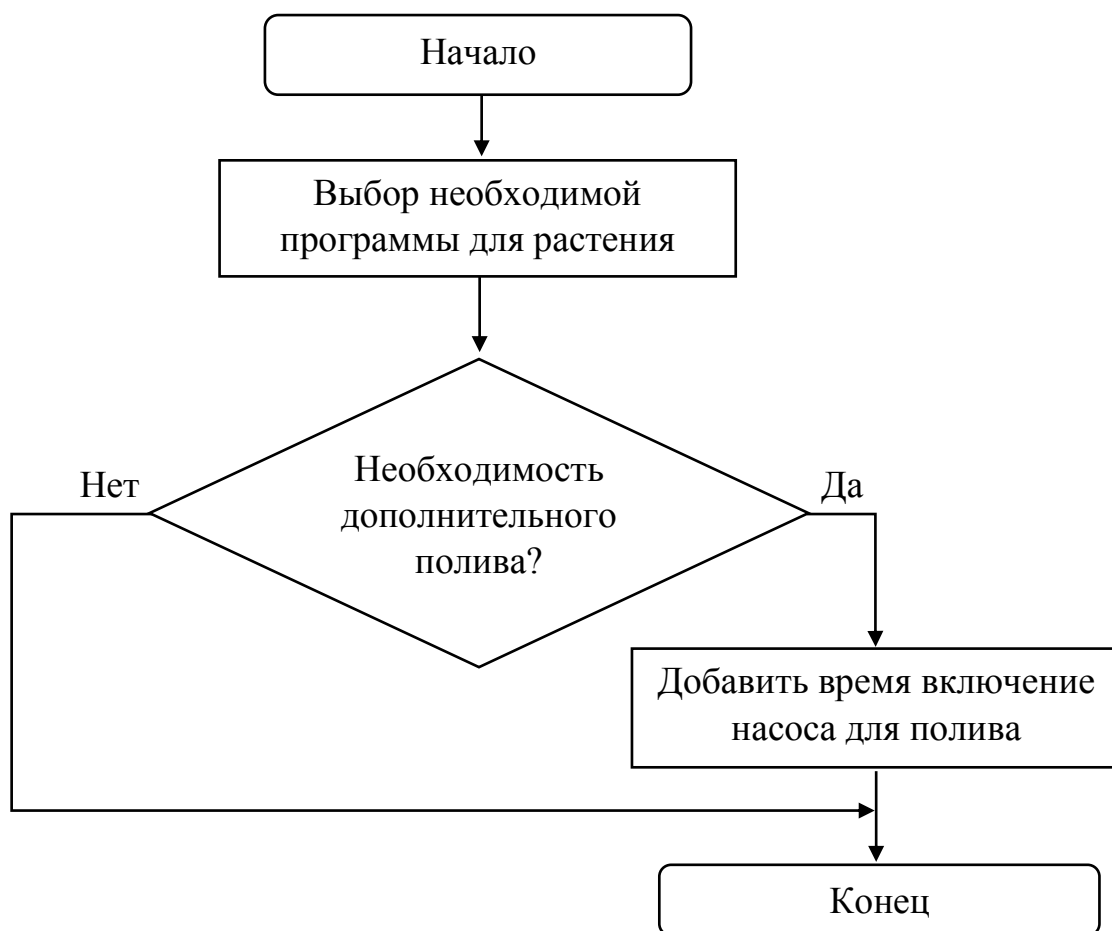


Рисунок 6.14 – Подпрограмма удаленного доступа

«Подпрограмма проветривая помещения представляет из себя датчик температуры и относительной влажности DHT 11 от которого будет получать информацию о влажности и на основании этих показаний включаем или выключаем двигатель для проветривания помещения. Данная подпрограмма будет показана на (рисунке 6.15)» [1]

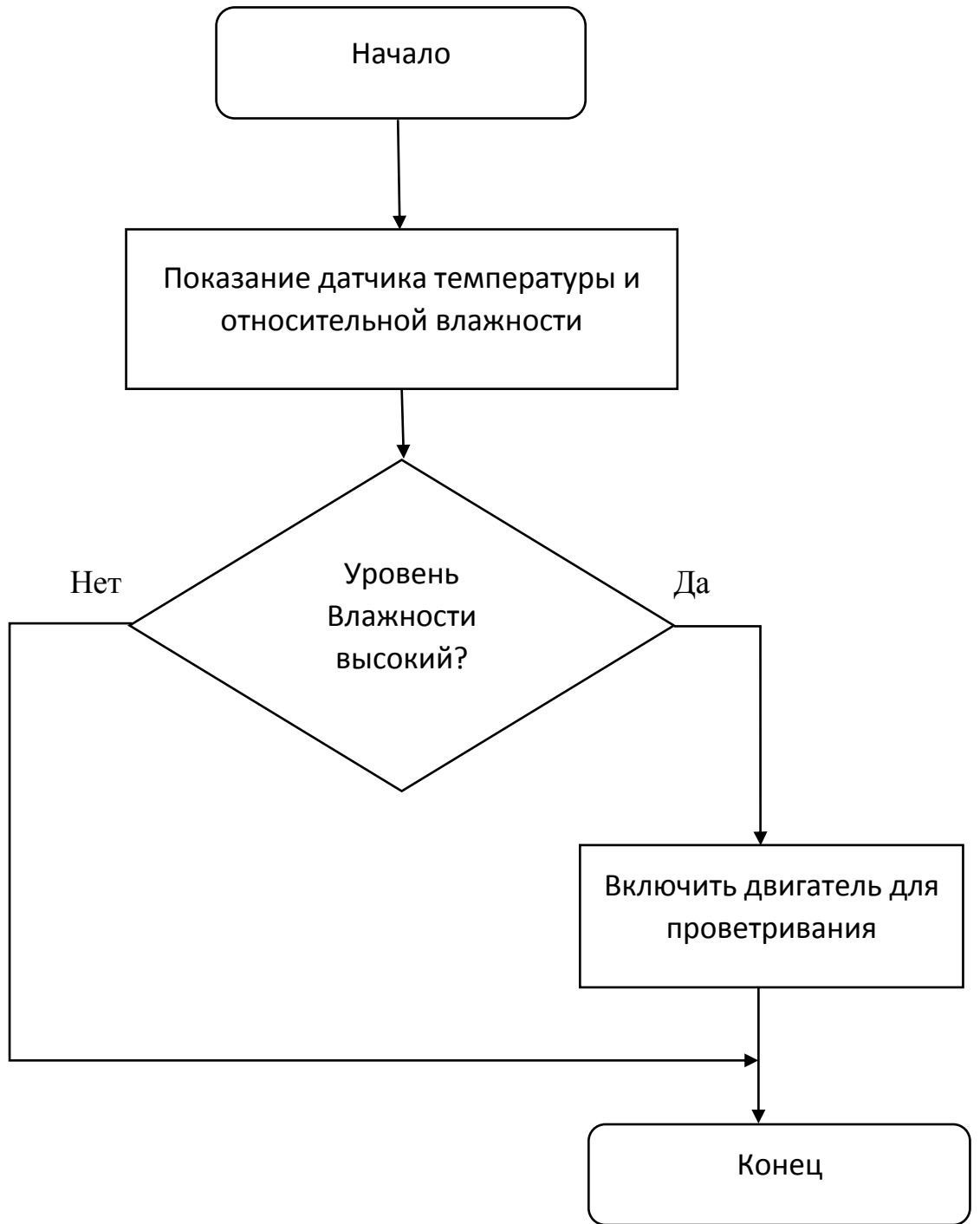


Рисунок 6.15 - Подпрограмма проветривания помещения

Еще одна из подпрограмм нужна для регулировки подогрева почвы в теплице. Данная подпрограмма состоит из датчика температуры земли DS18B20 и кулера для накачки воздуха в трубы. Если датчик температуры показывает что температуры замни снизилась до указанного значения

включается вентилятор, который нагнетает теплый воздух в трубы с помощью чего и осуществляется нагрев почвы. Подпрограмма указана на (рисунке 6.16).

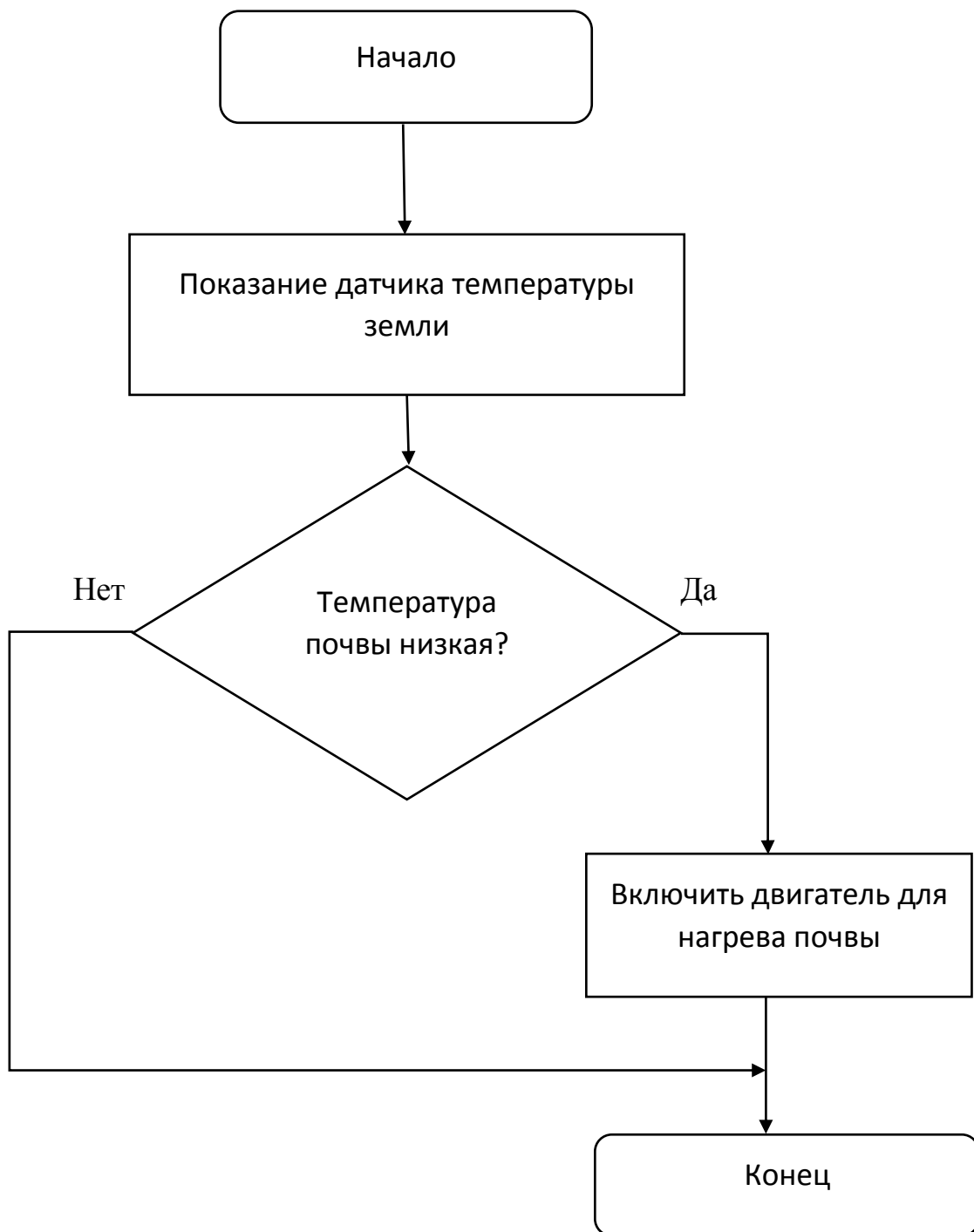


Рисунок 6.16 – Подпрограмма подогрева почвы

«Так же существует подпрограмма, отвечающая за уровень воды в емкости которая будет служить для полива растений. Данная подпрограмма состоит из датчика уровня воды с верхним и нижним уровнем воды. По показаниям датчика можно с помощью насоса набирать емкость для полива. Подпрограмма представлена на (рисунке 6.17)»[1].

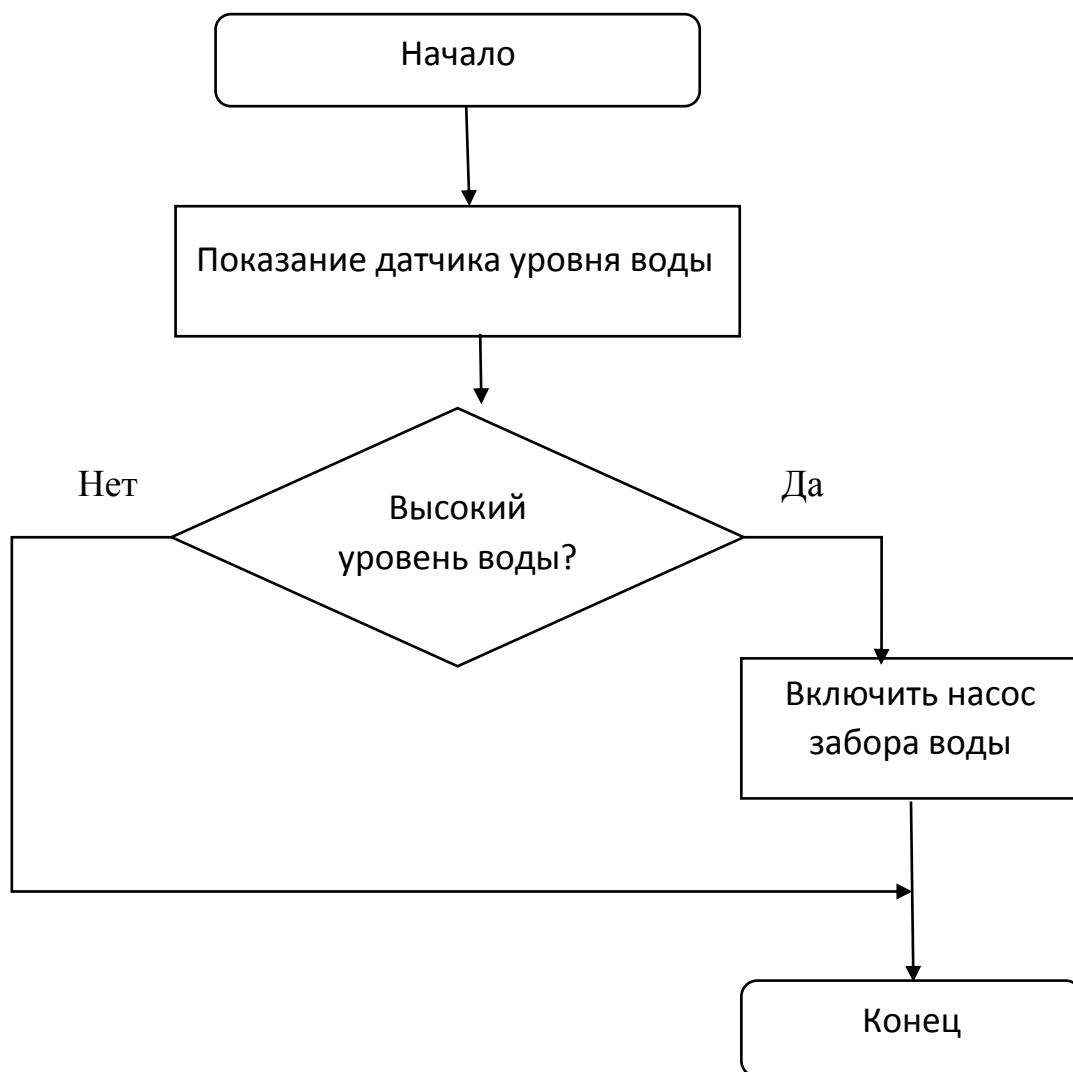


Рисунок 6.17 – Подпрограмма уровня воды

После написания программы и проверки ее в моделировании программы Tinkercad мы убедились, что программа работоспособная, после этого была написана подробная блок схема работы системы автоматического орошения, и после этого можно приступать к сборке макета и проверки программы на готовом устройстве системы автоматического орошения.

## Заключение

В магистерской диссертационной работе была представлена, разработана, смоделирована, изготовлена и описана система автоматического орошения растений.

В современном развивающемся мире, где каждый день происходят новые открытия в науке и технике для улучшения уровня жизни человека и автоматизации процессов в различных сферах деятельности человека. В нынешнее время имеются немало устройств и приборов, которые дают возможность сделать жизнь человека проще на работе и дома, а также экономить время и деньги. Данные устройства человек использует как дома такие как: автоматические пылесосы, мультиварка, климатические системы, автопилоты на автомобилях, так и на работе, где автоматизация процессов один из главных критериев повышения эффективности работы человека и производства.

В первом разделе был проведен анализ актуальности тема магистерской диссертации. Так же было описаны самые популярные режимы орошения для использования с система автоматического полива, из полученных данных был выбран метод капельного полива.

В разделе номер два был представлен и проанализирован нынешний рынок система автоматического полива которые являются прямыми конкурентами. После анализа готовых решений было выявлено, что этот вид устройства востребован и актуален в нынешнее время, но не может полностью удовлетворить желания пользователей. Основываясь на полученные результаты, был разработана, смоделирована и создана новая система автоматического полива растений с множеством дополнительных необходимых функций.

В третьем разделе после того, как был проанализирован рынок устройств для полива, были поставлены цели для магистерской диссертации. Главной

целью являлось разработка и изготовление системы автоматического орошения для промышленных и частных теплиц, который сможет в полном объёме удовлетворить потребности пользователей.

В четвертом разделе основываясь на полученной информации и поставленных целях была спроектирована структурная схема устройства, также были выбраны комплектующие из которых и должен состоять будущий макет системы автоматического орошения растений.

В пятом разделе были подробно описан каждый элемент, который используется в данном проекте. Вдобавок было описано программное обеспечение для программирования проекта, а также была описана программа с помощью которой было осуществлено моделирование системы орошения растений.

В шестом разделе была написана программа для проекта, после чего было проведено моделирование и проверка программы. После получения положительного результата была изготовлена готовая система автоматического орошения растений.

В заключении хотелось отметить, что все поставленные задачи были выполнены. Анализ, исследование, моделирование и разработка готовой системы автоматического полива было проведена успешная, а также был представлен готовый рабочий макет, который в дальнейшем сможет упростить жизнь человеку и повысить автоматизацию в области сельскохозяйственных растений.

## Список используемой литературы

1. Певнев И.В., Позднов М.В., Система автоматического полива с автономным водозабором, Тольяттинский государственный университет, Институт энергетики и электротехники, Кафедра Промышленная электроника, 2018. – 104 с
2. Курдюмов Н., Малышевский К., Умная теплица, Издательство: Владис, 2018.-19 с
3. Курдюмов Н., Малышевский К., Умная теплица, Издательство: Владис, 2017.-19 с
2. Кудинов, В.А. Техническая термодинамика и теплопередача: Учебник для бакалавров / В.А. Кудинов, Э.М. Карташов, Е.В. Стефанюк.. - Москва: Юрайт, 2018. - 566 с.
3. Мирам, А.О. Техническая термодинамика. Тепломассообмен Учебное издание / А.О. Мирам. - Москва: АСВ, 2019. - 352 с.
4. ФГОС СПО 13.03.01 Теплоэнергетика и теплотехника от 2015-06-28; № 21200 - Москва: Минобрнауки России, 2018. - №1081.
5. Волошенко А.В. Проектирование функциональных схем систем автоматического контроля и регулирования: учебное пособие / А.В. Волошенко, Д.Б. Горбунов – Томск: Томского политехнического университета, 2014. – 109с.
6. Автоматизация теплиц, цветочных хозяйств и оранжерей: [Электронный ресурс]: URL: [http://poltraf.ru/publications/otrasli\\_promyshlennosti/avtomatizatsiya\\_teplits\\_oranzherey/](http://poltraf.ru/publications/otrasli_promyshlennosti/avtomatizatsiya_teplits_oranzherey/) (дата обращения: 7.11.2019).
7. СНиП 2.04.02-84\*. Водоснабжение. Наружные сети и сооружения. - М.: Стройиздат, 2015.
8. Попкович Г.С., Гордеев Н.А. Автоматизация систем ВиВ. - М.: Высшая школа, 2014

9. Современная теплица: автоматизированная система: [Электронный ресурс]: URL: <http://vseoteplicah.ru/instrumenty/avtomatizaciya-teplicy-svoimi-rukami.html>. (дата обращения: 18.05.2018).
10. Автоматизация процессов управления микроклиматом тепличного блока: [Электронный ресурс]: URL: <https://www.2d-3d.ru/2d-galereia/automatika/465-avtomatizaciya-processov-upravleniya-mikroklimatom-teplichnogo-bloka.html> (дата обращения: 24.09.2018).
11. Ключев, А.С. Автоматизация настройки систем управления / А.С. Ключев, В.Я. Ротач, В.Ф. Кузицин. - Москва: Альянс, 2015. - 272 с.
12. Схиртладзе, А.Г. Автоматизация технологических процессов: Учебное пособие / А.Г. Схиртладзе, С.В. Бочкарев, А.Н. Лыков. - Старый Оскол: ТНТ, 2016. - 524 с.
13. Иванов, А.А. Автоматизация технологических процессов и производств: Учебное пособие / А.А. Иванов. - Москва: Форум, 2015. - 224 с.
14. Кангин, В.В. Промышленные контроллеры в системах автоматизации технологических процессов: Учебное пособие / В.В. Кангин. - Старый Оскол: ТНТ, 2013. - 408 с.
15. Автоматический полив - системы автоматического полива: [Электронный ресурс]. URL: <http://www.polivmaster.ru/tehno/> (дата обращения: 26.01.2020).
16. Контроль и поддержание микроклимата в теплицах на базе компьютера - автоматическая система: [Электронный ресурс]. URL: [http://www.rlda.ru/Examples\\_Greenhouse.htm](http://www.rlda.ru/Examples_Greenhouse.htm) (дата обращения: 16.05.2018).
17. Автоматика для теплицы сэкономит урожай от жары и холода : [Электронный ресурс]. URL: <http://moyateplica.ru/osnashchenie-teplicy/avtomatika-teplicy-sberezhet-urozhai-ot-zhary-i-holoda#h2-sistemy-obogreva-sovremennykh-teplits-i-vozmozhnost-avtomatizatsii-ikh-raboty>. (дата обращения: 3.05.2019).



18. Автоматическое управление влажностью воздуха и почвы, температурой поливной воды: [Электронный ресурс]. URL: <http://studopedia.org/8-17660.html>. (дата обращения: 27.11.2019).
19. Общие сведения о платформе Arduino UNO: [Электронный ресурс]: URL: <http://arduino.ru/Hardware/ArduinoBoardUno>. (дата обращения: 17.03.2018).
20. Описание и подключение Motor Shield к плате Arduino: [Электронный ресурс]: URL: <http://freeduino.ru/arduino/mshield.html>. (дата обращения: 19.03.2019).
21. Установка и настройка Arduino под Windows: [Электронный ресурс]: URL: <https://all-arduino.ru/arduino-ide/>. (дата обращения: 1.04.2019).
22. Модуль KY-015 датчик влажности и температуры DHT11: [Электронный ресурс]: URL: [http://zi-zi.ru/docs/modules/info\\_KY-015.pdf](http://zi-zi.ru/docs/modules/info_KY-015.pdf). (дата обращения: 23.03.2018).
23. Датчик уровня влажности почвы и автоматический полив на Arduino: [Электронный ресурс]: URL: [https://arduino-kit.ru/textpage\\_ws/pages\\_ws/4.3.-podklyuchaem-datchik-vlajnosti-pochvyi](https://arduino-kit.ru/textpage_ws/pages_ws/4.3.-podklyuchaem-datchik-vlajnosti-pochvyi). (дата обращения: 3.04.2019).
24. Определение уровня воды в емкости при помощи датчика уровня воды: [Электронный ресурс]: URL: <http://makerplus.ru/wiki/arduino-datchik-urovnya-vody>. (дата обращения: 28.03.2019).
25. Автоматическая система полива Aquarius: [Электронный ресурс]: URL: <https://hi-garden.ru/aquarius>. (дата обращения: 29.04.2019).
26. Система автоматического полива ОГО-Родник 1: [Электронный ресурс]: URL: <http://sovelteh.ru/shop/product/sistema-avtopoliva-dlya-teplitsy-ogo-rodnik-1-616/>. (дата обращения: 30.04.2019).
27. Инструкция к применению комплекта капельного полива Aqua Planet-60: [Электронный ресурс]: URL: <http://vologda.zavodnason.ru/instrukcii/instrukciya-aqua-planet-60-avtomat.html>. (дата обращения: 30.04.2019).

28. Maxim Integrated and the Maxim Integrated logo are trademarks of Maxim Integrated Products, Inc. DS18B20 Programmable Resolution 1-Wire Digital Thermometer 2015 Maxim Integrated Products, Inc. 20

29. Oser J., Blemings H. (2013) Water Tank Depth Sensor. In: Practical Arduino. Apress

30. Veltman A., Pulle D.W., De Doncker R.W. (2015) VOLTAGE SOURCE CONNECTED ASYNCHRONOUS MACHINES. In: Fundamentals of Electrical Drives. Power Systems. Springer, Dordrecht

31. *Ferraris, G. "Atti della Reale Accademia delle Scienze di Torino". Atti della R. Accademia delle Scienze di Torino: 360–375.*

32. Bulletin of the Atomic Scientists. *Educational Foundation for Atomic Science*. 6 June 1973. Retrieved 8 August 2014.

## Приложение А

### Программа системы автоматического полива растений

```
1 #include <AFMotor.h> // подключаем библиотеку для шилда
2 AF_DCMotor motor3(3); // подключаем мотор к клеммникам M3
3 AF_DCMotor motor4(4); // подключаем мотор к клеммникам M4
4
5 #include <ESP8266WiFi.h>
6 #include <MQTT.h>
7 #include <EEPROM.h>
8 #define AP_SSID      "xxx"
9 #define AP_PASSWORD "xxx"
10 #define EIOTCLOUD_USERNAME "xxx"
11 #define EIOTCLOUD_PASSWORD "xxx"
12 // создаем объект MQTT:
13 #define EIOT_CLOUD_ADDRESS "cloud.iot-playground.com"
14
15 // Подключаем библиотеку для работы с дисплеем
16 #include "QuadDisplay2.h"
17 // даём разумное для пина, к которому подключена помпа
18 #define POMP_PIN      4
19 // даём разумное для пина, к которому подключён датчик влажности почвы
20 #define HUMIDITY_PIN  A0
21 // минимальный порог влажности почвы
22 #define HUMIDITY_MIN  500
23 // максимальный порог влажности почвы
24 #define HUMIDITY_MAX  800
25 // интервал между проверкой на полив растения
26 #define INTERVAL     60000 *3
27 // даём разумное для пина, к которому подключён датчик влажности почвы
28 // переменная для хранения показания влажности почвы
29 unsigned int humidity = 0;
30 // статическая переменная для хранения времени
31 unsigned long waitTime = 0;
32 #define HUMIDITY_PIN1  A1
33 // минимальный порог влажности почвы
34 #define HUMIDITY_MIN1  400
35 // максимальный порог влажности почвы
36 #define HUMIDITY_MAX1  1000
37 // интервал между проверкой на полив растения
38 #define INTERVAL1     60000 * 3
39 // переменная для хранения показания влажности почвы
40 unsigned int humidity1 = 0;
41
42 // статическая переменная для хранения времени
43 unsigned long waitTime1 = 0;
44 // создаём объект класса QuadDisplay и передаём номер пина CS
45
46 // создаём объект класса QuadDisplay и передаём номер пина CS
47 QuadDisplay qd(9);
48
49 #define DEBUG 1 // Если поменять на "0", потребуется меньше памяти
50 #include <AFMotor.h> // подключаем библиотеку для шилда
```

```

51 AF_DCMotor motor2(2); // подключаем мотор к клеммникам M1
52 #include "DHT.h" // Стандартная библиотека, подключается в меню "Скетч"
53 #define DHTTYPE DHT11 // Если используется не DHT11, надо поменять
54 #include <Average.h>
55 #define relaypin 9 // Реле прицеплено к пину №9
56 #define dhtllpin 2 // Датчик прицеплен к пину №2
57
58 DHT dht(dhtllpin, DHTTYPE);
59 Average<float> ave(10); // Усреднения значений с датчика создаём массив на 10 шт
60
61 float treshold = 72; // Процент влажности, по достижению включаем вентилятор
62
63 unsigned long runInterval = 300000; // Если включаем вентилятор, на 5 минут
64 unsigned long idleInterval = 12000; // После выключения останавливаем на 2 минуты
65 // unsigned long previousMillis = 0;
66 const long interval = 1000; // Опрашиваем датчик с перерывами в одну секунду
67
68 unsigned int i = 0; // Счётчик
69
70 void setup(void)
71 {
72   // начало работы с дисплеем
73   qd.begin();
74   // пин помпы в режим выхода
75   pinMode(POMP_PIN, OUTPUT);
76   // выводим 0 на дисплей
77   qd.displayInt(0);
78   motor3.setSpeed(255); // задаем максимальную скорость мотора
79   motor3.run(RELEASE); // останавливаем мотор
80   motor4.setSpeed(255); // задаем максимальную скорость мотора
81   motor4.run(RELEASE); // останавливаем мотор
82
83 {
84   pinMode(relaypin, OUTPUT);
85   digitalWrite(relaypin, HIGH); // По умолчанию вентилятор не работает
86   if (DEBUG == 1) {
87     Serial.begin(9600);
88     Serial.print("Starting, humidity treshold set to: ");
89     Serial.print(treshold);
90     Serial.println("%.");
91   }
92   dht.begin();
93   motor2.setSpeed(255); // задаем максимальную скорость мотора
94   motor2.run(RELEASE); // останавливаем мотор
95 }
96
97 }
98

```

```

99 void loop(void)
100 {
101 // считываем текущее показания датчика влажности почвы
102 int humidityNow = analogRead(HUMIDITY_PIN);
103 // если показания текущей влажности почвы
104 // не равняется предыдущему запросу
105 if(humidityNow != humidity) {
106 // сохраняем текущие показания влажности
107 humidity= humidityNow;
108 // и выводим показания влажности на дисплей
109 qd.displayInt(humidityNow);
110 }
111 // если прошёл заданный интервал времени
112 // и значения датчика влажности меньше допустимой границы
113 if ((waitTime == 0 || millis() - waitTime > INTERVAL) && humidity < HUMIDITY_MIN ) {
114 // включаем помпу
115 motor3.run(FORWARD); // задаем движение вперед
116 motor3.setSpeed(255); // задаем скорость движения
117 delay(2000); // указываем время движения
118 motor3.run(RELEASE); // останавливаем мотор M1
119 delay(0); // Указываем время задержки
120 // приравниваем переменной waitTime
121 // значение текущего времени плюс 3 минуты
122 }
123
124 {
125 // считываем текущее показания датчика влажности почвы
126 int humidityNow1 = analogRead(HUMIDITY_PIN1);
127 // если показания текущей влажности почвы
128 // не равняется предыдущему запросу
129 if(humidityNow1 != humidity1) {
130 // сохраняем текущие показания влажности
131 humidity1= humidityNow1;
132 // и выводим показания влажности на дисплей
133 qd.displayInt(humidityNow1);
134 }
135 }
136
137 if ((waitTime1 == 0 || millis() - waitTime1 > INTERVAL1) && humidity1 < HUMIDITY_MIN1
138 // включаем помпу
139 motor4.run(FORWARD); // задаем движение вперед
140 motor4.setSpeed(255); // задаем скорость движения
141 delay(2000); // указываем время движения
142 motor4.run(RELEASE); // останавливаем мотор M1
143 delay(0); // Указываем время задержки
144 // приравниваем переменной waitTime
145 // значение текущего времени плюс 3 минуты
146

```

```

147 }
148 }
149
150 {
151   if (DEBUG == 1) {
152     Serial.print(i); i++;
153     Serial.print(": ");
154   }
155
156   float t = dht.readTemperature(); // Опрашиваем датчик
157   float h = dht.readHumidity();
158   if (isnan(h) || isnan(t)) {
159     Serial.println("Sensor error!");
160     delay(interval);
161     return;
162   } else {
163     ave.push(h); // Кладём результат измерения влажности в массив.
164     //В массиве помещается 10 измерений, новоприбывшее выталкивает самое старое.
165     if (DEBUG == 1) {
166       Serial.print("Humidity: ");
167       Serial.print(h);
168       Serial.print(" %\t");
169       Serial.print("Humidity running average: ");
170       Serial.print(ave.mean());
171       Serial.print(" %\t");
172       Serial.print("Temp.: ");
173       Serial.print(t);
174       Serial.println("C");
175     }
176
177     if (ave.mean() >= threshold) { // Вычисляем "скользящее среднее" по последним 10 замерам.
178       //сравниваем его с пороговым значением
179     }
180
181     motor2.run(FORWARD); // Если влажность высокая, включаем вентилятор...
182     motor2.setSpeed(255);
183
184     delay(15000);
185     motor2.run(RELEASE); // ...а потом даём ему остыть, какая бы ни оставалась влажность.
186     if (DEBUG == 1) {
187       Serial.println("Fan will cool down.");
188     }
189     // После того, как мы остудили мотор, надо "прочистить" массив с замерами влажности актуальными замерами.
190     //Иначе мотор будет врубаться снова и снова.
191     delay(idleInterval);
192   }
193 }
194 delay(interval);

```

```

195 }
196
197 #define CONFIG_START 0
198 #define CONFIG_VERSION "v01"
199
200 struct StoreStruct {
201     // это чтобы убедиться, что это ваши настройки:
202     char version[4];
203     // переменные для ваших настроек:
204     uint moduleId; // ID модуля
205 } storage = {
206     CONFIG_VERSION,
207     // дефолтный модуль «0»
208     0,
209 };
210
211 #define PARAM_HUMIDITY_TRESHOLD "/Sensor.Parameter1"
212 #define PARAM_MANUAL_AUTO_MODE "/Sensor.Parameter2"
213 #define PARAM_PUMP_ON "/Sensor.Parameter3"
214 #define PARAM_HUMIDITY "/Sensor.Parameter4"
215
216
217 #define MS_IN_SEC 1000 // 1S
218
219
220 MQTT myMqtt("", EIOT_CLOUD_ADDRESS, 1883);
221
222 // переменные:
223 int state;
224 bool stepOk = false;
225 int soilHumidityThreshold;
226 bool autoMode;
227 String valueStr("");
228 String topic("");
229 boolean result;
230 int lastAnalogReading;
231 bool autoModeOld;
232 int soilHumidityThresholdOld;
233 unsigned long startTime;
234 int soilHum;
235 int irrigatorCounter;
236
237 void setup() {
238     state = s_idle;
239     pinMode(PIN_PUMP, OUTPUT);
240     pinMode(PIN_BUTTON, INPUT);
241
242     autoMode = false;

```

```

243 stepOk = false;
244 soilHumidityThresholdOld = -1;
245 startTime = millis();
246 soilHum = -1;
247
248 Serial.begin(115200);
249
250 WiFi.mode(WIFI_STA);
251 WiFi.begin(AP_SSID, AP_PASSWORD);
252
253 Serial.println();
254 Serial.println();
255 Serial.print("Connecting to "); // "Подключение к "
256 Serial.println(AP_SSID);
257
258 while (WiFi.status() != WL_CONNECTED) {
259     delay(500);
260     Serial.print(".");
261 };
262
263 Serial.println("WiFi connected"); // "Подключились к WiFi"
264 Serial.println("Connecting to MQTT server");
265     // "Подключение к MQTT-серверу"
266
267
268 EEPROM.begin(512);
269 loadConfig();
270
271
272 // задаем ID клиента;
273 // генерируем название клиента на основе MAC-адреса
274 // и последних 8 бит счетчика микросекунд:
275 String clientName;
276 //clientName += "esp8266-";
277 uint8_t mac[6];
278 WiFi.macAddress(mac);
279 clientName += macToStr(mac);
280 clientName += "-";
281 clientName += String(micros() & 0xff, 16);
282 myMqtt.setClientId((char*) clientName.c_str());
283
284 Serial.print("MQTT client id:"); // "ID MQTT-клиента"
285 Serial.println(clientName);
286
287 // функции обратного вызова:
288 myMqtt.onConnected(myConnectedCb);
289 myMqtt.onDisconnected(myDisconnectedCb);
290 myMqtt.onPublished(myPublishedCb);

```



```

291 myMqtt.onData(myDataCb);
292
293 //Serial.println("connect mqtt...");
294 // "Подключение к MQTT..."
295 myMqtt.setUserPwd(EIOTCLOUD_USERNAME, EIOTCLOUD_PASSWORD);
296 myMqtt.connect();
297
298 delay(500);
299
300 Serial.print("ModuleId: ");
301 Serial.println(storage.moduleId);
302
303
304 // создаем модуль, если нужно:
305 if (storage.moduleId == 0)
306 {
307     // создаем модуль:
308     Serial.println("create module: /NewModule");
309     // "Создание модуля: /NewModule"
310
311     storage.moduleId = myMqtt.NewModule();
312
313     if (storage.moduleId == 0)
314     {
315         Serial.println("Module NOT created. Check module limit");
316         // "Модуль не создан. Проверьте лимит модулей"
317
318         while(1)
319             delay(100);
320     }
321
322     // задаем тип модуля:
323     Serial.println("Set module type"); // "Установка типа модуля"
324     myMqtt.SetModuleType(storage.moduleId, "ZMT_IRRIGATOR");
325
326     // задаем тип модуля:
327     Serial.println("Set module type"); // "Установка типа модуля"
328     myMqtt.SetModuleType(storage.moduleId, "ZMT_IRRIGATOR");
329
330     // создаем параметр Sensor.Parameter1;
331     // это пороговое значение для влажности почвы:
332     Serial.println("new parameter: /"+String(storage.moduleId)+ PARAM_HUMIDITY_TRESHOLD);
333     myMqtt.NewModuleParameter(storage.moduleId, PARAM_HUMIDITY_TRESHOLD);
334
335     // устанавливаем флаг «IsCommand»:
336     Serial.println("set isCommand: /"+String(storage.moduleId)+ PARAM_HUMIDITY_TRESHOLD);
337     myMqtt.SetParameterIsCommand(storage.moduleId, PARAM_HUMIDITY_TRESHOLD, true);
338

```

```

339
340 // создаем параметр Sensor.Parameter2;
341 // «0» - ручной режим; «1» - автоматический режим:
342 Serial.println("new parameter: /"+String(storage.moduleId)+ PARAM_MANUAL_AUTO_MODE);
343 myMqtt.NewModuleParameter(storage.moduleId, PARAM_MANUAL_AUTO_MODE);
344
345 // устанавливаем флаг «IsCommand»:
346 Serial.println("set isCommand: /"+String(storage.moduleId)+ PARAM_MANUAL_AUTO_MODE);
347 myMqtt.SetParameterIsCommand(storage.moduleId, PARAM_MANUAL_AUTO_MODE, true);
348
349
350 // создаем параметр Sensor.Parameter3;
351 // он отвечает за включение/выключение водяной помпы:
352 Serial.println("new parameter: /"+String(storage.moduleId)+ PARAM_PUMP_ON);
353 myMqtt.NewModuleParameter(storage.moduleId, PARAM_PUMP_ON);
354
355
356 // устанавливаем флаг «IsCommand»:
357 Serial.println("set isCommand: /"+String(storage.moduleId)+ PARAM_PUMP_ON);
358 myMqtt.SetParameterIsCommand(storage.moduleId, PARAM_PUMP_ON, true);
359
360
361 // создаем параметр Sensor.Parameter4;
362 // он отвечает за текущую влажность почвы:
363 Serial.println("new parameter: /"+String(storage.moduleId)+ PARAM_HUMIDITY);
364 myMqtt.NewModuleParameter(storage.moduleId, PARAM_HUMIDITY);
365
366
367 // задаем описание:
368 Serial.println("set description: /"+String(storage.moduleId)+ PARAM_HUMIDITY);
369 myMqtt.SetParameterDescription(storage.moduleId, PARAM_HUMIDITY, "Soil hum.");
370
371 // задаем единицу измерения:
372 Serial.println("set Unit: /"+String(storage.moduleId)+ PARAM_HUMIDITY);
373 myMqtt.SetParameterUnit(storage.moduleId, PARAM_HUMIDITY, "%");
374
375 // устанавливаем флаг «dbLogging»;
376 // он отвечает за запись логов в базу данных:
377 Serial.println("set Unit: /"+String(storage.moduleId)+ PARAM_HUMIDITY);
378 myMqtt.SetParameterDBLogging(storage.moduleId, PARAM_HUMIDITY, true);
379
380 // сохраняем ID нового модуля:
381 saveConfig();
382 }
383
384 subscribe();
385

```

```

386 lastAnalogReading = analogRead(PIN_HUM_ANALOG);
387
388 autoModeOld = !autoMode;
389 }
390
391 void loop() {
392   while (WiFi.status() != WL_CONNECTED) {
393     delay(500);
394 #ifdef DEBUG
395     Serial.print(".");
396 #endif
397   }
398
399   int in = digitalRead(PIN_BUTTON);
400
401   //Serial.println(in);
402   if (in == 0)
403   {
404     while(digitalRead(PIN_BUTTON) == 0)
405       delay(100);
406
407     if (state == s_idle || state == s_irrigation_start)
408       state = s_irrigation_start;
409     else
410       state = s_irrigation_stop;
411   }
412
413 #include <iarduino_RTC.h>
414 iarduino_RTC time(RTC_DS1307);
415 String day_RTC; // строка для хранения текущего дня.
416 int day_1, day_2; // строка для хранения заданных дней.
417 bool Push_M1;
418 int Hhh1_MAX, Hhh2_MAX, Hhh3_MAX, Hhh1_MIN, Hhh2_MIN, Hhh3_MIN; //Переменные для задания уровня влажности.
419 int Hhh; //Показания датчика.
420 int d; // Дни в int
421 int dd = 0;
422 void setup()
423 {
424   delay(300);
425   time.begin();
426   time.setTime(0,51,21,27,10,20,2); // 0 сек, 51 мин, 21 час, 27, октября, 2020 года, вторник
427 }
428
429 void loop()
430 {
431   day_RTC = time.getTime("d");
432   d = day_RTC.toInt();
433   if (d < day_1)
434   {

```

```

435 Contrl_Hhh (Hhh1_MIN, Hhh1_MAX);
436 }
437 if (d >= day_1 && dd == 1)
438 {
439 Contrl_Hhh (Hhh2_MIN, Hhh2_MAX);
440 }
441 if (d >= day_2)
442 {
443 Contrl_Hhh (Hhh3_MIN, Hhh3_MAX);
444 dayofday (dd);
445 }
446
447 }
448
449 int Contrl_Hhh (int Hhh_MIN, int Hhh_MAX)
450 {
451 if (Hhh < Hhh_MIN); // Если значение датчика влажности меньше чем значение уставки включаем насос.
452 {
453 Pusk_M1=1;
454 }
455 if (Hhh > Hhh_MAX); //Если значение датчика влажности больше чем значение уставки выключаем насос.
456 {
457 Pusk_M1=0;
458 }
459 return;
460 }
461
462 int dayofday (int dd)
463 {
464 dd=dd+1;
465 }
466

```

---