

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

ВЕБ-ОРИЕНТИРОВАННАЯ СИСТЕМА МОНИТОРИНГА ОБЪЕКТОВ

Выпускная квалификационная работа
обучающегося по направлению подготовки 09.03.02 Информационные системы
и технологии
очной формы обучения, группы 07001407
Мартынюка Сергея Леонидовича

Научный руководитель
к.г.н., доцент
Шамраев А.А.

БЕЛГОРОД 2018

РЕФЕРАТ

Веб-ориентированная система мониторинга объектов. Мартынюк Сергей Леонидович, выпускная квалификационная работа бакалавра, Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 54, включая приложения 55, количество рисунков 33, количество таблиц 2, количество использованных источников 28.

КЛЮЧЕВЫЕ СЛОВА: мониторинг, датчик, микроконтроллер, сигнал, питание.

ОБЪЕКТ ИССЛЕДОВАНИЯ: процесс веб-ориентированного мониторинга объекта средствами созданного решения.

ПРЕДМЕТ ИССЛЕДОВАНИЯ: спроектированная автоматизированная система мониторинга.

ЦЕЛЬ РАБОТЫ: автоматизация процесса удаленного мониторинга объектов.

ЗАДАЧИ ИССЛЕДОВАНИЯ:

- исследование предметной области;
- разработка общего перечня требований к системе мониторинга;
- проектирование программно-аппаратных модулей;
- разработка структурных схем;
- реализация аппаратно-программных модулей системы;
- тестирование.

МЕТОДЫ ИССЛЕДОВАНИЯ:

- методы определения влажности почвы;
- методы определения влажности и температуры воздуха;
- методы передачи данных от микроконтроллера к серверу;
- методы обработки полученной информации с датчиков;
- методы визуализации показаний с датчиков.

ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ: В работе исследованы принципы мониторинга с использованием различных датчиков установленных на микроконтроллер. Разработан сервер для мониторинга, принимающий показания, снятые с датчиков. Разработано клиентское веб-приложение, способное визуализировать полученные данные и отправлять уведомления с указанными критическими значениями.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Анализ состояния проблемы и формирование технического задания на разработку	8
1.1 Системы капельного полива	8
1.2 Системы управления микроклиматом	10
1.3 Котельное оборудование	13
2 Разработка функциональной схемы устройства	19
2.1 Описание выбранной платформы.....	19
2.2 Описание датчиков, схема их подключения	24
3 Разработка программного обеспечения веб–ориентированной системы.....	38
3.1 Выбор и обоснование средств реализации	38
3.2 Разработка алгоритмов функционирования системы	40
3.3 программная реализация разработанных алгоритмов.....	43
3.4 Тестирование	54
ЗАКЛЮЧЕНИЕ	56
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	57
ПРИЛОЖЕНИЕ А	60

ВВЕДЕНИЕ

Для принятия верного абсолютно любого управленческого решения – от изменения стоимости до открытия новой точки продажи или создания совершенно нового бизнеса нужна определённая информация. Однако решения приходится принимать постоянно, следовательно, так же требуется постоянная информация для их принятия, а маркетинговые исследования проводятся через довольно длительные промежутки времени. В то же время, в распоряжении любой организации зачастую находится почти вся необходимая информация, но она не систематизирована, не проанализирована и не готова к использованию.

Построение веб-ориентированной системы мониторинга объектов можно проводить на примере абсолютно любого объекта с возможностью подключения к интернету. Решение данной задачи будем проводить на примере системы «Умной теплицы» с возможностью удаленного мониторинга и управления, так как сельскохозяйственное направление является перспективным в нашей стране.

Практическая значимость заключается в том, что результаты работы, а также изученные методы системы маркетинговой информации могут принести практическую пользу разным предприятиям, фирмам и частным предпринимателям для планирования и осуществления анализа.

Актуальность идеи «Умной теплицы» как самостоятельной единицы и как части проекта «Умный дом» весьма высока именно сегодня, когда мы уже не просто узнаем об Интернете вещей с голубых экранов наших гаджетов, а уже частично используем элементы IoT у себя дома. Правда пока это еще мало похоже на единую слаженную систему работы всех наших приборов, но потребность уже в этом созрела. Так, например, уже сейчас среди населения нашей страны существует острая неудовлетворенная потребность в некоем «автономном домашнем помощнике» по уходу как за своими урожаями как в загородных теплицах, так и в квартирных

«цветниках». При этом управление данной системой должно быть понятно обычному пользователю, быть социализировано в современном информационном пространстве (социальные сети, электронные почты и т.д.). И система должна быть недорога в приобретении.

Оцениваем значимость данного исследования в перспективах данного проекта для практического внедрения в сельском хозяйстве: как в промышленных масштабах, так и для индивидуального пользования. Так же этот проект вызовет высокий интерес у крайне занятых жителей городских квартир, занимающихся выращиванием саженцев для своего огорода или просто декоративных растений для красоты и уюта в доме.

Особый интерес работы заключается в новизне включения процесса ухода за своим урожаем в социальную жизнь пользователя: теперь результатами своего труда (в виде фото урожая) можно делиться в социальных сетях. Что делает этот проект весьма привлекательным для молодой аудитории.

С другой, практической стороны, теперь у человека появилась возможность в режиме реального времени удалённо наблюдать за своими растениями, отслеживать параметры работы теплицы, отвечающие за здоровье урожая и, при необходимости, удаленно вмешиваться в автономную работу теплицы, осуществляя дополнительный полив или, например, увеличивая освещение в теплице.

Ожидаемыми результатами проекта являются создание коммерческого комплекта для сборки умной теплицы, с помощью которого можно будет автоматизировать и удаленно управлять выращиванием различных культур. Данную систему можно интегрировать любую, уже посторожённую, теплицу или домашний цветник. Управление теплицей будет достаточно понятным даже для пожилых людей, не умеющих работать с компьютером.

Целью работы является разработка веб-ориентированной системы мониторинга объектов.

Набор выполняемых задач:

- изучить предметную область;
- сравнить с конкурентами;
- спроектировать и собрать набор датчиков в единый прибор;
- реализовать систему программно;
- тестирование системы.

1 Анализ состояния проблемы и формирование технического задания на разработку

1.1 Системы капельного полива

Капельный полив по малообъемной технологии для овощных, цветочных и рассадных комплексов, орошение с возвратным раствором. На сегодняшний день ни один новый тепличный комбинат не обходится без системы капельного полива. Это основа основ, позволяющая подавать под растения строго дозированное количество удобрений и воды в нужное время в нужной пропорции. Современное капельное орошение предусматривает установку целого комплекса инженерных систем и сетей, обеспечивающих в автоматическом режиме растения требуемой подкормкой. Все технологическое оборудование работает под управлением одной компьютерной программы, с помощью которой легко можно проводить контроль процессов полива в реальном времени, а также просматривать архивные данные и вносить корректировки. Система капельного полива состоит из нескольких составляющих:

- приготовления воды (фильтрация, нагрев, стабилизация бикарбонатов и т.д.);
- приготовление раствора (растворный узел – мозг и сердце системы капельного полива);
- магистральный трубопровод и капельная сеть (раздача питательного раствора растениям);
- также к системе капельного полива можно отнести следующие составляющие;
- система повторного использования дренажа (подача использованного раствора обратно в полив);
- приготовление маточных растворов (растворение сухих удобрений в воде) [1].

На рисунке 1 представлена схема простой системы капельного полива.



Рисунок 1 – Система капельного полива

Современная технология предусматривает высокие требования к качеству поливочной воды. В ней должно содержаться определенное количество микро и макроэлементов, а также существуют требования к уровню pH и чистоте воды. Дополнительным фактором является то, что вода должна поступать к растению выдержанной температурой. Узлы фильтрации поливочной воды. Многоступенчатые узлы, состоящие из песчано-гравийных и дисковых фильтров, очищают воду от механических частиц до 130 микрон. Также фильтры помогают избавиться от органических соединений, часто попадающих в воду из открытых водоемов. Узел снабжен электроникой, которая по мере загрязнения фильтра осуществляет его автоматическую промывку. Фильтрация воды в системах капельного полива получила очень широкое распространение. Она необходима при подготовке воды к поливу, для предотвращения засорения капельниц, а также для обеспечения бесперебойной и длительной работы инженерного оборудования. Кроме этого, фильтрация используется в системах с рециркуляцией дренажного раствора. Как известно, дренаж перед повторным использованием должен

быть продезинфицирован, но любая дезинфицирующая установка, вне зависимости от типа, требует предварительной фильтрации от содержащихся в дренаже взвесей. Песчано-гравийные фильтры (любой производительности) очищают воду от примесей и взвесей, с которыми не могут справиться сетчатые или пластинчатые фильтры. Устройство выполнено на нержавеющей раме, как независимый модуль и легко интегрируется, как в существующие системы, так и в новые проекты. Для удобства использования промывка фильтра осуществляется в автоматическом режиме. Дополнительную очистку воды обеспечивает фильтр тонкой очистки (130 микрон). Надежные и практичные пластинчатые фильтры обеспечивают механическую очистку растворов от фракций более 130 мкм. Фильтрующие элементы легко промываются и имеют практически неограниченный срок службы. Устройство может поставляться, как с насосом, который выполняет функцию подачи воды из накопительной емкости в растворный узел, так и без него.

1.2 Системы управления микроклиматом

Получение высоких результатов при выращивании овощей, цветов, рассады, салата в теплицах напрямую зависит от качества обеспечения растений оптимальными условиями роста и развития. И одним из важнейших таких является четко сбалансированный микроклимат. Еще несколько лет назад управление микроклиматом могло осуществляться вручную оператором, но с появлением большого количества инженерных систем в теплице, а также с повышением требований к качеству поддержания микроклимата, на сегодняшний день, ни одна промышленная теплица не обходится без системы автоматического управления. Современная теплица включает в себя множество исполнительных инженерных систем, то есть устройств, которые позволяют управлять микроклиматом теплицы.

Система отопления. Основная задача системы отопления – это поддержание заданной агрономом температуры. Как правило, отопление теплицы состоит из нескольких отдельных контуров(систем). Регулирование температуры воздуха осуществляется с помощью изменения температуры воды в контурах, что осуществляется с помощью смесительного клапана, который смешивает в необходимой пропорции воду от теплоисточника (прямой теплоноситель) с водой, вернувшейся из теплицы (обратным теплоносителем).

Система форточной вентиляции. Для вентилирования воздуха внутри теплицы в кровле предусматриваются фрамуги(форточки), площадь которых составляет до 30% от общей площади остекления. Открытие и закрытие форточек производится с помощью моторизированных редукторов.

Система зашторивания. Практически все новые проекты теплиц комплектуются системой зашторивания, т.к. она позволяет экономить до 30% теплоресурсов, а также защитить растения от солнечных ожогов. Принцип ее работы заключается в том, что специальный полимерный материал сворачивается и разворачивается над растениями, отсекая тем самым кровлю. Свертывание и разворачивание производится с помощью специальных моторредукторов [2].

Система подкормки CO₂. Важнейшим параметром микроклимата наряду с температурой и влажностью является концентрация углекислого газа в воздухе. Это связано с тем, что углерод является основным строительным материалом для растения и процесс фотосинтеза (производства сухого вещества) без CO₂ невозможен. Наиболее экономичным решением является использование отходящих газов котельной для подкормки, и на большинстве тепличных комбинатах так и происходит. Но в некоторых случаях приходится прибегать к использованию жидкой углекислоты для подкормки, что является более дорогим решением [3].

Система досвечивания. Современная интенсивная технология выращивания овощей предусматривает установку системы

ассимиляционного освещения. А для выращивания цветов система досвечивания вообще является необходимым условием. Основная задача системы обеспечить определенный уровень освещенности для выращивания в те моменты, когда естественного света недостаточно.

Система рециркуляции воздуха. Для обеспечения воздухообмена в теплице в верхней ее части устанавливаются вентиляторы. При включении они обеспечивают движение воздуха, выравнивание теплового поля и ускоряют конвективный теплообмен;

Автоматическая система управления микроклиматом. Она предназначена для связи всех вышеперечисленных систем в единое целое, в единый процесс с централизованным управлением. Системы отопления и вентиляции, зашторивания и досвечивания, CO₂ и рециркуляция все работает под управлением специального компьютера, который отвечает за то, чтобы режим микроклимата в теплице точно совпадал с заданием агронома.

На рисунке 2 представлена автоматическая система управления микроклиматом.



Рисунок 2 – Системы управления микроклиматом

1.3 Котельное оборудование

Предлагаемые котельные для теплиц отвечают самым передовым требованиям, включают в себя теплоаккумулирующие емкости, позволяющие вырабатывать в дневные часы необходимый для растений углекислый газ. Специальный компьютер, в зависимости от требуемого технологического режима теплицы, автоматически управляет котельной. Как известно, большую часть себестоимости производимой в теплицах продукции составляют энергозатраты. Поэтому, качество используемого энергетического оборудования выходит на первый план. В качестве котлового оборудования используются такие мировые бренды, как «Crone», «Viessmann», «НКВ», которые позволяют получать высокое КПД горения с одновременно низкой эмиссией газов оксидов азота, а также большим процентом выхода CO_2 . Последний факт позволяет успешно использовать дымовые газы от данных котлов для подкормки растений углекислым газом. Современные котлы оборудованы рядом специальных подсистем, которые гарантируют длительную бесперебойную работу, например:

- каждый котел оборудован шунтирующим насосом, который обеспечивает циркуляцию воды внутри котла, делая процесс нагрева воды более равномерным;

- каждый котел оборудован демпферным насосом, который включается при низких температурах обратной воды и не допускает образование конденсата в котле [4].

В качестве горелочных устройств предлагаем известные марки «Zantingh», «Thermeta», «Weishaupt». Отличительной особенностью используемых горелок является широкий диапазон регулируемой мощности от 10% до 100%. Сравнивая новые горелки с предшествующими, необходимо отметить наличие частотных преобразователей на вентиляторах подачи воздуха, что позволяет существенно экономить на электроэнергии. Для повышения КПД горения, а также для возможности отбора CO_2 часть котлов,

либо все, оборудуются конденсаторами отходящих газов. Эффект повышения КПД заключается в том, что в конденсаторе отходящие газы охлаждаются с 180 гр до 60гр, нагревая при этом теплоноситель, а охлажденные таким образом газы после измерения уровня CO можно беспрепятственно подавать в теплицу. Использование CO₂ для подкормки растений дает до 15% прибавки урожайности. Вся работа котельной находится под управлением компьютера, который по запросу от теплиц автоматически увеличивает или уменьшает мощность котельной, подает CO₂. Особенность тепличного производства состоит в том, что углекислый газ нужен в солнечную погоду днем и не нужен ночью. Ситуация с отоплением обстоит наоборот: основное потребление тепла идет ночью, а днем потребности в нем практически нет или в малых количествах. Для обеспечения растений в дневное время CO₂ современная котельная оборудуется баком-аккумулятором. Принцип действия его заключается в том, что днем котельная включается на выработку CO₂, и выталкивая из бака холодную воду, замещает ее на горячую. В ночное же время, наоборот, горячая вода из бака поступает в теплицы, а охлажденная вода с теплиц возвращается в бак. Таким образом, растения снабжаются необходимым количеством углекислого газа без расхода тепла. Все управление баками аккумуляторами осуществляется в полностью автоматическом режиме [5].

Как можем заметить на сегодняшний день существует множество различных систем, которые помогают человеку управлять теплицей, но к сожалению, все это системы находятся разрозненно друг по отношению к другу. Поэтому создание устройства, которое объединит в единую экосистему все необходимые функции для удаленного управления теплицей довольно востребовано в данный момент.

Одно из преимуществ Arduino – ему не нужен программатор. Всё сделано так, чтобы программирование Arduino для начинающих не составляло труда. Написанный код можно загрузить в МК посредством USB-шнура. Достигается это преимущество не каким-то встроенным уже заранее

программатором, а специальной прошивкой – бутлоадером. Бутлоадер является специальным программным обеспечением, которая запускается сразу после подключения и слушает, будут ли какие-то команды, прошивать ли процессор, есть ли проекты Arduino или нет. Из использования бутлоадера выплывает несколько очень привлекательных плюсов:

- использование только одного канала связи, что не требует дополнительных затрат по времени;
- так, проекты Arduino не требуют, чтобы вы подключали множество различных проводов, и возникала путаница при их использовании;
- для успешной работы хватает одного USB-шнура;
- довести микроконтроллер до состояния бесполезной железяки с помощью прямой прошивки довольно легко, сильно напрягаться не надо. При работе с бутлоадером до потенциально опасных настроек вам не добраться.

Пользователь может самостоятельно запрограммировать загрузчик в чистый микроконтроллер. Для этого в IDE интегрирована поддержка программатора на основе проекта AVRDUDE. Поддерживается несколько типов популярных дешёвых программаторов [6].

Под торговой маркой Arduino выпускается несколько плат с микроконтроллером и платы расширения. Большинство плат с микроконтроллером снабжены минимально необходимым набором обвязки для нормальной работы микроконтроллера (стабилизатор питания, кварцевый резонатор, цепочки сброса и т. п.).

Arduino и Arduino-совместимые платы спроектированы таким образом, чтобы их можно было при необходимости расширять, добавляя в устройство новые компоненты. Эти платы расширений подключаются к Arduino посредством установленных на них штыревых разъёмов. Существует ряд плат с унифицированным конструктивом, допускающим конструктивно жесткое соединение процессорной платы и плат расширения в стопку через штыревые линейки. Кроме того, выпускаются платы уменьшенных габаритов

(например, Nano, Lilypad) и специальных конструктивов для задач робототехники. Независимыми производителями также выпускается большая гамма всевозможных датчиков и исполнительных устройств, в той или иной степени совместимых с базовым конструктивом Ардуино.

В концепцию Arduino не входит корпусной или монтажный конструктив. Разработчик выбирает метод установки и механической защиты плат самостоятельно [7].

Микроконтроллеры для Arduino отличаются наличием предварительно прошитого в них загрузчика (bootloader). С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов. Загрузчик соединяется с компьютером через интерфейс USB (если он есть на плате) или с помощью отдельного переходника UART-USB. Поддержка загрузчика встроена в Arduino IDE и выполняется в один щелчок мыши.

На случай затирания загрузчика или покупки микроконтроллера без загрузчика разработчики предоставляют возможность прошить загрузчик в микроконтроллер самостоятельно. Для этого в Arduino IDE встроена поддержка нескольких популярных дешевых программаторов, а большинство плат Arduino имеет штыревой разъем для внутрисхемного программирования.

В Arduino IDE от компании, базирующейся на сайте arduino.cc, встроена возможность создания своих программно-аппаратных платформ. Этой возможностью пользуются сторонние компании, добавляющие в Arduino IDE свои наборы плат и компиляторов-загрузчиков к ним. Компания на сайте arduino.org не поддерживает такую возможность [8].

Разновидности микроконтроллеров Arduino:

– AVR. В линейке устройств Arduino в основном применяются микроконтроллеры Atmel AVR: ATmega328, ATmega168, ATmega2560, ATmega32U4, ATTiny85 с частотой тактирования 16 или 8 МГц. В старых изделиях применялись ATmega8, ATmega1280 и другие;

- Есть также платы на процессоре ARM Cortex M;
- ESP8266. Сторонние разработчики портировали в Arduino поддержку популярного Wi-Fi микроконтроллера ESP8266. Теперь компилировать и загружать прошивку для ESP8266 со своими скетчами и поддержкой Wi-Fi можно прямо из Arduino IDE, получая одноплатную схему с поддержкой сети Wi-Fi. Подробное русскоязычное описание процесса установки и доступного API;
- Intel x86. В рамках сотрудничества со сторонними производителями в Arduino IDE была включена поддержка некоторых аппаратных средств Intel x86. Intel Galileo (процессор Intel Quark X1000 400 МГц), Intel Edison и Arduino 101 — Arduino-совместимые платы на Intel x86 архитектуре. Платы механически и электрически совместимы с периферийными платами Ардуино. Платы функционируют под собственной ОС Linux, поверх которой работает приложение, позволяющее загружать и исполнять скетчи Arduino [9].

Порты ввода–вывода микроконтроллеров оформлены в виде штыревых линеек. Никакого буферизирования, защиты, конвертации уровней или подтяжек, как правило, нет. Микроконтроллеры питаются от 5В или 3,3В, в зависимости от модели платы. Соответственно порты имеют такой же размах допустимых входных и выходных напряжений. Программисту доступны некоторые специальные возможности портов ввода-вывода микроконтроллеров, например, широтно-импульсная модуляция, аналогово-цифровой преобразователь, интерфейсы UART, SPI, I2C. Количество и возможности портов ввода-вывода определяются конкретным вариантом микропроцессорной платы.

Помимо портов на платах микроконтроллеров иногда устанавливается периферия в виде интерфейсов USB или Ethernet. Опциональный набор внешней периферии на модулях расширения включает в себя:

- USB Device (чаще всего как виртуальный COM порт через FTDI FT232, имеются также версии с эмуляцией USB HID Class клавиатур и мышек);

- проводной и беспроводной Ethernet как на основной плате, так и на платах расширения;

- модуль GSM и другие беспроводные интерфейсы;

- USB Host;

- SD card;

- модуль управления низковольтным мотором на базе L298.

Поддерживаются шаговый и коллекторный двигатели с напряжением до 12В и током до 2А на канал. Могут подключаться также реле, электромагниты и т.п. Модуль не имеет гальванической развязки;

- графический ЖКИ индикатор;

- модуль с макетным полем.

Сторонние производители выпускают широкую гамму датчиков и исполнительных устройств, подключаемых к Arduino. Например, гироскопы, компасы, манометры, гигрометры, термометры, релейные модули, индикаторы, клавиатуры и т. п. Всё это превращает Arduino в универсальное ядро системы, которое может быть сконфигурировано совершенно разнообразными способами.

Из выше сказанного напрашивается вывод что существующие системы либо очень дорогие, либо не выполняют необходимый функционал для решения поставленных задач. Оптимальным вариантом является создание системы на базе платформы Arduino за его достойные характеристики и простоту создания устройств.

2 Разработка функциональной схемы устройства

2.1 Описание выбранной платформы

Arduino UNO R3 выполнен на микроконтроллере ATmega328. У него (Таблица 1):

- 14 цифровых портов входа-выхода (6 из них поддерживают режим ШИМ модуляции);
- 6 аналоговых входов;
- частота тактирования 16 МГц;
- USB порт;
- разъем питания;
- разъем внутрисхемного программирования;
- кнопка сброса [10].

Таблица 1 – Характеристики платформы Arduino UNO R3

Характеристики	Значение
Напряжение питания микроконтроллера	5 В
Рекомендуемое напряжение питания платы	7-12 В
Предельно допустимое напряжение питания платы	6 – 20 В
Цифровые входы-выходы	14 (из них 6 поддерживают ШИМ)
Выходы ШИМ модуляции	6
Аналоговые входы	6
Допустимый ток цифровых выходов	20 мА
Допустимый ток выхода 3,3 В	50 мА
Объем флэш памяти (FLASH)	32 кБ (из которых 0,5 кБ используется загрузчиком)

Продолжение таблицы 1– Характеристики платформы Arduino UNO R3

Характеристика	Значение
Объем оперативной памяти (SRAM)	2 кБ
Объем энергонезависимой памяти (EEPROM)	1 кБ
Частота тактирования	16 МГц
Длина платы	68,6 мм
Ширина платы	53,4 мм
Вес	25 г

У платы есть все необходимые компоненты для обеспечения работы микроконтроллера (Приложение А). Достаточно подключить USB кабель к компьютеру и подать питание. Микроконтроллер установлен на колодке, что позволяет легко заменить его в случае выхода из строя.

Контроллер программируется из интегрированной среды программного обеспечения Arduino (IDE). Программирование происходит под управлением резидентного загрузчика по протоколу STK500. Аппаратный программатор при этом не требуется.

Микроконтроллер можно запрограммировать через разъем для внутрисхемного программатора ICSP, не используя загрузчик. Исходный код программы – загрузчика находится в свободном доступе.

Arduino UNO R3 (Рисунок 3), в отличие от предыдущих версий, не использует для подключения к компьютеру мост USB-UART FTDI. Эту функцию в нем выполняет микроконтроллер ATmega16U2.

Плата UNO может получать питание от USB порта или от внешнего источника. Источник питания выбирается автоматически. В качестве внешнего источника питания может использоваться сетевой адаптер или батарея. Адаптер подключается через разъем диаметром 2,1 мм. Батарея подключается к контактам GND и Vin разъема POWER (Таблица 2).

Напряжение внешнего источника питания может быть в диапазоне 6 – 20В. Но рекомендуется не допускать снижение напряжения ниже 7В из-за нестабильной работы устройства. Также нежелательно повышать напряжение питания более 12В, т.к. может перегреться стабилизатор и выйти из строя. Т.е. рекомендуемый диапазон напряжения питания 7 – 12 В [11].

Таблица 2 – Выводы для подключения питания

Пин	Описание
Vin	Питание платы от внешнего источника питания. Не связано с питанием 5В от USB или выходами других стабилизаторов. Через этот контакт можно получать питание для своего устройства, если плата питается от адаптера.
5 V	Выход стабилизатора напряжения платы. На нем напряжение 5В при любом способе питания. Питая плату через этот вывод не рекомендуется, т.к. не используется стабилизатор, что может привести к выходу микроконтроллера из строя.
3 V 3	Напряжение 3,3В от стабилизатора напряжения на плате. Предельно допустимый ток потребления от этого вывода 50 мА.
GND	Общий провод.
IOREF	На выводе информация о рабочем напряжении платы. Плата расширения может считать значение сигнала и переключиться на режим питания 5В или 3,3В.

У микроконтроллера три типа памяти:

- 32кБ флэш(FLASH);
- 2кБ оперативной памяти (SRAM);
- 1кБ энергонезависимой памяти (EEPROM).

Каждый из 14 цифровых выводов может быть использован в качестве выхода или входа. Уровень напряжения на выводах 5 В. Рекомендовано вытекающий и втекающий ток каждого вывода ограничивать на уровне 20 мА. Предельно допустимое значение этого параметра составляет 40 мА. Каждый вывод имеет внутренний подтягивающий резистор сопротивлением 20-50 кОм. Резистор может быть отключен программно.

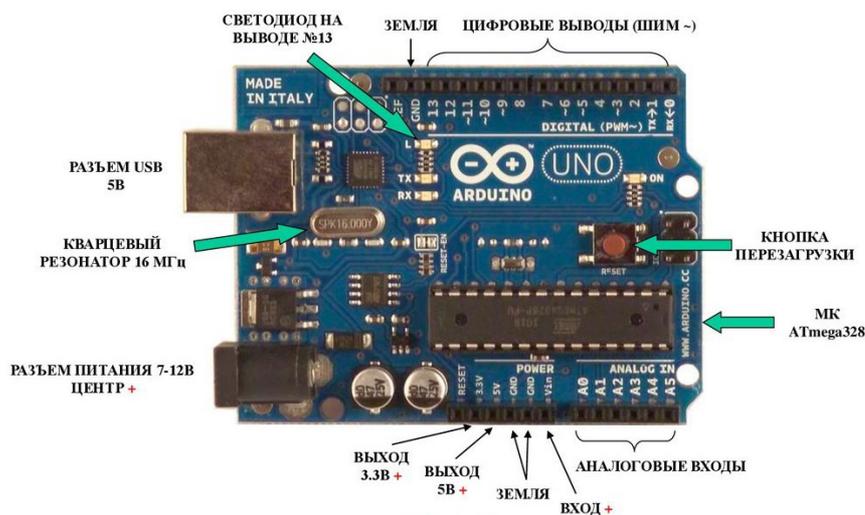


Рисунок 3 – микроконтроллер Arduino

Последовательный интерфейс: выводы 0 (Rx) и 1 (Tx). Используются для приема (Rx) и передачи (Tx) последовательных данных логических уровней TTL. Эти выводы подключены к выводам передачи данных микросхемы ATmega16U2, используемой в качестве моста USB-UART.

Внешние прерывания: выводы 2 и 3. Эти выводы могут быть использованы как входы внешних прерываний. Программно могут быть установлены на прерывание по низкому уровню, положительному или отрицательному фронту, или на изменение уровня сигнала.

ШИМ: выводы 3, 5, 6, 9, 10, 11. Могут работать в режиме ШИМ модуляции с разрешением 8 разрядов.

Последовательный интерфейс SPI: выводы 10(SS), 11(MOSI), 12(MISO), 13(SCK).

Светодиод: вывод 13. Светодиод, подключенный к выводу 13. Светится при высоком уровне сигнала на выводе.

Интерфейс TWI: вывод A4 или SDA и A5 или SCL. Коммуникационный интерфейс TWI.

У платы Arduino UNO есть 6 аналоговых входов, обозначенных A0-A5. Разрешающая способность аналогового цифрового преобразования 10 разрядов. По умолчанию, входное напряжение измеряется относительно

земли в диапазоне 0-5 В, но может быть изменено с помощью вывода AREF и программных установок.

Еще 2 вывода платы имеют функции:

- AREF. Опорное напряжение АЦП микроконтроллера;
- RESET. Низкий уровень на этом выводе вызывает сброс микроконтроллера.

Модуль Arduino UNO имеет средства для связи с компьютером, с другой платой UNO или с другими микроконтроллерами. Для этого на плате существует интерфейс UART с логическими уровнями TTL (5 В), связанный с выводами 0 (RX) и 1(TX). Микросхема ATmega16U2 на плате связывает UART интерфейс с USB портом компьютера. При подключении к порту компьютера, появляется виртуальный COM порт, через который программы компьютера работают с Arduino. Прошивка ATmega16U2 использует стандартные драйверы USB-COM, и установка дополнительных драйверов не требуется. Для операционной системы Windows необходим соответствующий .inf файл. В интегрированную среду программного обеспечения Arduino (IDE) включен монитор обмена по последовательному интерфейсу, который позволяет посылать и получать с платы простые текстовые данные. На плате есть светодиоды RX и TX, которые индицируют состояние соответствующих сигналов для связи через USB (но не для последовательного интерфейса на выводах 0 и 1).

Микроконтроллер ATmega328 также поддерживает коммуникационные интерфейсы I2C (TWI) и SPI.

Для того, чтобы не приходилось каждый раз перед загрузкой программы нажимать кнопку сброс, на плате UNO реализована аппаратная функция сброса, инициируемая с подключенного компьютера. Один из сигналов управления потоком данных (DTR) микросхемы ATmega16U2 подключен к выводу сброса микроконтроллера ATmega328 через конденсатор емкостью 0,1 мкФ. Когда сигнал DTR переходит в низкое состояние, формируется импульс сброса микроконтроллера. Это решение

позволяет загружать программу одним нажатием кнопки из интегрированной среды программирования Arduino (IDE).

Но такая функция может приводить к отрицательным последствиям. При подключении платы UNO к компьютеру с операционной системой Mac Os X или Linux, микроконтроллер будет сбрасываться при каждом соединении программы с платой. В течение половины секунды на плате UNO будет запущен загрузчик. Несмотря на то, что программа загрузчика игнорирует посторонние данные, она может принять несколько байтов из пакета сразу после установки соединения. Если в программе на плате Arduino предусмотрено получение каких-либо данных при первом запуске, необходимо отправлять данные с задержкой примерно на 1 секунду после соединения.

На модуле UNO существует дорожка, которую можно перерезать для отключения функции автоматического сброса. Дорожка маркирована надписью «RESET-EN». Автоматический сброс также можно запретить, подключив резистор сопротивлением 110 Ом между линией питания 5 В и выводом RESET.

В плате Arduino UNO линия питания от интерфейса USB защищена восстанавливаемым предохранителем. При превышении тока свыше 500 мА, предохранитель разрывает цепь до устранения короткого замыкания.

2.2 Описание датчиков, схема их подключения

Реле – электрическое или электронное устройство (ключ), предназначенное для замыкания или размыкания электрической цепи при заданных изменениях электрических или неэлектрических входных воздействий. На рисунке 4 представлено реле [12].



Рисунок 4 – Реле

В данном примере используется стандартный модуль реле ардуино, на котором уже установлены все необходимые элементы для подключения к. Схема подключения очень проста: модуль реле присоединяется к 5 пину платы Arduino. При этом для простоты мы можем даже не присоединять реальную нагрузку – реле будет щелкать при каждом изменении состояния, мы услышим эти щелчки и будем понимать, что скетч работает (Рисунок 5) [13].

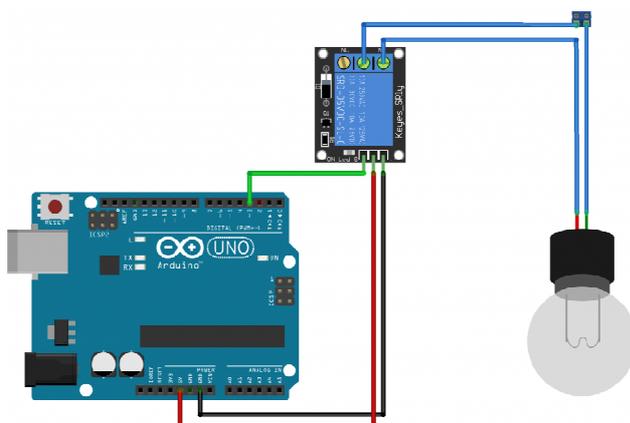


Рисунок 5 – Схема подключения реле к Arduino

Сервопривод – устройство, обеспечивающее преобразование сигнала в строго соответствующее этому сигналу перемещение (как правило, поворот) исполнительного устройства. Представляют собой прямоугольную коробку с мотором, схемой и редуктором внутри и выходным валом, который может поворачиваться на строго фиксированный угол, определяемый входным сигналом. Как правило, этот угол имеет предел в 60 градусов, иногда 180. На рисунке 6 представлен сервопривод [14].



Рисунок 6 – Сервопривод

Сервопривод обладает тремя контактами, которые окрашены в разные цвета. Коричневый провод ведет к земле, красный – к питанию +5В, провод оранжевого или желтого цвета – сигнальный. К Arduino устройство подключается через макетную указанным на рисунке образом. Оранжевый провод (сигнальный) подключается к цифровому пину, черный и красный – к земле и питанию соответственно. Подключение сервопривода к микроконтроллеру представлено на рисунке 7 [15].

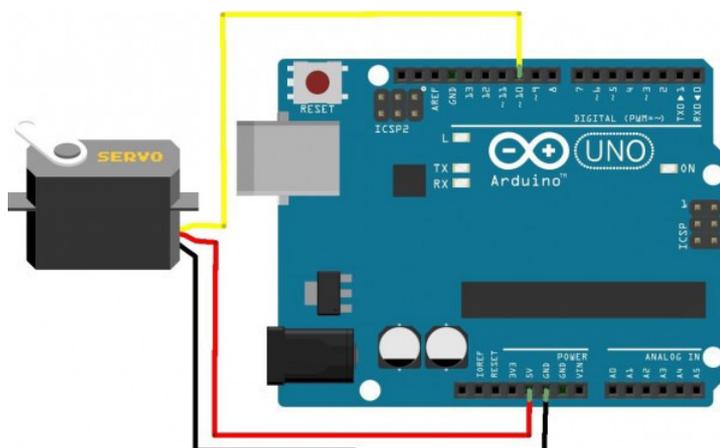


Рисунок 7 – Схема подключения сервопривод к Arduino

Название "геркон" происходит от словосочетания "герметичный контакт". И это объясняет его устройство. По сути, геркон – это два разомкнутых (или замкнутых) контакта, находящихся в вакуумной колбе, которые меняют своё состояние на противоположное при воздействии на них магнитного поля [16].



Рисунок 8 – Геркон

Герконы – очень популярные датчики, которые используются во многих задачах. Это и контроль открытия/закрытия дверей, разнообразные счётчики срабатываний, счётчики скорости и т.д. Давайте подключим геркон к Arduino и посмотрим, как он работает.

Подключим модуль с герконом к Arduino по приведённой схеме. Питание осуществляется от 5 В или от 3,3 В. Сигнал подключим к цифровому пину D2.

Модуль с герконом содержит переменный резистор номиналом 10 кОм. Этим резистором можно установить порог срабатывания геркона и таким образом отрегулировать чувствительность. Также модуль содержит компаратор LM393 для исключения ложных срабатываний магнитного датчика. На рисунке 9 представлен сервопривод [17].

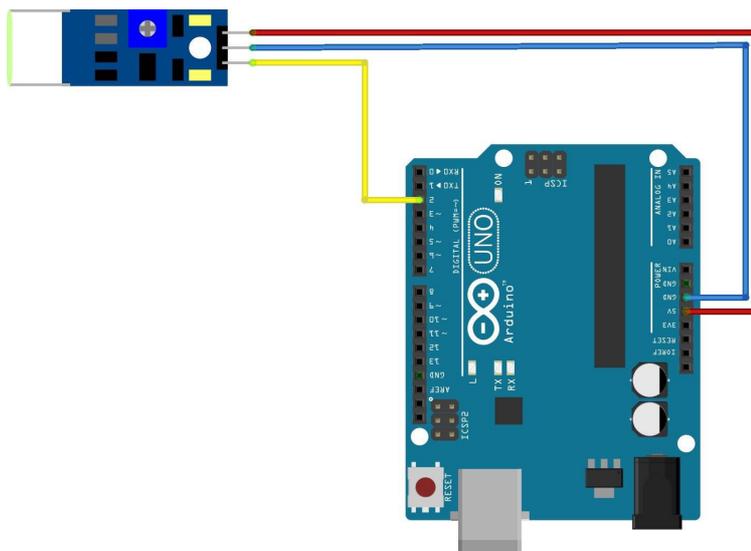


Рисунок 9 – Схема подключения геркона к Arduino

Для обнаружения протечек будем использовать датчик воды. На рисунке 6 представлен датчик уровня воды [18].

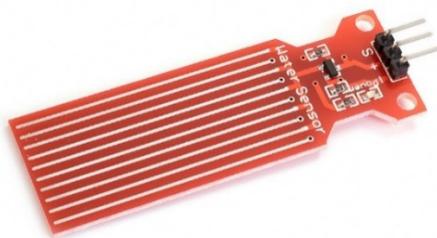


Рисунок 10 – Датчик уровня воды

Датчики воды предназначены для определения уровня воды в различных емкостях, где недоступен визуальный контроль, с целью предупреждения перенаполнения емкости водой через критическую отметку. Данный датчик воды – погружной. Чем больше погружение датчика в воду, тем меньше сопротивление между двумя соседними проводами (Рисунок 10).

Датчик имеет три контакта для подключения к контроллеру:

- питание датчика;
- земля;
- аналоговое значение.

На вывод S подается аналоговое значение, которое можно передавать в контроллер для дальнейшей обработки, анализа и принятия решений. Датчик имеет красный светодиод, сигнализирующих о наличие поступающего на датчик питания.

Аналоговый выход присоединяем к аналоговому пину микроконтроллера, например, A1. Цифровой выход, соответственно подключается к одному из цифровых пинов. Напряжение можно подать с вывода 5В платы ардуино, земля соединяется с землей. Подключение сервопривода к микроконтроллеру представлено на рисунке 11.

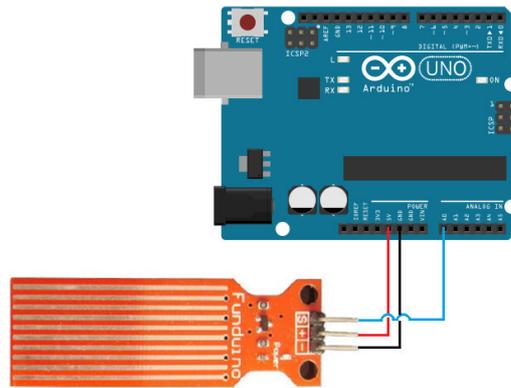


Рисунок 11 – Схема подключения датчика уровня воды к Arduino

Фоторезистор – это резистор, электрическое сопротивление которого изменяется под влиянием световых лучей, падающих на светочувствительную поверхность и не зависит от приложенного напряжения, как у обычного резистора. На рисунке 12 показан фоторезистор.

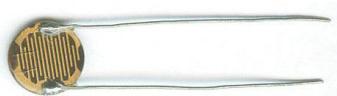


Рисунок 12 – Фоторезистор

Фоторезисторы чаще всего используются для определения наличия или отсутствия света, или для измерения интенсивности света. В темноте, их сопротивление очень высокое, иногда доходит до 1 МОм, но, когда датчик LDR подвергается воздействию света, его сопротивление резко падает, вплоть до нескольких десятков ом в зависимости от интенсивности света.

Фоторезисторы имеют чувствительность, которая изменяется с длиной волны света. Они используются во многих устройствах, хотя уступают по своей популярности фотодиодам и фототранзисторам. Некоторые страны запретили LDR из-за содержащегося в них свинца или кадмия по соображению экологической безопасности.

Самый простой вариант использования: подключить одну ногу к источнику питания, вторую – к земле через понижающий резистор. После

этого точка между резистором с постоянным номиналом и переменным резистором – фоторезистором – подключается к аналоговому входу микроконтроллера. На рисунке 13 показана схема подключения к Arduino [19].

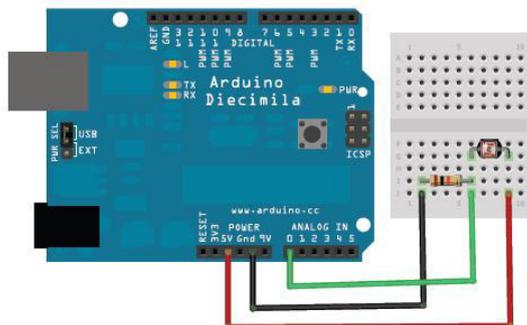


Рисунок 13 – Схема подключения фоторезистора к Arduino

Датчик влажности почвы Arduino предназначен для определения влажности земли, в которую он погружен. На рисунке 14 представлен датчик влажности почвы.

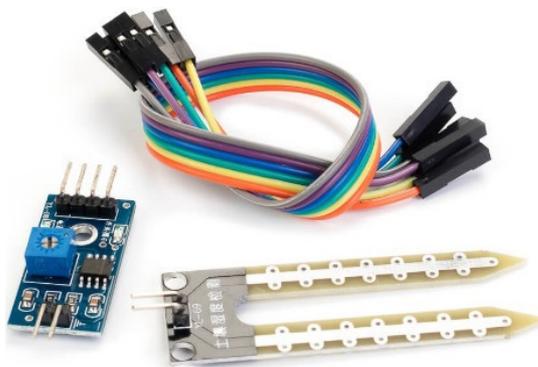


Рисунок 14 – Датчик влажности почвы

Он позволяет узнать о недостаточном или избыточном поливе ваших домашних или садовых растений. Подключение данного модуля к контроллеру позволяет автоматизировать процесс полива ваших растений, огорода или плантации. На рисунке 15 показана схема подключения к Arduino [20].

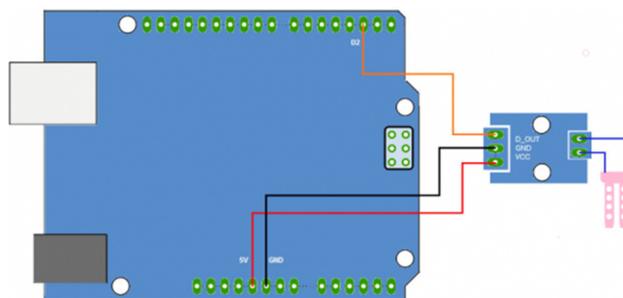


Рисунок 15 – Схема подключения датчика влажности почвы к Arduino

Модуль состоит из двух частей: контактного щупа YL-69 и датчика YL-38, в комплекте идут провода для подключения. Устройство его заключается в том, что между электродов создается напряжение. В случае недостатка влаги сопротивление будет выше, следовательно, ток будет меньше. Если земля влажная, то ток будет больше за счет понижения сопротивления. Отталкиваясь от этих значений создается аналоговый сигнал. Щуп YL-69 соединен с датчиком YL-38 по двум проводам.

Линия питания модуля +5V (VCC) соединена с питанием 5V платы Arduino. Земля GND соединена с землей Arduino. Цифровой выходной сигнал модуля DO подключается к линии D2 платы Arduino. Аналоговый выход модуля AO в данном случае не используется. Сам датчик состоит из двух штырей, на печатных платах которых имеются металлические полосы. При помещении датчика во влажную почву сопротивление между штырями мало, в сухой земле это сопротивление достаточно большое. С помощью потенциометра на модуле можно менять чувствительность датчика к тому или иному уровню сопротивления между штырями.

Большинство микросхем, таких как DS1307 используют внешний кварцевый генератор частотой 32кГц, но в них есть существенный недостаток, при изменении температуры меняется частота кварца, что приводит к погрешности в подсчете времени. Эта проблема устранена в чипе DS3231, внутрь которого установили кварцевый генератор и датчик температуры, который компенсирует изменения температуры, так что время остается точным. Так же чип DS3231 поддерживает секунды, минуты, часы,

день недели, дата, месяц и год информацию, а также следит за количеством дней в месяце и делает поправку на високосный год. Поддерживает работу часов в двух форматах 24 и 12, а также возможно запрограммировать два будильника. Модуль работает по двух проводной шине I2C. На рисунке 16 представлен модуль DS3231.

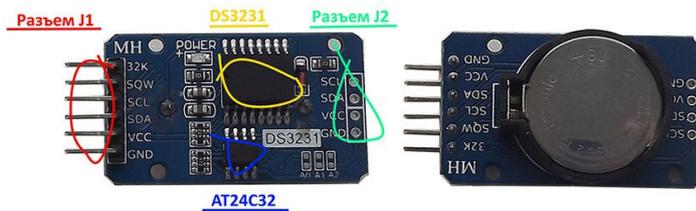


Рисунок 16 – часы реального времени

В данном примере буду использовать только модуль DS3231 и Arduino UNO R3, все данные будут передаваться в «Мониторинг порта». Необходимо всего четыре провода, сначала подключаем шину I2C, SCL в A4 (Arduino UNO) и SDA в A5 (Arduino UNO), осталось подключить питание GND к GND и VCC к 5V (можно записать и от 3.3В), схема собрана, теперь надо подготовить программную часть. На рисунке 17 показана схема подключения к Arduino [21].

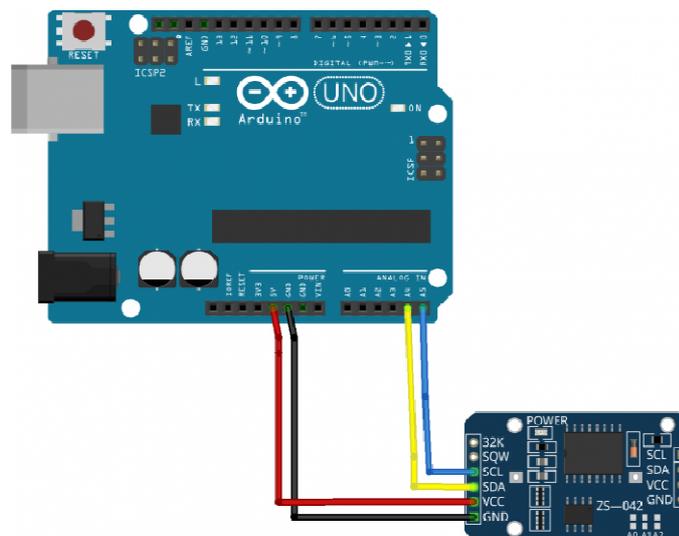


Рисунок 17 – Схема подключения часов реального времени к Arduino

Датчики DHT11 и DHT22 обладают довольно слабой точностью и скоростью работы, но их можно купить дешевле конкурентов, и они просты в использовании. Конструктивно датчик состоит из двух частей датчика влажности и емкостного термистора. Плата, на которой он установлен выполняет аналого-цифровые преобразования с выходным цифровым сигналом. На рисунке 18 представлен датчик температуры и влажности воздуха.

Сенсор DHT22:

- определение влажности в диапазоне 0–100%
- определение температуры от -40°C до $+125^{\circ}\text{C}$
- частота опроса 1 раз в 2 секунды

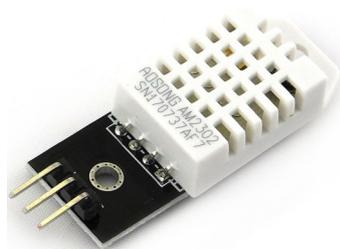


Рисунок 18 – Датчик влажности и температуры воздуха DHT22

Если у вас в руках готовый модуль датчика, то подключение его к Arduino предельно упрощается: подключаете VCC к +5В, GND – к земле, третий контакт – к любому свободному пину на плате Arduino. Номер пина нужно будет затем указать скетче. Таким образом можно подключить DHT11 к разным платам Arduino: Uno, Arduino Mega, Arduino Nano и другим [22].

Подключение модуля датчика производится по примеру показанном на рисунке 19:

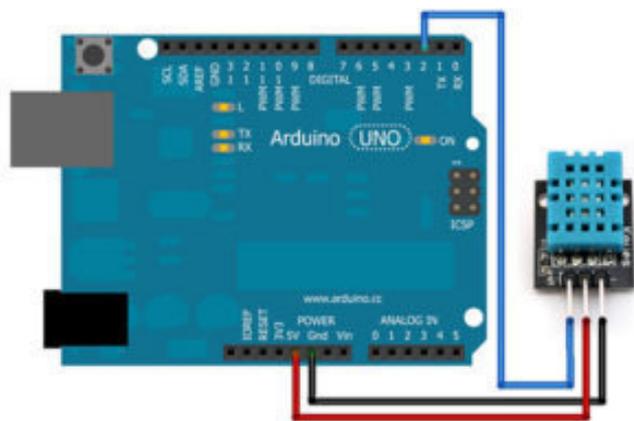


Рисунок 19 – Схема подключения DHT22 к Arduino

Модуль датчика качества воздуха MQ135 представлен на рисунке 20. Он предназначен для определения содержания и количества вредных и опасных газов в воздухе таких как: NH_3 , NO_x , пары алкоголя, бензина, дыма, CO_2 и т.д. Модуль построен на основе датчика MQ135. На плате модуля предусмотрен компаратор для определения порогового значения концентрации вредных веществ. Порог срабатывания задается потенциометром (Рисунок 20).



Рисунок 20 – Датчика качества воздуха MQ135

В датчике газа MQ-135 применяется диоксид олова (SnO_2), который имеет низкую проводимость на чистом воздухе. При появлении в воздухе паров бензина, углекислого газа, алкоголя или дыма и увеличении их концентрации, проводимость датчика увеличивается. А греется он для ускорения реакции. На рисунке 21 показана схема подключения к Arduino [23].

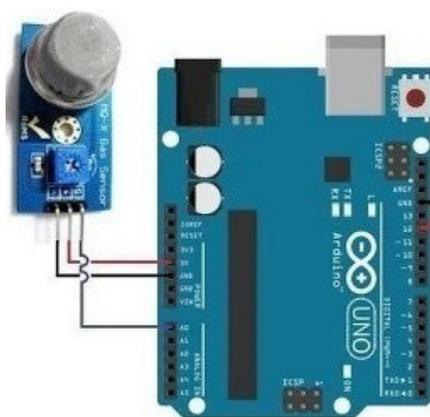


Рисунок 21 – Схема подключения MQ135 к Arduino

Плата расширения Arduino Ethernet Shield позволяет плате Arduino подключаться к интернету. Она построена на базе ethernet микросхемы Wiznet W510. Wiznet W5100 обеспечивает сетевой (IP) стек, способный работать и с TCP, и с UDP. Она поддерживает до четырех одновременных сокетных соединений. Для написания скетчей используйте библиотеку Ethernet, которая позволяет подключаться к интернету с помощью платы Ethernet Shield. Ethernet плата подключается к плате Arduino с помощью длинных разъемов, которые проходят сквозь плату расширения для дальнейшего использования. Это позволяет сохранить распиновку неизменной и использовать другие платы расширения, подключая их сверху (Рисунок 23).

Большинство последних версий платы предоставляет распиновку 1.0 версии 3 платы Arduino UNO.

Плата расширения Ethernet Shield обладает стандартным соединением RJ-45 с интегрированным линейным трансформатором и включенным питанием через Ethernet (PoE).

Встроенный слот карты памяти micro-SD может быть использован для хранения файлов, предназначенных для передачи по сети. Он совместим со всеми платами Arduino/Genuino. Встроенный micro-SD картридер доступен через SD Library. При работе с этой библиотекой SS находится на выводе 4.

Оригинальная версия платы расширения содержит полноразмерный слот SD картридера; он не поддерживается.

Плата расширения также включает в себя контроллер сброса, служащий для правильного сброса ethernet модуля W5100 при включении питания. Предыдущие версии платы расширения были не совместимы с Arduino Mega и нуждаются в ручном сбросе после подачи питания.

Текущая версия платы обладает модулем Power over Ethernet (PoE), предназначенным для получения питания через обычный ethernet кабель витой пары категории 5:

- совместим со стандартом IEEE802.3af;
- низкие пульсации и шум на выходе (100 мВ);
- диапазон входных напряжения: от 36В до 57В;
- защита от перегрузки и короткого замыкания;
- выходное напряжение: 9В;
- высокоэффективный преобразователь постоянного напряжения;
- изоляция 1500В (между входом и выходом).

Плата расширения поставляется без модуля PoE, это отдельно добавляемый компонент. Arduino Ethernet Shield представлен на рисунке 22 [24].

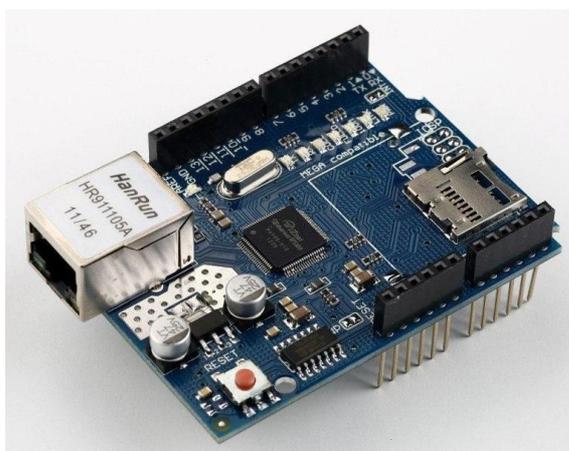


Рисунок 22 – Ethernet Shield

Arduino взаимодействует и с W5100, и SD картой с помощью шины SPI (через разъем ICSP). Он расположен на цифровых выводах 10, 11, 12 и 13 на платах Arduino Uno и на выводах 50, 51 и 52 на платах Arduino Mega. На обеих платах вывод 10 используется для выбора W5100, а вывод 4 – для SD карты. Эти выводы не могут использоваться в качестве обычных входов/выходов. На Arduino Mega аппаратный вывод SS (вывод 53) не используется для выбора W5100 или SD карты, но должен продолжать работать, как выход, иначе SPI интерфейс не будет работать. На рисунке 23 продемонстрировано как к Arduino подключается данный шилд.



Рисунок 23 –Подключенный Ethernet Shield к Arduino

Обратите внимание, из-за того, что W5100 и SD карта делят шину SPI, в какой-либо момент времени может быть активен только один из них. Если вы используете в своей программе оба периферийных устройства, ответственность за взаимодействие с ними берут соответствующие библиотеки. Но если вы не используете только одно из этих периферийных устройств, то должны явно отменить его выбор. Чтобы выполнить это для SD карты, настройте вывод 4 на выход и запишите в него высокий уровень. Для W5100 настройте цифровой вывод 10, как выход с высоким уровнем.

Вышеперечисленные датчики были подобраны для решение конкретной задачи и полностью для нее адаптированные.

3 Разработка программного обеспечения веб-ориентированной системы

3.1 Выбор и обоснование средств реализации

Arduino – это электронная платформа с открытым исходным кодом, основанная на простом в использовании оборудовании и программном обеспечении. Для этого вы используете Arduino язык программирования, и в Arduino программирования (IDE), основанная на обработке.

На протяжении многих лет Arduino был мозгом тысяч проектов, от повседневных объектов до сложных научных инструментов. Всемирное сообщество разработчиков – студентов, любителей, художников, программистов и специалистов – собрала вокруг этой открытой платформы, их вклад составляет невероятное количество знаний, которые могут быть большим подспорьем для новичков, так и экспертами.

Arduino родился в Ivrea Interaction Design Institute как простой инструмент для быстрого прототипирования, направленный на студентов без фона в электронике и программировании. Как только он достиг широкой общественности, платы Arduino начала меняться, чтобы адаптироваться к новым потребностям и нуждам. Во всех платах Arduino открытый исходный код для расширения прав и возможностей пользователей, чтобы спроектировать их самостоятельно и в конечном итоге адаптировать их к их конкретным потребностям. Программное обеспечение также является открытым исходным кодом, и растет благодаря вкладам пользователей во всем мире.

Благодаря простому и доступному пользовательскому опыту, Arduino используется в тысячах различных проектов и приложений. Программное обеспечение Arduino является простым в использовании для начинающих, но достаточно гибким для опытных пользователей. Он работает на Mac, Windows и Linux. Преподаватели и студенты используют его для создания недорогих научных инструментов, чтобы доказать принципы химии и

физики, или начать работу с программированием и робототехникой. Дизайнеры и архитекторы создают интерактивные прототипы, музыканты и художники используют его для инсталляции и экспериментирования с новыми музыкальными инструментами.

Существует множество микроконтроллеров и платформ для осуществления «physical computing». Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard и многие другие предлагают схожую функциональность. Все эти устройства объединяют разрозненную информацию о программировании и заключают ее в простую в использовании сборку. Arduino, в свою очередь, тоже упрощает процесс работы с микроконтроллерами, однако имеет ряд преимуществ перед другими устройствами для преподавателей, студентов и любителей:

Низкая стоимость – платы Arduino относительно дешевы по сравнению с другими платформами. Самая недорогая версия модуля Arduino может быть собрана вручную, а некоторые даже готовые модули стоят меньше 50 долларов.

Кроссплатформенность – программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows.

Простая и понятная среда программирования – среда Arduino подходит как для начинающих пользователей, так и для опытных. Arduino основана на среде программирования Processing, что очень удобно для преподавателей, так как студенты, работающие с данной средой, будут знакомы и с Arduino.

Программное обеспечение с возможностью расширения и открытым исходным текстом – ПО Arduino выпускается как инструмент, который может быть дополнен опытными пользователями. Язык может дополняться библиотеками C++. Пользователи, желающие понять технические нюансы, имеют возможность перейти на язык AVR C. Соответственно, имеется возможность добавить код из среды AVR-C в программу Arduino.

Аппаратные средства с возможностью расширения и открытыми принципиальными схемами – микроконтроллеры ATMEGA8 и ATMEGA168 являются основой Arduino. Схемы модулей выпускаются с лицензией Creative Commons, а значит, опытные инженеры имеют возможность создания собственных версий модулей, расширяя и дополняя их. Даже обычные пользователи могут разработать опытные образцы с целью экономии средств и понимания работы.

3.2 Разработка алгоритмов функционирования системы

Работа системы будет осуществляться таким образом, что микроконтроллер постоянно считывает показания датчиков и в зависимости от этого отдает команды исполнительным механизмам. На рисунке 21 представлена структурная схема системы.

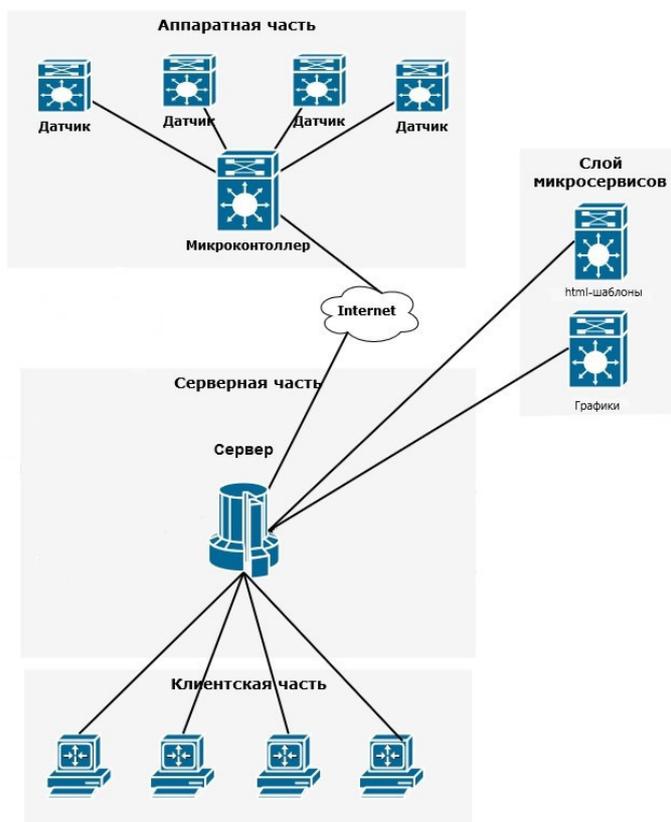


Рисунок 24 – Структурная схема устройства

Также микроконтроллер постоянно слушает запросы сервера. Если сервер присылает запрос, то контроллер обрабатывает его, выполняет и отправляет ответ в виде GET-запроса. На рисунке 25 представлена диаграмма системы в IDEF0 [26].

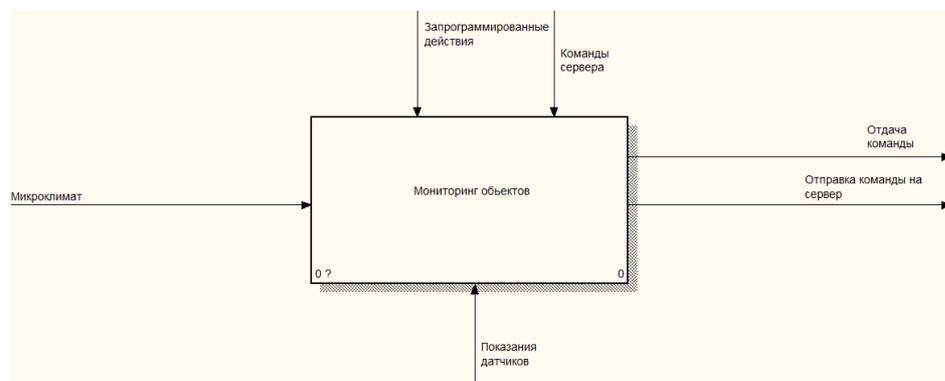


Рисунок 25 – Диаграмма системы в IDEF0

На систему действует внутренний климат теплицы. Микроконтроллер считывает показания датчиков и ждет команд от сервера, если он их отправит, после чего обрабатывает полученные данные, при необходимости выполняет необходимые действия и отправляет данные на сервер.

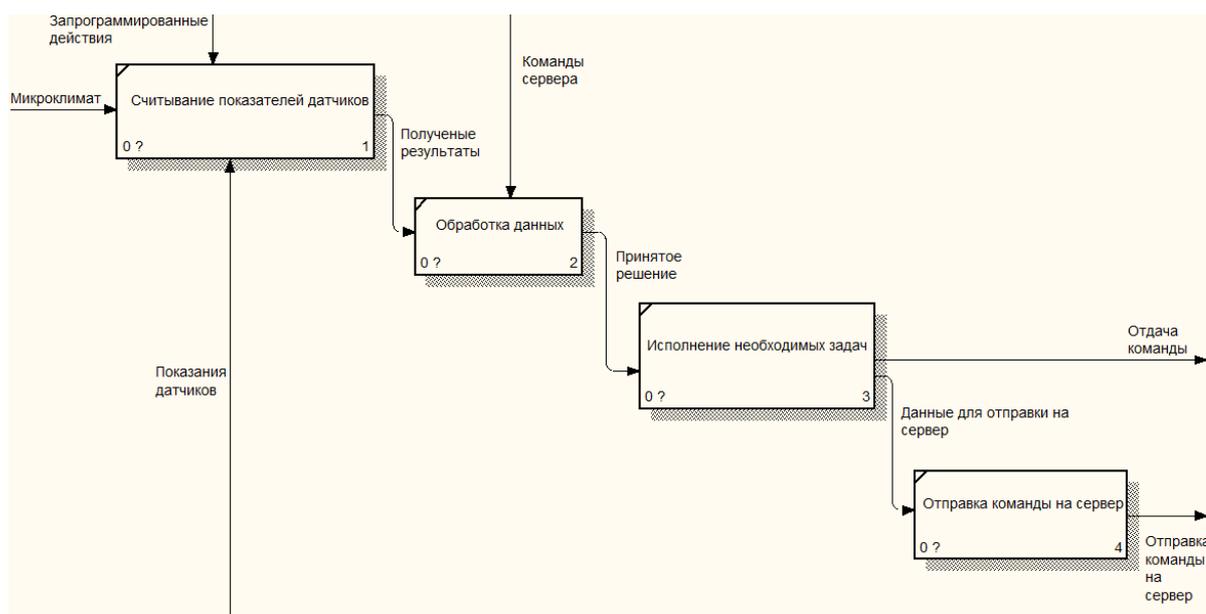


Рисунок 26 – Декомпозиция диаграмма в IDEF0

Для хранения всей необходимой нам информации необходимо использовать базу данных. Создание БД производилось в СУБД phpMyAdmin [25]. На рисунке 27 представлена созданная база данных.

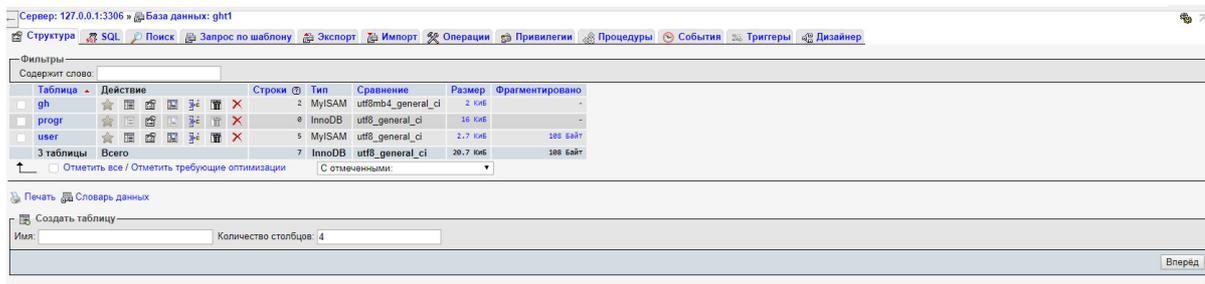


Рисунок 27 – Созданная БД

Первой была создана таблица «user». Каждый пользователь может иметь поля, хранящие ID теплиц и программ. На рисунке 28 представлена таблица «user».

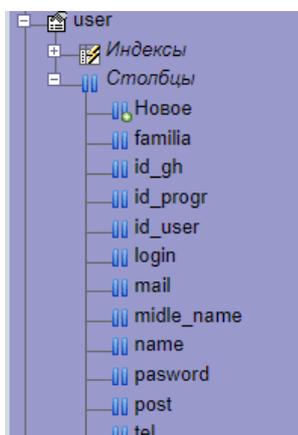


Рисунок 28 – Столбцы таблицы «user»

Далее была создана таблица «gh», в которой хранится вся информация с датчиков. На рисунке 29 представлена таблица «gh».

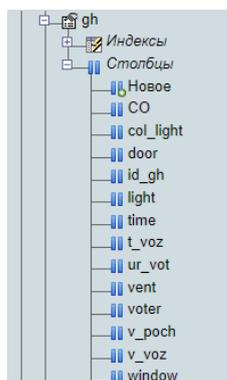


Рисунок 29 – Столбцы таблицы «gh»

На рисунке 30 представлена таблица «prog».

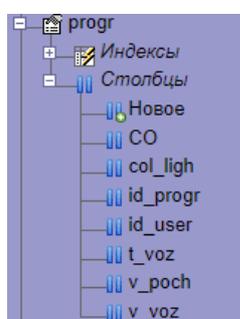


Рисунок 30 – Столбцы таблицы «prog»

Таблица «prog» хранит в себе значения датчиков, введенные пользователем. Если он хочет, чтобы для определенной культуры были необходимые показатели микроклимата.

3.3 Программная реализация разработанных алгоритмов

Библиотека SPI позволяет контроллеру Arduino взаимодействовать с устройствами, поддерживающими SPI протокол. Arduino в данном случае выступает в качестве ведущего устройства.

Serial Peripheral Interface (SPI) – это синхронный протокол последовательных данных, используемый микроконтроллерами для связи с одним или несколькими периферийными устройствами быстро на коротких

расстояниях. Его можно также использовать для сообщения между 2 микроконтроллерами.

С соединением SPI всегда один мастерский прибор (обычно микроконтроллер) который контролирует периферийные приборы. Обычно существует три линии, общие для всех устройств

- MISO (рабочие линии для передачи данных);
- MOSI (мастер-линию для передачи данных на периферийные устройства);
- SCK (часы, которые синхронизируют передачи данных, генерируемых мастером);
- SS (пин-код на каждом устройстве, которое можно использовать для включения и отключения определенных устройств).

Стандарт SPI свободен, и каждое устройство реализует его немного по-разному. Это означает, что вы должны обратить особое внимание на техническое описание устройства при написании кода.

Servo.h

Эта библиотека позволяет плате Arduino управлять сервомоторами RC (hobby). Сервоприводы имеют интегрированные шестерни и вал, который можно точно контролировать. Стандартные сервоприводы позволяют устанавливать вал под различными углами, обычно от 0 до 180 градусов. Непрерывное вращение сервоприводов позволяет вращать вал на различных скоростях.

Сервотека поддерживает до 12 двигателей на большинстве плат Arduino и 48 на Arduino Mega. На платах, отличных от Mega, использование библиотеки отключает функциональность analogWrite() (PWM) на выводах 9 и 10, независимо от того, есть ли на этих выводах Сервопривод или нет. На Mega, до 12 сервоприводов можно использовать без мешать с функциональностью PWM.

Моторы сервопривода имеют 3 провода: сила, земля, и сигнал. Провод питания, как правило, красный, и должны быть подключены к 5V pin на

плате Arduino. Провод заземления обычно черный или коричневый и должен быть подключен к заземляющему штифту на плате Arduino. Сигнальный штифт обычно желтый, оранжевый или белый и должен быть подключен к цифровому штифту на плате Arduino. Обратите внимание, что сервоприводы привлекают значительную мощность, поэтому, если вам нужно управлять более чем одним или двумя, вам, вероятно, нужно будет питать их от отдельной поставки (т. е. не +5V pin на Вашем Arduino). Обязательно соедините основания Arduino и внешнего источника питания вместе.

DHT.h

Библиотека позволяет получать данные о температуре и влажности от датчиков DHT-11 или DHT-22, по одной линии данных. Библиотека сама определяет, какой тип датчика используется. Если к одной плате Arduino подключено несколько датчиков, то для каждого из них, нужно создать свой объект, с указанием номера вывода, к которому подключен датчик.

Библиотека позволяет получать данные от датчиков на базе чипа AM2320, но только если они подключены по одному проводу, а не по шине I2C [27].

Ethernet.h

Эти библиотеки предназначены для работы с Arduino Ethernet Shield или Arduino Ethernet Shield 2 и Leonardo Ethernet. Библиотеки позволяют плате Arduino подключаться к интернету. Плата может служить либо сервером, принимающим входящие соединения, либо клиентом, производящим исходящие. Библиотеки поддерживают до четырех одновременных подключений (входящих или исходящих или комбинации). Библиотека Ethernet управляет обломком W5100, пока архив Ethernet2 (Ethernet2.h) управляет обломком W5500; все функции отстают этими же. Изменение используемой библиотеки позволяет перенести тот же код от Arduino Ethernet Shield на Arduino Ethernet 2 Shield или Arduino Leonardo Ethernet и наоборот.

Arduino связывает с экраном используя шину SPI. Это на цифровых выводах 11, 12 и 13 на ООН и выводах 50, 51 и 52 на Мега. На обеих досках, контакт 10 используется в качестве СС. На Мега оборудовании пин СС, 53, не используется для выбора w5100, но он должен быть как выход или интерфейс SPI не работает.

`begin()`

Инициализирует библиотеку UIPEthernet, задает основные настройки. Имеет форматы:

– `mac`: MAC-адрес модуля (массив из 6 байт). Может быть указан на наклейке модуля или задан программно;

– `ip`: IP-адрес сетевого устройства (массив из 4 байт);

– `dns`: IP-адрес DNS-сервера (массив из 4 байт). Параметр необязательный. При отсутствии аргумента, в качестве адреса DNS-сервера будет установлен IP-адрес модуля, в котором последний октет равен 1;

– `gateway`: IP-адрес сетевого шлюза (массив из 4 байт). Параметр необязательный. При отсутствии аргумента, в качестве адреса шлюза будет установлен IP-адрес модуля, в котором последний октет равен 1;

– `subnet`: маска подсети (массив из 4 байт). Параметр необязательный.

По умолчанию задается 255.255.255.0.

Если функция используется в формате с единственным аргументом `Ethernet.begin(mac)`, назначение IP-адреса модуля происходит автоматически, по протоколу DHCP. В этом случае, при неудачном DHCP-соединении функция возвращает 0, при успешном – 1. Остальные форматы функции `begin()` не возвращают никаких значений.

`localIP()`

`IPAddress localIP()`

Функция возвращает IP-адрес Ethernet-модуля. Используется для того, чтобы узнать IP-адрес модуля, назначенный автоматически по DHCP-протоколу.

`IPAddress ip = Ethernet.localIP(); //чтение IP-адреса`

subnetMask()

IPAddress subnetMask()

Функция возвращает маску подсети.

gatewayIP()

IPAddress gatewayIP()

Функция возвращает IP-адрес шлюза.

dnsServerIP()

IPAddress dnsServerIP()

Функция возвращает IP-адрес DNS-сервера.

maintain()

int maintain()

Иницирует обновление IP-адреса по протоколу DHCP. В результате может быть получен, как новый, так и прежний адрес. Как решит DHCP-сервер.

Возвращает значение типа byte:

- 0 (ничего не произошло);
- 1 (обновить аренду адреса не удалось);
- 2 (обновление аренды адреса прошло успешно);
- 3 (IP-адрес обновить не удалось);
- 4 (обновление IP-адреса прошло успешно).

Класс IP-Address – используется для работы с IP адресами.

Класс Server – применяется для создания серверов и работы с ними.

Server.

Базовый класс, непосредственно не вызывается. Используется при вызове его функций.

EthernetServer()

Создает сервер, ожидающий соединения через заданный порт.

EthernetServer(port)

port – номер порта.

EthernetServer server(1000); //создаем сервер, порт 1000

`begin()`

`void begin()`

Инициализирует сервер начать ожидание входящих соединений.

`server.begin()`

Аргументов и возвращаемых значений нет.

При наличии клиента, подключенного к серверу и имеющего непрочитанные данные, возвращает описывающий клиента объект `Client`.

`server.available()`

Если отсутствуют клиенты с непрочитанными данными, то функция возвращает `false`.

Отправляет данные всем клиентам, подключенным к серверу. Можно отправить один байт или последовательность байтов.

`server.write(val)`

`server.write(buf, len)`

– `val` (байт, который необходимо передать);

– `buf` (массив данных, который необходимо отправить);

– `len` (длина массива).

Возвращает количество отправленных байтов.

`server.write(0x23); //отправляем байт`

`print()`

Отправляет данные всем клиентам, подключенным к серверу. Данные передаются в текстовом виде (ASCII символы). Формат функции аналогичен одноименной функции класса `Serial`.

`server.print(data, BASE);`

– `data` (данные, которые необходимо передать (`char`, `byte`, `int`, `long` или `string`));

– `BASE` (система счисления, в которой необходимо выводить числа).

Функция возвращает количество отправленных байтов.

`server.print("Error")`

Аналогично функции `print()` отправляет данные всем подключенным клиентам в текстовом виде. В отличие от `print()` добавляет символы переноса строки (`\r`, код 13) и (`\n`, код 10).

Класс `Client` – используется для создания клиентов и работы с ними.

`Client`

Базовый класс, непосредственно не вызывается. Используется при вызове его функций.

`EthernetClient()`

`void EthernetClient()`

Функция создает клиента. Параметров и возвращаемых значений нет.

`EthernetClient client; //создаем клиента`

Показывает готов ли указанный клиент к работе.

`if(client)`

Если указанный клиент готов, возвращает `true`.

Функция показывает, подключен ли клиент к серверу. При наличии непрочитанных данных клиент считается подключенным к серверу, даже если соединение уже закрыто.

`client.connected()`

Если клиент подключен, функция возвращает `true`.

Функция устанавливает соединение с сервером через заданный порт. В качестве аргумента может использоваться как IP-адрес, так и доменное имя.

`client.connect(ip, port)`

`client.connect(URL, port)`

– `ip` (IP-адрес сервера, с которым необходимо установить соединение (массив из 4 байт));

– `URL` (доменное имя сервера, с которым необходимо установить соединение (строка, например "google.com"));

– `port` (порт, через который необходимо установить соединение).

При успешном подключении функция возвращает `true`.

Отправляет данные серверу, с которым было установлено соединение.

Можно отправлять один байт или последовательность байтов.

```
client.write(val);
```

```
client.write(buf,len)
```

Функция возвращает количество отправленных байтов.

Отправляет данные от клиента серверу. Данные передаются в текстовом виде (ASCII символы). Формат функции аналогичен одноименной функции класса Serial.

```
println()
```

Аналогично функции print() отправляет данные серверу в текстовом виде. В отличие от print() добавляет символы переноса строки (\r, код 13) и (\n, код 10).

```
client.print(data, BASE)
```

```
available()
```

```
int available()
```

Возвращает количество непрочитанных байтов, принятых клиентом от сервера.

```
client.available()
```

Возвращает количество непрочитанных байтов.

Считывает очередной байт из потока данных, принятых клиентом от сервера.

```
client.read()
```

```
client.read(buf,len)
```

Возвращает значение очередного байта, либо -1, если буфер пуст.

```
char chr = client.read()
```

```
peek()
```

```
int peek()
```

Возвращает очередной принятый от сервера байт, не удаляя его из буфера.

```
client.peek()
```

flush()

void flush()

Удаляет из буфера все непрочитанные данные, полученные от сервера.

client.flush()

Параметров и возвращаемого значения нет.

stop()

void stop()

Разрывает соединение с сервером.

client.stop()

Параметров и возвращаемого значения нет.

Класс EthernetUDP – поддерживает UDP протокол.

begin()

uint8_t begin(uint16_t);

Инициализирует UDP библиотеку, запускает прослушивание на указанном порту.

EthernetUDP.begin(port)

Возвращает 1 при успешной операции.

read()

Считывает данные в заданный буфер. Если аргументы функции не заданы, то возвращает очередной принятый символ.

Функцию можно использовать только после вызова parsePacket().

UDP.read()

UDP.read(buf, size)

Возвращает значение очередного байта, либо -1, если буфер пуст. В случае приема пакета, возвращает количество байтов, записанных в массив.

peek()

int peek()

Возвращает очередной принятый от сервера байт, не удаляя его из буфера.

UDP.peek()

write()

size_t write(uint8_t)

Посылает данные по протоколу UDP. Метод должен вызываться между вызовами функций beginPacket() и endPacket(). beginPacket() инициализирует пакет данных, но не отправляет его до вызова endPacket().

UDP.write(val)

UDP.write(buf, len)

Функция возвращает количество отправленных байтов.

beginPacket()

int beginPacket(IPAddress ip, uint16_t port)

int beginPacket(const char *host, uint16_t port)

Устанавливает соединение с устройством для отправки данных по протоколу UDP.

UDP.beginPacket(remoteIP, remotePort);

– remoteIP (IP-адрес удаленного устройства (4 байта));

– remotePort (номер порта на удаленном устройстве).

Возвращает 1, если операция прошла успешно, 0 – если возникли неразрешимые проблемы с адресом или портом.

endPacket()

int endPacket(void)

Данный метод вызывается после подготовки UDP-данных для отправки.

UDP.endPacket()

Возвращает 1, если пакет был послан успешно, 0 – если произошла ошибка.

parsePacket()

int parsePacket(void)

Функция проверяет наличие принятого UDP-пакета и возвращает его размер. Должна вызываться перед считыванием буфера методом UDP.read().

UDP.parsePacket()

Функция возвращает размер принятого UDP-пакета.

available()

int available(void)

Функция возвращает количество принятых байтов, доступных для считывания из буфера. Ее можно вызывать только после вызова UDP.parsePacket().

UDP.available()

Возвращает количество байтов, доступных для считывания.

stop()

void stop(void)

Разрывает соединение с сервером.

EtheretUDP.stop()

Параметров не имеет, значение не возвращает.

remoteIP()

IPAddress remoteIP(void)

Функция возвращает IP-адрес удаленного устройства. Можно вызывать только после вызова UDP.parsePacket().

UDP.remoteIP()

Возвращает IP-адрес удаленного устройства (4 байта).

remotePort()

uint16_t remotePort(void)

Функция возвращает номер порта удаленного устройства. Можно вызывать только после вызова UDP.parsePacket().

UDP.remotePort()

Функция возвращает номер порта UDP-соединения на удаленном устройстве

flush()

Удаляет из буфера все непрочитанные данные.

UDP.flush()

Параметров и возвращаемого значения нет [28].

3.4 Тестирование

Так как все вычисления происходят на стороне сервера, а не клиента, то второму необходимы минимальные системные требования, лишь для работы браузера. А именно:

- Windows 7, Windows 8, Windows 8.1, Windows 10 или более поздней версии;
- процессор Intel Pentium 4 или более поздней версии с поддержкой SSE2;
- свободное место на диске: 350Мб;
- оперативная память: 512Мб.

На рисунке 31 представлена страница авторизации.

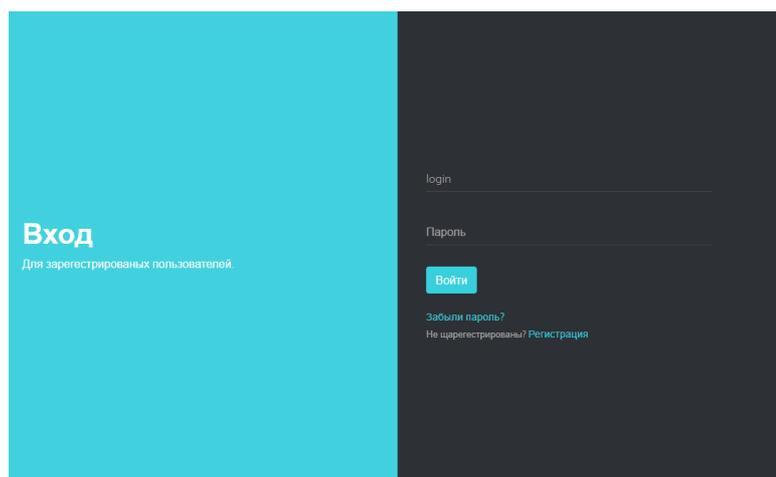


Рисунок 31 – Страница авторизации

В результате для пользования системой необходимо просто использовать функции сайта. При входе на сайт пользователя встречает страница входа в которой он вводит свой логин и пароль.

После того как пользователь проходит авторизацию он попадает на главную страницу сайта. Слева находится меню, в котором находятся страницы сайта. Над меню находится имя и должность под которым зашел пользователь. На рисунке 32 представлена главная страница.

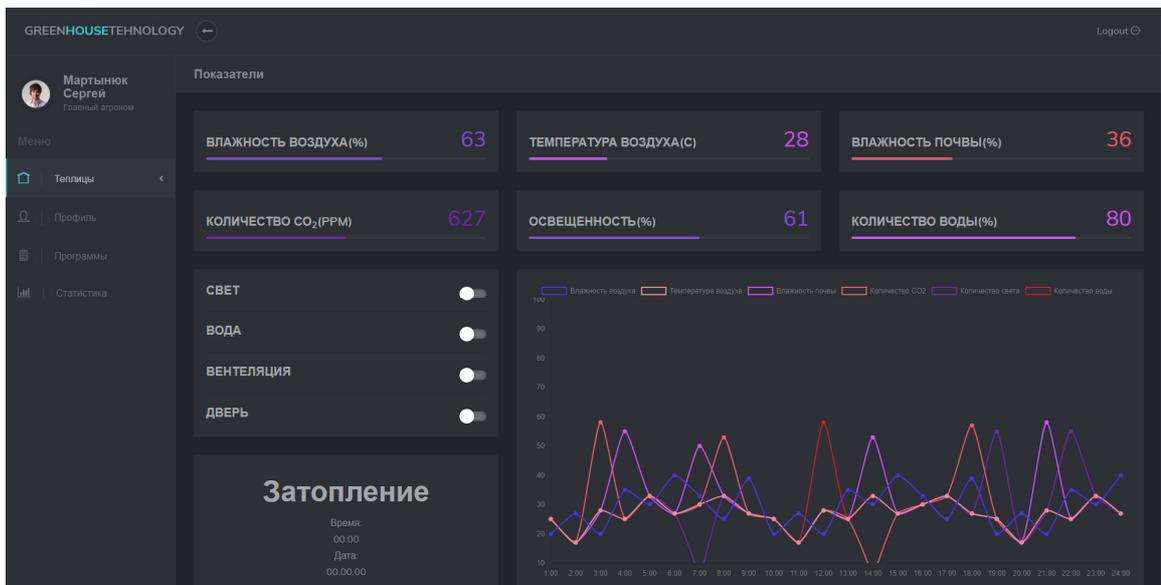


Рисунок 32– Главная страница

В главном окне страницы находятся показания датчиков, а также график изменения показателей за сутки. Отдельно выведено окно с информацией в случае затопления, а именно дата и время. На рисунке 33 представлена страница статистики.

Время	Влажность воздуха(%)	Температура воздуха(С)	Влажность почвы(%)	Количество CO ₂	Освещенность(%)	Количество воды(%)
01-05-18 00:00	64	25	36	627	22	80
01-05-18 01:00	63	25	36	633	21	80
01-05-18 02:00	63	25	36	67	22	80
01-05-18 03:00	65	26	36	636	22	80
01-05-18 04:00	63	25	36	641	31	80
01-05-18 05:00	64	26	36	636	46	80
01-05-18 06:00	63	26	36	635	53	80
01-05-18 07:00	65	27	36	632	72	80
01-05-18 08:00	62	27	36	628	75	80
01-05-18 09:00	64	27	36	625	76	80
01-05-18 10:00	65	27	36	622	74	80
01-05-18 11:00	65	28	36	617	77	80
01-05-18 12:00	64	27	36	620	76	80
01-05-18 13:00	63	28	36	623	61	80
01-05-18 14:00	63	28	36	622	78	80
01-05-18 15:00	63	28	36	622	73	80
01-05-18 16:00	65	29	36	624	72	80
01-05-18 17:00	63	28	36	626	73	80
01-05-18 18:00	64	27	36	637	72	80

Рисунок 33 – Страница статистики

На странице статистики хранятся собранные данные в виде таблицы.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе на тему «разработка веб-ориентированной системы мониторинга объектов», были выполнены следующие задачи:

- изучены преимущества систем автоматического регулирования;
- выбраны схемы регулирования для теплицы;
- выбрано оборудование для автоматизации и контроля процессов теплопотребления и определено место его установки;
- для получения максимального эффекта от тепличного оборудования были спроектированы алгоритмы регулирования;
- разработана аппаратная часть проекта;
- создан сайт для пользования системой;
- разработана методическая составляющая проекта.

Значение умной теплицы в нашей жизни, так же, как и её общественное значение трудно переоценить – это одна из альтернатив выращивания круглогодично сельскохозяйственных культур всей современной жизни. Потребление чисто выращенных овощей и фруктов, без применения химикатов, человеком с каждым годом растет, в связи с этим разрабатываются и применяются технологии, которые заключаются в нахождении новых технологий для их круглогодичного выращивания, и безотходной переработкой. Система автоматического регулирования процессов представляет собой систему из датчиков на основе платформы Arduino. Одним из преимуществ данной системы является ее способность управления теплицей через интернет. Даже рядовому пользователю не составит труда использовать эту систему так как интерфейс интуитивно понятен и прост.

Что касается экономической составляющей проекта то с уверенностью можно сказать то, что данная система является одной из самых дешевых подобных систем, а это еще больше привлекает потенциального покупателя.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Курдюмов, Н.А., Малышевский, К.С. Умная теплица: Учебник для бакалавров / В.А. Кудинов, Э.М. Карташов, Е.В. Стефанюк.. – Москва: Юрайт, 2012. – 473 с.
2. Кудинов, В.А. Техническая термодинамика и теплопередача: Учебник для бакалавров / В.А. Кудинов, Э.М. Карташов, Е.В. Стефанюк.. – Москва: Юрайт, 2013. – 566 с.
3. Мирам, А.О. Техническая термодинамика. Тепломассообмен: учебное издание / А.О. Мирам. – Москва: АСВ, 2011. – 352 с.
4. ФГОС СПО 13.03.01 Теплоэнергетика и теплотехника от 2015-0628; № 21200 – Москва: Минобрнауки России, 2015. – №1081. – 165 с.
5. Назарова В. И. Современные системы отопления: учебное пособие / В. И. Назарова. – Москва: РИПОЛ классик, 2011. – 320 с.
6. Бочарова, В. М. Bootloader Arduino [Электронный ресурс] / М.Ю. Лермантов. – 2014. – Дата доступа: 17.04.2018. – Режим доступа: <http://microsin.net/programming/avr/arduino-bootloader.html>
7. Берестова, Т.Ф. Особенности Arduino [Электронный ресурс] / Т.Ф. Берестова. – 2016. – Дата доступа: 17.04.2018. – Режим доступа: <http://developer.alexanderklimov.ru/arduino/spec.php>
8. Борозда, И.В. Почему Arduino [Электронный ресурс] / И.В. Борозда. – 2015. – Дата доступа: 17.04.2018. – Режим доступа: <http://arduino.ru/About>
9. Лермантов, М.Ю. Особенности Arduino [Электронный ресурс] / М.Ю. Лермантов. – 2016. – Дата доступа: 18.04.2018. – Режим доступа: <https://electrosam.ru/glavnaja/slabotochnye-seti/oborudovanie/arduino-uno/>
10. Гайденко, Т.А. Характеристики Arduino [Электронный ресурс] / Т.А. Гайденко. – 2018. – Дата доступа: 18.04.2018. – Режим доступа: <http://kip-world.ru/dostoinstva-i-nedostatki-arduino-uno-r3.html>

11. Соколов, А.Н. Аппаратная платформа Arduino [Электронный ресурс] / А.Н. Соколов. – 2010. – Дата доступа: 20.04.2018. – Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardUno>
12. Сычев, А.С. Реле [Электронный ресурс] / А.С. Сычев. – 2013. – Дата доступа: 20.04.2018. – Режим доступа: <https://www.drive2.ru/l/8865608/>
13. Галисеев, Г.В. Подключение реле к Arduino [Электронный ресурс] / Г.В. Галисеев. – 2016. – Дата доступа: 20.04.2018. – Режим доступа: <http://robotclass.ru/tutorials/arduino-relay/>
14. Саймон, Р. Сервоприводы [Электронный ресурс] / Р. Саймон. – 2015. – Дата доступа: 23.04.2018. – Режим доступа: <http://wiki.amperka.ru/робототехника:сервоприводы>
15. Дубейковский, В.И. Как подключить сервопривод к Arduino [Электронный ресурс] / В.И. Дубейковский. – 2012. – Дата доступа: 23.04.2018. – Режим доступа: <http://роботехника18.рф/как-подключить-сервопривод-к-ардуино/>
16. Губина, Е. А. Что такое герконы [Электронный ресурс] / Е. А. Губина. – 2015. – Дата доступа: 24.04.2018. – Режим доступа: <https://samelectrik.ru/chto-takoe-gerkony.html>
17. Коваленко, В.В. Подключение геркона к Arduino [Электронный ресурс] / В.В. Коваленко. – 2014. – Дата доступа: 24.04.2018. – Режим доступа: <https://soltau.ru/index.php/arduino/item/358-how-connect-reed>
18. Грекул, В. И. Подключение датчика воды к Arduino [Электронный ресурс] / В. И. Грекул. – 2017. – Дата доступа: 24.04.2018. – Режим доступа: <http://роботехника18.рф/подключение-датчика-воды-к-ардуино/>
19. Моисеенко, С. Подключение фоторезистора к Arduino [Электронный ресурс] / С. Моисеенко. – 2014. – Дата доступа: 24.04.2018. – Режим доступа: <http://arduino-diy.com/arduino-fotorezistor>
20. Маклаков, С.В. Подключение датчика влажности почвы к Arduino [Электронный ресурс] / С.В. Маклаков. – 2013. – Дата доступа: 25.04.2018. –

Режим доступа: http://digitrode.ru/computing-devices/mcu_cpu/289-arduino-i-datchik-vlazhnosti-pochvy.html

21. Джестон, Дж. Подключение часов реального времени к Arduino [Электронный ресурс] / Дж. Джестон. – 2015. – Дата доступа: 25.04.2018. – Режим доступа: <http://blog.rchip.ru/obzor-chasov-realnogo-vremeni-ds1307/>

22. Рябова, Ю.С. DHT22 подключение к Arduino [Электронный ресурс] / Ю.С. Рябова. – 2011. – Дата доступа: 28.04.2018. – Режим доступа: <https://geekelectronics.org/arduino/dht22-podklyuchenie-k-arduino.html>

23. Маклаков, С.В. Датчик широкого спектра газов MQ135 [Электронный ресурс] / С.В. Маклаков. – 2014. – Дата доступа: 28.04.2018. – Режим доступа: <http://wiki.amperka.ru/продукты:mq135>

24. Кузнецов, С.Д. Подключение Ethernet Shield к Arduino [Электронный ресурс] / С.Д. Кузнецов. – 2014. – Дата доступа: 28.04.2018. – Режим доступа: http://helpduino.ru/podklyuchenie_Ethernet_Shield_k_arduino.html

25. Голицына, О.Л. Администрирование MySQL [Электронный ресурс] / О.Л. Голицына. – 2016. – Дата доступа: 29.04.2018. – Режим доступа: <http://fb.ru/article/362994/administrirovanie-mysql>

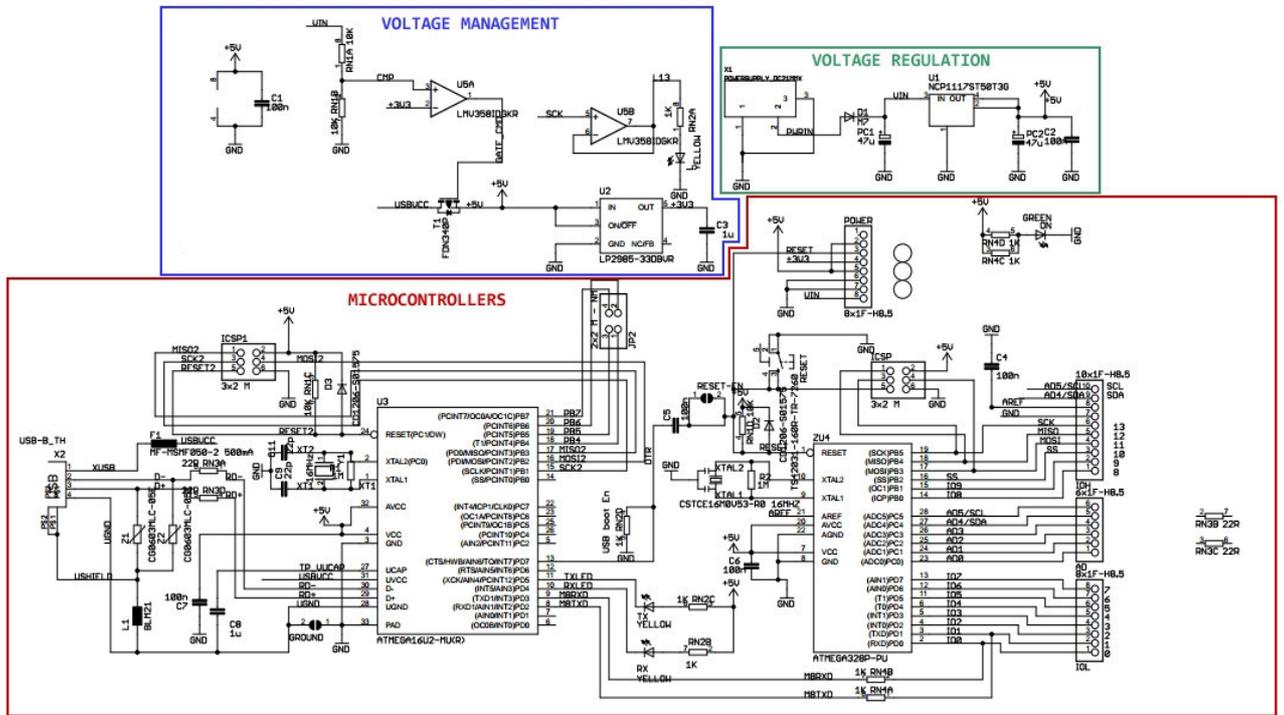
26. Репин, С.В. GET переменные [Электронный ресурс] / С.В. Репин. – 2011. – Дата доступа: 29.04.2018. – Режим доступа: <https://php.ru/manual/reserved.variables.get.html>

27. Белоусов, Н.А. Программирование микроконтроллерных плат Arduino [Электронный ресурс] / Н.А. Белоусов. – 2012. – Дата доступа: 29.04.2018. – Режим доступа: <http://mexalib.com/read/16868>

28. Гусев, А.В. Создание web-сервера для Arduino на базе Ethernet shield W5100 [Электронный ресурс] / А.В. Гусев. – 2010. – Дата доступа: 29.04.2018. – Режим доступа: http://helpduino.ru/podklyuchenie_Ethernet_Shield_k_arduino.html

ПРИЛОЖЕНИЕ А

Схема элементов Arduino UNO R3



Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«___» _____ Г.
