

Муниципальное бюджетное образовательное учреждение
«Гимназия № 11 г. Ельца»

Научно-исследовательская работа:
Теплица для комнатных растений

ФИО автора работы:
Галкина Ольга Игоревна,
Ученица 11 «А» класса
МБОУ «Гимназия № 11 Г. Ельца»

ФИО научного руководителя:
Австриевских Н. М., учитель физики МБОУ
«Гимназия №11 г. Ельца»
Злобин Максим Сергеевич, преподаватель
ГООУ «Центр поддержки одаренных детей «Стратегия»

Содержание

1. Введение
2. Теоретическая часть
 - 2.1. Факторы, влияющие на развитие растений
 - 2.2. Техническое описание
3. Практическая часть
4. Заключение
5. Список литературы
6. Приложение



Рис. 1 – Хлорофитум (лат. Chlorophytum)

1. Введение

Свою работу я хотела бы начать с того, что комнатные растения являются подарком человеку от природы. Красотой форм, приятным запахом и спокойной зелёной окраской они благотворно влияют на центральную нервную систему, помогая справиться с плохим настроением или стрессовым состоянием, а также создают уют в доме и наполняют его положительной энергетикой. Но даже за самой крохотной теплицей на подоконнике нужен уход, который люди по различным причинам не могут осуществлять, например, от того, что порой их не бывает дома. Тут на помощь может прийти устройство, которое является темой моего проекта – «Теплица для комнатных растений».

Проблема: изучение факторов, влияющих на рост и развитие растений; создание оптимизированной системы, ухаживающей за ними.

Гипотеза: данный прибор сможет автоматизировать уход даже за самым прихотливым растением.

Методы исследования: теоретический анализ научной литературы по данной проблеме (моделирование, синтез); эмпирический способ (эксперимент, наблюдение, сравнение); математический расчет (статистика, программирование).

Объектом моего исследования являются комнатные растения.

Цель проекта: создание действующего макета домашней теплицы для выращивания растений в автономном режиме с возможностью наблюдения и управления процессом ухода за растениями с применением энергосберегающих технологий и современных интеллектуальных систем.

Задачи

1. Рассмотреть условия, необходимые для жизни растений
2. Продумать модели, улучшающие эти условия
3. Описать работу каждой модели
4. Объединить модели в одну
5. Собрать схему
6. Написать скетч (программу)
7. Оценить работу устройства

Актуальность: весьма высока сегодня, поскольку проект представляет собой часть системы «Умный дом», элементы которого мы уже частично используем в своих жилищах. При этом управление «Теплицей для комнатных растений» понятно обычному пользователю, социализировано в современном информационном пространстве.

Оригинальность работы заключается в том, что разработанная мною автоматизированная система существенно упрощает жизнь людям, ухаживающим за комнатными растениями, не имея при этом достойных аналогов в своём ценовом сегменте.

Целевая аудитория: люди, которые не желают тратить много времени на уход за растениями / не имеют возможности осуществлять уход по причине длительного отсутствия: командировок или отпуска / люди-любители садоводства.

Ожидаемые результаты: создание комплекта для сборки умной теплицы, с помощью которой можно будет автоматизировать любой домашний цветник. А её управление будет легким для людей с любым уровнем компьютерной грамотности.

Мой проект имеет межпредметную направленность, так как затрагивает



Рис. 2 - Фуксия (лат. Fuchsia)

обширные разделы информатики (программирование), физики (электротехника), биологии (ботаника). Я использовала различные источники информации (научная и учебная литература, интернет). Проанализировав теоретический материал и проведя эксперимент, я пришла к выводу, что знание именно этих дисциплин необходимо для развития современного агарного комплекса.

2. Теоретическая часть

2.1. Факторы, влияющие на рост и развитие растений

Свет (рис. 3) – это источник энергии. Синий и красный совместно активируют процесс фотосинтеза; инфракрасный стимулирует поглощение питательных веществ; красный и инфракрасный влияют на рост и развитие растения и плодоношения. С уменьшением интенсивности света замедляются жизненные процессы.



Рис. 3

Температура (+10 - + 25°C). Тепло стимулирует рост, питание и процессы, происходящие в почве. Необходимое условие для всех растений: температура почвы не должна быть выше температуры помещения.

Вода (рис. 4) – третий важнейший фактор жизни. В ней растворены



Рис. 4

минеральные вещества, которыми питаются растения, вода служит для них «транспортным средством», необходимым для циркуляции питания, она регулирует процессы поглощения, преобразования и выделения веществ, температуру субстрата и воздуха.

Таким образом, растения в комнатной культуре полностью зависят от человека: не будем забывать об этом и создадим систему автоматического ухода за ними, тогда всем вместе будет комфортно жить рядом друг с другом.

2.2. Техническое описание проекта

Проект «Теплица для комнатных растений» выполнен на основе аппаратно-вычислительной платформы Arduino UNO. Теплица оборудована платой расширения и встроенной метеостанцией. Выбор элементов:

Для 1 фактора (свет). Выбирая источник света, я изучала спектры ламп накаливания и люминесцентных. В первых преобладают красный и желтый части спектра, но синего в них часто не хватает, в то время как во-вторых компенсируется этот недостаток. Такие люминесцентные лампы называются фито лампы, и они предназначены именно для искусственного освещения растений. У них множество преимуществ: концентрация излучения в нужном

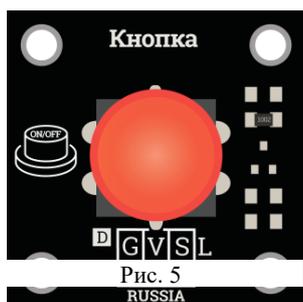
направлении, долговечность, небольшое энергопотребление, энергоэффективность, безопасность для человека и растений.

Для 2 фактора (температура). Долго думая о том, как же можно поддерживать нужную температуру автоматически, я пришла к выводу, что нет ничего лучше простого её измерения и самостоятельного наблюдения за ней.

Для 3 фактора (вода). Авто полив для комнатных цветов – это обобщенное название для принципиально разных технических решений, позволяющих заниматься поливом цветов значительно реже. Рассмотрев разные их виды и устройства, я выбрала самый практичный вариант, работа которого основана на мембранном насосе.

Для управления теплицей предусмотрены 2 способа:

1) Ручной (рис. 5). Инструкция:



Однократно нажатие на обе кнопки «L» и «M» - закрепление текущего состояние влажности почвы как минимальное и переход устройства в рабочий режим.

Нажатие и удержание обе кнопки «L» и «M» дольше 2 секунд - переход устройства в режим ввода значений:

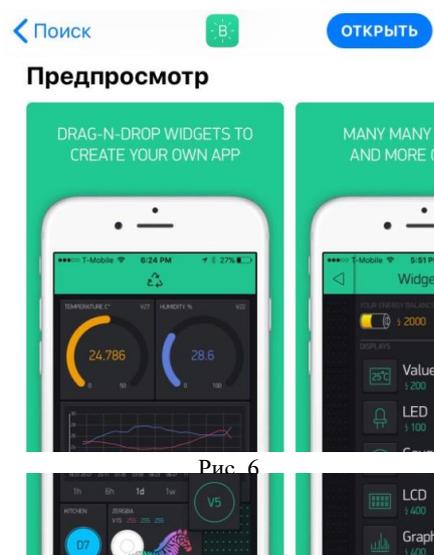
- установка минимальной влажности почвы
- установка длительности полива

Удержание кнопки «A» - уменьшение, а кнопки «B» - увеличение.

2) Интерактивный (рис. 6). Характеристика:

Blynk – контроллер аппаратно-вычислительной платформы Arduino UNO. Возможности этой программы: управление теплицей напрямую, получение показаний датчиков в реальном времени, возможность создавать графики для каждого сенсора, управление с помощью таймеров, а также удаленное управление через Интернет.

Все детали теплицы будут вырезаны из фанеры по макетам, спроектированным в программе Компас



3. Практическая часть

Основное оборудование: Arduino Uno, кабель USB, провода «папа-папа» (курсив - доп. оборудование)

Разобьем нашу систему на три части и рассмотрим сборку каждой в отдельности.

1. Освещение (скетч – приложение 1).

Фитолампа работает постоянно, поэтому напрямую подключаем ее к Arduino Uno, как показано на рис. 8. За фитолампу принимать Neo Pixel Ring 12, поскольку в данном симуляторе предложенное оборудование отсутствует.



Рис. 7 - Фитолампа

2. Вывод значения текущей температуры на дисплей (скетч – приложение 2).

Зададим формулу для преобразования выходного напряжения в температуру: $temperatureC = (voltage - 0.5) * 100$. Данные будем измерять с интервалом 5 секунд и значения выводить на *символьный ЖК дисплей*. Для работы с *модулем DHT11* будем использовать Arduino библиотеку DHT (LiquidCrystal). Объединим все детали в проект (+*резистор 10 кОм, малая макетная плата прототипирования*), показанный на рис. 8, затем соединим их в цепь. Процесс ее сборки вы можете увидеть на рис. 9.

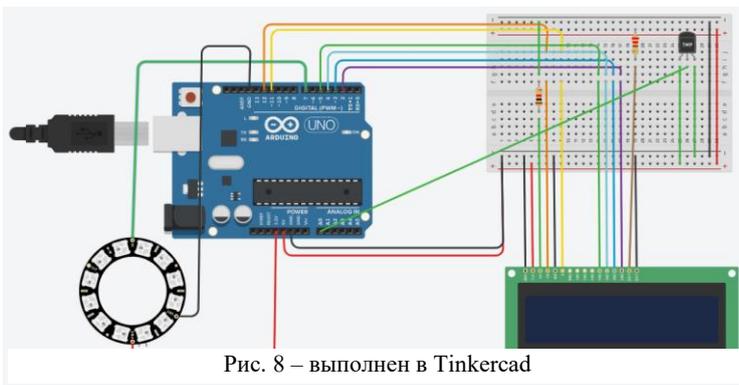


Рис. 8 – выполнен в Tinkercad

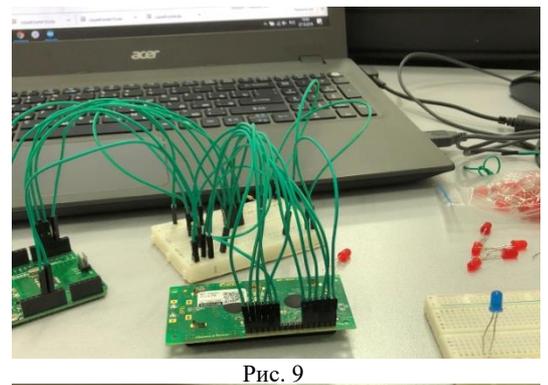


Рис. 9

3. Полив (скетч - приложение 3). Выделим три режима в работе устройства:

1) Действие. Вывод на *четырёхразрядный индикатор* данных влажности почвы, минимальной влажности почвы, времени прошедшего с последнего полива: `if(timSketch/1000%15<5) {dispLED...print(valMoisture);} else if(timSketch/1000%15<10) {dispLED...print(limMoisture,LEN4);} else {dispLED.light(7); if(timWatering) {dispLED.print(int.../1000%3600%60), TIME);} else{dispLED.print("----");}}`

Переходный момент: значения влажности почвы ниже значения минимальной влажности почвы – переход устройства в режим полива:
`if(valMoisture<=limMoisture) { timWatering=timSketch; modState=3; dispLED.light(7); analogWrite(pinPump,pwmPump); }break;`

2) Полив. Подача сигнала *Коннектора Power Jack с клемником на силовой ключ*, который включает *мембранный насос*. Значение скорости мотора насоса и длительность полива указывается в скетче (приложение 3). Вывод количества секунд до окончания полива: `dispLED.print(timDuration-...); dispLED.point(0, true); dispLED.point((timSketch/100%4) +1, true)}`

3) Ожидание – равномерное распределение влаги по грунту. Вывод надписи «STOP». Время нахождения в режиме ожидания указывается в скетче (приложение 3).

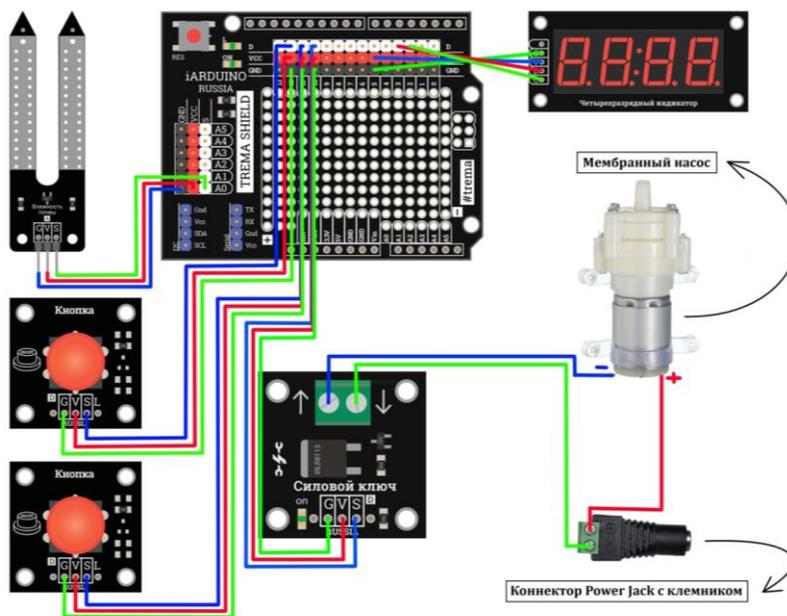


Рис. 10 – выполнен в Adobe Photoshop CC

Подключение элементов проекта (рис. 10): кнопки подключены к цифровым выводам 11 и 12, силовой ключ к цифровому выводу 10 (с ШИМ), датчик влажности почвы к аналоговому входу A0. В теоретической части моего проекта, был рассмотрен один из способов управления работы теплицы с помощью кнопок (стр. 5)

4. Объединим этапы в единую систему.
5. Создадим скетч, занесем в него код из листинга 1 и загрузим скетч на плату Arduino Uno.
6. Управление устройством с телефона (скетч – приложение 4).

Для реализации данной модели, нам необходим Bluetooth модуль HC-05. С его подключением вы можете ознакомиться на рис. 11. Включаем Bluetooth на телефоне, ищем новое устройство "HC-05" и подключаемся к нему.

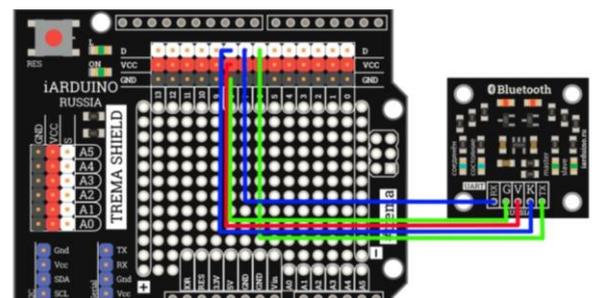


Рис. 11 – выполнен в Adobe Photoshop CC

Затем скачаем приложение «Blynk», преимущества которого описаны в теоретической части моего проекта (стр. 5). Выведем на экран смартфона необходимые нам данные: значение температуры и влажности почвы; вкл/выкл фитолампы и полива; установка минимальной влажности почвы и длительности полива; этап работы полива – как показано на рис. 12. Теперь мы можем автоматически регулировать работу «Теплицы для комнатных растений» и собирать её статистику через данное приложение.

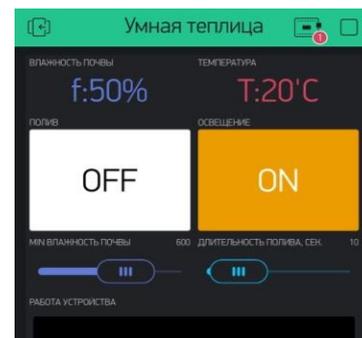


Рис. 12 – скриншот экрана на телефоне

7. Проверка полезности «Теплицы для комнатных растений».

На данных фотографии вы можете оценить полезность моего устройства, увидеть, как оно выглядит и сколько место занимает на столе.

Лучше ли растению? (эксперимент 7 дней)

1 День:



2 День:



3 День:



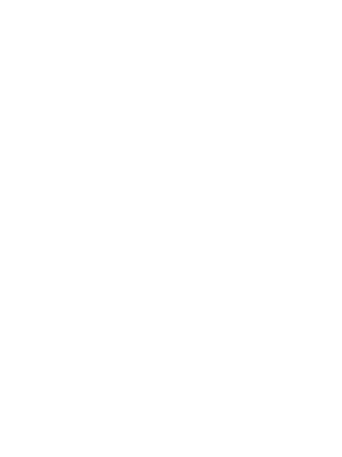
4 День:



5 День:



6 День:





7 День: Объект исследования – цитрус лимон (*C. limon*) – вечнозеленое плодовое дерево, происходящее из Юго-Восточной Азии. У него очередные, обычно широкоовальные, слегка заостренные листья. *Освещение:* светолубивые;

Полив: умеренный весь год; *Температура:* умеренная

Субстрат: дерновая и листовая земля, перегной, торф и песок разными порциями.

В ходе данного эксперимента, растение компенсировало нехватку дневного света и сохранило свой биоритм, необходимый для здорового роста и правильной деятельности всех его органов.

4. Заключение

Основная цель проекта полностью достигнута: создан действующий автономный макет домашней теплицы. В своей работе я, охватив множество аспектов научно-исследовательской деятельности, раскрыла практическую и теоретическую значимость ухода за растениями. Разумеется, я не

ограничиваюсь данным исследованием приведённой проблемы, надеюсь продолжить его в дальнейшем.

Ценность моей работы проявляется в двух её главных качествах:

- практическая ценность,
- научная ценность.

Оценивая значимость данного проекта для практического внедрения в сельское хозяйство или индивидуальное пользование, следует отметить, что у человека появилась возможность в режиме реального времени наблюдать за своими растениями, отслеживать параметры работы теплицы, отвечающие за их здоровье и, при необходимости, вмешиваться в автономную работу теплицы.

5. Список литературы

1. Гололобов В. Н. «Умный дом» своими руками. – М., 2007, стр.
2. Кашкаров А.П. Электронные схемы для "умного дома". - М.: НТ Пресс, 2007. - 256 стр.
3. Курдюмов Н.И., Малышевский К.Г. Умная теплица - М.: Владис, 2007. - 37 стр.
4. Роберт К. Элсенпитер, Тоби Дж. Велт Умный Дом строим сами. - М.: Кудиц-Образ, 2004. - 362 стр.
5. Массимо Банци. Arduino для начинающих волшебников. - М.: Рид Групп, 2012. - 128 стр.
6. Б.Н. Головкин, Е.С. Колобов, Л.П. Косюченко. Все о комнатных растениях, 2003. - 366 стр.
7. Тимошенко К. Фитолампа. - http://mbtclub.ru/index.php?route=product/product&path=97&product_id=160
8. <http://arduino.cc/>
9. <http://amperka.ru/page/what-is-arduino>
10. <http://farmerforum.ru/viewtopic.php?t=4>

6. Приложение

6.1. Приложение 1.

```
#include <Adafruit_NeoPixel.h>
#define PIN 7
#define count_led 12 // количество светодиодов
Adafruit_NeoPixel pixels = Adafruit_NeoPixel(count_led, PIN, NEO_GRB +
NEO_KHZ800);
void setup() {
pixels.begin();
pixels.show();} // Все светодиоды в состояние "Выключено"
void loop() {
pixels.setPixelColor(1, pixels.Color(0,150,0)); // светодиод цвет "Зеленый"
pixels.setPixelColor(2, pixels.Color(250,150,0)); // светодиод цвет "Красный"
pixels.setPixelColor(3, pixels.Color(0,0,250)); // светодиод цвет "Синий"
pixels.show(); }
```

6.2. Приложение 2.

```
#include <LiquidCrystal.h> // символьный ЖК дисплей
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int sensorPin1 = A1;
void setup() {
lcd.begin(16, 2); }
void loop() {
int reading = analogRead(sensorPin);
float voltage = reading * 5.0;
voltage /= 1024.0;
Serial.print(voltage);
Serial.println(" volts");
float temperatureC = (voltage - 0.5) * 100 ;
lcd.setCursor(0,0);
lcd.print("T: ");
lcd.setCursor(3, 0);
lcd.print(temperatureC);
delay(1000); }
```

6.3. Приложение 3.

```
#include <iarduino_4LED.h> // библиотека четырёхразрядного индикатора
iarduino_4LED dispLED(2,3); // указанием выводов индикатора (CLK , DIO)
```

```

const uint8_t pinSensor2 = A0; // датчик влажности почвы
const uint8_t pinButtonL = 12; // кнопка L
const uint8_t pinButtonM = 11; // кнопка M
const uint8_t pinPump = 10; // силовой ключ ( вывод с ШИМ )
uint8_t btnState; // переменная для хранения состояний кнопок: 0 - не нажаты,
1- нажата L, 2 - нажата M, 3 – нажаты две, 4 - удерживается L, 5 - удерживается
M, 6 – удерживаются две
uint16_t arrMoisture[10]; // массив для хранения 10 последних значений
влажности почвы
uint32_t valMoisture; // переменная среднего значения влажности почвы
uint32_t timWatering; // переменная времени начала последнего полива
uint32_t timSketch; // переменная времени прошедшего с момента старта
скетча
const uint8_t timWaiting = 60; // константа времени ожидания после полива
const uint8_t pwmPump = 100; // константа скорости вращения мотора
насоса
uint16_t timDuration = 5; // переменная длительности полива
uint16_t limMoisture = 0; // переменная минимальной влажности почвы
uint8_t modState = 0; // переменная для хранения состояния устройства: 0 -
не активно, 1 - ожидание, 2 - активно, 3 - полив, 4 - установка пороговой
влажности, 5 - установка времени полива
void setup(){
dispLED.begin(); // четырёхразрядный индикатор
pinMode(pinButtonL, INPUT); // переводим вывод pinButtonL в режим входа
pinMode(pinButtonM, INPUT); // переводим вывод pinButtonM в режим входа
pinMode(pinPump, OUTPUT); // переводим вывод pinPump в режим выхода
digitalWrite(pinPump, LOW); // выключаем насос
timWatering = 0; // сбрасываем время начала последнего полива
void loop(){
//Считывание значений:
btnState = Func_buttons_control(); // состояние кнопок
timSketch = millis(); // текущее время с момента старта скетча
if(timWatering>timSketch){timWatering=0;} // время начала последнего полива
= 0,
valMoisture = 0; for(int i=0; i<9; i++){arrMoisture[i]=arrMoisture[i+1];}
arrMoisture[9]=analogRead(pinSensor2); for(int i=0; i<10;
i++){ valMoisture+=arrMoisture[i];} valMoisture/=10; // вычисляем среднее
значение влажности почвы
//Управление устройством

```

```

switch(modState){
    // Устройство не активно
    case 0: if(btnState){ // если зафиксировано нажатие или удержание кнопок
if(btnState==6){modState=4;}
if(btnState==3){modState=2; limMoisture=valMoisture;} }
if(timSketch%100==0){ // если начинается десятая доля секунды
if(timSketch/1000%2){dispLED.print(valMoisture);}else{dispLED.print(" ");} }
break;

    // Устройство в режиме ожидания
    case 1: if(btnState){ // если зафиксировано нажатие или удержание кнопок
if(btnState==6){modState=4;}
if(btnState==1){modState=2;}
if(btnState==2){modState=2;}
if(btnState==3){modState=2;} }
if(timSketch%100==0){ // если начинается десятая доля секунды
dispLED.print("stop");
dispLED.point((timSketch/100%4)+1,true);}
if(timDuration+timWaiting-((timSketch-timWatering)/1000)<=0){ // если
закончилось время ожидания
modState=2;}
break;

    // Устройство активно
    case 2: if(btnState){ // если зафиксировано нажатие или удержание кнопок
if(btnState==6){modState=4; dispLED.light(7);} }
if(timSketch%100==0){ // если начинается десятая доля секунды
if(timSketch/1000%15<5 ){dispLED.light(7); dispLED.print(valMoisture);}else
if(timSketch/1000%15<10){dispLED.light(1);
dispLED.print(limMoisture,LEN4);}else
{dispLED.light(7); if(timWatering){dispLED.print(int((timSketch-
timWatering)/1000%3600/60),int(( timSketch-timWatering)/1000%3600%60),
TIME);}else{dispLED.print("----");} } }
if(valMoisture<=limMoisture){ // если текущая влажность почвы меньше
минимальной
timWatering=timSketch; modState=3; dispLED.light(7);
analogWrite(pinPump,pwmPump);}
break;

    // Устройство в режиме полива
    case 3: if(btnState){ // если зафиксировано нажатие или удержание кнопок
if(btnState==6){modState=4;}else{modState=1;} analogWrite(pinPump,0);}

```

```

if(timSketch%100==0){ // если начинается десятая доля секунды
dispLED.print(timDuration-((timSketch-timWatering)/1000));
dispLED.point(0,true);
dispLED.point((timSketch/100%4)+1,true); }
if(timDuration-((timSketch-timWatering)/1000)<=0){// если закончилось время
полива
modState=1; analogWrite(pinPump,0); }
break;
// Устройство в режиме установки минимальной влажности почвы
case 4: if(btnState){ // если зафиксировано нажатие или удержание кнопок
if(btnState==6){ modState=5; }
if(btnState==1){ if(limMoisture>0 ){limMoisture--;} }
if(btnState==2){ if(limMoisture<999){limMoisture++;} }
if(btnState==3){ limMoisture=valMoisture; }
if(btnState==4){ while(digitalRead(pinButtonL)){ if(limMoisture>0 ){limMoisture--
;} delay(100); dispLED.print(limMoisture);} }
if(btnState==5){ while(digitalRead(pinButtonM)){ if(limMoisture<999){limMoisture
++;} delay(100); dispLED.print(limMoisture);} } }
if(timSketch%100==0){ // если начинается десятая доля секунды
dispLED.print(limMoisture); }
break;
// Устройство в режиме установки длительность полива
case 5: if(btnState){ // если зафиксировано нажатие или удержание кнопок
if(btnState==6){ modState=2; }
if(btnState==1){ if(timDuration>0 ){timDuration--;} }
if(btnState==2){ if(timDuration<99){timDuration++;} }
if(btnState==4){ while(digitalRead(pinButtonL)){ if(timDuration>0 ){timDuration--;}
delay(100); dispLED.print(timDuration);} }
if(btnState==5){ while(digitalRead(pinButtonM)){ if(timDuration<99){timDuration++
;} delay(100); dispLED.print(timDuration);} } }
if(timSketch%100==0) // если начинается десятая доля секунды
{ dispLED.print(timDuration); dispLED.point(0,true); }
break; } }
// Функция определения состояния кнопок
uint8_t Func_buttons_control(){
uint8_t a=0, b=0; // время удержания кнопок L и M (в десятых долях секунды)
while(digitalRead(pinButtonL)||digitalRead(pinButtonM)){ // если нажата кнопка L
и/или кнопка M, то создаём цикл, пока они нажаты

```

```

if(digitalRead(pinButtonL)){if(a<200){a++;}} // если удерживается кнопка L, то
увеличиваем время её удержания
if(digitalRead(pinButtonM)){if(b<200){b++;}} // если удерживается кнопка M, то
увеличиваем время её удержания
if(a>20 && b>20){dispLED.print("----");} // если обе кнопки удерживаются
дольше 2 секунд, выводим на экран прочерки, указывая что их пора отпустить
if(a>20 && b==0){return 4;} // если кнопка L удерживается дольше 2 секунд,
возвращаем 4
if(a==0 && b>20){return 5;} // если кнопка M удерживается дольше 2 секунд,
возвращаем 3
delay(100); } // задержка на 0,1 секунды, для подавления дребезга
if(a>20 && b>20){return 6;} // если обе кнопки удерживались дольше 2
секунд, возвращаем 6
if(a> 0 && b> 0){return 3;}else // если обе кнопки удерживалась меньше 2
секунд, возвращаем 5
if(a> 0 && b==0){return 1;}else // если кнопка L удерживалась меньше 2
секунд, возвращаем 2
if(a==0 && b> 0){return 2;}else // если кнопка M удерживалась меньше 2
секунд, возвращаем 1
{return 0;} // если ни одна из кнопок не была нажата, возвращаем 0}

```

6.4. Приложение 4.

```

#define BLYNK_PRINT Serial
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>
char auth1 = "ArduinoUno";
const uint8_t pinSensor1 = A1; // датчик температуры
const uint8_t pinSensor2 = A0; // датчик влажности почвы
const uint8_t pinPump = A10; // силовой ключ
#define PIN 7; // фитолампа
V1 = uint16_t timDuration; // переменная длительности полива
V2 = uint16_t limMoisture ; // переменная минимальной влажности почвы
uint8_t modState ; // переменная для хранения состояния устройства: 0 - не
активно, 1 - ожидание, 2 - активно, 3 - полив, 4 - установка пороговой
влажности, 5 - установка времени полива
WidgetTerminal terminal(V3);
void setup()

```

```
{Serial.begin(9600);  
Blynk.begin();  
BLYNK_CONNECTED() {Blynk.syncVirtual(A7);}   
BLYNK_CONNECTED() {Blynk.syncVirtual(A10);}   
BLYNK_READ(A0)   
BLYNK_READ(A1)   
BLYNK_WRITE(A7) {int buttonState = param.asInt();}   
BLYNK_WRITE(A10) {int buttonState = param.asInt();}   
{Blynk.virtualWrite(A10, val);}   
{Blynk.virtualWrite(A7, val);}   
{Blynk.virtualWrite(A1, val);}   
{Blynk.virtualWrite(A0, val);}   
Blynk.virtualWrite(uint16_t limMoisture, 1000);   
Blynk.virtualWrite(uint16_t timDuration, 120);   
BLYNK_WRITE(V3)   
terminal.println(F("uint8_t modState "));   
terminal.flush();}   
{Blynk.run();}
```