

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
имени Гумарбека Даукеева
Кафедра «Электроника и робототехника»

«Допущен к защите»
Зав. кафедрой «Электроника и робототехника»

Т.О. Чигамбаев к.т.н., доц.
«___» _____ 2020г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Автоматизация управления теплицей на базе Raspberry Pi с альтернативными источниками питания»

Специальность «5В071600 – Приборостроение»

Выполнил _____ ст. гр. ПС-16-3 Овденко В.

Научный руководитель _____ Е.О. Елеукулов доц.

Консультанты:

по экономической части:

Боканова Г.Ш. старший преподаватель

_____ «___» _____ 20__ г.

по безопасности жизнедеятельности:

Приходько Н.Г., профессор

_____ «___» _____ 20__ г.

Нормоконтролер: Фазылова А.Р. ст.преп.

_____ «___» _____ 20__ г.

Рецензент: Д.Р.Пя к.т.н, проф.

_____ «___» _____ 20__ г.

(подпись)

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт «Космическая технология и телекоммуникация»
Специальность «5В071600 –Приборостроение»
Кафедра «Электроника и робототехника»

ЗАДАНИЕ
на выполнение дипломного проекта

Студента В.Н. Овденко

Тема проекта «Автоматизация управления теплицей на базе Raspberry Pi с альтернативными источниками питания» утверждена приказом ректора № ____ от ____ сентября 2020 г.

Срок сдачи законченной работы « 15» мая 2020г.

Исходные данные к проекту (требуемые параметры результатов проектирования) и исходные данные:

- 1) разновидности функционала «умных» теплиц;
- 2) перспектива развития тепличной отрасли в Казахстане;

Перечень подлежащих разработке в дипломном проекте вопросов или краткое содержание дипломного проекта:

1. Поиск необходимой литературы
2. Выбор аппаратной базы для разработки теплицы
3. Создание микроклимата в теплице
4. Создание веб-сайта
5. Описание опытного образца
6. Техничко–экономическая часть (расчет стоимости программного обеспечения, затрат на установку системы, экономического эффекта и срока окупаемости).
7. Общие вопросы безопасности и жизнедеятельности.

Перечень графического материала (с точным указанием обязательных чертежей): в данной работе содержится 64 рисунка и 15 таблиц.

Рекомендуемая основная литература:

- 1) Германович В., Турилин А. Альтернативные источники энергии и энергосбережение— СПб.: Наука и Техника, 2014
- 2) Лифшиц Е. М., Питаевский Л. П., Статистическая физика, ч. 2, М., 1978;
- 3) Князевский Б.А. Охрана труда. – М.: Высшая школа, 2002. – 365 с.
- 4) Базылов К.Б., Алибаева С.А., Бабич А.А. Выпускная работа бакалавров. Экономический раздел. – Алматы: АИЭС, 2008. - 20 с.

Консультанты по проекту с указанием относящихся к ним разделов работы

Раздел	Консультант	Сроки	Подпись
Поиск необходимой литературы	Елеукулов Е.О.	03.03.20	
Выбор аппаратной базы для разработки теплицы	Елеукулов Е.О.	05.04.20	
Создание микроклимата в теплице	Елеукулов Е.О.	19.04.20	
Создание веб-сайта	Елеукулов Е.О.	02.05.20	
Экономическая часть	Боканова Г.Ш.	17.03.20	
Безопасность жизнедеятельности	Приходько Н.Г.	23.04.20	

Г РАФИК
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	2	3	4
1	Поиск необходимой литературы	03.03.20	
2	Выбор аппаратной базы для разработки теплицы	05.04.20	
3	Создание микроклимата в теплице	19.04.20	
4	Создание веб-сайта	22.04.20	
5	Экономическая часть	02.05.20	
6	Безопасность жизнедеятельности	17.03.20	

Дата выдачи задания «__» _____ 2020 г.

Заведующий кафедрой _____ Т.О. Чигамбаев

Руководитель _____ Е.О. Елеукулов

Задание принял к исполнению
студент _____ Овденко В.Н.
(подпись)

Содержание

Введение	7
1 Выбор аппаратной базы для создания «умной теплицы»	8
1.1 «Умная теплица»	8
1.2 Обзор технологий для управления автоматизацией	9
1.2.1 Микроконтроллеры	10
1.2.2 Ардуино	11
1.2.3 Одноплатные компьютеры на базе Raspberry Pi	15
2 Создание микроклимата в теплице	20
2.1 Мониторинг показателей в теплице	20
2.1.1 AM2320	21
2.1.2 BH1750	22
2.1.3 BMP180	23
2.1.4 DS18B20	24
2.1.5 Аналогово-цифровой преобразователь ADS1115	25
2.1.6 MQ-2 Gas Sensor	26
2.1.7 Moisture sensor	27
2.2 Сборка теплицы	29
4 Создание сайта	33
4.1 Язык программирования PHP	34
4.2 CSS	35
4.3 Оформление сайта	36
5 Бизнес план	39
5.1 Резюме	39
5.2 Характеристика предлагаемого продукта	39
5.3 Расчет суммы инвестиций для разработки проекта	40
5.3.1 Заработная плата и социальные отчисления	40
5.3.2 Материальные затраты	41
5.3.3 Расчет себестоимости проекта	43
6 Безопасность жизнедеятельности	46
6.1 Анализ условий труда	46
6.1.2 Условия безопасности при пайке и лужении	47
6.1.3 Противопожарная безопасность	49
6.1.4 Эргономика рабочего места	50
6.2 Расчет системы зануления	52
6.3 Расчет кондиционирования в лабораторном помещении.	54
Заключение	57
Список литературы	58
Приложение А	59
Приложение В	65

Аңдатпа

Бұл дипломдық жұмыста жылыжай саласында қашықтықтан басқаруды қолдану тенденциясы талқыланады. Қазіргі таңда бұл Қазақстанда өзекті мәселе болып табылады. Жылыжайда микроклимат құру үшін микроконтроллерлер, Ардуино, Raspberry Pi сияқты құрылғылар салыстырылды.

Сонымен қатар, өсімдіктердің өсіп-жетілуіне неғұрлым тиімді өсу үшін қажетті жағдайларды жасауға және қолдауға мүмкіндік беретін сәйкес сенсорларға шолу жасалды.

Салыстырулар мен шолулар негізінде Raspberry Pi және Ардуино негізінде автоматтандырылған жүйе құрылды, олардың синхронды түрде жұмыс жасау үшін және жылыжайдағы құрылғыларды тікелей басқару мақсатында арнайы сайт жасалған.

Аннотация

В дипломной работе рассмотрена перспектива использования удаленного управления в тепличной сфере, которая очень актуальна в Казахстане.

Также был проведено сравнение таких устройств, как микроконтроллеры, Ардуино, Raspberry Pi, для создания микроклимата в теплице.

В дополнении был произведен обзор более актуальных датчиков, которые позволяли создавать и поддерживать необходимые условия для более эффективного роста и созревания растения.

На основе проделанных сравнений и обзоров была создана автоматизированная система на базе Raspberry Pi и Ардуино, с веб-сайтом для их синхронизации и непосредственного управления устройствами в теплице.

Annotation

The diploma paper discusses the prospect of using remote control in the greenhouse sphere, which is very relevant in Kazakhstan.

A comparison was also made of devices such as microcontrollers, Ардуино, Raspberry Pi, to create a microclimate in a greenhouse.

In addition, a review of more relevant sensors was made, which made it possible to create and maintain the necessary conditions for more efficient plant growth and maturation.

Based on the comparisons and reviews, an automated system was created based on the Raspberry Pi and Ардуино, with a website for synchronizing them and directly managing devices in the greenhouse.

Введение

Масштаб технологических открытий человечества за последний век сложно переоценить. Темп развития технологий нарастает всё с большими оборотами,

Четвертая промышленная революция меняет масштабы производства. Сейчас каждое первое предприятие стремится минимизировать человеческий труд, заменяя его машинами, аппаратами, роботами, так как это намного более выгодный и уверенный путь к успешному развитию. Всё, что необходимо для исправного функционирования машин – это питание и правильное техническое обслуживание специалистом. В отличие от человека, он не требует заработной платы, социальных нужд, выходных, поощрений, к тому же, минимизируются ошибки, допускаемые людьми из-за невнимательности, непрофессионализма, и простого человеческого фактора.

На данный момент одной из основных целей любой компаний автоматизация процесса. Это позволяет не только ускорить процесс производства, но и исключить определенные риски.

На ряду с автоматизацией важным пунктом является установка удаленного управления. В век всего «умного» - домов, часов, гаджетов – дистанционное управление устройствами открывает перед человечеством невероятные горизонты. Исключается любая привязанность к чему-либо, все необходимые дела можно решить на расстоянии.

Основной целью моей дипломной работы является создание макета теплицы с автоматизированным поливом контроля микроклимата, который должен питаться от альтернативных источников энергии и городской сети с помощью буферного обмена.

1 Выбор аппаратной базы для создания «умной» теплицы

Как и перед созданием любого проекта, сначала необходимо провести анализ возможностей теплицы, сделать более эффективный выбор аппаратной базы и обосновать его.

1.1 «Умная» теплица

Одной из основных целей автоматизации теплицы, конечно же, является наблюдение за работой устройств теплицы и возможность менять определенные значения приборов, с целью изменения условий для более эффективного роста растущих растений, для которых отсутствие необходимой влажности, освещенности, температуры может сказаться очень пагубно. Рассмотрим конкретнее, какие именно характеристики необходимо учитывать.

Автоматизированная теплица наблюдает за следующими параметрами:

- температура;
- влажность;
- освещенность.

Также помогают в создании микроклимата теплицы нижеуказанные пункты:

1. Орошение;
2. Вентиляция;
3. Затенение;
4. Опрыскивание;
5. Удобрение;
6. Охлаждение;
7. Отопление.

1.2 Обзор технологий для управления автоматизацией

После этого, как мы решим, по каким показателям мы хотим отслеживать и управлять состоянием теплицы, необходимо выбрать аппаратную базу для внедрения её в нашу систему. Для этого рассмотрим основные методы и технологии, проанализируем их достоинства и недостатки.

Микроконтроллер [4] представляет собой специальную микросхему, предназначенную для управления различными электронными устройствами. Это уже не процессор, но и не компьютер. В современной терминологии это система на чипе (SoC). Микроконтроллер содержит одно или несколько процессорных ядер, а также память и программируемые периферийные устройства ввода-вывода. Программная память в виде сегнетоэлектрического ОЗУ, флеш-памяти NOR или OTP-ПЗУ также часто включается в чип, а также в небольшой объем ОЗУ. Микроконтроллеры предназначены для встроенных систем, в отличие от микропроцессоров, используемых в персональных компьютерах или других приложениях общего назначения, состоящих из различных специальных микросхем. Микроконтроллеры используются в автоматизированных устройствах, таких как системы управления автомобильными двигателями, имплантируемые медицинские устройства,

панели дистанционного управления, офисные машины, электрические приборы, электроинструменты, игрушки и другие встроенные системы. Благодаря уменьшению размеров и стоимости по сравнению с конструкцией, в которой используются отдельные устройства, такие как микропроцессор, память и устройства ввода / вывода, микроконтроллеры позволяют более экономично управлять еще большим количеством устройств и процессов. Таким образом, микроконтроллеры обеспечивают низкое энергопотребление, достаточную надежность системы, а также очень гибкую и настраиваемую логику благодаря своему программированию. На рисунке 1.1 представлен один из популярных микроконтроллеров.

Хотя микроконтроллеры в большинстве случаев имеют низкую стоимость, это преимущество может быть перечеркнуто сложностью разработки и создания прототипа для непромышленных и частных проектов, а также отсутствием универсальности в некоторых случаях.

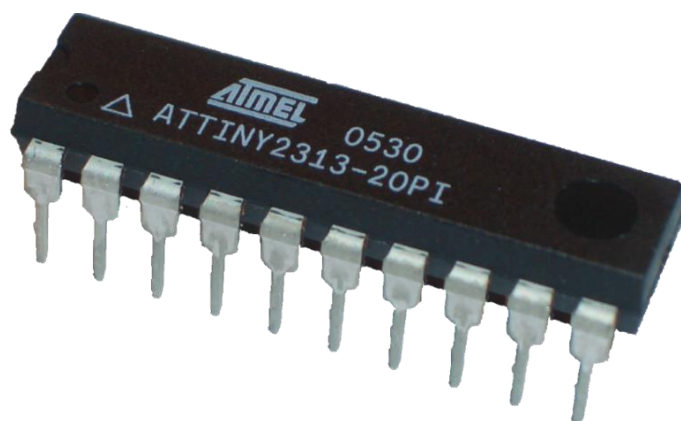


Рисунок 1.1 – Микроконтроллер ATtiny2313 американской фирмы Atmel

Но появление таких решений, как, например, Arduino и одноплатные компьютеры, вывело домашнюю и непромышленную автоматизацию на новый уровень. Значительно снижена стоимость этих систем без ущерба для надежности. В связи с массовым внедрением в повседневную жизнь стоимость датчиков (сенсоров) для них также снизилась.

1.2.2 Ардуино

Arduino [5] является платформой с открытым исходным кодом и может использоваться для создания проектов электроники. Создание программного обеспечения для программирования и создания компьютерных платформ (IDC), которое работает на вашем компьютере и работает для хранения и загрузки компьютерного кода. Платформа Arduino довольно популярна среди людей, только начинающих работать с электроникой. В отличие от большинства ранее разработанных плат, USB-кабель можно использовать для программирования нового кода на платной основе. На рисунке 1.2 представлена одна из моделей Ардуино.



Рисунок 1.2 – Ардуино Uno – одна из наиболее популярных плат в семействе Ардуино

Кроме того, в среде Arduino IDE (рис. 1.3) используется упрощенная версия C ++, которая упрощает обучение разработке для этой платы. Аппаратное и программное обеспечение Arduino было разработано для художников, дизайнеров, любителей, хакеров, начинающих и всех, кто заинтересован в создании интерактивных объектов или сред. Arduino может взаимодействовать с кнопками, светодиодами, моторами, динамиками, устройствами GPS, камерами, Интернетом и даже вашим смартфоном или телевизором. Эта гибкость в сочетании с тем фактом, что программное обеспечение Arduino является бесплатным, аппаратное обеспечение довольно дешевым, а программное и аппаратное обеспечение легко изучаемо, привело к большому сообществу пользователей, которые написали код и выпустили инструкции для огромного разнообразия проектов, основанных на Arduino. Arduino может использоваться в качестве «мозга» практически любого проекта в области электроники - для чего угодно, от роботов и нагревательного одеяла для обогрева до умных гаджетов.

```

Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);             // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);             // wait for a second
}
  
```

Рисунок 1.3 – скриншот Ардуино IDE

Существует много разновидностей плат Arduino, которые можно использовать для разных целей. Некоторые платы немного отличаются от приведенных ниже, но имеют в виду следующие компоненты (рисунок 1.4):

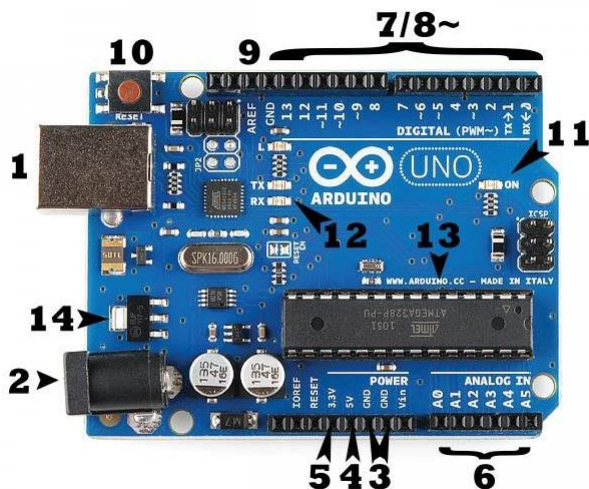


Рисунок 1.4 – Компоненты платы Ардуино

Каждая плата Arduino нуждается в способе подключения к источнику питания. Устройство Arduino UNO может получать питание от USB-кабеля, идущего от вашего компьютера, или от аналогичного источника питания. На рисунке выше обозначено USB-соединение (1), а бочковое гнездо (2). USB-соединение также является способом загрузки кода на плату Arduino. Рекомендуемое напряжение для большинства моделей Arduino составляет от 6 до 12 вольт.

Контакты на Arduino - это места, где вы соединяете провода для построения схемы (возможно, в сочетании с макетом и некоторыми проводами. Они обычно имеют черные пластиковые «заголовки», которые позволяют просто вставить провод непосредственно в плату. несколько различных типов контактов, каждый из которых отмечен на плате и используется для различных функций.

- GND (3): сокращение от "grund". На Arduino есть несколько выводов GND, каждый из которых может использоваться для заземления вашей цепи;

- 5 В (4) и 3,3 В (5): как можно догадаться, 5-контактный разъем подает 5 вольт, а 3,3-контактный - 3,3 вольт. Большинство простых компонентов, используемых с Arduino, работают как от 5, так и от 3,3 вольт;

- Аналоговый (6): площадь контактов от A0 до A5 на UNO является выходом «Аналоговый вход». Эти выходы могут считывать сигнал с аналогового датчика (например, датчика температуры) и преобразовывать его в цифровое значение, которое мы можем прочесть;

- Цифровой (7): напротив аналоговых контактов расположены цифровые выходы (от 0 до 13 в UNO). Эти контакты можно использовать как для цифрового входа (например, нажатием кнопки), так и для цифрового выхода (например, при включенном светодиоде);

- P W M (8): вы могли заметить тильду (~) рядом с некоторыми

цифровыми выводами (3, 5, 6, 9, 10 и 11 в UNO). Эти шаги действуют как обычные цифровые контакты, но также могут использоваться для так называемой широтно-импульсной модуляции (ШИМ). Эти контакты могут имитировать аналоговый выход (например, затухание светодиода);

- AREF (9): иногда используется для установки внешнего напряжения (от 0 до 5 вольт) в качестве верхнего предела для аналоговых входных контактов.

Arduino имеет кнопку сброса (10). Нажатие этой кнопки временно соединяет контакт сброса с землей и перезапускает любой код, загруженный в Arduino. Это может быть очень полезно, если ваш код не повторяется, но вы хотите проверить его несколько раз;

Под и справа от слова «UNO» на доске есть крошечный индикатор рядом со словом «ON» (11). Этот светодиод должен гореть каждый раз, когда вы подключаете Arduino к источнику питания.

TX - для передачи, RX - для приема. Эти маркировки появляются в электронике довольно редко, чтобы указать контакты, ответственные за последовательную связь. В нашем случае в Arduino UNO есть два места, где появляются TX и RX - один раз с цифровыми контактами 0 и 1, а второй - рядом со светодиодами с индикаторами TX и RX (12). Эти светодиоды загораются всякий раз, когда наш Arduino получает или передает данные (например, когда мы загружаем новую программу на плату).

Черная часть со всеми металлическими ножками - это IC или интегральная схема (13). Можно сказать, что это «мозги» Arduino. Основная IC на Arduino немного отличается для разных типов, но, как правило, это линия ATmega от IC от компании ATELEL. Это может быть важно, так как вам может потребоваться узнать тип микросхемы (вместе с типом вашей платы) перед загрузкой новой программы из программного обеспечения Arduino. Эту информацию обычно можно найти в письменном виде на верхней стороне IC.

Регулятор напряжения (14) не совсем то, с чем вы можете (или должны) взаимодействовать на Arduino. Но потенциально полезно знать, что это такое и зачем это нужно. Регулятор напряжения выполняет то, что сказано в его названии - он контролирует величину напряжения, которое подается на плату Arduino. Это отключит дополнительное напряжение, которое может повредить цепь.

Используя простой код, Arduino может управлять и взаимодействовать с широким спектром датчиков (Рисунок 1.5) - измерение яркости света, температуры, степени изгиба, давления, близости, ускорения, угарного газа, радиоактивности, влажности, атмосферного давления. Рисунок 1.5 – Некоторые датчики, которые легко взаимодействуют с Ардуино. Кроме того, есть такие платы (рисунок 1.6), которые называются shields – в основном это заранее подготовленные монтажные платы, которые помещаются поверх Ардуино и предоставляют дополнительные возможности – с двигателями, подключение к Интернету, предоставление сотовой или другой беспроводной связи, управление ЖК-экраном и многое другое.

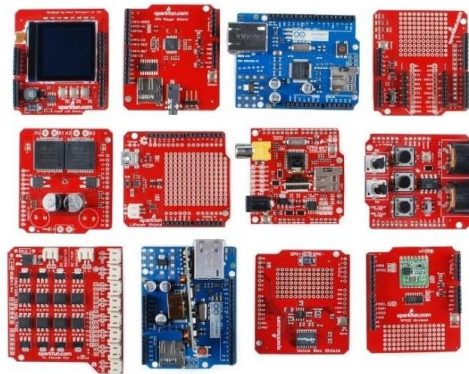


Рисунок 1.6 – Частичный выбор доступных плат для расширения возможностей Ардуино

Таким образом, можно выделить следующие достоинства использования Ардуино:

- готовность к использованию: готовая к использованию структура делает ее очень простой в использовании. Вам не нужно заботиться о программаторе, настройках предохранителей, программных средствах серийного монитора и т. д.
- примеры кода: много стандартных библиотек, которые облегчают и ускоряют разработку. Также стоит отметить, что это open-source продукт.
- легкие функции: в программном обеспечении Ардуино есть много функций, которые делают кодирование настолько простым и быстрым, что это невозможно при использовании простого микроконтроллера.
- большое сообщество: есть много форумов в Интернете, на которых люди говорят о Ардуино. Инженеры, любители и профессионалы делают свои проекты через Ардуино. Вы можете легко найти помощь во всем. Кроме того, сам сайт Ардуино объясняет все функции Ардуино.
- хорошая поддержка IoT-концепции.

Также стоит выделить недостатки этой платформы:

- структура Ардуино также является ее недостатком. Обычно во время создания проекта вы должны постараться сделать его размер как можно меньшим. Но с большими структурами Ардуино мы должны придерживаться больших размеров печатных плат. Если вы работаете на небольшом микроконтроллере, таком как ATmega8, вы можете легко сделать свою РСВ настолько маленькой, насколько это возможно.
- стоимость. Самый важный фактор, который нельзя отрицать – это стоимость. Это проблема, с которой сталкивается каждый любитель, инженер или профессионал. Иногда для некоторых решений выгоднее организовать решение на микроконтроллере.

1.2.3 Одноплатные компьютеры на примере Raspberri Pi

В последнее время в области автоматизации и робототехники стали популярны одноплатные компьютеры [6].

Одноплатный компьютер - это полноценный компьютер, построенный на одной плате, с микропроцессором, памятью, вводом / выводом (I / O) и

другими функциями, необходимыми для функционального компьютера. Одноплатные компьютеры создавались в качестве демонстрационных систем или систем разработки, для образовательных систем или для использования в качестве встроенных компьютерных контроллеров. Многие типы домашних компьютеров или ноутбуков объединяют все свои функции на одной плате.

В отличие от настольного персонального компьютера, компьютеры с одноплатными компьютерами часто не имеют слотов расширения для периферийных функций или устройств. Одноплатные компьютеры построены с использованием широкого спектра микропроцессоров. Простые конструкции, такие как созданные компьютерными энтузиастами, часто используют статическую оперативную память и недорогие 8- или 16-разрядные процессоры. Другие типы, такие как blade-серверы, работают как серверный компьютер в более компактном формате.

В настоящее время на рынке представлено довольно много популярных моделей от разных компаний. Один из самых популярных (различные модели и варианты): Raspberry Pi, BeagleBone, Orange Pi, Vana Pi, Asus Tinker Board. Все они отличаются по цене, техническим характеристикам, дизайну и т. Д. Мы будем ориентироваться на лидера по продажам и популярности - Raspberry Berry, у которого относительно средние конкуренты по соотношению цены и производительности, но в то же время у него самое большое сообщество и программное обеспечение для добычи руды

Raspberry Pi [7] - это крошечный компьютер размером с кредитную карту, на плате есть процессор, оперативная память и типичные аппаратные порты, которые вы найдете на большинстве компьютеров. Основная операционная система для Pi - RASBIAN, который основан на Debian. Хотя основной поддерживаемой операционной системой является RASBIAN, вы можете установить другие операционные системы, такие как Ubuntu Mac, Ubuntu Core, OSMC, RIS OS, Winds 10 IoT и многое другое. На сегодняшний день существует несколько версий этого компьютера, которые отличаются по размеру, цене и техническим характеристикам.

Raspberry Pi 3 (рис. 1.7) - это третье поколение. Он заменил Raspberry Pi 2 Model B в феврале 2016 года. По сравнению с Raspberry Pi 2 он имеет: 1.2-гигабитный 64-разрядный четырехъядерный процессор;

- ARMv8 802.11n Wireless LAN;
- Bluetooth 4.1;
- Bluetooth Low Energy (BLE); Как и Pi 2, он также имеет:
- 1 ГБ ОЗУ;
- 4 порта USB;
- порт Full HDMI;
- Порт Ethernet;
- Комбинированный 3,5-мм аудиоразъем и композитный видеоинтерфейс;
- Интерфейс камеры (CSI);
- Интерфейс дисплея (DSI);
- Гнездо для карты Micro SD (теперь push-pull, а не push-push).

Raspberry Pi 3 имеет идентичный форм-фактор с предыдущими Pi 2 (и Pi 1 Model B +) и имеет полную совместимость с Raspberry Pi 1 и 2. Также у всего семейства есть много внешних модулей, которые расширяют функциональность этого компьютера. Например, плата Gertboard (рисунок 1.8) позволяет расширить число линий ввода / вывода лучше, чем другие возможные решения.

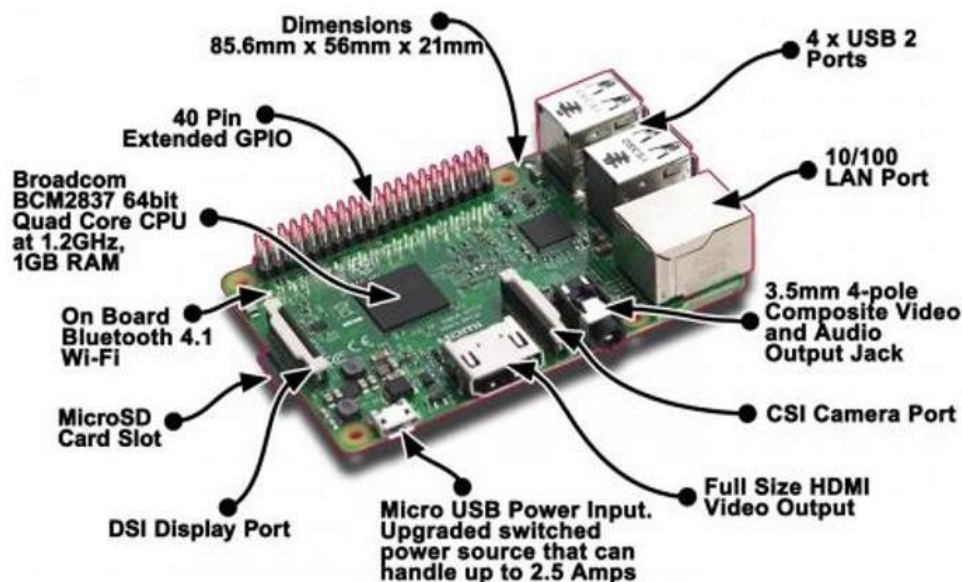


Рисунок 1.7 – Распберри Pi 3 model B

Это означает, что это 28-контактный микроконтроллер Atmega (любая из моделей spicс-cledogo AtmEGA 48A / PA, 88A / PA, 168A / PA или 328 / P). Благодаря подключению к разработке, разработке и внедрению новых плат разработки Arduino, теперь до и после.



Рисунок 1.8 – Gertboard для Распберри Pi

1.3 Обоснование выбора

Основываясь на обзоре возможных решений, вы должны сделать выбор в пользу создания базы автоматизации теплицы. Поскольку технологии движутся в направлении Интернета вещей, мы также выбираем такой вектор. Среди основных компонентов IoT - аппаратная поддержка, которая поддерживает все датчики и поддержку подключенного программного обеспечения. Наиболее распространенными аппаратными платформами для IoT являются Raspberry Pi и Arduino, поэтому мы немедленно отказываемся от варианта с «голым» микроконтроллером, потому что он по меньшей мере трудоемок и сложен в разработке, а изначально низкая стоимость микроконтроллера может не быть стоит дополнительных затрат. клиентская часть будущего проекта.

Давайте рассмотрим плюсы и минусы Arduino и Raspberry Pi для организации автоматизации с IoT:

Преимущества Arduino:

1. Архитектура с низким энергопотреблением (по сравнению с Raspberry Pi);
2. Легко начать с отличной онлайн-поддержки, быстрого и простого прототипирования;
3. Простое сопряжение с датчиками и сбор данных;
4. Дешевле, чем Raspberry Pi (для продуктов, которые не имеют подключения к Интернету);
5. Вы можете отправлять данные по беспроводной сети через Bluetooth, Rf и т. Д. На сервер через компьютер;
6. Множество GPIO с функциями ШИМ и дружественными к разработчикам;
7. Вы можете использовать IDE, python, ruby, embedd C и т. Д. Для программирования;
8. Полностью открытый исходный код.

Ардуино минусы:

1. Ограничение памяти (крайне мало по сравнению с Raspberry Pi);
2. Для подключения к Интернету вам понадобятся дополнительные экраны;
3. Менее мощный по сравнению с Raspberry Pi. Вы не можете запустить много тяжелых алгоритмов или запрограммировать сенсорный экран и использовать дополнительные карты расширения. Например, вы не можете запустить Open CV.

Плюсы Raspberry Pi:

1. Хорошая производительность. Расширяемая память;
2. ОС на базе Linux, а теперь даже Windows 10, которая может работать поверх нее, чтобы сделать обработку более удобной для пользователя;
3. Набор доступных GPIO и чем больше доступно GPIO, тем больше датчиков вы можете подключить;

4. Если у вас есть опыт работы с Linux, начать разработку очень легко, иначе потребуется некоторое время, чтобы понять это;

5. Python, C, C ++, Ruby, Go и многие другие могут использоваться для программирования Pi точно так же, как вы можете программировать на любом компьютере;

6. Люди успешно используют Raspberry Pi для запуска Open CV, алгоритмов интеллектуального анализа данных и т. Д. И связывают результаты с различными приложениями;

7. С точки зрения стоимости лучше, чем Arduino с картой расширения Ethernet;

8. Отличное интернет-сообщество и бесконечные возможности того, что вы можете с ним сделать.

Минусы Raspberry Pi:

1. Необходимо хорошее знание систем Linux;

2. Для большинства приложений производительность будет чрезмерной, поскольку в основном будет использоваться только передача данных;

3. Закрытый исходный код;

4. Требуется больше энергии.

Таким образом, можно сделать вывод, что обе платформы подходят для их выбора на аппаратной основе. Но для автоматизации теплицы с решением поставленных задач, таких как мониторинг показателей датчиков из любой точки мира (в основном с помощью веб-сайта) и коррекция поведения в режиме реального времени, целесообразно использовать Raspberry Berry или другой одноразовый бортовой компьютер с ОС по следующим причинам:

- Одно из основных преимуществ: Raspberry Pi можно использовать в качестве веб-сервера, что снимает с нас задачу поиска, настройки и управления дополнительным оборудованием. Кроме того, он несет дополнительные расходы;

- В дополнение к серверному компоненту существуют платформы с открытым исходным кодом для организации клиентской части для Raspberry Pi

без глубоких знаний в этой области (например, WEBIORi, который также отвечает за серверную часть);

- Доступных портов GPIO достаточно для подключения любых цифровых датчиков, а для аналоговых существует возможность использования дополнительных недорогих АЦП;

- Поддержка многозадачности;

- Есть много возможностей для организации и подключения к различным веб-сервисам, которые могут быть полезны (SMS-оповещения, твиттер-сообщения и т. Д.).

После выбора аппаратной базы нам нужно определиться с такими компонентами, как датчики и оборудование для климат-контроля теплицы.

2 Создание микроклимата в теплице

Для автоматизации теплицы в первую очередь нужны данные, на основе которых будут включаться модули для создания необходимых условий микроклимата для роста растений. Для этого нам потребуются различные датчики, а также другое оборудование, например, аналогово-цифровой передатчик для подключения аналоговых датчиков, а также камера, с помощью которой мы будем наблюдать за происходящим.

За управление теплицей, а если быть точным, создания нужного микроклимата, может отвечать различное периферийное оборудование в зависимости от необходимых требований и финансовых возможностей, но основные функции, которые это оборудование должно выполнять: полив, увлажнение воздуха, проветривание, регулировка освещения и поддержка необходимой температуры.

В этой главе описаны используемые датчики в нашем проекте, а также возможные варианты создания необходимых условий с помощью автоматизации для создания микроклимата в теплице.

2.1 Мониторинг показателей в теплице

Датчики, также называемые датчиками, огромны по количеству и, в первую очередь, имеют различное назначение.

Основные датчики (по назначению):

- датчик давления - для определения физического давления, например, при выжимании, сжатии, толкании;
- фотоэлементы - используются для измерения уровня освещенности, обнаружения объекта проекта по принципу свет / темнота;
- датчик температуры - для определения температуры окружающей среды или, например, жидкого кремния;
- датчик вибрации - используется для обнаружения движения;
- датчик движения - обнаружение двигателей активной активности, таких как животные или люди;
- Термопары - используются для измерения температуры, обычно выше 150 ° C.

Это далеко от предполагаемых датчиков. Датчики для определения влажности, огня, дыма и т. Д. Также доступны. В принципе, вы можете подобрать датчики практически для любой задачи.

У нас есть датчики, которые используют методы взаимодействия. Некоторые могут быть подключены к плате; соединение других требует дополнительных манипуляций. Например, могут потребоваться резиновые или дополнительные источники питания.

Датчики имеют аналоговый или цифровой сигнал.

Выбрав даты, вы заметите, что это чаще всего упоминается в статье с Arduino и не упоминается в списке с Distribution of Ri. Если датчик выдает цифровой сигнал, чтобы его можно было подключить к портам GPIO Raspberry Pi. Для того чтобы датчик включил аналоговый сигнал, такой

датчик можно подключить к распределителю только с помощью аналого-цифрового преобразователя (АЦП).

Различные датчики могут быть оценены, так как датчик может стоить 1 или 5 долларов, датчики могут быть очень точными.

Ниже приведен список датчиков, которые я выбрал для контроля теплотворной способности и того, что они измеряют.

цифровой:

- AM2320 [8] - влажность и температура воздуха;
- BH1750 [9] - освещение;
- BMR180 [10] - температура и давление воздуха
- DS18B20 [11] - температура почвы.

Аналог:

- MQ-2 [12] - детектор утечки газа;
- Moistur Sensor [13] - влажность почвы.

Поскольку у Distribution нет аналоговых входов, нам потребовался аналого-цифровой преобразователь, который был выпущен в форме ADS1115 [14].

Была также камера-камера Rapprerie Ree [15], с помощью которой мы будем делать снимки и отображать пользовательские интерфейсы.

Для того чтобы датчики были написаны на языке Riton, пример кода некоторых из предварительно определенных в Приложении А. Подключение тех же датчиков с назначением контактов описано и показано в Приложении В.

2.1.1 AM2320

AM2320 – это цифровой комбинированный датчик температуры и влажности (рисунок 2.1).

Датчик состоит из емкостного элемента измерения влажности и интегрального элемента измерения температуры. Оба этих элемента соединяются с высокопроизводительным процессором, который преобразовывает измеренные аналоговые значения в цифровой сигнал. Датчик имеет хорошее качество изготовления, быстрое время отклика и обладает помехозащищенностью для применения в промышленности.

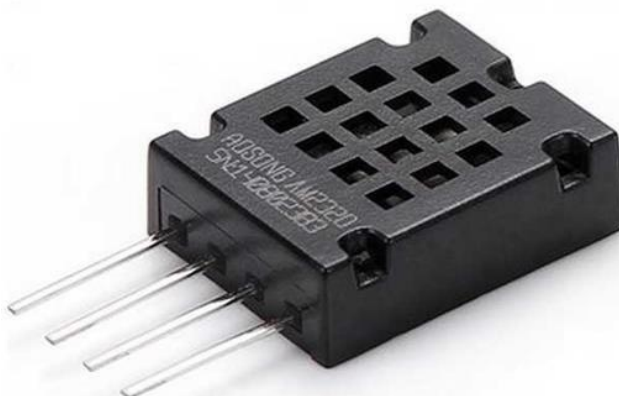


Рисунок 2.1 – AM2320

Таблица 2.1 – Спецификация AM2320

Характеристика	Значения
Напряжение питания	3,1 – 5,5 В
Потребляемый ток	20 мА
Диапазон измерения температуры	от – 40 до 80 °С
Максимальная погрешность измерения температуры	0,5 °С
Диапазон измерения влажности	от 0 до 100 %
Усредненная погрешность измерения влажности	3%

Характеристики:

- Очень малый размер;
- Экономичная работа;
- Сверхнизкое напряжение;
- Отличная долговременная стабильность;
- Стандарт I2C и выход на одну шину;
- Погрешность $\pm 0,5^{\circ}\text{C}$ / $\pm 3\% \text{ RH}$;
- Меньший размер, чем D H T11 /12, но более высокая точность.

2.1.2 ВН1750

Иногда возникает необходимость получить данные об уровне освещенности конкретного места, объекта. Для таких целей используются специальные приборы - люксометры. Но это не такое распространенное оборудование. Например, вы можете измерить свет с помощью фоторезистора GL5516. Но в этом случае необходимо изготовить измерительную схему и откалибровать данные, полученные от такого датчика. Датчик поможет нам упростить задачу. С помощью цифрового интерфейса он будет производить готовые данные в определенных единицах измерения - ВН1750 (рисунок 2.2).

Этот датчик находится в корпусе для поверхностного монтажа WSOF6I. Внутри сам фотодатчик выполнен в виде фотодиода, усилителя сигнала фотодиода, АЦП (аналого-цифрового преобразователя) и некоторой логики, которая обрабатывает данные, переводит все в единицы Люкс и передает их через I2C на устройство управления



Рисунок 2.2 – ВН1750

ВН1750 обладает следующими характеристиками:

- Цифровой интерфейс – I² C;
- Высокое разрешение – до 0,5 Лк;
- Малый потребляемый ток и функция спящего режима;
- Фильтрация световых шумов 50/60 Гц;
- Малая зависимость от источника света (лампа накаливания, свето диод и так далее);
- Малое влияние инфракрасного излучения;

2.1.3 BMP180

Этот высокоточный датчик от Bosch является лучшим недорогим решением для измерения барометрического давления и температуры. Датчик припаивается к печатной плате с регулятором 3,3 В, реле уровня I²C и нагрузочными резисторами на клеммах I² C. BMP180 (рисунок 2.3) - это датчик нового поколения от Bosch, который заменяет BMP085. Он полностью идентичен BMP 085 с точки зрения прошивки / программного обеспечения. Эта плата соответствует стандарту 5 В - регулятор 3.3 В и цепь переключения уровня I²C включены, так что вы можете безопасно использовать этот датчик с 5 В логикой и питанием.

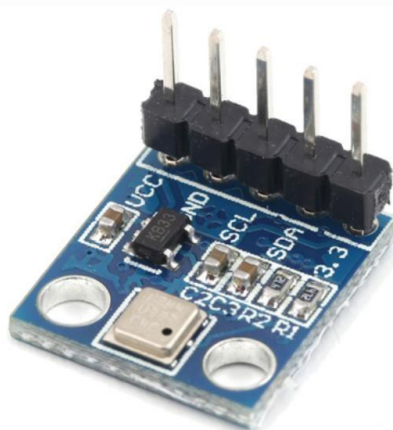


Рисунок 2.3 – BMP180

Характеристики датчика BMP180:

- Диапазон измерения давления: 300 – 1100hPa;
- Напряжение питания: от 3.3 и 5V;

2.1.4 DS18B20

DS18B20 (рисунок 2.4) – цифровой термометр с программируемым разрешением, от 9 до 12-bit, которое может сохраняться в E EP ROM памяти прибора. DS 18 B20 обменивается данными по 1 - Wire шине и при этом может быть, как единственным устройством на линии, так и работать в группе.

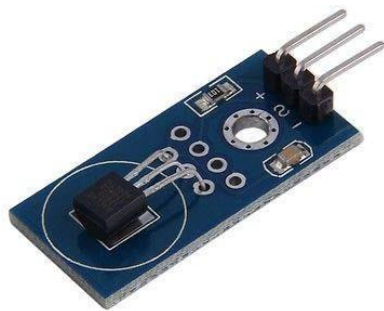


Рисунок 2.4 – DS18B20

Характеристики DS18B20:

- Диапазон измеряемых температур: $-55...+125\text{ }^{\circ}\text{C}$;
- Точность: $\pm 0,5^{\circ}\text{C}$ (в пределах $-10...+85\text{ }^{\circ}\text{C}$);
- Время получения данных: 750 мс при 12-битном разрешении; 94 мс при 9-битном разрешении;
- Напряжение питания: 3–5,5 В;
- Потребляемый ток при бездействии: 750 нА;
- Потребляемый ток при опросе: 1 мА.

2.1.5 Аналогово-цифровой преобразователь ADS1115

ADS1115 (рисунок 2.5) - отличный аналого-цифровой преобразователь, который позволяет легко и просто использовать с распределенными аналоговыми датчиками, а также его коммуникационную шину I2C. ADS1115 - это высокоточный 16-разрядный АЦП с 4-мя доступными.



Рисунок 2.5 –АЦП ADS1115

Данное устройство предполагает установку адреса для I2C шины при помощи замыкания вывода ADDR на цифровые сигнальные выводы или выводы шины питания (рисунок 2.6). Установка различных адресов I2C предназначена для того, чтоб при подключении двух и более различных устройств не происходило конфликта между ними, т. к. в лучшем случае модули у которых адреса совпадут просто не будут работать, а в худшем – работа всех модулей на шине может стать непредсказуемой.

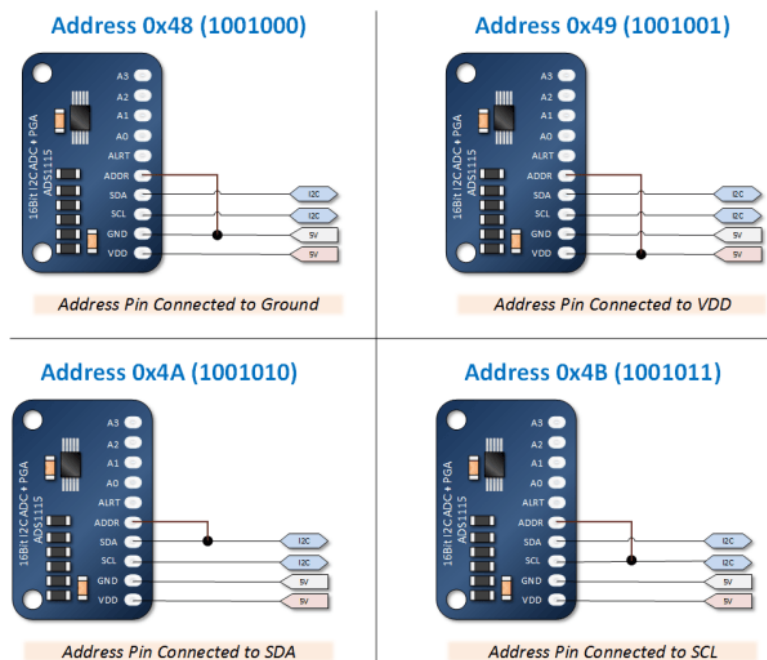


Рисунок 2.6 – Выбор адреса для АЦП ADS1115

Характеристики АЦП ADS1115:

- Широкий диапазон электропитания: от 2,0 до 5,5 В;
- Малое потребление тока, непрерывный режим: только 150 мкА;
- Программируемая скорость передачи данных: 8sps - 860sps;
- Внутреннее низкотемпературное напряжение;
- Внутренний осциллятор;

2.1.6 MQ-2 Gas Sensor

Датчик газа, построенный на базе газоанализатора MQ-2 (рисунок 2.7) позволяет обнаруживать наличие в окружающем воздухе углеводородных газов (пропан, метан, н-бутан), дыма (взвешенные частицы, являющиеся результатом горения), водорода.

Датчик можно использовать для обнаружения утечек промышленного газа и задымления. Выходным результатом является аналоговый сигнал, пропорциональный содержанию газов, к которым восприимчив газоанализатор. Чувствительность может быть настроена с помощью триммера на плате датчика.

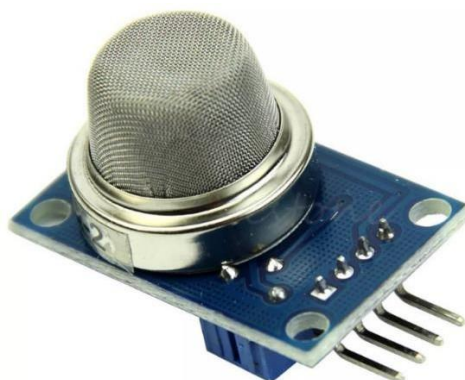


Рисунок 2.7 – MQ-2

Показания сенсора подвержены влиянию температуры и влажности окружающего воздуха. Поэтому в случае использования датчика газа в изменяющейся среде, при необходимости получения точных показаний, понадобится реализовать компенсацию этих параметров.

Диапазон измерений:

- Пропан: 0,2 – 5 промилле;
- Бутан: 0,3 – 5 промилле;
- Метан: 5 – 20 промилле;

2.1.7 Moisture sensor

Moisture Sensor (рисунок 2.8) - датчик для измерения влажности почвы. Издание выполнено в виде вилки, которую для измерений легко втыкать в почву. Сопротивление сухой почвы очень высокое. С увеличением влажности сопротивление почвы уменьшается. Датчик установлен на окне усилителя переходного тока. Он был включен в цепь между коллектором и транзистором биполярного трансформатора n-pn. Когда влажность почвы изменяется, сопротивление между коллектором, к которому подключен положительный уровень мощности, и основанием, изменяется. Изменения в базе данных, ток течет от коллектора через излучатель на землю. В результате изменяется выходное аналоговое напряжение, которое можно оцифровать с помощью АЦП. Продукт может быть использован в Сістемах автоматического полива, что нам и нужно.

Характеристики Moisture Sensor:

- Глубина обнаружения: 38мм;
- Напряжение питания: 2В – 5В;
- Монтажные отверстия: 2мм;
- Размеры: 20 x 51мм.



Рисунок 2.8 – Moisture Sensor

3 Сборка теплицы

Если процесс мониторинга показателей еще можно как-то стандартизировать и создать универсальную систему, то процесс создания микроклимата для каждой конкретной теплицы уникален. Также мы не ограничены в количестве каналов, в настоящее время на рынке представлены модули как на 4 канала (рисунок 2.10), так и на 8, 16 и более (рисунок 3.1). Проблема подключения реле к портам GPIO Распберри Pi не составляет никаких проблем, и нам остается только посылать логические «0» или «1» для смены состояния нужного модуля.

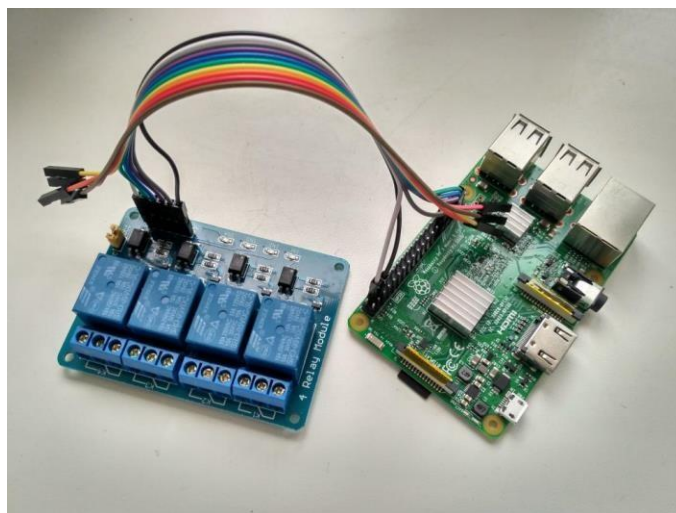


Рисунок 3.1 – Модуль реле с 4 каналами

Но в независимости от оборудования, сами алгоритмы автоматизации можно обобщить. Поэтому далее сделаем небольшой обзор на варианты реализаций некоторых функций, которые могут быть использованы для автоматизации теплицы, не рассматривая конкретное оборудование.

Автоматический полив: самый простой вариант для автоматизации полива – это задать время начала полива и его продолжительность. Но в таком случае система не является гибкой, никак не учитываются показания с датчиков. Поэтому более оптимальный вариант – полив при достижении влажности почвы ниже контрольного значения. При этом надо учесть факт того, что наш датчик влажности почвы может перестать давать корректные значения (например, поломка), из-за чего может возникнуть опасность затопления. Возможные варианты выхода из этой ситуации:

- После очередного полива отключать его возможность на некоторый промежуток времени;
- Задать некоторое число поливов и каждый раз через малый промежуток времени делать контрольный замер влажности почвы. Если влажность почвы не изменяется – значит с нашим датчиком что-то случилось, поэтому следует отключить возможность полива до устранения причины неисправности.

Проветривание: проблема аналогична поливу – можно открывать окна

или включать вентиляторы на определенное время в нужный час, а можно ориентироваться на показатели датчиков, чтобы достичь нужной влажности и температуры воздуха с настройкой запасного варианта на случай отказов датчиков. Также при наличии датчика-анализатора газов можно ориентироваться на определенный уровень углекислого газа или создать определенный алгоритм для полного проветривания в случае обнаружения токсичных веществ или дыма.

На веб-сайте мы можем определить заданные значения температуры и влажности. Также возможно управлять каждым приводом вручную и просматривать их состояния.

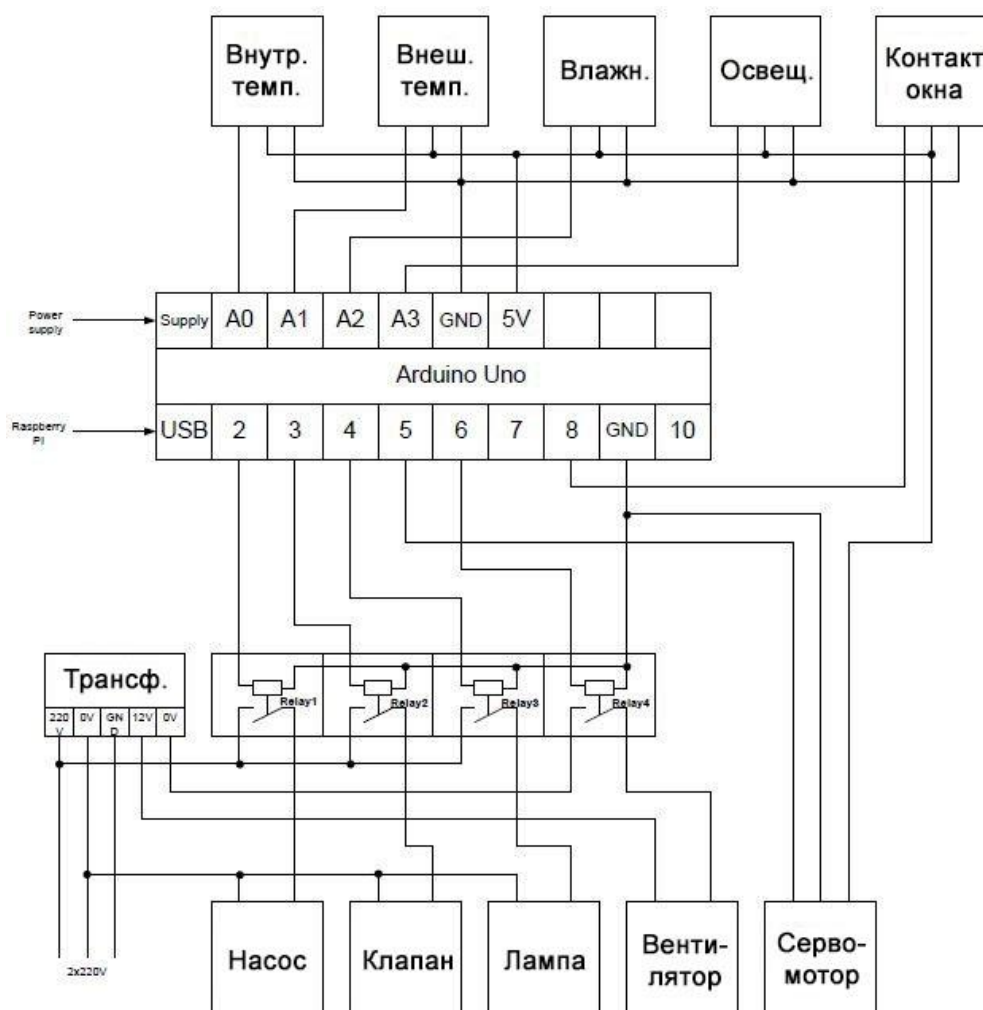


Рисунок 3.2 – Схема подключения устройств

Одной из особенностей моей дипломной работы является питание от альтернативных источников энергии. В своей работе я использовал солнечные панели (рисунок 3.3) и ветрогенератор (рисунок 3.4).

Солнечные батареи уже сейчас используются для питания самой разнообразной техники: от мобильных гаджетов до электромобилей. Как устроены, какими бывают и на что способны современные солнечные батареи, вы узнаете из этой статьи.

Солнечные же батареи производят непосредственно электричество, что

намного эффективнее. При прямой трансформации теряется значительно меньше энергии, чем при многоступенчатой, как у коллекторов (концентрация солнечных лучей, нагрев воды и выделение пара, вращение паровой турбины и только в конце выработка электричества генератором).

Современные солнечные батареи состоят из цепи фотоэлементов – полупроводниковых устройств, преобразующих солнечную энергию напрямую в электрический ток. Процесс преобразования энергии солнца в электрический ток называется фотоэлектрическим эффектом.

Данное явление открыл французский физик Александр Эдмон Беккерель в середине XIX века. Первый же действующий фотоэлемент спустя полвека создал русский ученый Александр Столетов. А уже в двадцатом столетии фотоэлектрический эффект количественно описал не требующий представления Альберт Эйнштейн.

На работу солнечных батарей может негативно влиять ряд факторов. К примеру, с ростом температуры снижается КФП фотоэлементов. Это при том, что солнечные батареи как раз-то и устанавливают в жарких солнечных странах. Получается своеобразная палка о двух концах. А если затемнить часть солнечной панели, то неактивные фотоэлементы не только прекращают вырабатывать электричество, но и становятся дополнительной, вредной нагрузкой.

Лидерами глобального производства солнечных батарей являются компании Suntech, Yingli, Trina Solar, First Solar и Sharp Solar. Первые три представляют Китай, четвертая – США, а пятая, как нетрудно догадаться, является подразделением японской корпорации Sharp.

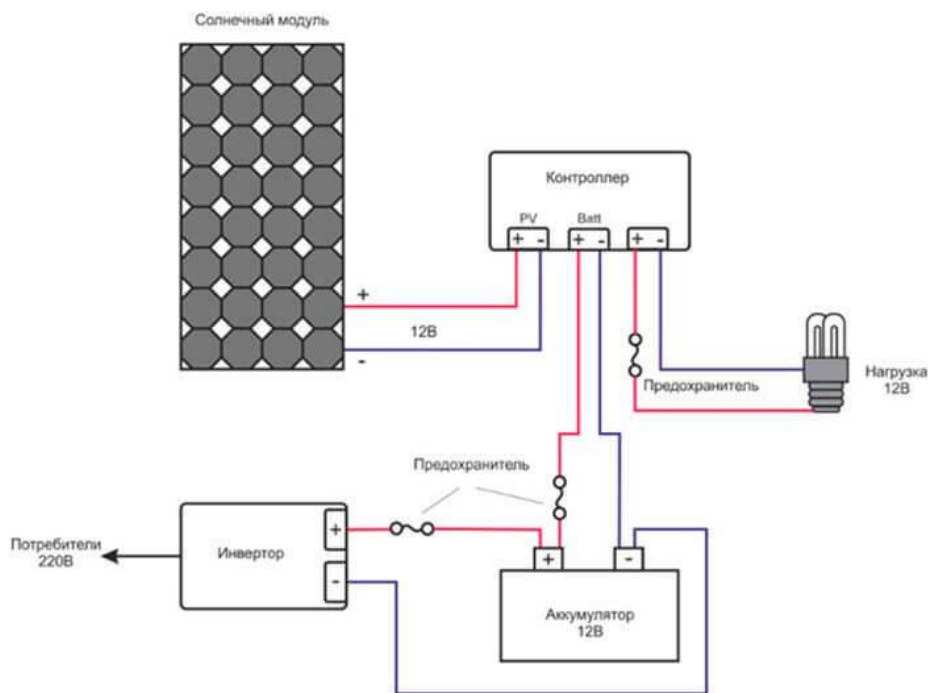


Рисунок 3.3 – Подключение солнечной панели

Ветрогенератор (ветроэлектрическая установка или сокращенно ВЭУ)

– это прибор для превращения энергии ветра в электрическую.

Сначала он превращает кинетическую энергию ветра в механическую энергию ротора, а затем в электрическую энергию. Мощность ветрогенератора может быть от 5 КВт до 4500 КВт. Современные устройства генерируют энергию даже очень слабого ветра – от 4 м/с. Ветроэлектрические установки могут входить в состав частной независимой электростанции и позволяют продавать излишнюю энергию государству по условиям «зеленого тарифа». Такие сооружения могут быть источником энергии для локальных и островных объектов, так как решают проблемы энергоснабжения автономно.

Потоки ветра вращают лопасти ветрогенератора: проходят через турбину, приводит её в действие и она начинает вращаться. На валу турбины возникает энергия, которая будет пропорциональна ветровому потоку. Чем сильнее ветер, тем большее количество энергии возникает. Далее энергия передается по валу ротору на мультипликатор (если он есть), который её генерирует. Учтите, что более продуктивными являются устройства без мультипликатора, который ускоряет вращение оси, потому что не создается, а, естественно, и не растрачивается лишняя энергия, а скорости ветра вполне достаточно для оптимальной работы ветрогенератора. Генератор превращает механическую энергию в электрическую. Мощность ветряка измеряется «ометаемой» площадью турбины.

Чем больший размер лопастей, тем большую мощность он создает.

По мощности и области применения ветрогенераторы бывают:

- промышленные (мощность от 500 КВт);
- бытовые (мощность 0-10 КВт).

Устройства с мощностью от 10 до 500 КВт используются крайне редко.



Рисунок 3.4 – Подключение ветрогенератора

4 Создание сайта

Наличие сайта в наше время не то что бы желательно, а просто необходимо. Крупные компании вкладывают огромные деньги в развитие своего сайта для того, чтобы клиент, посещая сайт, мог найти всю необходимую для него информацию в кратчайшие сроки, при этом не наткнувшись на какие-либо ошибки в работе сайта и его функционале. Также важную роль в оценке сайта играет его дизайн, так как эта составляющая очень важна для вербального восприятия, что тоже очень сильно сказывается на первом ощущении клиента при посещении сайта.

Я, не собираясь отставать от трендов, в своей дипломной работе так же создал веб-сайт для управления функционалом теплицы. Сайт написан на языке программирования PHP.

Основная идея такова: датчики собирают информацию в теплице о температуре, влажности, освещенности, передают их на Ардуино, он обрабатывает эти данные, и посылает необходимые сигналы на исполнительные устройства. И далее вся картина о состоянии устройств передается на Raspberri, после чего отправляется на веб-сайт (рисунок 4.1)

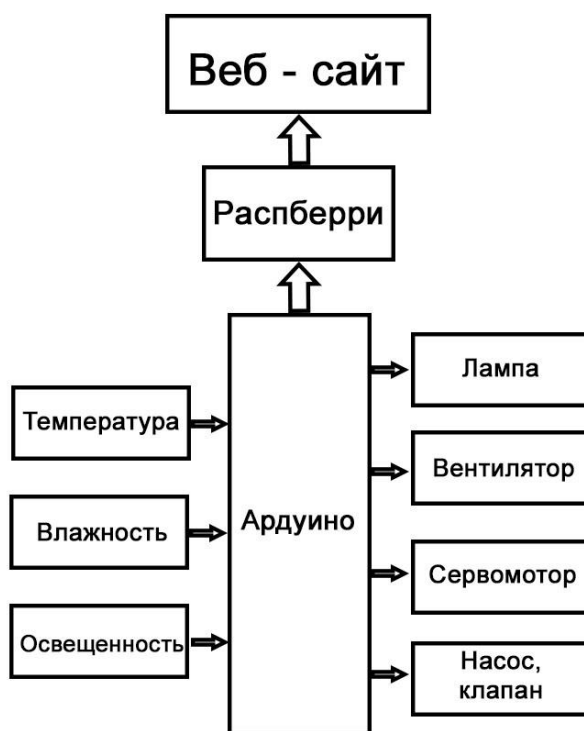


Рисунок 4.1 – общая схема работы устройств

4.1 Язык программирования PHP

PHP - широко используемый универсальный язык сценариев с открытым исходным кодом.

Проще говоря, язык программирования, специально разработанный для написания веб-приложений (скриптов), работающих на веб-сервере.

Аббревиатура PHP означает «Hypertext Preprocessor (Препроцессор

гипертекста)». Синтаксис языка происходит от C, Java и Real. RHR достаточно прост для изучения. преимущества

Важным преимуществом языка PHP над такими языками, как языки Perl и C, является возможность создавать документы HTML со встроенными командами PHP. Подробнее об этой функции читайте здесь.

Например, JavaScript означает, что сценарии RHR выполняются на стороне сервера. Вы можете настроить свой сервер так, чтобы файлы HTML обрабатывались процессором PHP, чтобы клиенты не могли узнать, получают ли они обычный файл HTML или результат сценария.

PHP известен как язык программирования на стороне сервера. Это означает, что он работает на веб-сервере. JavaScript, работает на стороне клиента, это означает, что они работают в веб-браузере.

JavaScript - например, работа с файлами, базой данных или работа с изображением. Я должен сказать, что JavaScript очень быстро распространился в наши дни.

Хотя многие языки могут быть использованы для создания веб-приложений. PHP имеет много полезных веб-функций, таких как:

- а) чтение и обработка веб-форм и файлов cookie;
- б) функции создания и работы с графикой;
- в) установление связей с популярными базами данных, такими как MySQL;
- г) функции для работы с HTML.

Вы можете включить блокировку кода RHR на странице HTML. Вы можете изолировать блоки RHR с помощью специальных символов. Когда веб-сервер получает информацию о странице, все блоки страницы запускаются в браузере.

Эта функция позволяет легко создавать интерактивные обычные веб-страницы. Отличный инструмент для обратной связи.

4.2 CSS

Каскадные таблицы стилей (CSS) - это язык, который отвечает за визуальное представление документов пользователю.

Под документом мы понимаем набор языковой информации, описываемой языком разметки.

Форма документа пользователя, в свою очередь, означает его форму. Браузеры, такие как Firefox, Chrome или Internet Explorer, были созданы для визуального представления документов, например, на экране компьютера, проекторе или через принтер.

Потребность в CSS была признана консорциумом W3C в 1990-х годах. В 1996 году был принят стандарт CSS1, позволяющий изменять настройки шрифта, цвет, атрибуты текста, выравнивание и отступы. В 1998 году был выпущен CSS2, добавлены функции для использования блочной разметки, звуковых таблиц, сгенерированного контента, указателей, медиа страниц. Версия CSS3 значительно расширила возможности: стало возможным создавать анимированные элементы без использования JavaScript, появилась

поддержка сглаживания, теней, градиентов и т. Д. Специализация была разделена на модули. С 2011 года модули CSS4 находятся в стадии разработки. Возможности все еще доступны в проекте переговоров.

CSS (каскадные таблицы стилей, каскадные таблицы стилей) - это язык для описания внешнего вида HTML-документа. Это одна из базовых технологий в современном интернете. Почти ни один сайт не может обойтись без CSS, поэтому HTML и CSS действуют в одной ссылке.

Каскадные таблицы работают в HTML, но это совершенно другой язык. HTML структурирует документ и организует информацию, а CSS взаимодействует с браузером, чтобы придать документу вид.

В HTML мы создадим текст разметки - документ с гиперссылками, таблицами, маркированными списками, различными начальными шрифтами, заголовками, подзаголовками и так далее. Мы получаем «лист» текста с таблицами и иллюстрациями. Ученые изобрели Интернет, и для них такое положение дел было приемлемым. Но все изменилось, когда WWW пошла в массы, и их увлечения стали создавать обычные пользователи, которые также хотели коммерческие компании со своими корпоративными стандартами дизайна. В целом, веб-страницам требовался индивидуальный стиль.

4.3 Оформление сайта

Итак, для повышения ценности моей работы был создан веб-сайт, с помощью которого можно будет управлять функционалом теплицы. Мой сайт состоит из двух страниц. Первая – главная - будет содержать сведения о состоянии устройств (датчиков температуры, влажности, освещенности, полива) и о проводимых этими устройствами измерениях (рисунок 3.1)

В рамке «Измерения» находятся показания измерений датчиков, а именно: внешняя температура, внутренняя температура, уровень влажности почвы, освещенность, количество запусков насоса, количество запусков вентилятора, количество включений лампы, количество открытий клапана, количество открытий окна.



Рисунок 4.2 – Главная страница сайта

В рамке «Состояние устройств» находятся действующие режимы работы

устройств, которые отвечают за создание и сохранение микро-климата в теплице. Имеется 2 режима работы устройств: ручной и автоматический.

При ручном режиме отключается скрипт, ответственный за автополив, освещение, температуру – в зависимости от прибора. Ручной режим был введен для расширения возможностей управления устройствами, а также по следующим причинам: в случае выращивания более комплексных и капризных культур, не всегда будет полезно, к примеру, поливать их по строгому графику. Поэтому здесь больше пригодится широкий человеческий разум, интуиция и знания.

Автоматический же наоборот – включает/выключает скрипт, по которому работают приборы.

Вторая страница называется «Команды и параметры» (рисунок 3.2). В ней уже содержатся не просто данные с датчиков и приборов, а так же кнопки, переключатели. Эта страница показывает всю суть автоматизированных теплиц – здесь можно задать, к примеру, необходимые значения температуры, при достижении которых вентилятор будет включаться/выключаться.

Первая рамка – «Команды» подразделена на 5 окон, каждый из которых содержит сведения о соответствующем приборе и кнопки для управления им. Функционал данных рамок – это изображение действующего режима, состояния и кнопки для смены режима.

Во второй рамке – «Параметры» можно задать определенное значение для изменения порога включения приборов. Содержит два окна: «Внутренняя температура» и «Влажность почвы»

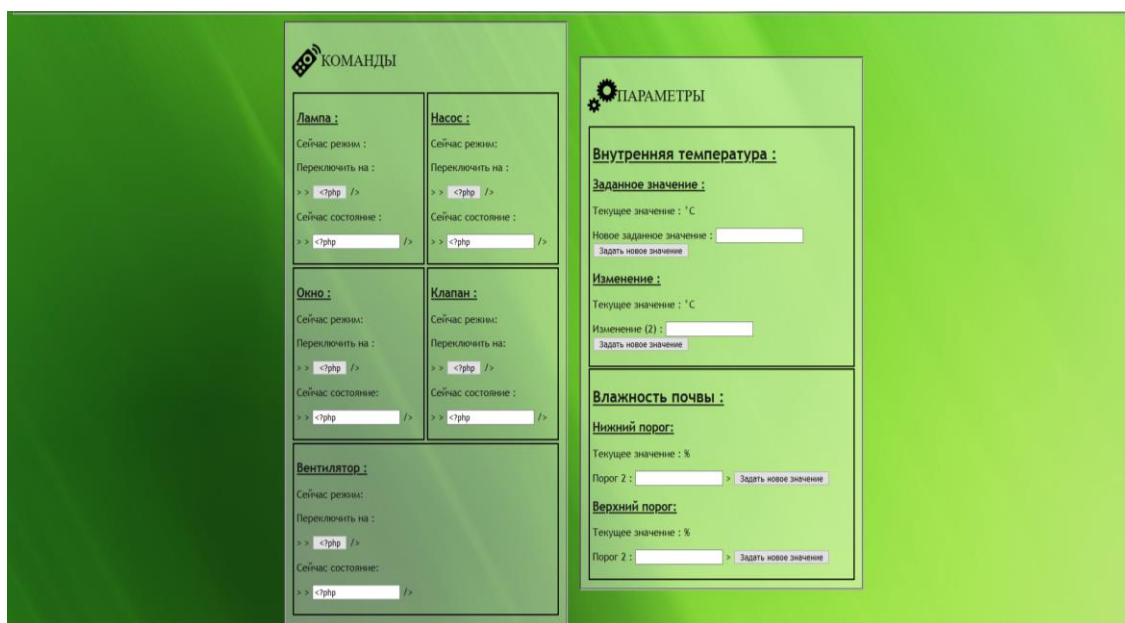


Рисунок 4.3 – Страница «Команды и параметры»

5 Бизнес-план

5.1 Резюме

Казахстан является одним из ведущих регионов в области сельского хозяйства, обладающим огромным аграрным потенциалом. По данным

Агентства Республики Казахстан по статистике в 2013 г. удельный вес овощей и бахчевых, корнеплодов и клубнеплодов в валовом выпуске продукции (услуг) сельского хозяйства составлял 15,3%, что в абсолютных показателях равно 364 982 млн. тенге, при этом темп роста объема составил к предыдущему году 106,5%.

В целом, Казахстан обеспечивает себя овощами. Единственная проблема заключается в ранних овощах. Не хватает овощей, которые выращиваются в закрытом грунте, то есть в теплицах. Поэтому государство сейчас стимулирует организацию теплиц. Только за последние два года количество теплиц у нас в десятки раз увеличилось. Чтобы достичь необходимого уровня обеспечения внутреннего рынка ранними овощами, необходимо дополнительно на 80% увеличить производство ранних овощей.

Основными конкурентами нового тепличного хозяйства могут выступить имеющиеся тепличные хозяйства, которые завоевали 8 определенных клиентов в области, привозная продукция из соседних областей и южных республик, а также новые тепличные хозяйства.

5.2 Характеристика предлагаемого продукта

В разрабатываемом мной проекте под автоматизацией управления теплицы имеется в виду обеспечение растений всем необходимым. Основным назначением автоматизации теплицы являются мониторинг различных показателей датчиков и управление оптимальными условиями для роста растений, которые требуют определенного диапазона температуры, влажности почвы, света, влажности и состава воздуха (кислород, двуокись углерода и азот), а также питательных веществ для роста. Тепличное покрытие помогает контролировать многие из этих факторов, чтобы помочь в развитии растений и увеличении количества и качества урожая.

Автоматизация теплицы будет осуществляться с помощью ряда приборов и датчиков. Они будут отвечать за получение и отправку данных на персональный компьютер, далее на веб-сервер. В качестве платформы для веб-сервера был выбран одноплатный компьютер Raspberry Pi 3, так как он имеет упрощенный мануал и широкий спектр возможностей.

В качестве сборщика данных с датчиков я выбрал Ардуино Uno. К его плюсам можно отнести:

1. Архитектура с низким энергопотреблением (по сравнению с Raspberry Pi).
2. Простое сопряжение с датчиками и сбор данных.
3. Можно отправлять данные без проводов, используя Bluetooth, Rf и т. д. на сервер через компьютер.
4. Множество GPIO с возможностями PWM и дружественным к разработчикам.
5. Полностью открытый исходный код

5.3 Расчет суммы инвестиций для разработки проекта

Для расчета суммы инвестиций в проект по автоматизации теплицы нам необходимо узнать размер заработной платы, которую получит разработчик данной системы за весь период его работы, а также себестоимость материалов, которые необходимо будет закупить для реализации системы.

5.3.1 Заработная плата и социальные отчисления

Одной из основных статей расчета себестоимости разработки системы автоматизации теплицы является оплата труда разработчикам.

Для расчета основной ЗП необходимо установить среднюю ставку разработчика – 100000 тг./мес.

Размер фонда заработной платы, выплачиваемый работникам, определяется по следующей формуле:

$$\Phi_{\text{от}} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (5.1)$$

где $Z_{\text{осн}}$ – базовая заработная плата, тенге;

$Z_{\text{доп}}$ – дополнительная заработная плата, тенге.

Дополнительная заработная плата составляет 10% от базовой заработной платы и определяется по следующей формуле:

$$Z_{\text{доп}} = Z_{\text{осн}} \times H_{\text{д}} / 100, \quad (5.2)$$

$$Z_{\text{доп}} = 100000 \cdot 0,1 = 10000 \text{ (тг)},$$

где $H_{\text{д}}$ – коэффициент дополнительной заработной платы.

Фонд заработной платы, выплачиваемый работникам, равен сумме основной заработной платы $Z_{\text{осн}}$ и дополнительной заработной платы $Z_{\text{доп}}$:

$$\Phi_{\text{от}} = Z_{\text{осн}} + Z_{\text{доп}}, \quad (5.3)$$

$$\Phi_{\text{от}} = 100000 + 10000 = 110000 \text{ (тг)}.$$

Социальный налог составляет 9,5% от дохода работника и рассчитывается по следующей формуле:

$$C_{\text{н}} = (\Phi_{\text{от}} - \text{ПО}) \times 9,5\%, \quad (5.4)$$

где ПО – пенсионные отчисления, которые составляют 10% фонда заработной платы и не облагаются социальным налогом:

$$\text{ПО} = \Phi_{\text{от}} \times 10\%, \quad (5.5)$$

$$\text{ПО} = 110000 \times 10\% = 11000 \text{ (тг)}.$$

Из этого следует:

$$C_n = (100000 - 10000) \times 9,5\% = 8550 \text{ (тг.)}$$

5.3.2 Материальные затраты

Для разработки системы автоматизации теплицы необходимо закупить комплектующие, на основе которых эта система будет создана. В стоимость входит только сама плата Raspberries Pi и датчики для мониторинга показателей, т.к. управляющее оборудование покупается и настраивается отдельно под нужды клиента. Основные компоненты с закупочными ценами (с онлайн-магазина AliExpress) на них представлены в таблице 5.1:

Таблица 5.1 – Материальные затраты

Наименование	Цена тг./шт.
Распберри Pi 3 model B	14250
AM2320	960
BH1750	350
DS18B20	555
MQ-2 Gas Sensor	550
Moisture Sensor	350
ADS1115	565
Распберри Pi Camera Board	3500
SD карта 8 Гб	1235
Прочие расходные материалы	2000
Итого	24315

Общие капитальные затраты на разработку автоматизированной теплицы включают капитальные затраты на оборудование, капитальные затраты на программное обеспечение, капитальные затраты на установку и транспортные расходы. Общие капитальные затраты рассчитываются по формуле:

$$\sum K = K_{\text{тм}} + K_{\text{м}} + K_{\text{к}}, \quad (5.6)$$

где $\sum K$ - общая сумма капитальных вложений;
 $K_{\text{тм}}$ - капитальные вложения в оборудование;
 $K_{\text{м}}$ - капитальные затраты на монтажные работы (20% от стоимости оборудования);
 $K_{\text{к}}$ - суммарные капитальные затраты на транспортные расходы (5% от стоимости оборудования).

$$K_{\text{м}} = 24315 \times 0,2 = 4865 \text{ (тг.)},$$

$$K_{\text{к}} = 24315 \times 0,05 = 1215 \text{ (тг.)},$$

$$\sum K = 24315 + 4865 + 1215 = 30395 \text{ (тг)}.$$

Расходы на материалы и запасные части составляют 0,5% в год от суммы капитальных затрат:

$$M = 0.005 \times K, \quad (5.7)$$

$$M = 0.005 \times 30395 = 152 \text{ (тг)}.$$

где M - материалы и детали,

Амортизация - процесс переноса по частям стоимости основных средств и нематериальных активов по мере их физического или морального износа на себестоимость производимой продукции (работ, услуг). Отчисления амортизации A определяются по следующей формуле:

$$A = (K \times H) / 100, \quad (5.8)$$

где H - среднегодовая норма амортизации (20%);

K – сумма капитальных затрат, (тг).

Рассчитаем амортизационные отчисления для разработки автоматизированной теплицы:

$$A = (30395 \times 20) / 100 = 6080 \text{ (тг)}.$$

Так как создание теплицы невозможно без использования электрических приборов и устройств, а именно: паяльной станции, персонального компьютера, необходимо рассчитать затраты на электроэнергию (тарифы г. Алматы), они определяются по формуле:

$$\Xi = W \times \text{Ц} \times t, \quad (5.9)$$

где W - мощность оборудования;

C - тариф за 1 кВт / ч электроэнергии, потребляемой в Алматы (17,81 тенге) [1];

t - время работы оборудования на 1 месяц (240 часов).

Мощность паяльной станции W_1 - 100 Вт / час, [2]

Мощность персонального компьютера W_2 составляет 450 Вт / час. [3]

Мощность всего рассчитанного оборудования:

$$W = W_1 + W_2, \quad (5.10)$$

$$W = 100 + 450 = 0,55 \text{ кВт/ч.}$$

Рассчитаем затраты на электроэнергию:

$$\Xi = 17,81 \times 0,55 \times 240 = 2351 \text{ (тг)}.$$

Накладные расходы – это дополнительные затраты, не относящиеся напрямую к основному производству, не входящие в оплату труда основного персонала и в стоимость сырья. Для предприятия такие затраты важны не меньше, чем прямые расходы, поскольку позволяют обеспечить работу всего предприятия. Они состоят из расходов на охрану, освещение, отопление). Он равен 40% от общего фонда заработной платы и определяется по формуле:

$$H = Z_{\text{осн}} \times H_{\text{к}}, \quad (5.11)$$

где $H_{\text{к}}$ - норматив накладных расходов в организациях, %.

$$H = 100000 \times 0,4 = 40000 \text{ (тг)}.$$

После проведения всех необходимых расчетов, можно рассчитать общие затраты на разработку проекта:

$$C = 110000 + 8550 + 6080 + 510 + 3099 + 42000 = 167133 \text{ (тг)}.$$

Таблица 5.3 - Результаты стоимости для разработки теплицы:

Раздел затрат	Обозначение	Цена, теңге
Фонд заработной платы	Фот	110000
Социальный налог	$C_{\text{н}}$	8550
Материалы и детали	М	152
Расходы на электроэнергию	Ξ	2351
Накладные расходы	Н	40000
Амортизационные отчисления	А	6080
Итого		167133

5.3.3 Расчет себестоимости проекта

Себестоимость продукции – это сумма затрат на производство и реализацию (сбыт) этой продукции. Она складывается из затрат, связанных с использованием в процессе производства основных средств, сырья, материалов, топлива и энергии, труда работников и прочих затрат на производство и реализацию. Она определяется по формуле:

$$C_{\text{п}} = C + K, \quad (5.12)$$

Рассчитаем себестоимость автоматизированной теплицы:

$$C_{\text{п}} = 167133 + 30395 = 197528 \text{ (тг)}$$

Определим стоимость с наценкой (рентабельность). То есть уровень рентабельности (наценка) для телекоммуникационной отрасли колеблется от 20% до 40%.

$$\Pi_n = C + K \times (1 + P) / 100, \quad (5.13)$$

где P - выход (20% - 40%);
 Π_n – стоимость с наценкой.
 Следовательно:

$$\Pi_n = (167133 + 30395) \times (1 + 20 / 100) = 237033,6 \text{ (тг)}.$$

Далее стоимость продаж рассчитывается с учетом НДС.

$$\Pi_{\pi} = \Pi_n + \text{НДС}, \quad (5.14)$$

Поскольку налог на добавленную стоимость составляет 12% (2020):

$$\Pi_{\pi} = 237033,6 \times 1,12 = 265477,632 \text{ (тг)}.$$

Далее необходимо посчитать доход от продаж продукции. Он определяется по формуле:

$$K_t = N_k \times \Pi_t, \quad (5.15)$$

где N_k - количество проданных инструментов;
 Π_t - цена товара.

Следовательно, доход от продажи 1 единицы товара равен следующей стоимости:

$$K_t = 1 \cdot 265477,632 = 265477,632 \text{ (тг)}.$$

Чистая прибыль (также - чистый доход) — часть балансовой прибыли предприятия, остающаяся в его распоряжении после уплаты налогов, сборов, отчислений и других обязательных платежей в бюджет. То есть, чистую прибыль можно выразить как разность доходов и расходов:

$$\Pi_{\pi} = K_t - (C + K), \quad (5.16)$$

$$\Pi_{\pi} = 265477,632 - 197528 = 67949,632$$

Коэффициент экономической эффективности рассчитывается по формуле:

$$E = \Pi_{\pi} / K, \quad (5.17)$$

$$E=67949,632/30395 = 2,23$$

Срок окупаемости рассчитываем по формуле:

$$T = 1/E, \quad (5.18)$$

$T = 1/2,23 = 0,45$ лет, что равно примерно 5 месяцам.

6 Безопасность жизнедеятельности

6.1 Анализ условий труда

Представленная диссертация посвящена разработке теплицы, которая содержит не малое количество датчиков, приборов, ветрогенератора, системы управления, различных преобразователей, аккумуляторов, контроллера заряда, сетей. Рабочее место оператора - диспетчерский центр, с помощью которого настраиваются режимы работы ветрогенератора и контролируются параметры. Учитывайте вредные и опасные факторы, наблюдаемые в процессе установки.

Для анализа условий труда мы предлагаем помещение размером 7м * 11м., Площадь которого $S = 77 \text{ м}^2$. Высота $H = 3 \text{ м}$. Объем помещения $V = 231 \text{ м}^3$. В комнате естественное одностороннее освещение и один выход.

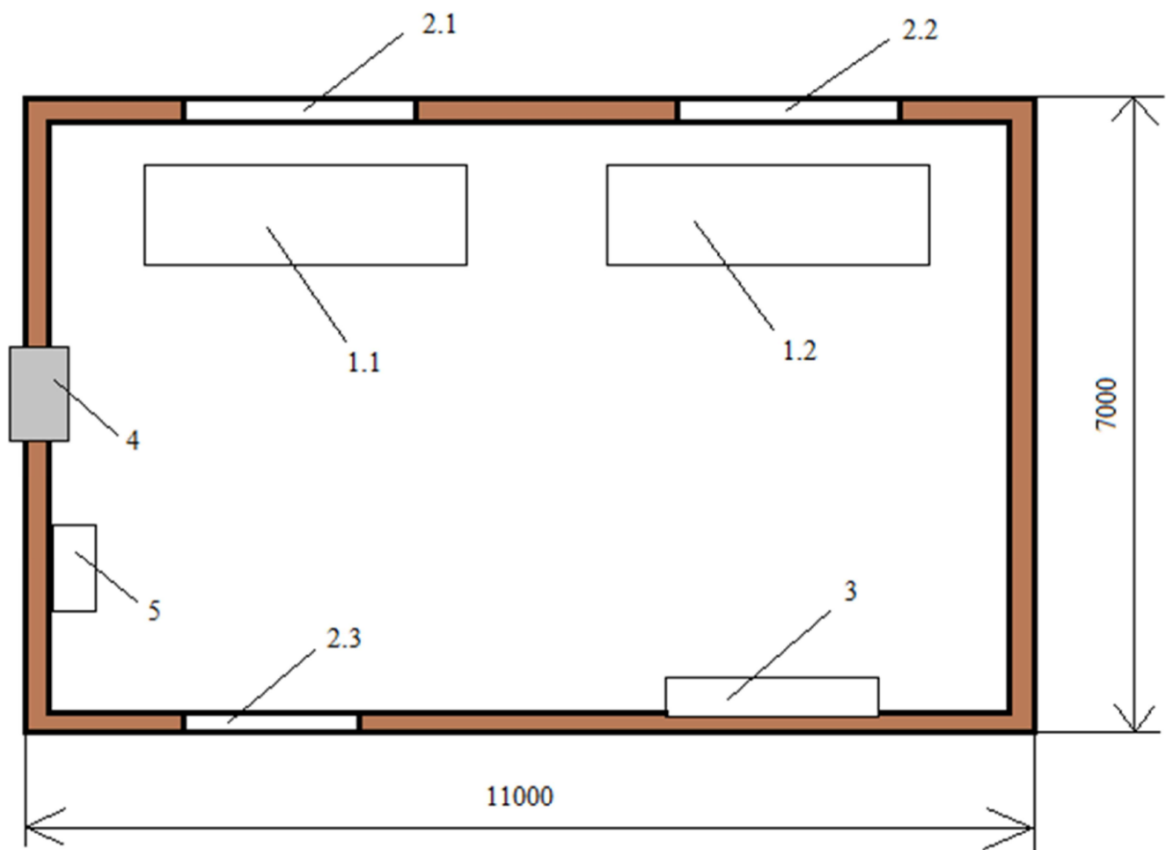
Б / у оборудование:

- персональный компьютер Lenovo;
- печатающие устройства типа HP Laser Jet Pr M426fdn;
- разные виды паяльников для микросхем.

Оборудование размещено таким образом, чтобы обеспечить беспрепятственный доступ ко всем рабочим местам. Основная работа в комнате связана с ПК и досками. Мы изучаем планировку и размещение рабочих мест. В соответствии с эргономическими требованиями ГОСТ 12.2.049-80, все производственное оборудование должно отвечать всем антропометрическим, физическим, психофизиологическим, эмоциональным свойствам человека, а также соответствующим гигиеническим требованиям для поддержания здоровья человека, снижения его утомляемости и повышения высокой производительности труда. Согласно ГОСТ 12.2.032-78 «Рабочее пространство, при ведении бизнеса сидя», при правильном использовании и размещении вышеперечисленного оборудования, с учетом конструкции рабочего места, размеры и взаимное расположение элементов процесса обеспечиваются, абсолютно Безопасность дел должна соответствовать требованиям эргономики и технической эстетики.

Высота рабочих станций, предназначенных для диагностики, составляет 850 мм в высоту, 1450 мм в длину, 800 мм в ширину, высота стула 450 мм над землей и высота помещения для ног 650 мм. Эти параметры позволяют создавать комфортные и физиологические условия для рабочего пространства. Уровень и структура рабочего стола были выбраны с учетом их рабочих функций, что позволяет легко переходить из сидячего состояния в стоячее (рисунок 4.1).

На самом деле необходимо указать, что только лицам, достигшим совершеннолетия и прошедшим специальную подготовку, разрешено вести дела, связанные с пайкой и лужением. Они обязаны совершенствовать аристократичность по критериям охраны труда, безвредные методы ведения дел, уметь правильно обращаться с инструментами.



1 – первое рабочее место; 1.2 – второе рабочее место; 2.1 – первое окно; 2.2 – второе окно; 2.3 – третье окно; 3 – пожарный щит; 4 – вентиляционная отдушина; 5 – силовой щит.

Рисунок 6.1 - Планировка помещения с указанием рабочих мест.

6.2 Условия безопасности при пайке и лужении.

Принципиально с высокой степенью точности общепризнанные стандарты техники безопасности. В неприятной ситуации это может нанести вред здоровью.

Должно быть указано, что это должно быть только индивидуально, отточено и специально обучено. Они проводят специальное исследование. Они отвечают требованиям совершенствования охраны труда, без обязательных методов ведения бизнеса. Если у работника возникают какие-либо проблемы во время пайки или на том же газоне, он должен немедленно связаться с боссом, а не пытаться решить дилемму самостоятельно.

В высшей степени принципиальная степень является основополагающей для соблюдения правил безопасности. К ним относятся: чрезмерное загрязнение газа химическими парами

вещества, пожароопасность, брызги воздуха и припой, температура воздуха рабочей зоны.

В работе использованы преимущественно качественные материалы и инструменты. Итак, вы должны соответствовать всем стандартам. Припой используются при пайке изделий из латуни, бронзы и меди. Ведущие лица проводят разумный инструктаж по работе с этими инвентарем.

Специальные испытания и проверки должны быть проведены. Условия

производства и строительные категории должны быть соблюдены. Все это должно быть защищено от случайных механических повреждений.

Работы, связанные с пайкой и газом, проводились заранее подготовленными и специально оборудованными помещениями. Вам нужна система вентиляции, звуковая и световая сигнализация.

Особое значение имеет подготовка рабочего пространства, которое должно быть оборудовано системой вентиляции и освещения с непрозрачными отражателями. Не рекомендуется паять и газон без использования специальных защитных очков. Осветительные приборы в глазах работника.

В предоставленной лаборатории используются паяльные станции YX852D + (рисунок 4.2). Цифровая индикация фактической температуры струи воздуха и наконечника пикальника позволяет добиться минимальных отклонений при эксплуатации. Индикатор состояния теплоносителя для дополнительного удобства во время работы

Широта функциональной температуры позволяет использовать это устройство по всем критериям. Станция оснащена четырьмя сменными насадками для собственного использования.

Керамическое нагревательное вещество обладает высокой способностью образовывать электрические компоненты. Практичный дизайн позволяет заменить жало без каких-либо дополнительных усилий.

Технические характеристики:

- напряжение питания паяльной станции, В: 220-240;
- приложенная мощность, Вт: 350;
- приложенная мощность паяльника, Вт: 50;



Рисунок 6.2 - Паяльная станция YX852D+

6.3 Противопожарная безопасность

Противопожарная безопасность регулируется ГОСТ 12.1.004-85

“Пожарная безопасность” и ГОСТ 12.1.010-85 “Взрывобезопасность. Совместные запросы.”.

Сообразно Приказа № 32 от 31.10.95 (введен 1.01.96) лаборатория ЭВМ принадлежит к помещениям категории “Д”, т.е. здание, содержащее огнестойкие препараты и материалы в прохладном состоянии.

Пожарная защищенность объекта в согласовании с ГОСТ 12.1.004-85 гарантируется системами предотвращения пожара и противопожарной обороны, учтены организационно – технические события.

Противопожарные системы должны характеризоваться уровнем противопожарной защиты людей и имущества.

В целях предотвращения противопожарной защиты принимаются соответствующие меры:

- вентиляция взрывозащищенного исполнения;
- строительство зданий из негорючих материалов;
- лаборатория оснащена переносным огнетушителем;
- есть пожарный гидрант в общем коридоре;
- в случае пожара существует план эвакуации;
- пожарная сигнализация содержит датчики ДИП - 215 ЗМЗ;
- замечать свет и звук;
- осведомленность персонала о пожарной безопасности.

В лаборатории, а также в коридоре учебного лабораторного корпуса имеются методы пожаротушения в непредвиденных ситуациях (огнетушители, ведра, лопаты, ящики с песком). Лаборатория оснащена порошковым огнетушителем марки ОПС-10, размещена в видимом пространстве и легко доступна для использования. Для различных помещений существуют общепринятые меры первых средств пожаротушения. На каждые 100 м² пола производственных помещений требуется 1-2 огнетушителя. В рабочем помещении установлена пожарная сигнализация с детектором тепла. Коридоры не оборудованы посторонними предметами, затрудняющими эвакуацию людей в случае пожара.

Расстояние от мастерской до пожарного гидранта составляет около 8 метров.

6.4 Эргономика рабочего места

В соответствии с ГОСТ 12.2.032 - 78 к психофизиологически небезопасным и вредным моментам в работе оператора компьютера можно отнести нервное и психическое состояние тела, вызванное недостаточным освещением и однообразием труда, и, например, ужасное оборудование рабочего пространства.

Эти моменты имеют все шансы стать предпосылкой нервно-психологических перегрузок. Исходя из этого, оборудование и приборы в лаборатории разрабатываются с точки зрения физических и психических данных человека. Были учтены умственные усилия работника, использующего созданное устройство, чрезмерная осторожность и физиологический стресс при работе с электронными приборами учета электроэнергии и счетчиками электроэнергии.

Все виды оборудования пригодны для использования. Размещение органов управления позволяет сохранить движение, устраняет неудобные усилия в положении тела.

Соответствующие запросы были разработаны в отношении рабочего пространства оператора компьютера:

- разумный выбор рабочей зоны;
- выбор условий труда;
- выбор более выгодного расположения материалов.

Основные элементы рабочего места показаны на рисунке 4.3.

Высота плоскости сиденья определяется высотой подколенной ямки над полом. Высота стула 400 мм. Высота рабочей плоскости 710 ± 5 мм считается хорошей. Монитор компьютера установлен на регулируемой подставке, что позволяет вводить его таким образом, чтобы обеспечить комфортный визуальный контроль и комфорт работы.

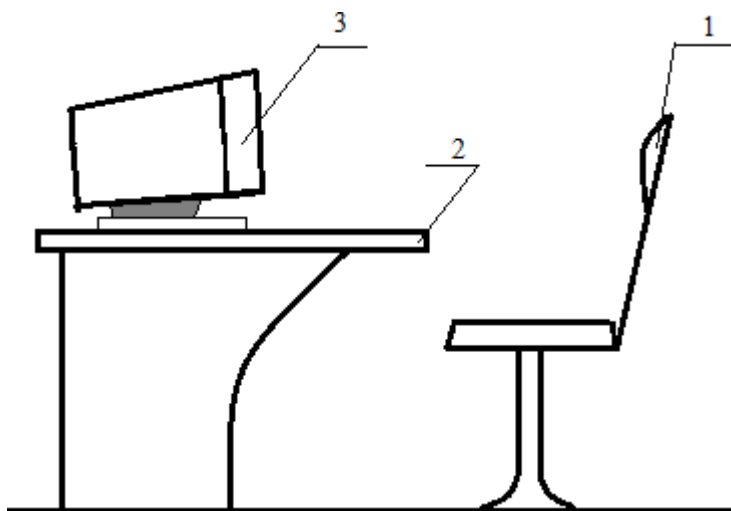


Рисунок 6.3 - Основные элементы рабочего места: 1 - рабочее кресло; 2 - рабочая поверхность; 3 – ЭВМ

Были приняты к сведению подбора проектирования прибора были предусмотрены надлежащие моменты:

- состояние тела оператора;
- месторасположение органов управления;
- величина и конфигурация органов управления;
- назначение, амплитуда и линия движения их движения;
- отношение величины движения ручек управления к величине движения указателя указателя и т.п.

Величина зоны приложения труда лимитируются площадью, снаряженной технологической оснасткой, инструментами и приспособлениями, в согласовании с ГОСТ 12.0.003 – 74 При месторасположении составляющих рабочего пространства учтены нужные способы обороны проектировщика от небезопасных и вредоносных моментов. Обоюдное месторасположение составляющих рабочего пространства содействует подходящему режиму труда и развлечений, понижению утомления, предупреждению возникновения

неверных поступков.

Выполнение движений оператором в границах подходящей зоны значительно уменьшает мышечное усилие. При формировании посадочной площадки за панелью управления предусмотрено, что поле обзора в горизонтальной плоскости без поворота головы занимает 1200, а поворот - 1300.

Приемлемый вертикальный угол обзора 1300. Удобное и рациональное расположение материалов, инструментов, устройств исключает ненужные, непроизводительные движения. Продукты, обработанные материалы и продукты размещаются на рабочем месте с учетом частоты их использования: наиболее часто используемые устройства находятся в пределах досягаемости рук без наклона тела, менее часто используемые устройства находятся в более отдаленной зоне.

Основная задача состоит в том, чтобы свести к минимуму количество несчастных случаев и травм во время эксплуатации изделия, рабочее пространство соответствует общепризнанным стандартам технической и противопожарной защиты, а проектировщик должен соблюдать общепризнанные стандарты и требования по охране труда и технике безопасности. не способствуют созданию чрезвычайных ситуаций.

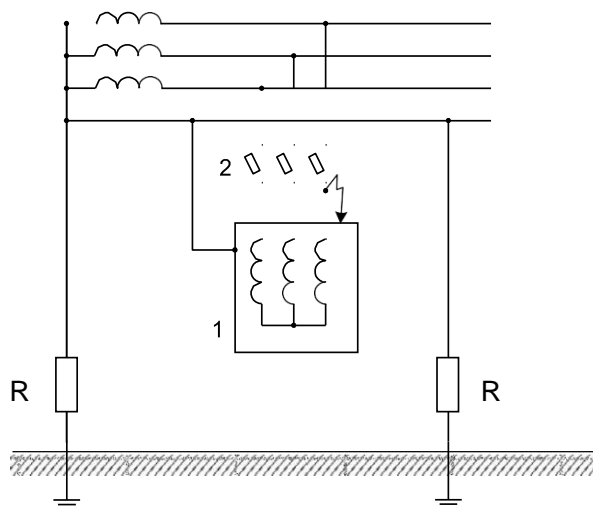
6.5 Расчет системы зануления

Для того чтобы работника не ударило электрическим током я рассчитала систему зануления, это наиболее действенный метод уберечь рабочий коллектив от удара электрическим током. И так защитное зануление полностью не ограждает от удара током, но может уменьшить его, прикоснувшись к корпусу, которое равно половине фазного напряжения U_{ϕ} , благодаря занулению автоматически отключается участок цепи на котором совершился пробой, сбой

Далее показан проект обнуление электрооборудования с нарицательным усилием 380 В. и нарицательным током 10 А.

Для этого для того чтобы электрооборудование питалось от силового трансформатора применяется красновато-желтый кабель калибр которого составляет $d=10$ мм (сечением $S=78,54$ мм²). Клиент подсоединен к трансформатору 11 килоВ·А., длина участка цепи $L=0,1$ км. Нулевой кабель состоит из металлической покрышки разрезом $S_{\text{ст}}=20 \cdot 4$ мм². Расстояние прокладки его равна $D=50$ см вдали от провода. Схема подключения выпрямителя представлена на рисунке 4.2.

Эти моменты имеют все шансы стать предпосылкой нервно-психологических перегрузок. Исходя из этого, оборудование и приборы в лаборатории разрабатываются с точки зрения физических и психических данных человека. Были учтены умственные усилия работника, использующего созданное устройство, чрезмерная осторожность и физиологический стресс при работе с электронными приборами учета электроэнергии и счетчиками электроэнергии.



Tr – насыщающий преобразователь; R_0 – противодействие заземления нейтрали;

R_{Π} – противодействие повторного заземления; 1 – блок-корпус; 2 – механические выключатели

Рисунок 6.4 – Схема зануления оборудования

Вычисление сближается к нахождению критерий, при каких поддерживаются стремительно включение предельно-токовой предохранения и выключение испорченной конструкции от сети. С целью достоверного отключения испорченной монтажа следует, для того чтобы электроток в коротко замкнутой цепочки существенно превышал электроток монтажа автомата, т.е.

$$I_{K3} \geq k \cdot I_{НОМ}, \quad (6.1)$$

где k – это коэффициент.

Теперь испытаем концепцию обнуления в отключ. Умение.

Прогнозируемый электроток кратковременного $I_{K3} \geq 1,4 \cdot I_{НОМ}$ Его номинальная мощность равна:

$$P_{НОМ} = 3 \cdot U_{Л} \cdot I_{Л}, \quad (6.2)$$

Номинальный ток определим как:

$$I_{НОМ} = P_{Н} / 3 \cdot U_{\phi} = 11 \cdot 10^3 / 3 \cdot 380 = 9,65 \text{ (A)}, \quad (6.3)$$

Прогнозируемый электроток кратковременного равен:

$$I_{K30} \geq 1,4 \cdot 9,65 = 13,5 \text{ (A)}, \quad (6.4)$$

Аппарат охраны на пятнадцать А.

Найдем противодействие трансформатора Z_T при схеме объединения обмоток: D/Y_H $Z_T = 0,906 \text{ Ом}$.

Сопротивление фазных проводов можно определить по формуле:

$$R_{\Phi} = \rho * L / S, \quad (6.5)$$

где L - длина кабеля, м;
 S - разрез кабеля, мм²;
 ρ - удельное сопротивление использованного материала (с целью меди $\rho = 0,018 \text{ Ом}$).

$$R_{\Phi} = 0,018 * 100 / 78,54 = 0,023 \text{ (Ом)}, \quad (6.6)$$

Индуктивное сопротивление фазного кабеля $X_{\Phi} = 0$.

Плотность тока в нулевом проводнике изготовленный из стальной шины имеющий размеры $S_{\text{ст}} = 20 \cdot 4 \text{ мм}$ (сечением $S = 80 \text{ мм}^2$):

$$\delta = I_{\text{кз0}} / S_{\text{НСТ}} = 13,5 / 80 = 0,169 \text{ (А/мм}^2\text{)}, \quad (6.7)$$

Присутствие этой густоты тока я найду активное сопротивление $R_{\omega} = 5,24 \text{ Ом/км}$ и индуктивное сопротивление $X_{\omega} = 3,14 \text{ Ом/км}$.

Рассчитаем сопротивление нулевого проводника (длина которого $L = 0,3 \text{ км}$):

$$R_{\text{H}} = R_{\omega} * L = 5,24 * 0,3 = 0,524 \text{ (Ом)}, \quad (6.8)$$

Рассчитаем индуктивное сопротивление нулевого проводника:

$$X_{\text{H}} = X_{\omega} * L = 3,14 * 0,3 = 0,314 \text{ (Ом)}, \quad (6.9)$$

Линия длиной один км, с частотой тока равной 50 Гц, наружное индуктивное сопротивление силуэта «стадия-ноль» равно:

$$X = 0,1256 * \lg(2 * D / d) = 0,1256 * \lg(2 * 500 / 10) = 0,578 \text{ Ом/км}, \quad (6.10)$$

Наружное индуктивное сопротивление силуэта $X_{\text{в}}$ «стадия-ноль» определим как:

$$X_{\text{в}} = L * X_{\text{вЛ}} = 0,3 * 0,578 = 0,578 \text{ (Ом)}, \quad (6.11)$$

Определим полное сопротивление цепи «фаза-нуль»:

$$Z_{\text{кз}} = Z_{\Phi} + Z_{\text{H}} + j * X_{\text{П}} = (R_{\Phi} + R_{\text{H}}) + j * (X_{\Phi} + X_{\text{H}} + X_{\text{в}}), \quad (6.12)$$

$$Z_{\text{кз}} = (0,023 + 0,524) + j * (0 + 0,314 + 0,0578) = 0,547 + j * 0,372$$

Далее определим модуль полного сопротивления цепи «фаза-нуль»:

$$|Z_{\text{кз}}| = \sqrt{(R_{\Phi} + R_{\text{H}})^2 + (X_{\Phi} + X_{\text{H}} + X_{\text{в}})^2}, \quad (6.13)$$

$$|Z_{\text{кз}}| = \sqrt{(0,023 + 0,524)^2 + (0 + 0,314 + 0,0578)^2} = \sqrt{0,3} + 0,14 = 0,66 \text{ (Ом)}$$

Действительный ток КЗ можно определить по формуле:

$$I_{кз}=U_{ф}/(Z_T/3+ Z_{кз})=380/(0,906/3+0,66)=395 \text{ (А)}, \quad (6.14)$$

Согласно приобретенным вычислениям очевидно то что, приобретенный электроток КЗ более прогнозируемого $IKЗ>1,4*INOM$ (395,13,5 А). В таком случае, то что я приобрели с поддержкой расчетов демонстрирует то что, требование срабатывания охраны осуществится и отключится монтаж с узы.

6.6 Расчет кондиционирования в лабораторном помещении.

Дабы исполнить расплата кондиционирования, нужно аристократия численность тепла, поступающего в комнату. Эти данные просто определить, в случае если счесть высоту и площадь комнаты, численность поступающего с улицы света в комнате, численность людей, живущих в жилплощади, численность офисных и домашних устройств, обогревательных и осветительных устройств.

Модель кондиционера в этом случае обязана владеть подобной или же немногим большей мощностью.

- паяльная станция – 1 ($Q = 350 \text{ Вт}$);
- подвесная электро отвертка – 1 ($Q = 30 \text{ Вт}$);
- ноутбук – 1 ($Q = 100 \text{ Вт}$);
- светильник – 3 ($Q = 40 \text{ Вт}$).

Для выполнения расчета кондиционирования можно воспользоваться простой формулой 4.9:

$$Q_{\text{общ}}= T_1+T_2 +T_3, \quad (6.12)$$

где: $T_1 = S * h * k$ (S – это площадь помещения,

h – высота от пола до потолка,

k – коэффициент, который равняется 30 Вт для умеренного освещения, 35 Вт для среднего освещения, 40 Вт для ярко освещенного помещения, формула 4.10).

$$T_1=77*3*30=6930 \text{ (Вт)} \quad (6.13)$$

$T_2= T_{\text{ср}} * N$ ($T_{\text{ср}}$ – количество тепла, которое выделяется человеком и оно колеблется от 130 Вт до 440 Вт, в зависимости от количества движения. N – количество людей, формула 4.11).

$$Q_2=130*1=130 \text{ (Вт)} \quad (6.14)$$

$T_3= Q_1+Q_2+...Q_n$ (Q – 30% мощности от каждого прибора. n – количество приборов, формула 4.12).

$$Q_3=6930+130+70+28*3=7158 \text{ (Вт)} \quad (6.15)$$

Расчет кондиционирования измеряется в особой единице – BTU. Кол-

во BTU равна 0,293 Вт.

Для верного выбора системы кондиционирования нужно еще принимать во внимание сплошное численность комнат и их назначение.

Выполнив расплата кондиционирования, подыскиваем необходимый по мощности кондиционер. Кондиционер с мульти сплит системой OTEX OWM- 24RN. Свойства:

- кол-во комнат – 2;
- обслуживаемая площадь – 80 ;
- режим подогрева – есть;
- мощность – 7500 Вт.



Рисунок 6.5 - OTEX OWM-24RN

Произведя расчеты основных факторов, влияющих на безопасность, надежность, комфортность при работе с ветроустановкой от самого начала этапа ее создания - проектирования, можно сделать следующие выводы: при проектировании системы были необходимы безопасные и комфортные условия труда для оператора. В связи с этим проведен анализ общего микроклимата , а также расчет оптимального режима освещения при работе над проектированием системы.

Заключение

В данной дипломной работе рассмотрена перспектива сооружения теплиц, в которых соблюдение правил выращивания растений соблюдается с помощью системы приборов и датчиков.

В первую очередь был произведен обзор и выбор аппаратной базы для дальнейшей разработки системы автоматизации теплицы с возможностью мониторинга и управления через сеть Интернет. В ходе данного обзора были рассмотрены достоинства и недостатки таких решений как микроконтроллеры, Ардуино и одноплатные компьютеры, после чего был сделан выбор в пользу последних, по причине получения сразу «все в одном»: базу для подключения датчиков, производительную вычислительную систему, сервер с ОС Linux, доступ через Интернет в случае невозможности использования реальных IP-адресов и доменных имен.

После выбора аппаратной базы были описаны используемые датчики и другое периферийное оборудование для мониторинга показателей теплицы, а также варианты алгоритмизации создания микроклимата на основе этих показателей.

Ключевым результатом выпускной квалификационной работы является построение автоматизированной системы с веб-интерфейсом с помощью Raspberry Pi, который объединил ранее написанные программы на языке Python для считывания показаний с датчиков, позволил реализовать наглядные алгоритмы создания микроклимата, а также использовался для построения веб-приложения с графиками и интерфейсом для мониторинга и управления оборудованием в режиме online.

Список литературы

- 1 Интернет вещей готовится к земле. Новости Интернета вещей – <https://iot.ru/selskoe-khozyaystvo/internet-veshchey-gotovitsya-k-zemle> (дата обращения: 20.02.2017).
- 2 Википедия – свободная энциклопедия // [https://en.wikipedia.org/wiki/Data_collection_system]/«Data collection system» [дата обращения – 27.03.2020]
- 3 Cyber Security Intelligence // [<https://www.cybersecurityintelligence.com/blog/car-hacking-and-data-collection-4178.html>] / «Car Hacking & Data Collection» [дата обращения – 27.03.2020]
- 4 Анатомия армии // [<http://army.armor.kiev.ua/hist/blackbox.shtml>] / «Черный ящик» [дата обращения – 27.03.2020]
- 5 ТУСУР // [https://storage.tusur.ru/files/10865/10865/Статья_по_ГПО_Поручиков_Подгородецкий.pdf] / «Устройство сбора информации с датчиков автомобиля» [дата обращения – 27.03.2020]
- 6 vc.ru – бизнес, технологии, идеи, модели роста, стартапы // [<https://vc.ru/flood/12170-telematics>] / «Интернет машин: 11 стартапов для сбора данных о движении и состоянии автомобилей» [дата обращения – 27.03.2020]
- 7 Legal IT Group // [<https://legalitgroup.com/ru/regulirovanie-voprosov-bezopasnosti-svyazannyh-s-sistemami-podklyuchennyh-avtomobilej-i-dannyh-poluchennyh-iz-nih/>] / «GDPR и персональные данные» [дата обращения – 27.03.2020]
- 8 Системы современного автомобиля// [<http://systemsauto.ru/active/active.html>] / «Системы активной безопасности» [дата обращения – 27.03.2020]
- 9 ТехАвтоПорт // [<https://techautoport.ru/sistemy-bezopasnosti/aktivnaya-sistema-aktivnoy-bezopasnosti-avtomobilya.html>] / «Описание и функции системы активной безопасности автомобиля» [дата обращения – 27.03.2020]
- 10 Новостной сайт KP.RU // [<https://www.kp.ru/guide/sistema-kontrolja-avtomobilja.html>] / «Оригинал статьи: <https://www.kp.ru/guide/sistema-kontrolja-avtomobilja.html>» [дата обращения – 27.03.2020]
- 11 Новостной сайт KP.RU // [<https://www.kp.ru/guide/kontrol-avtomobil-nogo-transporta.html>] / «Оригинал статьи: <https://www.kp.ru/guide/kontrol-avtomobil-nogo-transporta.html>» [дата обращения – 27.03.2020]
- 12 Drom // [<https://www.drom.ru/info/misc/22676.html>] / «Видеорегистраторы» [дата обращения – 27.03.2020]
- 13 Habr.com // [<https://habr.com/ru/news/t/451526/>] / ««Глонасс»: для безопасности транспорта нужна единая система хранения данных об автомобилях» [дата обращения – 27.03.2020]
- 14 Википедия – свободная энциклопедия // [https://ru.wikipedia.org/wiki/Распберри_Pi] / «Распберри Pi» [дата обращения – 01.04.2020]
- 15 RaspberriPi // [<https://www.Распберриpi.org/documentation/hardware/camera/>] / «Camera Module» [дата обращения – 01.04.2020]

16 ELM327ODB2 // [<https://elm327-obd2.ru/obzory/elm327-bluetooth-mini.html>] / «Обзор ELM327 Bluetooth 1.5» [дата обращения – 01.04.2020]

17 Ex-hort.ru // [<https://ex-hort.ru/Рaspбepри-pi-command>] / «Список команд Рaspбepри Рi» [дата обращения – 01.04.2020]

ПРИЛОЖЕНИЕ А

Исходные тексты программ

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import MySQLdb #Bibliotheque pour base de donnees
import serial #Bibliotheque pour la communication serie
import time #Bibliotheque pour le delai

try:
    ser = serial.Serial('/dev/ttyACM0', 9600,timeout=0.5)
except:
    ser = serial.Serial('/dev/ttyACM1', 9600,timeout=0.5)

time.sleep(4) #on attend un peu, pour que l'Arduino soit prêt.

Compteur=0

'''Initialisation au demarrage'''
def RAZ():
    db = OuvertureDB()
    curs = db.cursor()

    curs.execute ("UPDATE commandes SET
Etat='0',NombreDem='0',Mode='0',ComManu='0' WHERE
Equipement='Lampe_infrarouge'")
    curs.execute ("UPDATE commandes SET
Etat='0',NombreDem='0',Mode='0',ComManu='0' WHERE Equipement='Pompe'")
    curs.execute ("UPDATE commandes SET
Etat='0',NombreDem='0',Mode='0',ComManu='0' WHERE
Equipement='Servomoteur'")
    curs.execute ("UPDATE commandes SET
Etat='0',NombreDem='0',Mode='0',ComManu='0' WHERE Equipement='Vanne'")
    curs.execute ("UPDATE commandes SET
Etat='0',NombreDem='0',Mode='0',ComManu='0' WHERE
Equipement='Ventilateur'")

    curs.execute ("UPDATE types SET Consigne='40',DeltaT='1' WHERE
Type='Temperature_int'")
    curs.execute ("UPDATE types SET SeuilBas='20',SeuilHaut='60' WHERE
Type='Humidite'")
```

```
curs.execute ("TRUNCATE TABLE mesures")
```

```
ser.write('CPRESET')
```

```
db.commit()
```

```
FermetureDB(db)
```

```
"""Analyse de l'arduino"""
```

```
def AnalyseArduino():
```

```
    MajDBMesures('TPEXT','Temperature_ext')
```

```
    MajDBMesures('TPINT','Temperature_int')
```

```
    MajDBMesures('HUMID','Humidite')
```

```
    MajDBMesures('LUMIN','Luminosite')
```

```
ComLampePrec,ComPompePrec,ComServoPrec,ComVannePrec,ComVentiPrec,Mo  
deLampePrec,ModePompePrec,ModeServoPrec,ModeVannePrec,ModeVentiPrec =  
LectureEquiDB()
```

```
    SPPompePrec,DeltaPompePrec,SeuilBasHumPrec,SeuilHautPrec =  
LectureTypeDB()
```

```
    MajDBEquip('LAMPEET','Lampe_infrarouge')
```

```
    MajDBEquip('POMPEET','Pompe')
```

```
    MajDBEquip('SERVOET','Servomoteur')
```

```
    MajDBEquip('VANNEET','Vanne')
```

```
    MajDBEquip('VENTIET','Ventilateur')
```

```
ComLampePrec,ComPompePrec,ComServoPrec,ComVannePrec,ComVentiPrec,Mo  
deLampePrec,ModePompePrec,ModeServoPrec,ModeVannePrec,ModeVentiPrec =  
LectureEquiDB()
```

```
    SPPompePrec,DeltaPompePrec,SeuilBasHumPrec,SeuilHautPrec =  
LectureTypeDB()
```

```
    MajCompteur('LAMPEET','Lampe_infrarouge')
```

```
    MajCompteur('POMPEET','Pompe')
```

```
    MajCompteur('SERVOET','Servomoteur')
```

```
    MajCompteur('VANNEET','Vanne')
```

```
    MajCompteur('VENTIET','Ventilateur')
```

```
"""Mise a jour des etats et des compteurs de chaque equipement"""
```

```
def MajDBEquip(Commande,Equip):
```

```
    db = OuvertureDB()
```

```
    curs = db.cursor()
```

```

while True:
    try:
        ser.write(Commande)
        #time.sleep(0.25)
        RetCom=str(ser.readline())

        curs.execute("UPDATE commandes SET Etat=%s WHERE
Equipement=%s",(int(RetCom),Equip))
        if Equip == 'Ventilateur' or Equip == 'Servomoteur':
            print 'Mise à jour réussie du {0}'.format(Equip)
        else:
            print 'Mise à jour réussie de {0}'.format(Equip)
        break
    except:
        if Equip == 'Ventilateur' or Equip == 'Servomoteur':
            print 'Mise à jour du {0} : échec'.format(Equip)
        else:
            print 'Mise à jour de {0} : échec'.format(Equip)

db.commit()
FermetureDB(db)

''' Mise à jour des mesures de l'arduino'''
def MajDBMesures(Commande,Mesure):
    db = OuvertureDB()
    curs = db.cursor()

    ValId=0

    if Measure == 'Temperature_ext':
        ValId=1
    elif Measure == 'Temperature_int':
        ValId=2
    elif Measure == 'Luminosite':
        ValId=3
    elif Measure == 'Humidite':
        ValId=4

while True:
    try:
        while True: # ''' MODIF VINCENT'''
            ser.write(Commande)
            #time.sleep(0.25)
            RetCom2=str(ser.readline())

```

```

        if not RetCom2=="": break

        curs.execute ("INSERT INTO mesures(IdType,Date,Valeur)
VALUES(%s,%s,%s)",(ValId,time.strftime('%y/%m/%d
%H:%M:%S',time.localtime()),RetCom2))
        if ValId==4:
            print 'Enregistrement de 1 {0}'.format(Mesure)
            print(RetCom2)
        else:
            print 'Enregistrement de la {0}'.format(Mesure)
            print(RetCom2)
        if RetCom2!=0:
            break
    except:
        print('Erreur')

    db.commit()
    FermetureDB(db)

def MajCompteur(Commande,Equip):
    db = OuvertureDB()
    curs = db.cursor()

    while True:
        try:
            ser.write(Commande)
            #time.sleep(0.25)
            RetCom=int(ser.readline())

            curs.execute("UPDATE commandes SET NombreDem=%s WHERE
Equipement=%s",(RetCom,Equip))
            if Equip == 'Ventilateur' or Equip == 'Servomoteur':
                print 'Mise à jour du compteur du {0} réussie'.format(Equip)
            else:
                print 'Mise à jour du compteur de la {0} réussie'.format(Equip)
            break
        except:
            if Equip == 'Ventilateur' or Equip == 'Servomoteur':
                print 'Mise à jour du compteur du {0} : échec'.format(Equip)
            else:
                print 'Mise à jour du compteur de la {0} : échec'.format(Equip)
    db.commit()
    FermetureDB(db)

```

'''Fonction d'ouverture du lien avec la base de donnees'''


```

def OuvertureDB():
    db = MySQLdb.connect(host = "localhost",
                          user = "root",
                          passwd = "root",
                          db = "db_basilic")

    return db

'''Fonction de fermeture du lien avec la base de donnees'''
def FermetureDB(db):
    db.close

'''Fonction de lecture des commandes des equipements dans la base de donnees'''
def LectureEquiDB():
    db = OuvertureDB()

    curs = db.cursor()

    curs.execute ("SELECT * FROM commandes")

    Lecture = 0

    for reading in curs.fetchall():
        if Lecture == 0:
            ComLampe = reading[4]
            ModeLampe = reading[3]
        elif Lecture == 1:
            ComPompe = reading[4]
            ModePompe = reading[3]
        elif Lecture == 2:
            ComServo = reading[4]
            ModeServo = reading[3]
        elif Lecture == 3:
            ComVanne = reading[4]
            ModeVanne = reading[3]
        elif Lecture == 4:
            ComVenti = reading[4]
            ModeVenti = reading[3]
        Lecture = Lecture + 1

    FermetureDB(db)

    return
ComLampe,ComPompe,ComServo,ComVanne,ComVenti,ModeLampe,ModePompe,

```

ModeServo,ModeVanne,ModeVenti

""Detection changement d'etat d'un equipement""

def

DetectionFrontEquip(ComLampePrec,ComPompePrec,ComServoPrec,ComVannePrec,ComVentiPrec,ModeLampePrec,ModePompePrec,ModeServoPrec,ModeVannePrec,ModeVentiPrec):

ComLampeActu,ComPompeActu,ComServoActu,ComVanneActu,ComVentiActu,ModeLampeActu,ModePompeActu,ModeServoActu,ModeVanneActu,ModeVentiActu = LectureEquiDB()

ACK=""

""Si changement mode equipement""

if ModeLampePrec != ModeLampeActu:

while True:

try:

ser.write('LAMPEMANU')

time.sleep(0.25)

ACK=str(ser.readline())

if not ACK=="OK": break

except:

""""

print('LAMPEMANU')

ACK=""

if ModePompePrec != ModePompeActu:

while True:

try:

ser.write('POMPEMANU')

time.sleep(0.25)

ACK=str(ser.readline())

if not ACK=="OK": break

except:

""""

print('POMPEMANU')

ACK=""

if ModeServoPrec != ModeServoActu:

while True:

try:

ser.write('SERVOMANU')

time.sleep(0.25)

ACK=str(ser.readline())

if not ACK=="OK": break

except:

```

        """
print('SERVOMANU')
ACK=""
if ModeVannePrec != ModeVanneActu:
    while True:
        try:
            ser.write('VANNEMANU')
            time.sleep(0.25)
            ACK=str(ser.readline())
            if not ACK=="OK": break
        except:
            """

print('VANNEMANU')
ACK=""
if ModeVentiPrec != ModeVentiActu:
    while True:
        try:
            ser.write('VENTIMANU')
            ACK=str(ser.readline())
            if not ACK=="OK": break
        except:
            """

print('VENTIMANU')
ACK=""

"Si demande commande manu"
if ComLampePrec != ComLampeActu:
    while True:
        try:
            ser.write('LAMPEON')
            ACK=str(ser.readline())
            if not ACK=="OK": break
        except:
            """

print('LAMPEON')
ACK=""
if ComPompePrec != ComPompeActu:
    while True:
        try:
            ser.write('POMPEON')
            ACK=str(ser.readline())
            if not ACK=="OK": break
        except:
            """

print('POMPEON')

```

```

    ACK=""

    if ComServoPrec != ComServoActu:
        while True:
            try:
                ser.write('SERVOON')
                ACK=str(ser.readline())
                if not ACK=="OK": break
            except:
                """"

        print('SERVOON')
        ACK=""

    if ComVannePrec != ComVanneActu:
        while True:
            try:
                ser.write('VANNEON')
                ACK=str(ser.readline())
                if not ACK=="OK": break
            except:
                """"

        print('VANNEON')
        ACK=""

    if ComVentiPrec != ComVentiActu:
        while True:
            try:
                ser.write('VENTION')
                ACK=str(ser.readline())
                if not ACK=="OK": break
            except:
                """"

        print('VENTION')
        ACK=""

    return
ComLampeActu,ComPompeActu,ComServoActu,ComVanneActu,ComVentiActu,ModeLampeActu,ModePompeActu,ModeServoActu,ModeVanneActu,ModeVentiActu

```

"Fonction de lecture des types dans la base de donnees "

```

def LectureTypeDB():
    db = OuvertureDB()

    curs = db.cursor()

```

```

curs.execute ("SELECT * FROM types")

Lecture = 0

for reading in curs.fetchall():
    if Lecture == 1:
        SPPompe = reading[3]
        DeltaPompe = reading[4]
    elif Lecture == 3:
        SeuilBasHum = reading[5]
        SeuilHautHum = reading[6]

    Lecture = Lecture + 1

FermetureDB(db)

return SPPompe,DeltaPompe,SeuilBasHum,SeuilHautHum

"Detection changement d'etat d'une donnee de type"

def
DetectionFrontType(SPTempPrec,DeltaTempPrec,SeuilBasHumPrec,SeuilHautPrec):

    ACK=""

    SPTempActu,DeltaTempActu,SeuilBasHumActu,SeuilHautActu =
    LectureTypeDB()

    "Si changement donnee type"

    if SPTempPrec != SPTempActu:
        while True: # " MODIF VINCENT"
            try:
                ser.write("TEMMO%s#" % SPTempActu)
                time.sleep(0.25)
                ACK=str(ser.readline())
                if not ACK=="OK": break
            except:
                """"

        print "TEMMO%s#" % SPTempActu
        ACK=""
    if DeltaTempPrec != DeltaTempActu:
        while True: # " MODIF VINCENT"
            try:
                ser.write("DELTAT%s#" % DeltaTempActu)

```

```

        time.sleep(0.25)
        ACK=str(ser.readline())
        if not ACK=="OK": break
    except:
        """"

    print "DELTAT%s#" % DeltaTempActu
    ACK=""
if SeuilBasHumPrec != SeuilBasHumActu:
    while True: # "" MODIF VINCENT""
        try:
            ser.write("HUMBS%s#" % SeuilBasHumActu)
            time.sleep(0.25)
            ACK=str(ser.readline())
            if not ACK=="OK": break
        except:
            """"

    print "HUMBS%s#" % SeuilBasHumActu
    ACK=""
if SeuilHautPrec != SeuilHautActu:
    while True: # "" MODIF VINCENT""
        try:
            ser.write("HUMHT%s#" % SeuilHautActu)
            time.sleep(0.25)
            ACK=str(ser.readline())
            if not ACK=="OK": break
        except:
            """"

    print "HUMHT%s#" % SeuilHautActu
    ACK=""

```

```

    return SPTempActu,DeltaTempActu,SeuilBasHumActu,SeuilHautActu

```

```

""Initialisation""
print('Initialisation equipement')

```

```

RAZ()
AnalyseArduino()

```

```

ComLampePrec,ComPompePrec,ComServoPrec,ComVannePrec,ComVentiPre
c,ModeLampePrec,ModePompePrec,ModeServoPrec,ModeVannePrec,ModeVentiPr
ec = LectureEquiDB()

```

```

    SPPompePrec,DeltaPompePrec,SeuilBasHumPrec,SeuilHautPrec =
LectureTypeDB()
reset=0

```

```

while True:
    if Compteur == 5:
        AnalyseArduino()
        Compteur = 0
    if time.strftime('%H:%M',time.localtime()) == "00:00" and reset==0: #modif
Vincent
        ser.write('CPRESET')
        reset=1
    if time.strftime('%H:%M',time.localtime()) == "00:01" and reset==1:
        reset=0

ComLampePrec,ComPompePrec,ComServoPrec,ComVannePrec,ComVentiPrec,Mo
deLampePrec,ModePompePrec,ModeServoPrec,ModeVannePrec,ModeVentiPrec =
DetectionFrontEquip(ComLampePrec,ComPompePrec,ComServoPrec,ComVannePr
ec,ComVentiPrec,ModeLampePrec,ModePompePrec,ModeServoPrec,ModeVannePr
ec,ModeVentiPrec)
    SPPompePrec,DeltaPompePrec,SeuilBasHumPrec,SeuilHautPrec =
DetectionFrontType(SPPompePrec,DeltaPompePrec,SeuilBasHumPrec,SeuilHautPre
c)
    Compteur = Compteur +1
    time.sleep(1)

```

ПРИЛОЖЕНИЕ Б

```
#include<Servo.h>
Servoservomoteur;
//Declarationsdesfonctions
boolset(bool);
booltemporaire=false;
boolstoprefroidissement=false;
//Initialiserdesvariables
inttpscycle=10;//tempsdecycledelabouclearduino
inttpsvanne=0;//timerd'ouverturedel'electrovanne
inttpsfdc=1000;//tempsavantFDCvanneok-simulation
intvitserial=9600;//vitessebaudrate
boolfdcvanne=0;//booleanetatvanne
charlecture=0;//lecturedusérie
Stringrequete="";//enregistrelacommandedanscettevariable
intlect_temp_int=0;//lecturetempératureintérieureserre
intlect_temp_ext=0;//lecturetempératureextérieure
intlect_humidite=0;//lecturetauxd'humidité
intlect_luminosite=0;//lectureluminosité
inttemp_int=0;//températureintérieureserre
inttemp_ext=0;//températureextérieure
inthumidite=0;//tauxd'humidité
intluminosite=0;//luminosité
inthumbas=20;//limitebasd'humidite
inthumhaut=60;//limitehautd'humidite
Stringvartempo;//variabletemporairepourl'humidité
//régulationtempérature
inttempmoyenne=20;//consignetempératuremoyenne
intdeltat=5;//de la T° à la consigne
//commandelogiquedel'arduino
boolcmdlog5=0;//pompe
boolcmdlog6=0;//ventilateur
boolcmdlog7=0;//lampeIR
boolcmdlog8=0;//servomoteur
boolcmdlog9=0;//e-v
//resultatdecommande
boolresultcmd5=0;//pompe
boolresultcmd6=0;//ventilateur
boolresultcmd7=0;//lampeIR
boolresultcmd8=0;//servomoteur
boolresultcmd9=0;//e-v
//memoiredel'étatprécédentdurésultatdecommande
boolfmmemcmd5=0;//pompe
```



```

boolfmmemcmd6=0;//ventilateur
boolfmmemcmd7=0;//lampeIR
boolfmmemcmd8=0;//servomoteur
boolfmmemcmd9=0;//e-v
boolfmmemrefroid=0;//refroidissement
//flancmontantdurésultatdecommande
boolfmcmd5=0;//pompe
boolfmcmd6=0;//ventilateur
boolfmcmd7=0;//lampeIR
boolfmcmd8=0;//servomoteur
boolfmcmd9=0;//e-v
boolfmrefroid=0;//refroidissement
//compteurpourflancmontant
intcomptfm5=0;//pompe
intcomptfm6=0;//ventilateur
intcomptfm7=0;//lampeIR
intcomptfm8=0;//servomoteur
intcomptfm9=0;//e-v

//modemanuel
boolmanucmd5=0;//pompe
boolmanucmd6=0;//ventilateur
boolmanucmd7=0;//lampeIR
boolmanucmd8=0;//servomoteur
boolmanucmd9=0;//e-v
//ModeON
boolcmdfor5=0;//pompe
boolcmdfor6=0;//ventilateur
boolcmdfor7=0;//lampeIR
boolcmdfor8=0;//servomoteur
boolcmdfor9=0;//e-v
//servomoteur
boolfdcfenetre=0;//booleanetatfenetre
intsensservo=0;
intdelay servo=440;
intetattemperature=0;
voidsetup(){
//permetdedéfinirlemodesérieetlebaudratedel'arduino
Serial.begin(9600);
//Initialiserlessorties
pinMode(2,OUTPUT);//pompe
pinMode(3,OUTPUT);//ventilateur
pinMode(4,OUTPUT);//lampe
pinMode(6,OUTPUT);//electrovanne
pinMode(7,OUTPUT);//imagefdcvanne

```

```

pinMode(8,INPUT);//fdcfenetre
servomoteur.attach(5);
servomoteur.write(90);
delay(500);
}
voidloop(){
//*****Partiel
ectureentréesanalogique
    lect_temp_int=analogRead(0);
    temp_int=((float(lect_temp_int)*0.004882)-0.50)*100;
    lect_temp_ext=analogRead(1);
    temp_ext=((float(lect_temp_ext)*0.004882)-0.50)*100;
    lect_humidite=analogRead(2);
    humidite((((float)lect_humidite)/950)*100);
    lect_luminosite=analogRead(3);
    luminosite((((float)lect_luminosite)/10);
//*****Partiel
ectureentréesdigitale
    fdcfenetre=digitalRead(8);
//*****Partiel
ecturesérie
    if(Serial.available()>0)
    {
        lecture=Serial.read();
        if(lecture!=10)
        {
            requete.concat(lecture);
        }
    }
//*****Intepr
étationdelacommande
    if(requete=="POMPEON")
    {
        cmdfor5=set(cmdfor5);
        Serial.println("OK");
        requete="";
    }
    elseif(requete=="POMPEET")
    {
        Serial.println(resultcmd5);
        requete="";
    }
    elseif(requete=="POMPEMANU")
    {
    }
    elseif(requete=="LAMPEON")

```

```

{
cmdfor7=set(cmdfor7);
Serial.println("OK");
requete="";
}
elseif(requete=="LAMPEET")
{
Serial.println(resultcmd7);
requete="";
}
elseif(requete=="LAMPEMANU")
{
manucmd7=set(manucmd7);
if(manucmd7==1)
{
Serial.println("OK");
}
else
{
Serial.println("OK");
}
requete="";
}
elseif(requete=="LAMPECP")
{
Serial.println(comptfm7);
requete="";
}
elseif(requete=="SERVOON")
{
cmdfor8=set(cmdfor8);
Serial.println("OK");
requete="";
}
elseif(requete=="SERVOET")
{
//Serial.println(resultcmd8);
Serial.println(fdcfenetre);
requete="";
}
elseif(requete=="SERVOMANU")
{
manucmd8=set(manucmd8);
if(manucmd8==1)
{

```

```

Serial.println("OK");
}
else
{
Serial.println("OK");
}
requete="";
}
elseif(requete=="SERVOCP")
{
Serial.println(comptfm8);
requete="";
}
elseif(requete=="VANNEON")
{
cmdfor9=set(cmdfor9);
Serial.println("OK");
requete="";
}
elseif(requete=="VANNEET")
{
Serial.println(resultcmd9);
requete="";
}
elseif(requete=="VANNEMANU")
{
manucmd9=set(manucmd9);
if(manucmd9==1)
{
Serial.println("OK");
}
else
{
Serial.println("OK");
}
requete="";
}
elseif(requete=="VANNECP")
{
Serial.println(comptfm9);
requete="";
}
elseif(requete=="TPINT")
{
//Serial.println("Tempint");

```

```

Serial.println(temp_int);
requete="";
}
elseif(requete=="TPEXT")
{
//Serial.println("Tempext");
Serial.println(temp_ext);
requete="";
}
elseif(requete=="HUMID")
{
//Serial.println("Humidite");
Serial.println(humidite);
requete="";
}
elseif(requete.startsWith("HUMBS")==true)
{
if(requete.endsWith("#")==true)
{
humbas=convstrint(vartempo);
vartempo="";
Serial.println("OK");
requete="";
}
if(isdigit(lecture)==true)
{
vartempo.concat(lecture);
}
}
elseif(requete.startsWith("HUMHT")==true)
{
if(requete.endsWith("#")==true)
{
humhaut=convstrint(vartempo);
vartempo="";
Serial.println("OK");
requete="";
}
if(isdigit(lecture)==true)
{
vartempo.concat(lecture);
}
}
elseif(requete.startsWith("TEMMO")==true)
{

```

```

if(requete.endsWith("#")==true)
{
tempmoyenne=convstrint(vartempo);
vartempo="";
Serial.println("OK");
requete="";
}
if(isdigit(lecture)==true)
{
vartempo.concat(lecture);
}
}
elseif(requete.startsWith("DELTAT")==true)
{
if(requete.endsWith("#")==true)
{
deltat=convstrint(vartempo);
if(deltat==0)
{
deltat=1;
}
vartempo="";
Serial.println("OK");
requete="";
}
if(isdigit(lecture)==true)
{
vartempo.concat(lecture);
}
}
elseif(requete=="LUMIN")
{
//Serial.println("Luminosite");
Serial.println(lect_luminosite);
requete="";
}
elseif(requete=="RESET")
default:break;
}
//*****Com

```

mandeArduino

```

//*****logiqustandarddecommande
resultcmd6=logiqustandard(manucmd6,cmdfor6,cmdlog6);
resultcmd7=logiqustandard(manucmd7,cmdfor7,cmdlog7);
resultcmd8=logiqustandard(manucmd8,cmdfor8,cmdlog8);

```

```

resultcmd9=logiqustandard(manucmd9,cmdfor9,cmdlog9);
//*****Detectiondesflancsmontantsdecommande
fmcmd5=fm(resultcmd5,&fmmemcmd5);
fmcmd6=fm(resultcmd6,&fmmemcmd6);
fmcmd7=fm(resultcmd7,&fmmemcmd7);
fmcmd8=fm(resultcmd8,&fmmemcmd8);
fmcmd9=fm(resultcmd9,&fmmemcmd9);
//*****Compteurdesflancsmontant
comptfm5=comptfm(fmcmd5,comptfm5);
comptfm6=comptfm(fmcmd6,comptfm6);
comptfm7=comptfm(fmcmd7,comptfm7);
comptfm8=comptfm(fmcmd8,comptfm8);
comptfm9=comptfm(fmcmd9,comptfm9);
//*****Partiecommandesortiesviacommandeforcéeou
logique
    commande(2,resultcmd5);
    commande(3,resultcmd6);
    commande(4,resultcmd7);
    //commande(5,resultcmd8);
    commandeinverse(6,resultcmd9);
    //*****fonctiondeconversionStringtoint
    intconvstrint(Stringmystring)
    {
        intmyStringLength=mystring.length()+1;

        charmyChar[myStringLength];
        mystring.toCharArray(myChar,myStringLength);

        intresult=atoi(myChar);

        returnresult;

    }

    voidfenetre(intcommande_logic,intcommande_ext,boolfdc,boolmodemanuel)
    {

        if((commande_logic==1andmodemanuel==0andfdc==true)or(commande_ext
        ==1andmodemanuel==1andfdc==true))//demanded'ouverture
        {
            servomoteur.attach(5);
            servomoteur.write(0);
            delay(500);
            servomoteur.detach();

```

```
    delay(200);
  }
  elseif((commande_logic==0andmodemanuel==0andfdc==false)or(commande
_ext==1andmodemanuel==1andfdc==false))//fenetreouverte
  {
    servomoteur.attach(5);
    servomoteur.write(180);
    delay(300);
    servomoteur.detach();
    delay(200);
  }
}
```


ПРИЛОЖЕНИЕ С

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="main_style.css" />
    <title>Овденко Валентин Дипломная работа</title>
    <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>
  </head>
  <body onload='refresh_div();'>
    <div id="bloc_page">
      <header>
        <div id="titre_principal">
          <div id="logoHelha">
            
          </div>
          <h1>Автоматизация управления теплицей </h1>
        </div>
        <nav>
          <ul>
            <li><a href="main.php">Главная страница</a></li>

            <li><a href="commands.php">Команды и
параметры</a></li>
          </ul>
        </nav>
        <?php
        try
        {
          //$bdd = new
PDO('mysql:host=192.168.1.12;dbname=db_basilic;charset=utf8','root','root');
          $bdd = new
PDO('mysql:host=localhost;dbname=db_basilic;charset=utf8','root','root');
        }
        catch (Exception $e)
        ?>
      </header>
      <section>
        <div id="Mesures">
        <!-- Affichage des mesures -->
        <!--<?php
          $reponse = $bdd->query('SELECT * from commandes');
```

```

        while ($donnees = $reponse->fetch())
        {
            echo $donnees['Equipement'].'<br />';
        }
        $reponse->closeCursor(); // Termine le traitement de la requête
    ?> -->
    <h1>Измерения</h1>
    <p>Внешняя температура : <span class = "valeur_ext_temp">
<strong> <?php echo GetMeasure('1',$req); ?> </strong></span> °C
    </p>
    <p>Внутренняя температура : <span class = "valeur_int_temp">
<strong><?php echo GetMeasure('2',$req); ?></strong></span> °C
    </p>
    <p>Уровень влажности почвы : <span class =
"valeur_soil_moist"> <strong><?php echo GetMeasure('4',$req);
?></strong></span> %
    </p>
    <p>Освещенность: <span class = "valeur_lum"> <strong><?php
echo GetMeasure('3',$req); ?></strong></span> Люмен
    </p>
    <!-- Affichage des nombres de démarrages -->
    <?php
;
    ?>
    <p>Кол-во запусков насоса : <span class
="counter"><strong><?php echo GetStartCount('Pompe',$req); ?>
</strong></span> <!-- voir pour faire une fonction pour le s -->
    <?php $device='Pompe' ?>
    </p>
    <p>Кол-во запусков вентилятора : <span class
="counter"><strong><?php echo GetStartCount('Ventilateur',$req);
?></strong></span> <!-- voir pour faire une fonction pour le s -->
    </p>
    <p>Кол-во включений лампы : <span class
="counter"><strong><?php echo GetStartCount('Lampe_infrarouge',$req);
?></strong></span> <!-- voir pour faire une fonction pour le s -->
    </p>
    <p>Кол-во открытий клапана : <span class
="counter"><strong><?php echo GetStartCount('Vanne',$req); ?></strong></span>
<!-- voir pour faire une fonction pour le s -->
    </p>
    <p>Кол-во открытий окна : <span class
="counter"><strong><?php echo GetStartCount('Servomoteur',$req);
?></strong></span> <!-- voir pour faire une fonction pour le s -->

```

```

        </p>
    </div>
    <div id="Etat">
        <!-- Affichage des états -->
        <h1>Состояние устройств</h1>
        <?php $PumpMode=GetDevMode('Pompe',$req); ?>
        <p>Режим работы насоса: <span class = "mode">
<strong><?php echo $PumpMode; ?></strong></span>
        </p>
        <p>Состояние насоса : <span class = "dev_sate">
<strong><?php echo GetDevState('Pompe',$req); ?></strong></span>
        </p>
        <?php $FanMode=GetDevMode('Ventilateur',$req); ?>
        <p>Режим работы вентилятора : <span class = "mode">
<strong><?php echo $FanMode; ?></strong></span>
        </p>
        <p>Состояние вентилятора : <span class = "dev_sate">
<strong><?php echo GetDevState('Ventilateur',$req); ?></strong></span>
        </p>
        <?php $LampMode=GetDevMode('Lampe_infrarouge',$req); ?>
        <p>Режим работы лампы : <span class = "mode">
<strong><?php echo $LampMode; ?></strong></span>
        </p>
        <p>Состояние вентилятора : <span class = "dev_sate">
<strong><?php echo GetDevState('Lampe_infrarouge',$req); ?></strong></span>
        </p>
        <?php $WindowMode=GetDevMode('Servomoteur',$req); ?>
        <p>Режим поднятия окна : <span class = "mode">
<strong><?php echo $WindowMode; ?></strong></span>
        </p>
        <p>Состояние окна: <span class = "dev_sate"><strong><?php
echo GetDevState('Servomoteur',$req); ?></strong></span>
        </p>
        <?php $ValveMode=GetDevMode('Vanne',$req); ?>
        <p>Режим работы клапана : <span class = "mode">
<strong><?php echo $ValveMode; ?></strong></span>
        </p>
        <p>Состояние клапана : <span class =
"dev_sate"><strong><?php echo GetDevState('Vanne',$req); ?></strong></span>
        </p>
    </div>
</section>

```

```

<footer>
<div id="Liens">
    <div class="nav2">
        <div>
        </div>
    </div>
</div>
</body>
</html>
<script>
function refresh_div()
{
    var xhr_object = null;
    if(window.XMLHttpRequest)
    { // Firefox
    }
    else if(window.ActiveXObject)
    { // Internet Explorer
        xhr_object = new ActiveXObject('Microsoft.XMLHTTP');
    }
    var method = 'GET';
    var filename = 'main.php';
    xhr_object.open(method, filename, true)
    xhr_object.onreadystatechange = function(
        if(xhr_object.readyState == 4)
            var tmp = xhr_object.responseText;
            document.getElementById('bloc_page').innerHTML = tmp;
        }
    }
    xhr_object.send(null);
    setTimeout('refresh_div()', 1000);
}
</script>
</script>

```