**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 6](#_Toc184131204)

[1 ОБЗОР ЛИТЕРАТУРЫ 7](#_Toc184131205)

[1.1 Аналоги устройства 7](#_Toc184131206)

[1.2 Основные компоненты 9](#_Toc184131207)

[1.3 Плата с микроконтроллером 10](#_Toc184131208)

[1.4 Датчик температуры и влажности воздуха 10](#_Toc184131209)

[1.5 Датчик влажности почвы 10](#_Toc184131210)

[1.6 Дисплей 10](#_Toc184131211)

[1.7 Элементы управления 11](#_Toc184131212)

[1.8 Водяная помпа 11](#_Toc184131213)

[1.9 Светодиодная лента 11](#_Toc184131214)

[2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА 12](#_Toc184131215)

[2.1 Перечень блоков 12](#_Toc184131216)

[2.2 Взаимодействие блоков 12](#_Toc184131217)

[3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА 14](#_Toc184131218)

[3.1 Аппаратная платформа 14](#_Toc184131219)

[3.2 Датчик температуры и влажности 14](#_Toc184131220)

[3.3 Датчик влажности почвы 14](#_Toc184131221)

[3.4 Датчик освещения 15](#_Toc184131222)

[3.5 Дисплей 15](#_Toc184131223)

[3.6 Светодиодная лента 15](#_Toc184131224)

[3.7 Водяная помпа 16](#_Toc184131225)

[3.8 Управление и структура меню 16](#_Toc184131226)

[4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА 18](#_Toc184131227)

[4.1 Разработка системы питания 18](#_Toc184131228)

[4.2 Описание входов и выходов микроконтроллера 19](#_Toc184131229)

[5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 20](#_Toc184131230)

[5.1 Требования к программе 20](#_Toc184131231)

[5.2 Текст программы 20](#_Toc184131232)

[6 РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ 21](#_Toc184131233)

[6.1 Основные этапы создания печатной платы 21](#_Toc184131234)

[ЗАКЛЮЧЕНИЕ 22](#_Toc184131235)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc184131236)

[ПРИЛОЖЕНИЕ А 24](#_Toc184131237)

[ПРИЛОЖЕНИЕ Б 25](#_Toc184131238)

[ПРИЛОЖЕНИЕ В 26](#_Toc184131239)

[ПРИЛОЖЕНИЕ Г 27](#_Toc184131240)

[ПРИЛОЖЕНИЕ Д 28](#_Toc184131241)

[ПРИЛОЖЕНИЕ Е 35](#_Toc184131242)

[ПРИЛОЖЕНИЕ Ж 36](#_Toc184131243)

[ПРИЛОЖЕНИЕ И 37](#_Toc184131244)

ВВЕДЕНИЕ

Микроклимат является критически важным фактором, влияющим на здоровье и развитие растений. Основные параметры, такие как температура и влажность воздуха, влажность почвы и уровень освещенности, играют ключевую роль в обеспечении оптимальных условий для роста растений. Нарушение допустимых границ этих показателей может привести к снижению урожайности, ухудшению состояния растений и их гибели.

Современные системы контроля микроклимата позволяют не только отслеживать ключевые параметры, но и автоматизировать процессы, такие как полив и освещение. В рамках данного курсового проекта разрабатывается устройство, которое будет выполнять следующие задачи:

– измерение температуры и влажности воздуха: устройство будет измерять эти параметры и выводить информацию на дисплей в градусах Цельсия и процентах;

– измерение влажности почвы: параметр будет отображаться в процентах, что позволит точно контролировать состояние почвы;

– измерение уровня освещенности: устройство будет фиксировать уровень освещения и выводить его на дисплей;

– автоматическое освещение и полив: при необходимости устройство будет включать освещение и автоматическую систему полива, обеспечивая оптимальные условия для растений;

– настройка допустимых параметров: с помощью кнопок пользователи смогут настраивать границы допустимых значений для каждого из параметров.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Аналоги устройства

В настоящее время на рынке представлено множество устройств для контроля микроклимата растений, каждое из которых имеет свои уникальные функции и особенности. Рассмотрим три популярных аналога, их преимущества и недостатки.

**1.1.1 Xiaomi Smart Flower Pot**

Xiaomi Smart Flower Pot — это современное устройство, которое сочетает в себе функции контроля микроклимата и автоматического полива. Оно оснащено датчиками, которые измеряют влажность почвы, освещенность и температуру воздуха. Устройство легко интегрируется с экосистемой умного дома Xiaomi, что позволяет пользователям управлять микроклиматом через мобильное приложение.

Среди его достоинств можно отметить наличие автоматической системы полива, которая помогает поддерживать оптимальный уровень влажности. Однако, несмотря на свои преимущества, устройство имеет и недостатки. Оно не поддерживает измерение углекислого и угарного газа, что ограничивает контроль за качеством воздуха. Кроме того, высокая цена может стать препятствием для пользователей с ограниченным бюджетом.

На рисунке 1.1 изображено Xiaomi Smart Flower Pot.



Рисунок 1.1 – Xiaomi Smart Flower Pot

**1.1.2 Parrot Flower Power**

Parrot Flower Power — это удобное устройство для ухода за растениями, которое позволяет отслеживать их состояние через Bluetooth. Оно оснащено датчиками для измерения влажности почвы, освещённости, температуры и содержания удобрений, что помогает поддерживать оптимальные условия для роста растений. Устройство синхронизируется с мобильным приложением, предоставляющим актуальную информацию и рекомендации по уходу через интуитивно понятный интерфейс.

Одним из главных преимуществ Parrot Flower Power является его долговечность — датчики могут работать без подзарядки до 6 месяцев. Однако устройство имеет свои ограничения: оно не поддерживает автоматические функции, такие как полив и освещение, а также требует периодической замены батареи, что может быть неудобно для пользователей с большим количеством растений.

На рисунке 1.2 изображено устройство Parrot Flower Power.



Рисунок 1.2 – Устройство Parrot Flower Power

**1.1.3 PlantLink**

PlantLink — это современная система контроля за состоянием растений, разработанная для автоматизации полива и облегчения ухода за растениями. Она состоит из набора высокочувствительных датчиков, которые отслеживают уровень влажности почвы, и центрального блока управления, который регулирует подачу воды в соответствии с заданными параметрами. Благодаря подключению к мобильному приложению, пользователи могут с лёгкостью настраивать параметры полива и контролировать состояние почвы в режиме реального времени.

Одним из ключевых преимуществ PlantLink является автоматизация процесса полива. Устройство способно своевременно обеспечивать растения водой, что значительно сокращает необходимость в ручном уходе, особенно для тех, кто занят или отсутствует длительное время. Мобильное приложение также предоставляет актуальную информацию и рекомендации по уходу, помогая создать оптимальные условия для роста растений.

Тем не менее, у PlantLink есть и определенные ограничения. Например, система не отслеживает температуру окружающей среды и уровень освещённости, что может быть недостатком для ухода за растениями, требующими комплексного подхода. Также для работы устройство требует постоянного подключения к Wi-Fi, что может быть неудобно в местах с ограниченным доступом к сети или при наличии перебоев в соединении.

На рисунке 1.3 изображено устройство PlantLink.

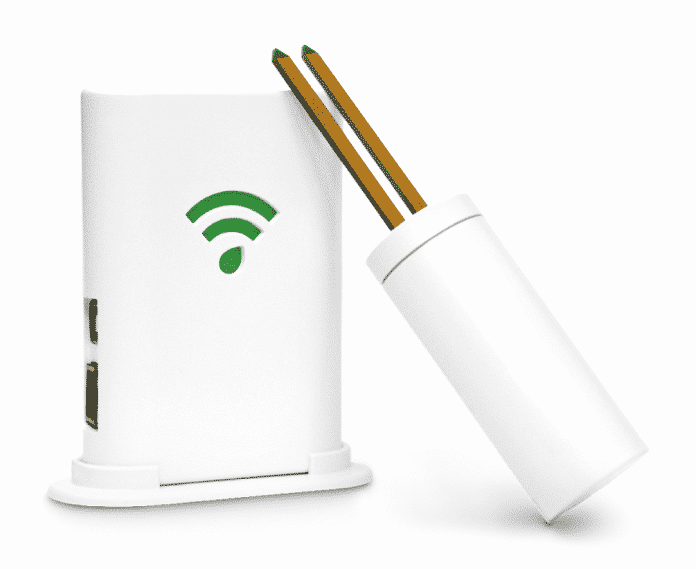


Рисунок 1.3 – Устройство PlantLink

1.2 Основные компоненты

Основные компоненты устройства:

– плата с микроконтроллером;

– датчик температуры и влажности воздуха;

– датчик влажности почвы;

– фоторезистор;

– жидкокристаллический дисплей;

– водяная помпа;

– светодиодная лента;

– 3 кнопки.

1.3 Плата с микроконтроллером

Среди микроконтроллеров доступных на рынке была рассмотрена плата Arduino NANO V3.0 CH340. Основные характеристики:

– микроконтроллер ATmega328;

– напряжение питания 5В;

– входное напряжение (рекомендуемое) 7-12В;

– входное напряжение (предельное) 6-20В;

– цифровой ввод-вывод 14 линии (6 поддерживают ШИМ);

– аналоговый ввод 8 линий;

– постоянный ток на линиях ввода-вывода 40мА;

– постоянный ток на линии 3.3В 50мА;

– flash-память 32КВ, 2 КВ из них использованы для загрузчика;

– SRAM-память 2КВ;

– EEPROM-память 1КВ;

– тактовая частота 16МГц [1].

1.4 Датчик температуры и влажности воздуха

На рынке распространены устройства, которые совмещают в себе и датчик температуры и датчик влажности. Одним из таких решений являются датчики серии DHT. Был рассмотрен DHT11. Основные характеристики:

– питание: DC 3,5 – 5,5 В;

– ток питания: в режиме измерения 0.3mA. В режиме ожидания 60μA;

– определение влажности 20–80 % с точностью 5 %;

– определение температуры 0–50 °С c точностью 2 %;

– частота опроса не более 1 Гц [2].

1.5 Датчик влажности почвы

Для определения влажности почвы был выбран датчик Water Sensor Detector. Этот тип датчика прост в использовании и предоставляет информацию о текущем состоянии влажности почвы, что помогает оптимизировать процесс полива. Основные характеристики:

– питание: DC 3,3 – 5 В;  
– ток потребления: менее 20 мА;  
– тип выхода: аналоговый сигнал;  
– материал электродов: антикоррозийные покрытия;

– время отклика: менее 1 секунды [3].

1.6 Дисплей

Для реализации вывода информации с датчиков в понятной для человека форме в проекте необходим дисплей. Был рассмотрен жидкокристаллический дисплей LCD1602, отображающий ограниченное количество символов. Основные характеристики:

– цвет подсветки: Синий;

– количество символов в строке: 16;

– количество строк: 2;

– язык: по умолчанию поддерживает латиницу;

– интерфейсы: Arduino;

– напряжение питания: 5В [4].

1.7 Элементы управления

Для управления устройством был выбран способ управления тремя кнопками, с помощью которых пользователь может регулировать граничные значения параметров микроклимата.

1.8 Водяная помпа

Для системы автоматического полива в проекте была выбрана помпа DC 12V 120L-H, благодаря её простоте использования и надежности. Эта помпа идеально подходит для задач, связанных с поливом растений, так как работает на низком напряжении, что обеспечивает безопасность в эксплуатации. Основные характеристики:

– напряжение питания: DC 9 - 12 В;

– ток потребления: 100 - 200 мА;

– высота подъема воды: 0.3 - 0.8 м;

– производительность: 120-180 л/ч;

– диаметр выходной трубки: внутренний 4.5 мм, внешний 7 мм;

– диаметр входной трубки: внутренний 4.5 мм, внешний 6.8 мм.

1.9 Светодиодная лента

В проекте для освещения будет использоваться светодиодная лента LS-3528W60 благодаря её универсальности и легкости установки. Гибкая структура позволяет адаптировать ленту к различным формам, а её влагостойкость делает её подходящей для условий с повышенной влажностью. Основные характеристики:

– цвет: Белый;

– влагозащита: IP65;

– количество светодиодов: 60 шт/м;

– мощность: 3,6 Вт;

– ширина: 8 мм;

– тип ленты: неуправляемая;

– напряжение питания: 9В.

2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

Структурная схема устройства приведена в приложении А.

2.1 Перечень блоков

В данном устройстве можно выделить 7 основных блоков:

– блок датчиков;

– микроконтроллер;

– блок полива;

– блок освещения;

– блок управления;

– блок отображения информации;

– блок питания.

В состав блока датчиков входят датчики влажности и температуры воздуха, влажности почвы, освещенности.

Блок полива состоит из помпы, которая будет обеспечивать необходимый уровень влажности почвы.

Блок освещения состоит из светодиодной ленты, которая будет обеспечивать необходимое освещение.

В состав блока управления входят кнопки, с помощью которых пользователь может управлять режимами работы помпы и светодиодной ленты, регулировать граничные значения параметров микроклимата.

Блок отображения информации состоит из жидкокристаллического дисплея.

Блок питания обеспечивает необходимое напряжение для всех компонентов устройства.

2.2 Взаимодействие блоков

Датчики снимают показания влажности воздуха, температуры воздуха, влажности почвы и уровня освещенности, передавая эту информацию в микроконтроллер.

Микроконтроллер обрабатывает данные, полученные от датчиков, и передаёт их в блок отображения информации, где они визуализируются для пользователя. Если значения, переданные датчиками, выходят за допустимые границы, микроконтроллер принимает решение и посылает сигналы на включение или выключение соответствующих исполнительных устройств, таких как блоки полива и освещения.

Блок управления позволяет пользователю настраивать пороговые значения для параметров, таких как минимальная влажность почвы или уровень освещенности. Эти параметры передаются микроконтроллеру, который на основе этих данных регулирует работу блоков полива и освещения, а также обновляет информацию, передаваемую на блок отображения.

Блок отображения информации преобразует данные, переданные микроконтроллером, в удобную для пользователя форму, показывая текущие значения параметров микроклимата — влажность, температуру и освещенность.

Блок полива, получая команды от микроконтроллера, отвечает за автоматическое увлажнение почвы. Он активируется в случае, если уровень влажности падает ниже заданного порога. Это позволяет поддерживать оптимальные условия для роста растений, предотвращая их пересыхание и обеспечивая необходимое количество влаги.

Блок освещения управляется микроконтроллером и включается при низком уровне освещенности. Он обеспечивает растениям достаточное количество света, что особенно важно в условиях недостатка естественного солнечного света.

Блок питания отвечает за снижение входного напряжения до необходимого уровня для работы различных компонентов системы. Он преобразует высокое входное напряжение в более низкое, подходящее для питания таких элементов, как датчики, микроконтроллер и блоки управления.

3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА

Функциональная схема устройства приведена в приложении Б.

3.1 Аппаратная платформа

Для курсового проекта была выбрана плата Arduino Nano благодаря своей компактности и удобству в использовании. Она легко интегрируется с различными сенсорами, что позволяет создать эффективную систему мониторинга, адаптированную под конкретные задачи. Благодаря своей архитектуре и большому количеству доступных библиотек, Arduino Nano также проста в программировании и совместима с Arduino IDE, что значительно упрощает процесс разработки.

3.2 Датчик температуры и влажности

Для курсового проекта был выбран датчик DHT11 благодаря его надежности и экономичности. Этот датчик позволяет измерять как температуру, так и влажность воздуха, что критически важно для мониторинга условий, в которых находятся растения. DHT11 прост в использовании и легко интегрируется с платами Arduino, такими как Nano. Его подключение не требует сложных схем, что делает его подходящим для образовательных проектов и прототипов.

Датчик использует однопроводной интерфейс для передачи данных, что упрощает подключение и минимизирует количество необходимых проводов. Хотя DHT11 не является самым точным датчиком на рынке, его точности достаточно для базового контроля микроклимата. Частота опроса составляет около 1 раза в секунду, что, хотя и не является высокой, полностью удовлетворяет требованиям для контроля параметров микроклимата. Такой режим позволяет регулярно получать актуальные данные о температуре и влажности, что способствует своевременному реагированию на изменения в условиях содержания растений.

3.3 Датчик влажности почвы

Для измерения влажности почвы был выбран Water Sensor Detector благодаря его простоте использования, доступности и способности точно измерять уровень влажности. Датчик работает по принципу делителя напряжения, где его сопротивление изменяется в зависимости от уровня влажности почвы. При повышении влажности сопротивление датчика уменьшается, что приводит к увеличению выходного напряжения. Считываемое выходное напряжение позволяет определить, насколько влажна почва.

Water Sensor Detector легко интегрируется с микроконтроллером, обеспечивая надежные данные для автоматизации полива. Такой подход позволяет эффективно контролировать уровень влаги в почве, что критически важно для поддержания оптимальных условий для роста растений.

3.4 Датчик освещения

В качестве датчика освещения был выбран фоторезистор, так как он представляет собой более экономичное решение по сравнению со специализированными датчиками. Фоторезистор прост в использовании и не требует сложной настройки, что делает его идеальным выбором.

Работа фоторезистора основана на изменении его сопротивления в зависимости от уровня освещения: при увеличении яркости света сопротивление фоторезистора снижается. Для подключения к микроконтроллеру используется делитель напряжения, что позволяет легко преобразовать изменения сопротивления в выходное напряжение. Это выходное напряжение можно быстро считывать с микроконтроллера, что дает возможность эффективно контролировать уровень освещения в окружающей среде.

3.5 Дисплей

LCD1602 был выбран как дисплей для системы, поскольку он сочетает доступность, удобство использования и наглядное отображение данных. Экран с двумя строками по 16 символов предоставляет достаточно места для показа основных параметров, таких как влажность, температура и уровень освещенности, что позволяет пользователю легко отслеживать важные показатели в реальном времени. Дисплей удобно подключается к микроконтроллеру как через параллельный интерфейс, так и через I2C-модуль, что даёт гибкость в использовании портов ввода-вывода и облегчает интеграцию в систему. Настраиваемый потенциометр позволяет регулировать контрастность экрана, обеспечивая хорошую читаемость в разных условиях освещенности. Поддержка библиотек, таких как LiquidCrystal для Arduino, делает программирование LCD1602 простым и быстрым, что минимизирует время на разработку интерфейса.

3.6 Светодиодная лента

Для курсового проекта была выбрана светодиодная лента LS-3528W60 благодаря своей универсальности и простоте установки. Эта лента обеспечивает эффективное освещение, необходимое для создания оптимальных условий для роста растений.

Лента работает от внешнего питания, и для её управления используется напряжение с микроконтроллера, которое подается на транзистор. Транзистор выполняет роль переключателя, позволяя контролировать подачу питания на светодиоды. Это обеспечивает возможность включения и выключения ленты.

Гибкая конструкция ленты позволяет адаптировать её к различным формам и размерам, а также устанавливать в ограниченных пространствах. LS-3528W60 обладает влагостойкостью, что делает её подходящей для использования в условиях повышенной влажности.

3.7 Водяная помпа

Для курсового проекта была выбрана помпа DC 12V 120L-H, благодаря своей компактности и простоте эксплуатации. Этот компонент идеально подходит для автоматизации полива, обеспечивая растения необходимым количеством влаги.

Помпа работает от внешнего источника питания и управляется с помощью транзистора, который позволяет контролировать её включение и выключение. Хотя мощность помпы не очень высокая, её вполне достаточно для выполнения задач по поливу, обеспечивая стабильную подачу воды.

3.8 Управление и структура меню

В проекте предусмотрено управление системой через интуитивно понятное меню, которое позволяет пользователю легко настраивать параметры микроклимата и полива. Структура меню организована так, чтобы обеспечивать максимальную простоту и доступность. Для навигации по меню используются кнопки, которые позволяют легко перемещаться между выбранными пунктами. Пользователь может просматривать текущие значения параметров и вносить изменения при необходимости. Также предусмотрены функции активации и деактивации автоматического полива и освещения, что дает возможность адаптировать систему под конкретные условия. Структура меню представлена в таблице 3.1.

Таблица 3.1 – Структура меню

|  |  |  |  |
| --- | --- | --- | --- |
| Уровень меню | Кнопка 1 | Кнопка 2 | Кнопка 3 |
| Главное меню | Переключение на управление светодиодной лентой | – | Включение и выключение подсветки LCD |
| Управление светодиодной лентой | Переключение на управление помпой или порог света (если AUTO) | Переключение режима светодиодной ленты: OFF-AUTO-ON | Переключение режима светодиодной ленты: OFF-ON-AUTO |
| Порог света | Переключение на управление помпой | Уменьшение значения порога света | Увеличение значения порога света |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| Уровень меню | Кнопка 1 | Кнопка 2 | Кнопка 3 |
| Управление помпой | Переключение на главное меню или порог влажности (если AUTO) | Переключение режима помпы: OFF-AUTO-ON | Переключение режима помпы: OFF-ON-AUTO |
| Порог влажности | Переключение на главное меню | Уменьшение значения порога влажности | Увеличение значения порога влажности |

Кнопка 1 используется для переключения уровней меню, позволяя пользователю переходить между различными функциями и настройками системы.

Кнопка 2 отвечает за уменьшение параметров и переключение режимов работы светодиодной ленты и помпы.

Кнопка 3 отвечает за увеличение параметров и переключение режимов работы светодиодной ленты и помпы.

4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ УСТРОЙСТВА

Принципиальная схема устройства приведена в приложении В.

4.1 Разработка системы питания

Для расчёта характеристик блока питания была составлена таблица 4.1.

Таблица 4.1 – Расчет характеристик блока питания

|  |  |  |  |
| --- | --- | --- | --- |
| Название устройства | Рабочее напряжение, В | Максимальный потребляемый ток, мА | Максимальная потребляемая мощность, мВт |
| Arduino Nano | 5 | 20 | 100 |
| Помпа DC 5V 120L-H | 9 | 200 | 1800 |
| Water Sensor Detector | 5 | 20 | 100 |
| LS-3528W60 (длина 0,3 м) | 9 | 120 | 1080 |
| DHT11 | 5 | 0,3 | 1,5 |
| LCD1602 | 5 | 26 | 130 |

Максимальный потребляемый ток устройства составил 386,3 мА, максимальная потребляемая мощность – 3211,5 мВт. Для устройства, разрабатываемого в проекте необходимо обеспечить 2 уровня входного напряжения – 5 В и 9 В. В качестве напряжения источника питания было выбрано 9 В.

При выборе блока питания для устройства, необходимо учитывать максимальный потребляемый ток в 386,3 мА и максимальную потребляемую мощность в 3211,5 мВт. Рекомендуется заложить запас мощности в 20% для обеспечения надежной работы устройства. С учетом этого запаса, максимальная мощность блока питания должна составлять 3853,8 мВт, а максимальный ток – 463,6 мА.

Таким образом, блок питания должен обеспечивать выходное напряжение 9 В и возможность выдачи 463,6 мА на уровне 9 В. Эти характеристики гарантируют надежность и долговечность работы устройства.

Для получения уровня напряжения 5 В будет использоваться линейный стабилизатор напряжения L7805ABV [5]. Он обеспечивает выходной ток до 1,5 А, которого достаточно для обеспечения работы устройства, и фиксированное выходное напряжение 5 В, которое необходимо для питания большинства компонентов устройства. Схема подключения L7805ABV приведена на рисунке 4.1.

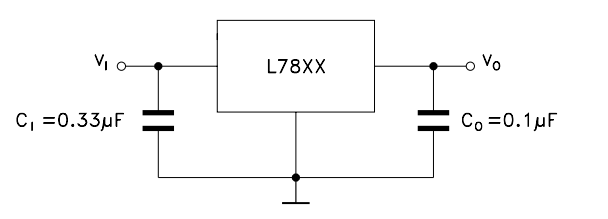


Рисунок 4.1 – Схема подключения L7805ABV

4.2 Описание входов и выходов микроконтроллера

Питание микроконтроллера обеспечивается на входах 5V и GND.

К микроконтроллеру подключены 3 кнопки. Эти кнопки заведены на цифровые входы D2, D3 и D4 соответственно. На данных входах активирован встроенный в плату подтягивающий резистор. Вследствие этого, когда кнопка не нажата, на соответствующем входе регистрируется логическая единица, в противном случае – логический ноль.

Цифровой выход D5 заведен на транзистор, управляющий светодиодной лентой LS3528W60.

Цифровой выход D6 заведен на транзистор, управляющий помпой DC 12V 120L-H.

На цифровой вход D7 подается сигнал DATA с датчика DHT1.

На аналоговые входы A0 и A1 заведены сигналы с фоторезистора и датчика Water Sensor Detector соответственно.

Через аналоговые выходы A4 и A5 по интерфейсу I2C подключен дисплей устройства LCD1602.

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Требования к программе

Программа, управляющая устройством, должна реализовывать следующий функционал:

– периодическое считывание показателей с датчиков;

– обработка считанной информации и вывод на дисплей;

– управление включением и выключением светодиодной ленты;

– управление включением и выключением помпы;

– обработка нажатия кнопок;

– поддержка настройки пользователем режима работы устройства и пороговых значений.

– четкий и понятный интерфейс;

– сохранение пользовательских настроек в энергонезависимой памяти;

– обработка ошибок;

– минимизация потребления энергии.

Схема программы приведена в приложении Г.

5.2 Текст программы

Текст программы приведен в приложении Д.

6 РАЗРАБОТКА ПЕЧАТНОЙ ПЛАТЫ

Печатная плата служит основой для надежного соединения всех компонентов устройства. Она обеспечивает механическую поддержку, упрощает электрические соединения между датчиками, микроконтроллером, помпой и светодиодной лентой, а также ускоряет процесс разработки и отладки. Использование печатной платы повышает устойчивость устройства к внешним воздействиям и организует пространство внутри корпуса, что делает его более аккуратным и компактным.

Схема печатной платы приведена в приложении Е.

6.1 Основные этапы создания печатной платы

Основные этапы создания печатной платы включают в себя:

1 Проектирование платы. Определение требований и использование ПО для проектирования для создания схемы и компоновки.

2 Обработка текстолита.

3 Печать зеркально отраженной схемы на термотрансферной бумаге.

4 Перенос схемы на текстолит. Обеспечивается нагревом термотрансферной бумаги на текстолите.

5 Травление платы в специальном растворе.

6 Очистка платы.

7 Проверка дорожек на наличие повреждений и соответствие схеме.

8 Сверление необходимых отверстий [6].

ЗАКЛЮЧЕНИЕ

В результате работы над курсовым проектом было разработано устройство для контроля параметров микроклимата растений, которое эффективно решает задачи мониторинга и автоматизации. Устройство позволяет измерять ключевые параметры микроклимата, такие как температура и влажность воздуха, влажность почвы и уровень освещенности, отображая их на дисплее. Автоматизация процессов освещения и полива значительно упрощает уход за растениями, создавая оптимальные условия для их роста и развития.

Реализованная система управления позволяет пользователям настраивать допустимые пороги для каждого параметра, что обеспечивает гибкость и адаптивность устройства к различным условиям. Это особенно актуально для домашних садоводов и профессиональных агрономов, стремящихся повысить урожайность и улучшить состояние своих растений.

Разработанное устройство представляет собой эффективное решение для контроля микроклимата, способствуя устойчивому развитию растений и облегчая процесс ухода за ними. В дальнейшем проект может быть расширен за счет интеграции дополнительных функций и улучшения пользовательского интерфейса, что сделает его еще более удобным и полезным.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Спецификация Arduino Nano [Электронный ресурс]. – Режим доступа: https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf

[2] Спецификация датчика DHT11 [Электронный ресурс]. – Режим доступа: http://robocraft.ru/files/datasheet/DHT11.pdf

[3] Спецификация дисплея LCD1602 [Электронный ресурс]. – Режим доступа: https://static.chipdip.ru/lib/157/DOC035157816.pdf

[4] Спецификация датчика влажности Water Sensor Detector [Электронный ресурс]. – Режим доступа: https://curtocircuito.com.br/datasheet/sensor/nivel\_de\_agua\_analogico.pdf

[5] Спецификация стабилизатора напряжения L7805ABV [Электронный ресурс]. – Режим доступа: https://static.chipdip.ru/lib/649/DOC011649092.pdf

[6] Изготовление печатной платы в домашних условиях [Электронный ресурс]. – Режим доступа: https://eax.me/peroxide-etching/

ПРИЛОЖЕНИЕ А

**(обязательное)**

**Схема электрическая структурная**

ПРИЛОЖЕНИЕ Б

**(обязательное)**

**Схема электрическая функциональная**

ПРИЛОЖЕНИЕ В

**(обязательное)**

**Схема электрическая принципиальная**

ПРИЛОЖЕНИЕ Г

**(обязательное)**

**Схема программы**

ПРИЛОЖЕНИЕ Д

**(обязательное)**

**Текст программы**

001 #include <DHT.h>

002 #include <EEPROM.h>

003 #include <LiquidCrystal\_I2C.h>

004

005 // Инициализация LCD дисплея с адресом 0x27 и размерами 16x2

006 LiquidCrystal\_I2C lcd(0x27, 16, 2);

007 bool screenOn = true; // состояние подсветки экрана

008

009 // Определение пинов для кнопок и сенсоров

010 const int button1Pin = 2; // Кнопка для переключения режимов

011 const int button2Pin = 3; // Кнопка для увеличения параметра

012 const int button3Pin = 4; // Кнопка для уменьшения параметра

013 const int dhtPin = 7; // Пин для DHT11

014 const int soilMoisturePin = A1; // Пин для датчика влажности почвы

015 const int lightSensorPin = A0; // Пин для датчика света

016 const int ledPin = 5; // Пин для управления светодиодной лентой

017 const int pumpPin = 6; // Пин для управления помпой

018

019 // Перечисление состояний меню

020 enum MenuState {

021 MAIN,

022 LED\_CONTROL,

023 LIGHT\_THRESHOLD,

024 PUMP\_CONTROL,

025 HUMIDITY\_THRESHOLD

026 };

027

028 MenuState currentState = MAIN; // Текущее состояние меню

029

030 // Перечисление состояний светодиодов

031 enum LEDState {

032 LED\_OFF,

033 LED\_ON,

034 LED\_AUTO

035 };

036

037 LEDState ledState = LED\_OFF; // Начальное состояние светодиодов

038 int lightThreshold = 0; // Порог освещенности

039

040 // Перечисление состояний помпы

041 enum PumpState {

042 PUMP\_OFF,

043 PUMP\_AUTO

044 };

045

046 PumpState pumpState = PUMP\_OFF; // Начальное состояние помпы

047 int humidityThreshold = 0; // Порог влажности

048 const unsigned long pumpRunTime = 2000; // Время работы

049 // Время ожидания перед следующим включением

050 const unsigned long pumpWaitTime = 60000;

051

052 unsigned long lastPumpRunTime = 0; // Время последнего включения помпы

053 bool pumpRunning = false; // Флаг, указывающий, работает ли помпа

054

055 unsigned long lastButtonPress = 0; // Время последнего нажатия кнопки

056 // Задержка для устранения дребезга кнопок

057 const unsigned long debounceDelay = 200;

058

059 // Переменные для хранения данных с сенсоров

060 float humidity = 0.0; // Уровень влажности

061 float temperature = 0.0; // Температура

062 float soilMoisture = 0; // Влажность почвы

063 float lightLevel = 0; // Уровень света

064

065 float lastHumidity = 0.0; // Предыдущее значение влажности

066 float lastTemperature = 0.0; // Предыдущее значение температуры

067

068 DHT dht(dhtPin, DHT11); // Инициализация DHT11

069

070 void setup() {

071 // Настройка пинов кнопок как входов с подтяжкой к питанию

072 pinMode(button1Pin, INPUT\_PULLUP);

073 pinMode(button2Pin, INPUT\_PULLUP);

074 pinMode(button3Pin, INPUT\_PULLUP);

075

076 pinMode(ledPin, OUTPUT); // Настройка пина светодиодов как выхода

077 pinMode(pumpPin, OUTPUT); // Настройка пина помпы как выхода

078 dht.begin(); // Инициализация DHT

079 lcd.begin(16, 2); // Инициализация LCD

080 lcd.backlight(); // Включение подсветки

081 lcd.print("Initializing..."); // Сообщение об инициализации

082 initializeEEPROM(); // Инициализация EEPROM

083 }

084

085 void saveToEEPROM() {

086 // Сохранение текущих параметров в EEPROM

087 EEPROM.put(0, pumpState);

088 EEPROM.put(1, ledState);

089 EEPROM.put(2, lightThreshold);

090 EEPROM.put(3, humidityThreshold);

091 }

092

093 void initializeEEPROM() {

094 // Загрузка параметров из EEPROM

095 EEPROM.get(0, pumpState);

096 EEPROM.get(1, ledState);

097 EEPROM.get(2, lightThreshold);

098 EEPROM.get(3, humidityThreshold);

099

100 // Проверка и корректировка значений

101 if (pumpState != PUMP\_OFF && pumpState != PUMP\_AUTO) {

102 pumpState = PUMP\_OFF;

103 EEPROM.put(0, pumpState);

104 }

105 if (ledState != LED\_OFF && ledState != LED\_ON && ledState !=

106 LED\_AUTO) {

107 ledState = LED\_OFF;

108 EEPROM.put(1, ledState);

109 }

110 if (lightThreshold < 0 || lightThreshold > 100) {

111 lightThreshold = 0;

112 EEPROM.put(2, lightThreshold);

113 }

114 if (humidityThreshold < 0 || humidityThreshold > 100) {

115 humidityThreshold = 0;

116 EEPROM.put(3, humidityThreshold);

117 }

118 }

119

120 void loop() {

121 // Чтение состояния кнопок

122 bool button1State = digitalRead(button1Pin) == LOW;

123 bool button2State = digitalRead(button2Pin) == LOW;

124 bool button3State = digitalRead(button3Pin) == LOW;

125

126 // Обработка нажатия кнопок с учетом задержки

127 if (button1State && millis() - lastButtonPress > debounceDelay) {

128 handleButton1();

129 lastButtonPress = millis();

130 }

131

132 if (button2State && millis() - lastButtonPress > debounceDelay) {

133 handleButton2();

134 lastButtonPress = millis();

135 }

136

137 if (button3State && millis() - lastButtonPress > debounceDelay) {

138 handleButton3();

139 lastButtonPress = millis();

140 }

141

142 // Чтение сенсоров раз в секунду

143 static unsigned long lastReadTime = 0;

144 if (millis() - lastReadTime >= 1000) { // Проверка на каждую секунду

145 readSensors();

146 lastReadTime = millis(); // Обновляем время последнего чтения

147 updateLCD(); // Обновляем информацию на LCD

148 }

149

150 // Управление светодиодной лентой и помпой

151 controlLED();

152 controlPump();

153 delay(100); // Небольшая задержка

154 }

155

156 void updateLCD() {

157 lcd.clear(); // Очистка дисплея

158 switch(currentState) {

159 case MAIN:

160 // Проверка на ошибки чтения данных

161 if (isnan(humidity) || isnan(temperature)) {

162 lcd.setCursor(0, 0); // Сообщение об ошибке

163 lcd.print("DHT ERROR!");

164 } else {

165 // Выводим влажность и температуру на первой строке

166 lcd.setCursor(0, 0);

167 lcd.print("H:");

168 lcd.print(round(humidity));

169 lcd.print("% T:");

170 lcd.print(temperature, 2);

171 lcd.print("C");

172 lastHumidity = humidity; // Сохраняем последнее значение

173 lastTemperature = temperature;

174 }

175 // Выводим влажность почвы и уровень света на второй строке

176 lcd.setCursor(0, 1);

177 lcd.print("Soil:");

178 lcd.print(round(soilMoisture/1023\*100));

179 lcd.print("% L:");

180 lcd.print(round(lightLevel/1023\*100));

181 lcd.print("%");

182 break;

183 case LED\_CONTROL:

184 lcd.setCursor(0, 0);

185 lcd.print("LED: ");

186 lcd.setCursor(0, 1);

187 // Выводим текущее состояние светодиодов

188 if (ledState == LED\_ON) {

189 lcd.print("ON");

190 } else if (ledState == LED\_OFF) {

191 lcd.print("OFF");

192 } else {

193 lcd.print("AUTO");

194 }

195 break;

196

197 case PUMP\_CONTROL:

198 lcd.setCursor(0, 0);

199 lcd.print("Pump: ");

200 lcd.setCursor(0, 1);

201 // Выводим текущее состояние помпы

202 if (pumpState == PUMP\_OFF) {

203 lcd.print("OFF");

204 } else {

205 lcd.print("AUTO");

206 }

207 break;

208

209 case LIGHT\_THRESHOLD:

210 lcd.setCursor(0, 0);

211 lcd.print("Light threshold:");

212 lcd.setCursor(0, 1);

213 lcd.print(lightThreshold); // Выводим порог освещенности

214 break;

215

216 case HUMIDITY\_THRESHOLD:

217 lcd.setCursor(0, 0);

218 lcd.print("Soil threshold:");

219 lcd.setCursor(0, 1);

220 lcd.print(humidityThreshold); // Выводим порог влажности

221 break;

222 }

223 }

224

225 void controlPump() {

226 unsigned long currentTime = millis();

227

228 if (pumpState == PUMP\_AUTO) {

229 // Проверка состояния почвы и управление помпой

230 if (soilMoisture/1023\*100 < humidityThreshold && !pumpRunning) {

231 digitalWrite(pumpPin, HIGH); // Включаем помпу

232 lastPumpRunTime = currentTime; // Запоминаем время включения

233 pumpRunning = true; // Устанавливаем состояние, что помпа работает

234 }

235

236 // Проверяем, нужно ли выключить помпу после работы

237 if (pumpRunning && (currentTime - lastPumpRunTime >= pumpRunTime)) {

238 digitalWrite(pumpPin, LOW); // Выключаем помпу

239 }

240

241 // Проверяем, нужно ли ожидать перед следующим включением

242 if (pumpRunning && (currentTime - lastPumpRunTime >=

243 pumpRunTime + pumpWaitTime)) {

244 pumpRunning = false; // Сбрасываем состояние помпы

245 }

246 } else if (pumpState == PUMP\_OFF) {

247 digitalWrite(pumpPin, LOW); // Выключаем помпу

248 pumpRunning = false; // Обнуляем состояние помпы

249 }

250 }

251

252 void controlLED() {

253 switch (ledState) {

254 case LED\_OFF:

255 digitalWrite(ledPin, LOW); // Выключаем светодиодную ленту

256 break;

257 case LED\_ON:

258 digitalWrite(ledPin, HIGH); // Включаем светодиодную ленту

259 break;

260 case LED\_AUTO:

261 // Автоматическое управление лентой в зависимости от света

262 if (lightLevel/1023\*100 < lightThreshold) {

263 digitalWrite(ledPin, HIGH); // Включаем при недостатке света

264 } else {

265 digitalWrite(ledPin, LOW); // Выключаем при достаточном свете

266 }

267 break;

268 }

269 }

270

271 void readSensors() {

272 // Чтение данных с DHT11

273 humidity = dht.readHumidity();

274 temperature = dht.readTemperature();

275 // Чтение влажности почвы

276 soilMoisture = analogRead(soilMoisturePin);

277 lightLevel = analogRead(lightSensorPin); // Чтение уровня света

278 }

279

280 void handleButton1() {

281 // Переключение между меню

282 switch (currentState) {

283 case MAIN:

284 currentState = LED\_CONTROL; // Переход к управлению светодиодами

285 break;

286 case LED\_CONTROL:

287 // Если в состоянии AUTO, переходим в выбор освещенности

288 if (ledState == LED\_AUTO) {

289 currentState = LIGHT\_THRESHOLD;

290 } else {

291 currentState = PUMP\_CONTROL; // Переход к управлению помпой

292 }

293 break;

294 case LIGHT\_THRESHOLD:

295 currentState = PUMP\_CONTROL; // Переход к управлению помпой

296 break;

297 case PUMP\_CONTROL:

298 // Если в состоянии AUTO, переходим в выбор порога влажности

299 if (pumpState == PUMP\_AUTO) {

300 currentState = HUMIDITY\_THRESHOLD;

301 } else {

302 currentState = MAIN; // Возврат к главному меню

303 }

304 break;

305 case HUMIDITY\_THRESHOLD:

306 currentState = MAIN; // Возврат к главному меню

307 break;

308 }

309 }

310

311 void handleButton2() {

312 // Уменьшение значений в зависимости от состояния

313 switch (currentState) {

314 case MAIN:

315 break;

316 case LED\_CONTROL:

317 // Переключение между состояниями

318 switch(ledState) {

319 case LED\_OFF:

320 ledState = LED\_AUTO;

321 break;

322 case LED\_ON:

323 ledState = LED\_OFF;

324 break;

325 case LED\_AUTO:

326 ledState = LED\_ON;

327 break;

328 }

329 break;

330 case LIGHT\_THRESHOLD:

331 // Уменьшение порога освещенности

332 if (lightThreshold > 0) {

333 lightThreshold--;

334 }

335 break;

336 case PUMP\_CONTROL:

337 // Переключение между состояниями

338 switch(pumpState) {

339 case PUMP\_OFF:

340 pumpState = PUMP\_AUTO;

341 break;

342 case PUMP\_AUTO:

343 pumpState = PUMP\_OFF;

344 break;

345 }

346 break;

347 case HUMIDITY\_THRESHOLD:

348 // Уменьшение порога влажности

349 if (humidityThreshold > 0) {

350 humidityThreshold--;

351 }

352 break;

353 }

354 saveToEEPROM(); // Сохранение изменений в EEPROM

355 }

356

357 void handleButton3() {

358 // Увеличение значений в зависимости от состояния

359 switch (currentState) {

360 case MAIN:

361 // Переключение состояния подсветки

362 if (screenOn) {

363 lcd.noBacklight();

364 lcd.clear(); // Очищаем экран

365 } else {

366 lcd.backlight(); // Включаем подсветку

367 }

368 screenOn = !screenOn; // Переключаем состояние

369 break;

370 case LED\_CONTROL:

371 // Переключение между состояниями

372 switch(ledState) {

373 case LED\_OFF:

374 ledState = LED\_ON;

375 break;

376 case LED\_ON:

377 ledState = LED\_AUTO;

378 break;

379 case LED\_AUTO:

380 ledState = LED\_OFF;

381 break;

382 }

383 break;

384 case LIGHT\_THRESHOLD:

385 // Увеличение порога освещенности

386 if (lightThreshold < 100) {

387 lightThreshold++;

388 }

389 break;

390 case PUMP\_CONTROL:

391 // Переключение между состояниями

392 switch(pumpState) {

393 case PUMP\_OFF:

394 pumpState = PUMP\_AUTO;

395 break;

396 case PUMP\_AUTO:

397 pumpState = PUMP\_OFF;

398 break;

399 }

400 break;

401 case HUMIDITY\_THRESHOLD:

402 // Увеличение порога влажности

403 if (humidityThreshold < 100) {

404 humidityThreshold++;

405 }

406 break;

407 }

408 saveToEEPROM(); // Сохранение изменений в EEPROM

409 }

ПРИЛОЖЕНИЕ Е

**(обязательное)**

**Схема печатной платы**

ПРИЛОЖЕНИЕ Ж

**(обязательное)**

**Перечень элементов**

ПРИЛОЖЕНИЕ И

**(обязательное)**

**Ведомость документов**