

Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

Курсовая работа

Дисциплина: Объекто-ориентированное программирование

Тема: Проектирование автоматической работы проекта “Умная теплица”

Студент гр. 3331506/00401

Аграновский Д.А.

Преподаватель

Ананьевский М.С.

Санкт-Петербург

2023

Оглавление

| | |
|---|----|
| Цель | 3 |
| 1. ВВЕДЕНИЕ | 4 |
| 2. ХОД РАБОТЫ | 7 |
| 2.1 Первый этап работы | 7 |
| 2.2 Контур управления освещением | 8 |
| 2.3 Контур управления вентиляцией | 10 |
| 2.4 Контур управления поливом | 12 |
| 3. РЕАЛИЗАЦИЯ | 14 |
| 4. ИТОГИ РАБОТЫ | 18 |
| СПИСОК ЛИТЕРАТУРЫ | 19 |

Цель:

- Проанализировать принцип работы Умной теплицы OMEGAGROW и ее компонентов;
- Изучить способы автоматизации работы теплицы с использованием современных технологий и устройств;
- Разработать проект автоматизации Умной теплицы OMEGAGROW, учитывая особенности ее работы и потребности клиента;
- Разработать алгоритм работы Умной теплицы, которая будет оптимально поддерживать условия для роста и развития растений.

1. ВВЕДЕНИЕ

Сегодня технологии автоматизации в сельском хозяйстве становятся все более популярными и востребованными. Автоматизация позволяет упростить и ускорить процессы, увеличить эффективность работы и сократить затраты. Одним из ярких примеров автоматизации в сельском хозяйстве является умная теплица.

Умные домашние теплицы – это технологически продвинутые устройства для выращивания растений внутри помещения, которые обычно включают в себя системы контроля и управления растениеводством.

Существует несколько разновидностей умных домашних теплиц, отличающихся своими особенностями и функциями:

- Гидропонные теплицы – это системы, в которых растения выращиваются без почвы, в специальной среде, например, в воде с добавлением питательных веществ.
- Аэропонные теплицы – это системы, в которых растения выращиваются в воздушной среде, обогащенной питательными веществами. Корни растений находятся в специальной камере, где они получают необходимые элементы питания из распыленных растворов.
- Аквапонные теплицы – это комбинированные системы гидропоники и аквакультуры, в которых растения выращиваются в воде, а рыбы - в той же самой воде. Рыбы выделяют нитраты, которые используются растениями в качестве питательных веществ, а в свою очередь растения очищают воду для рыб.
- Традиционные теплицы с автоматизацией – это системы, в которых растения выращиваются в почве, но имеют автоматическую систему контроля и управления, которая может регулировать температуру, влажность и освещение внутри теплицы.

Все эти разновидности умных теплиц могут быть использованы для выращивания широкого спектра растений, включая овощи, фрукты, цветы и травы, что делает их полезными для любителей здорового образа жизни, а также для тех, кто желает получить свежие овощи и фрукты круглый год. Именно умные теплицы стали самым распространенным объектом “Интернета вещей” (IoT).

Интернет вещей (Internet of Things, IoT) – это концепция, описывающая взаимодействие между физическими устройствами, объектами и системами через сеть Интернет. С помощью технологий IoT, устройства могут обмениваться данными между собой и с другими приложениями, что позволяет создавать интеллектуальные системы, способные автоматически реагировать на изменяющиеся условия и выполнять определенные действия без участия человека.

В данной курсовой работе мы будем рассматривать создание алгоритма работы умной теплицы с использованием технологий IoT. Мы будем использовать датчики для сбора информации о температуре, влажности и освещенности, а также управляющие устройства для автоматического управления режимом работы теплицы.

Цель данной курсовой работы - разработать алгоритм программы традиционной умной теплицы OMEGAGROW. Умная теплица OMEGAGROW – это современная система автоматизации, которая позволяет контролировать и оптимизировать множество процессов, необходимых для выращивания растений в закрытом помещении. Она включает в себя контроль температуры, влажности, освещения и полива, а также мониторинг изменений этих параметров в реальном времени и принятие решений на основе собранных данных.

Курсовая работа по написанию алгоритма программы умной теплицы OMEGAGROW предполагает использование платформы Arduino для программирования устройств управления теплицей.

Arduino – это открытая платформа для разработки электронных устройств, которая позволяет создавать простые и сложные системы управления на основе микроконтроллеров.

Использование Arduino позволяет создать надежную и гибкую систему управления теплицей, которая может автоматически контролировать температуру, влажность, освещение и другие параметры, что позволит увеличить урожайность и качество растений.

В автоматизированной системе поддержания роста растений умной теплицы OMEGAGROW имеются три независимые системы контроля условий окружающей среды. Они включают:

- контроль освещенности, который осуществляется с помощью модуля фоторезистора и фитодиодной ленты;
- контроль влажности почвы, который осуществляется с помощью датчика влажности почвы и системы капельного полива;
- контроль температуры воздуха, который осуществляется с помощью датчика температуры и влажности воздуха, а также вентилятора с нагревательным элементом.

Умная теплица OMEGAGROW представлена на рисунке 1 и 2.

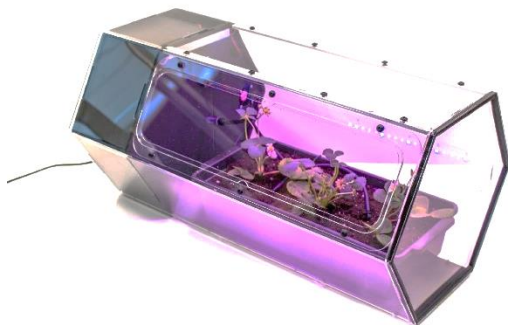


Рисунок 1 – Общий вид



Рисунок 2 – Вид сбоку

2. ХОД РАБОТЫ

2.1 Первый этап работы

Первый этап работы заключался в исследовании схемы питания и элементов управления теплицы. Для этого была произведена анализ электронной оснастки теплицы, которая представлена на фотографии в статье. Главным источником питания является блок питания от сети 220В с выходными характеристиками 12В и 8А. Силовая часть теплицы, включая все ее исполнительные устройства, питается напрямую от блока питания.

Для питания логических компонентов теплицы, включая программный контроллер, был использован регулируемый понижающий DC-DC преобразователь на основе микросхемы LM2596HVS. Благодаря этому, входящее напряжение с блока питания понижается и создается источник для питания логических компонентов теплицы.

Для управления исполнительными устройствами теплицы был использован блок из четырех электромеханических реле с опторазвязкой. Выходной сигнал этого блока может принимать только значения "включить" и "выключить". Кроме того, были установлены различные датчики, модули и системы контроля температуры, освещенности, влажности почвы и другие элементы, которые необходимы для автоматического управления теплицей.

На данном этапе были изучены основные компоненты и принципы работы электронной оснастки теплицы, которые позволят продолжить работу над созданием алгоритма управления теплицей.

Теплица в работе использует три контура управления: освещением, вентиляцией и поливом:

- Контур управления освещением основан на модуле датчика освещенности на основе фоторезистора. Когда уровень освещения в теплице становится ниже заданного порога, контроллер Arduino включает фитодиодные ленты, которые обеспечивают растениям необходимое количество света.

- Контур управления вентиляцией включает вентилятор и нагревательный элемент, и основан на системе контроля температуры воздуха в теплице. Если температура поднимается выше установленного порога, вентилятор начинает работать, чтобы обеспечить естественную циркуляцию воздуха в теплице. Если температура продолжает расти, то включается нагревательный элемент.
- Контур управления поливом основан на модуле аналогового датчика влажности почвы. Когда уровень влажности в почве становится ниже заданного порога, программа управления теплицей включает воздушно-водяной насос, который подает воду в почву до тех пор, пока уровень влажности не достигнет заданного уровня.

Каждый контур управления имеет свое исполнительное устройство, которое включается и выключается с помощью электромеханических реле. Реле управляются контроллером Arduino, который выдает управляющий сигнал на базу транзистора, который в свою очередь управляет реле.

2.2 Контур управления освещением



Рисунок 3 – Аналоговый датчик освещенности



Рисунок 4 – Фитодиодная лента

Для реализации реакции системы на изменение внешней освещенности был использован модуль датчика освещенности, который подключался к микроконтроллеру через цифровой выход. Низкий логический уровень

соответствовал высокому уровню внешней освещенности, а высокий логический уровень — низкому.

Однако, для работы с микроконтроллером датчик освещенности должен был быть подключен к аналоговому входу. Для решения этой проблемы был выбран следующий режим работы освещения: при появлении высокого сигнала на цифровом выходе датчика, реле отправляло высокий уровень, замыкающий разрыв цепи и подавал питание на фитодиодную ленту.

Фитодиодная лента — это светодиодная лента специального спектра, созданная для обеспечения оптимального освещения для растений в теплице. Пример такой ленты представлен на рисунке 4.

Алгоритм работы контура представлен на рисунке 5.

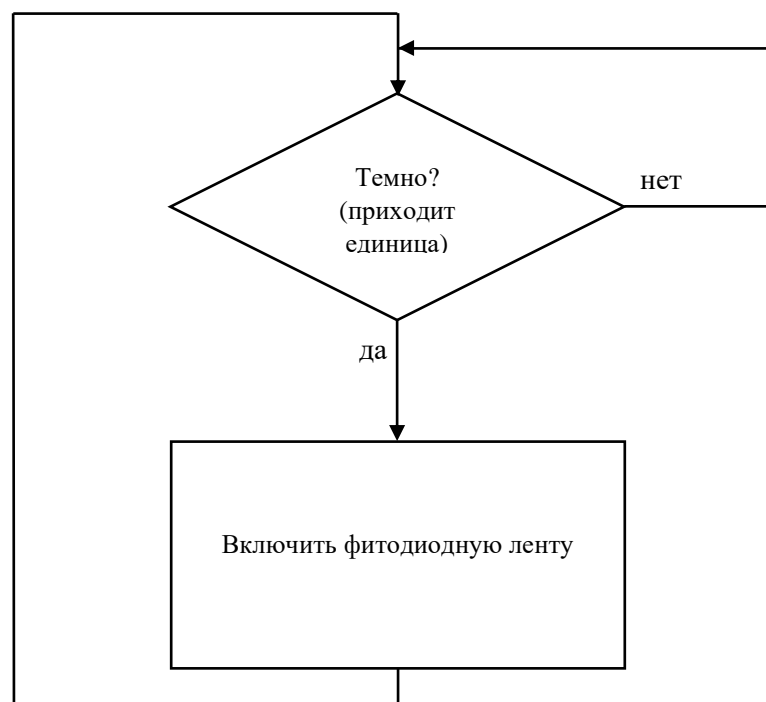


Рисунок 5 – Алгоритм контура управления освещением

2.3 Контур управления вентиляцией

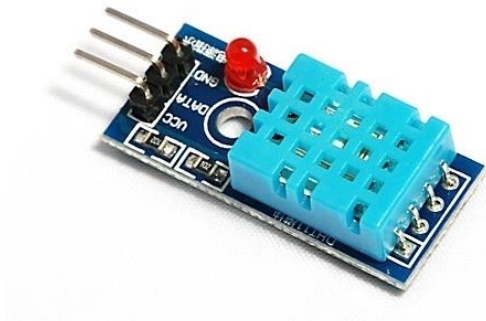


Рисунок 6 – Датчик модели GHT-11



Рисунок 7 – Вентилятор и нагревательный элемент

Для контроля температуры воздуха в умной теплице может использоваться DHT датчик, представленный на рисунке 7. DHT датчик позволяет измерять температуру и относительную влажность воздуха внутри теплицы.

Для реализации контроля температуры воздуха необходимо подключить DHT датчик к микроконтроллеру. После подключения датчика к микроконтроллеру необходимо написать программу, которая будет считывать данные с датчика и управлять работой вентилятора и нагревателя.

В программе нужно определить пороговые значения для температуры воздуха. Когда температура воздуха превышает установленный порог, необходимо включить вентилятор для охлаждения теплицы. Когда температура воздуха опускается ниже установленного порога, необходимо выключить вентилятор.

Кроме того, в программе нужно определить пороговые значения для относительной влажности воздуха. Когда относительная влажность воздуха превышает установленный порог, необходимо включить нагреватель для увеличения температуры и снижения влажности. Когда относительная

влажность воздуха опускается ниже установленного порога, необходимо выключить нагреватель.

Для управления вентилятором и нагревателем используются реле. Когда программа определяет необходимость включения вентилятора или нагревателя, она отправляет сигнал на соответствующее реле, которое замыкает контакты и включает соответствующее устройство.

Алгоритм работы контура представлен на рисунке 8.

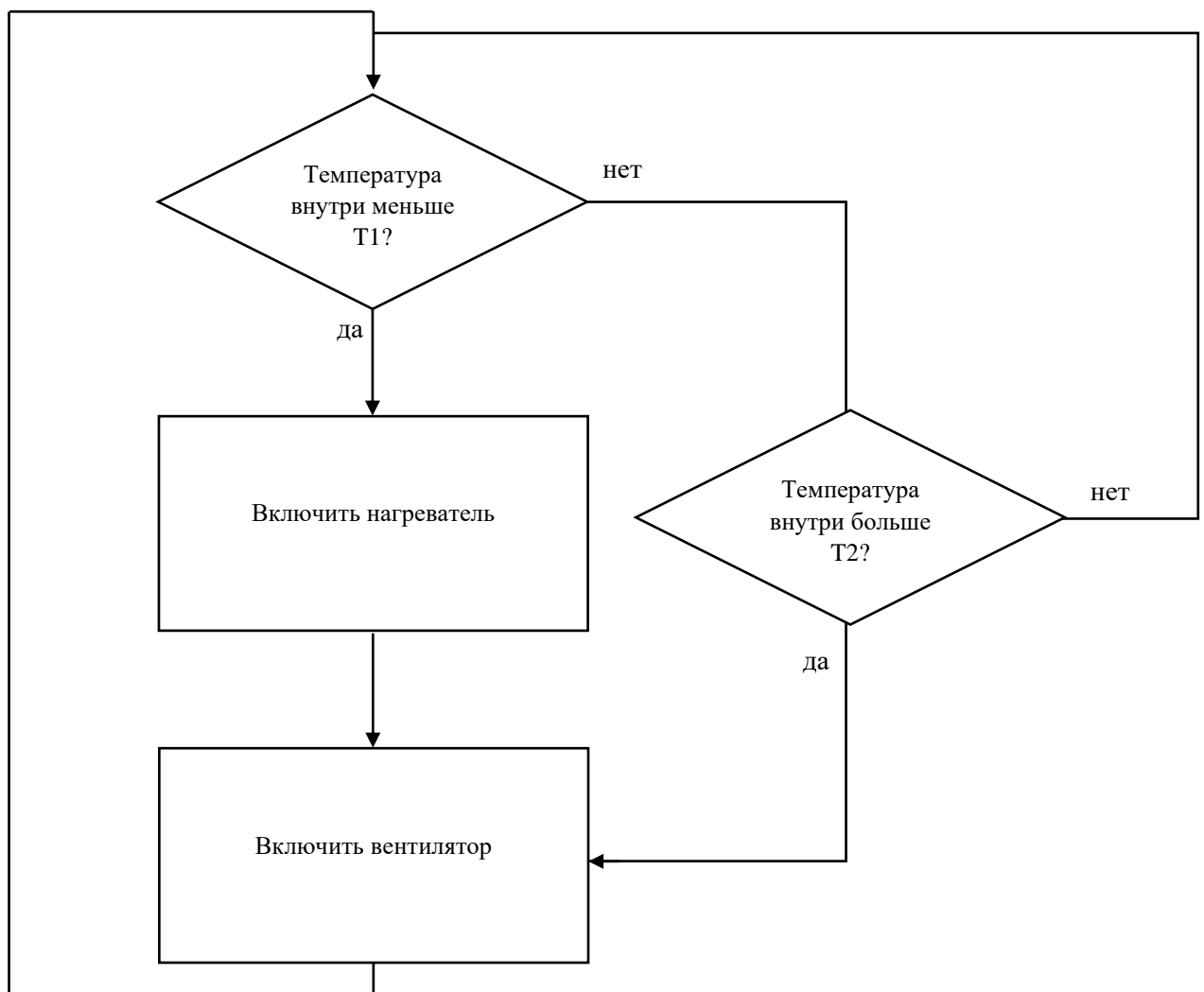


Рисунок 8 – Алгоритм контура управления температурой

2.4 Контур управления поливом



Рисунок 9 – Аналоговый датчик влажности почвы



Рисунок 10 – Водяная помпа

Для реализации контроля влажности почвы в умной теплице используется аналоговый датчик влажности почвы (рисунок 9), который соединен с микроконтроллером через аналоговый вход. Значение, полученное с датчика, интерпретируется как аналоговое напряжение, пропорциональное уровню влажности почвы.

Для поддержания оптимального уровня влажности почвы используется водяная помпа, которая подключена к микроконтроллеру и управляется программно. Если уровень влажности почвы понижается ниже заданного порогового значения, то микроконтроллер отправляет сигнал на включение помпы через реле, и вода начинает подаваться в почву в течение определенного времени. Далее ожидается распространение воды в течение 10 минут.

Таким образом, система контроля влажности почвы включает в себя аналоговый датчик влажности почвы, который позволяет определять уровень влажности, и водяную помпу, которая поддерживает оптимальный уровень

влажности почвы путем автоматического полива. Все управление происходит программно на микроконтроллере, что обеспечивает надежную и эффективную работу системы.

Алгоритм работы контура управления представлен на рисунке 11.

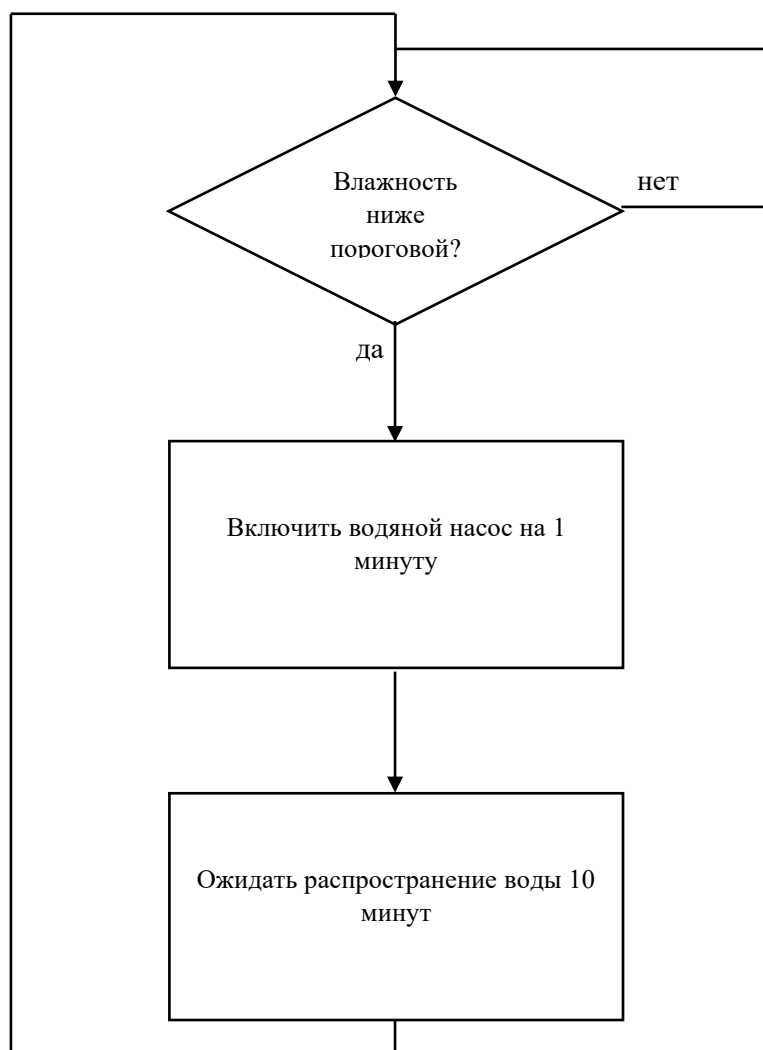


Рисунок 11 – Алгоритм контура управления поливом

3. РЕАЛИЗАЦИЯ

Алгоритм управления был реализован с помощью среды разработки Arduino IDE. К ней была подключена библиотека для датчика DGT-11 <TroykaDHT.h>. Было принято решение реализовать программу с выносом каждого контура в отдельную функцию, а так же с возможностью выбора одного из трех вариантов выращиваемых растений.

Код программы представлен далее:

```
#define SENSOR_LED_PIN A0 //датчик освещенности
// 0 - светло
// 1 - темно
#define LED_PIN 6 //светодиодная лента
#define SENSOR_HUMIDITY_PIN A1//датчик влажности почвы
// 0 - влажно
// 1023 - сухо
#define WATER_PUMP_PIN 5 //водяная помпа
#define SENSOR_DHT_PIN 2 //датчик влажности и температуры воздуха
#define HEAT_PIN 4 //нагревательный элемент вентилятора
#define FAN_PIN 7 //вентилятор
#include <TroykaDHT.h>
DHT dht_sensor(SENSOR_DHT_PIN, DHT21);

int TYPE;//тип растения 1, 2 или 3;

unsigned long DELAY = 2000;// задержка
int MIN_LIGHT; // меньше данной освещенности включается светодиодная лента если бы
был аналоговый....
float MIN_T;// меньше данной температуры включается вентилятор+нагреватель
float MAX_T;// выше данной температуры включается вентилятор
int MIN_HUMIDITY;// меньше данной относительной влажности почвы включается помпа
const int DRY_SOIL_HUMIDITY = 0; //значение влажной почвы
const int WET_SOIL_HUMIDITY = 1023; //значение сухой почвы
// 0 - влажно
// 1023 - сухо
const unsigned long int WATERING_TIME = 10000; //время полива
const unsigned long int DELAY_TIME = 80000; //ожидание распространения воды

void setup() {
    Serial.begin(9600);
    dht_sensor.begin();

    pinMode(SENSOR_LED_PIN, INPUT);
    pinMode(SENSOR_HUMIDITY_PIN, INPUT);
    pinMode(SENSOR_DHT_PIN, INPUT);
}
```

```

pinMode(LED_PIN, OUTPUT);
pinMode(HEAT_PIN, OUTPUT);
pinMode(FAN_PIN, OUTPUT);
pinMode(WATER_PUMP_PIN, OUTPUT);

Serial.println("Введите тип растения из списка:");
Serial.println("1 - клубника");
Serial.println("2 - огурцы");
Serial.println("3 - арбуз");
Serial.write(TYPE);

switch(TYPE){
    case 1 : Serial.println("Тип 1 - клубника");
        MIN_LIGHT = 400;
        MIN_T = 20.0;
        MAX_T = 30.0;
        MIN_HUMIDITRY = 50;
        break;

    case 2 : Serial.println("Тип 2 - огурцы");
        MIN_LIGHT = 600;
        MIN_T = 15.0;
        MAX_T = 40.0;
        MIN_HUMIDITRY = 60;
        break;

    case 3 : Serial.println("Тип 3 - арбуз");
        MIN_LIGHT = 800;
        MIN_T = 35;
        MAX_T = 50;
        MIN_HUMIDITRY = 70;
        break;

    default : Serial.println("вы выбрали неверный тип, введите число от 1 до 3");
        break;
}
}

void loop() {
    DHT_function(DELAY, MIN_T, MAX_T);
    LED_function(DELAY);
    HUM_function(DELAY, MIN_HUMIDITRY, DRY_SOIL_HUMIDITRY, WET_SOIL_HUMIDITRY,
    WATERING_TIME, DELAY_TIME);
}

void DHT_function(unsigned long delay, float min_temp, float max_temp){
    dht_sensor.read();
    float temperature;
    float humid_air;
    unsigned long int time_dht_read = 0;

```

```

// проверяем состояние данных
switch(dht_sensor.getState()) {
  // всё OK
  case DHT_OK:
    // выводим показания влажности и температуры
    temperature = dht_sensor.getTemperatureC();
    humid_air = dht_sensor.getHumidity();
    Serial.print("Temperature = ");
    Serial.print(temperature);
    Serial.println(" C \t");
    Serial.print("Temperature = ");
    Serial.print(dht_sensor.getTemperatureK());
    Serial.println(" K \t");
    Serial.print("Temperature = ");
    Serial.print(dht_sensor.getTemperatureF());
    Serial.println(" F \t");
    Serial.print("Humidity = ");
    Serial.print(humid_air);
    Serial.println(" %");
    digitalWrite(HEAT_PIN, LOW);
    digitalWrite(FAN_PIN, LOW);
    if(temperature <= min_temp){
      Serial.println("Temperature is too low. Need to turn on fan and
heater.\n");
      digitalWrite(FAN_PIN, HIGH);
      digitalWrite(HEAT_PIN, HIGH);
    } else {
      if(temperature >= max_temp){
        Serial.println("Temperature is too high. Need to turn on fan.\n");
        digitalWrite(FAN_PIN, HIGH);
      } else {
        Serial.println("Temperature is normal.\n");
      }
    }
  }
  if (millis() - time_dht_read > delay){
    time_dht_read = millis();
    Serial.println("Waiting");
  }
  break;
  // ошибка контрольной суммы
  case DHT_ERROR_CHECKSUM:
    Serial.println("Checksum error");
    break;
  // превышение времени ожидания
  case DHT_ERROR_TIMEOUT:
    Serial.println("Time out error");
    break;
  // данных нет, датчик не реагирует или отсутствует
  case DHT_ERROR_NO_REPLY:
    Serial.println("Sensor not connected");

```



```

        break;
    }
}

void LED_function(unsigned long delay){
    unsigned long int time_led_read = 0;
    bool led_lvl = 0;
    led_lvl = digitalRead(SENSOR_LED_PIN);

    if (led_lvl == 0) {
        Serial.println("It is bright\n");
        digitalWrite(LED_PIN, LOW);
    } else {
        Serial.println("It is dark. Need to turn on the light\n");
        digitalWrite(LED_PIN, HIGH);
    }
    if (millis() - time_led_read > delay){
        time_led_read = millis();
        Serial.println("Waiting");
    }
}

void HUM_function(unsigned long delay, int min_humidity, int dry_soil_humidity,
int wet_soil_humidity, unsigned long int watering_time, unsigned long int
delay_time){
    unsigned long int time_hum_read = millis();
    unsigned int hum_lvl = 0;
    hum_lvl = digitalRead(SENSOR_HUMIDITY_PIN);
    unsigned int countled;
    countled = map(hum_lvl, wet_soil_humidity, dry_soil_humidity, 0, 100);
    // индикация уровня влажности
    if (hum_lvl <= MIN_HUMIDITY){
        digitalWrite(WATER_PUMP_PIN, HIGH);
        if (millis() - time_hum_read > watering_time){
            time_hum_read = millis();
            Serial.println("Watering in progress");
        }
        digitalWrite(WATER_PUMP_PIN, LOW);
        time_hum_read = millis();
        if (millis() - time_hum_read > delay_time){
            time_hum_read = millis();
            Serial.println("Waiting for the water to spread");
        }
    }
}
}

```

4. ИТОГИ РАБОТЫ

В ходе работы были выявлены следующие проблемы:

- Неправильная полярность подключения насоса, который качает воду в обратную сторону;
- Питание Arduino подается с выхода понижающего преобразователя, что приводит к подаче напряжения около 5В на VIN-пин платы, вместо рекомендуемых 7-15В через пин 5V;
- Неудобный кабель-менеджмент, который не предоставляет возможности доступа к отдельным проводам и может являться источником помех для передачи данных с датчиков и вызывает трудности в нахождении неполадок и изучения платформы;
- Неудобство присоединения крышки теплицы, которая напрямую связана с корпусом теплицы при помощи двух проводов от порта для подключения блока питания;
- Отсутствие места для подключения отдельных-дополнительных датчиков для модифицирования теплицы.
- Отсутствие свободных гнезд в клеммах для подключения дополнительных проводов, так как имеющиеся провода питания занимают место в имеющихся гнездах;
- Использование реле, вместо мосфета.

Эти проблемы могут привести к неполадкам в работе теплицы, ошибкам с датчиками, ухудшению качества урожая и увеличению времени на обслуживание и ремонт теплицы.

Модификация платы была бы возможна при наличии места в корпусе под другие датчики, а также расширение Arduino UNO на Arduino MEGA и другие. Так же большим плюсом играло наличие LED-дисплея и кнопок, при помощи которых можно было бы видеть статус теплицы в момент времени и выбирать режим работы.

СПИСОК ЛИТЕРАТУРЫ

1. Умная теплица OMEGAGROW — <https://omegabot.ru/product/28>
2. Интернет вещей (Материал из Википедии) — https://ru.wikipedia.org/wiki/Интернет_вещей
3. Arduino UNO R3 — Product Reference Manual: <https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf>
4. Описание датчиков Arduino Master — <https://arduinomaster.ru/datchiki-arduino/photorezistor-arduino-datchik-sveta/>
5. Arduino IDE 2 Tutorials — <https://docs.arduino.cc/software/ide-v2>
6. LM2596HV 60V 3A 150kHz Step-Down Voltage Regulator Datasheet — <http://amperkot.ru/static/3236/uploads/datasheets/LM2596HV.PDF>
8. Datasheet для SRD-05VDC-SL-C, реле 1 переключатель 5VDC/10A, 125VAC — <https://static.chipdip.ru/lib/642/DOC012642672.pdf>
9. Temperature and humidity module AM2301 Product Manual — <https://smartep.ru/datasheets/sensors/DHT21%20AM2301.pdf>
10. Библиотека для Arduino, позволяющая считывать данные датчиков серии DHT — <https://github.com/amperka/TroykaDH>