

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»

НАУЧНО-ОБРАЗОВАТЕЛЬНОЕ СОРЕВНОВАНИЕ «ШАГ В БУДУЩЕЕ, МОСКВА»

22195

регистрационный номер

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

ИНТЕЛЛЕКТУАЛЬНЫЙ ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС КАК ИННОВАЦИОННОЕ СРЕДСТВО ВЕДЕНИЯ АГРОПРОМЫШЛЕННОГО ХОЗЯЙСТВА

Автор:

Крахотин Игорь Борисович

ГБОУ «Школа №1517» 11 класс

Научный руководитель:

Смирнов Иван Алексеевич

ГБОУ «Школа №1517»

Москва - 2023

АННОТАЦИЯ

Расчетно – пояснительная записка: 38 стр., 37 рис., 11 источников.

Цель работы – разработка интеллектуального программно-аппаратного комплекса, позволяющего существенно упростить ведение агропромышленного хозяйства.

В результате был разработан прототип, способный самостоятельно выращивать растительные культуры. Основным методом обмена информацией с пользователем является способность отправлять данные о состоянии окружающей среды растения при помощи сервера. Человек способен изменять такие параметры, как: период считывания данных всех установленных датчиков, скорость вращения прототипа, автоматическое и механическое вращение. Так же обмен данными позволит человеку постоянно отслеживать показатели всех датчиков прототипа как через определенный период времени, так и в режиме реального времени.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. Анализ существующих решений	5
2. План реализации изделия	6
3. Реализация схмотехнической части проекта	8
3.1 Выбор элементной базы прототипа.....	8
3.2 Создание монтажной электрической схемы	16
3.3 Сборка макета прототипа	16
4. Реализация схмотехнической части проекта	17
4.1 Реализация основных функций.....	17
4.2 Составление плана программного обеспечения, адаптированного под пользовательский интерфейс	20
4.3 Изучение протокола MQTT	20
4.4 Разработка бизнес-логики системы.....	22
5. Реализация проектно – конструкторской части проекта	29
5.1 Разработка трехмерной модели прототипа	29
5.2 Определение внешнего вида и компоновки изделия	29
5.3 Разработка корпуса блока электроники и внешнего корпуса аппарата	31
5.4 Разработка резервуара для воды.....	32
5.5 Разработка емкости для посева растения	32
5.6 Алгоритм сборки изделия	33
ЗАКЛЮЧЕНИЕ.....	35
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	37

ВВЕДЕНИЕ

Продовольственный кризис – одна из главных проблем последних лет, продолжающаяся до сих пор и характеризующаяся резким дефицитом продуктов питания в определенных частях планеты. Согласно докладу ООН "Положение дел в области продовольственной безопасности и питания", число людей, страдающих от нехватки продовольствия, в 2021 году достигло 828 млн, что на 46 млн больше, чем годом ранее. Таким образом, с каждым годом многократно увеличивается процент голодающих людей в мире. Причиной данного кризиса являются такие факторы, как:

1. отсутствие возможности выращивать зерновые культуры в определенных районах планеты;
2. отсутствие обеспечения достаточных условий существования для выращивания растений, активно используемых человеком в качестве продукта питания.

В рамках работы предлагается разработать автономную систему дистанционного контроля за широкой номенклатурой посаженных культур и растений. В качестве апробации был разработан прототип, представляющий собой часть подсистемы, выполняющей те же самые функции, но в рамках одного растения.

Целью работы является разработка и исследование интеллектуального автоматизированного программно – аппаратного комплекса жизнеобеспечения одного растения.

Поставленная цель в работе достигается решением следующих научных задач:

1. изучение аналогов прототипа;
2. планирование функционала создаваемого прототипа;
3. создание дорожной карты прототипа;
4. описание конструкции отладочного макета прототипа;
5. описание схмотехнической компоновки прототипа;
6. создание макета прототипа;

7. составление кода и плана программного обеспечения, адаптированного под пользовательский контроль;
8. изготовление прототипа.

1. Анализ существующих решений

В качестве обзора аналогов было принято решение рассматривать уменьшенные по размеру системы и прототипы, которые могут быть масштабированы.

На сегодняшний день, номенклатура систем, отвечающих поставленным требованиям, представлена линейкой умных горшков. В качестве примера было принято решение рассмотреть фирму Lechuza (Рисунок 1). [1]



Рисунок 1 - Прототипы

Подробно изучив аналоги, созданные на данный момент, было выявлено наличие в них таких компонентов, как:

1. автополив воды;
2. индикатор уровня воды;
3. емкость для воды внутри кашпо.

Так же было выявлено отсутствие следующих компонентов:

1. отсутствие датчиков освещенности растения и датчиков влажности почвы;
2. отсутствие автоматического и механического поворота для растений, требующих равномерный обогрев почвы;

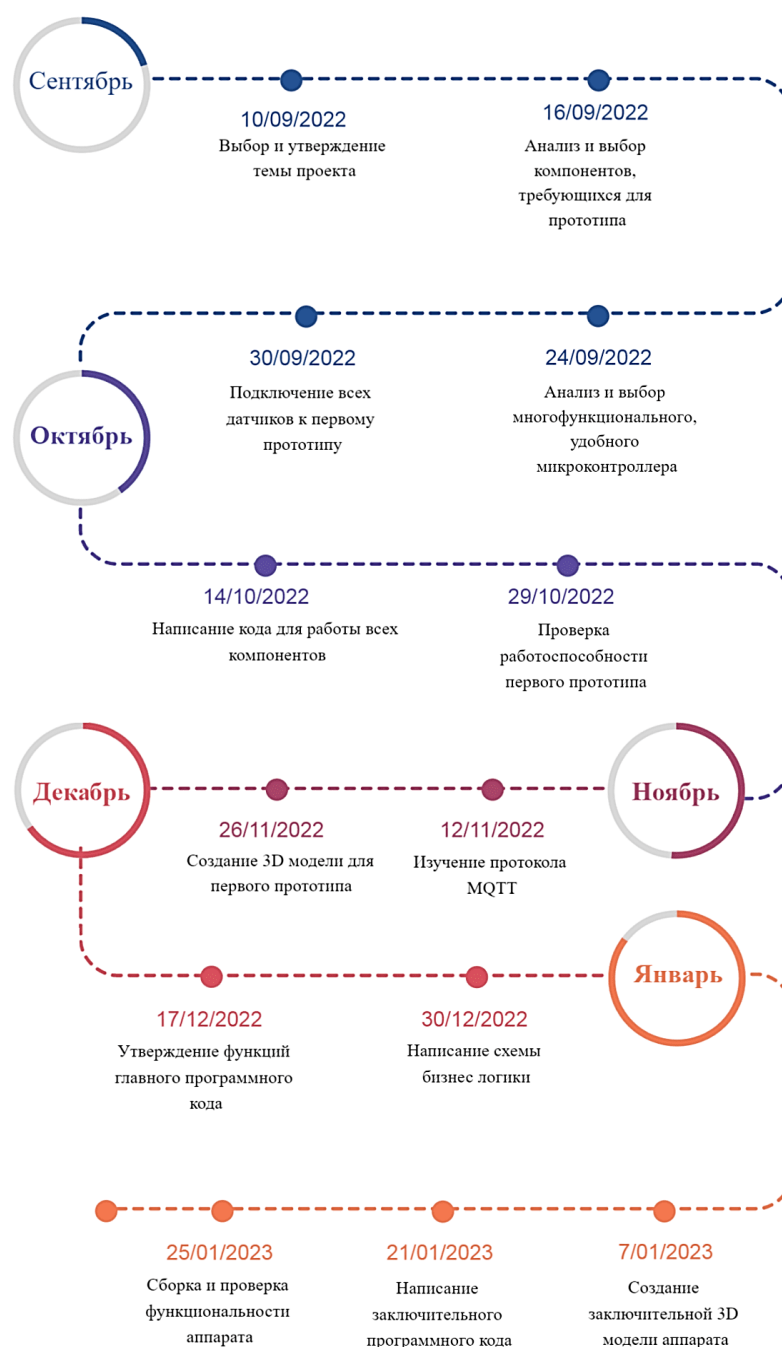
3. отсутствие дистанционного отслеживания состояния почвы и освещенности растения;

4. отсутствие возможности дистанционного управления функционалом прототипа.

Принимая во внимание вышеперечисленные недостатки, было принято решение разработать прототип с их учетом и исправлением.

2. План реализации изделия

Работа над данным проектом была разделена на следующие этапы:



Создание первого (отладочного) макета прототипа включает следующие этапы:

1. создание цифрового 3d макета прототипа;
2. объединение всех схмотехнических компонентов, составление макетной схемы прототипа;
3. написание программного кода для совместной работы всех используемых компонентов.

3. Реализация схмотехнической части проекта

3.1 Выбор элементной базы прототипа

В качестве элементной базы было принято решение использовать следующие компоненты: [2]

1. отладочная плата nodemcu на базе микроконтроллера esp8266;
2. модули 16-битного АЦП ADS1115;
3. фоторезисторы mlg5516b;
4. шаговый двигатель 28BYJ-48;
5. модуль влажности почвы;
6. датчик температуры и влажности внешней среды dht22;
7. водяной погружной насос dc 3-5v.

Фоторезисторы отвечают за считывание параметров освещенности, шаговый двигатель за автоматическое/механическое вращение прототипа, модуль влажности почвы отвечает за считывание показаний почвы, датчик температуры и влажности внешней среды отвечает за считывание параметров внешней среды, насос отвечает за автополив растения, в зависимости от показаний модуля влажности почвы.

Отладочная плата NodeMCU на базе микроконтроллера ESP8266

NodeMcu (Рисунки 2 - 4) – платформа на основе ESP8266 для создания различных устройств интернета вещей. Модуль умеет отправлять и получать информацию в локальную сеть либо в интернет при помощи Wi-Fi. [3]

Данная плата была выбрана исходя из следующих критериев:

1. маленькие размеры - 48.8 x 25.6 x 4.2 мм
2. встроенный модуль wi-fi;

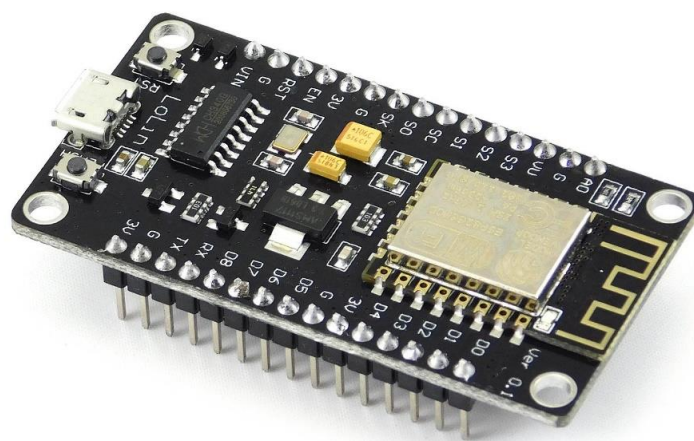


Рисунок 2 – Отладочная плата ESP 8266

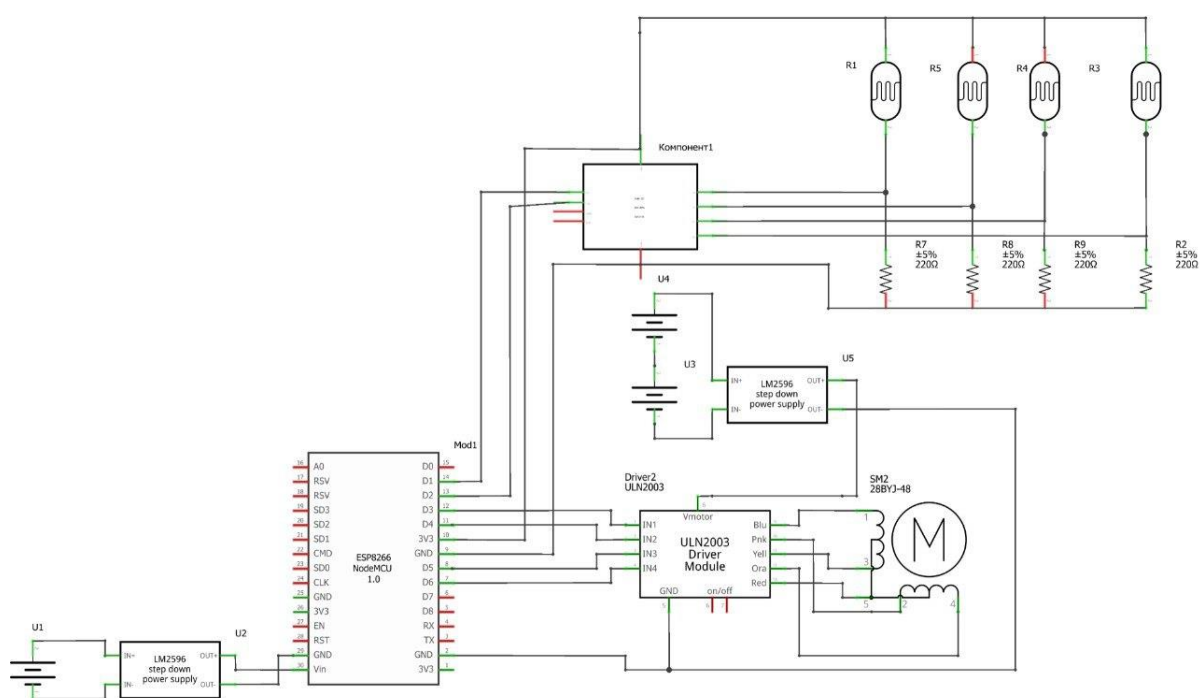


Рисунок 3 – ESP 8266 – схема электрическая принципиальная

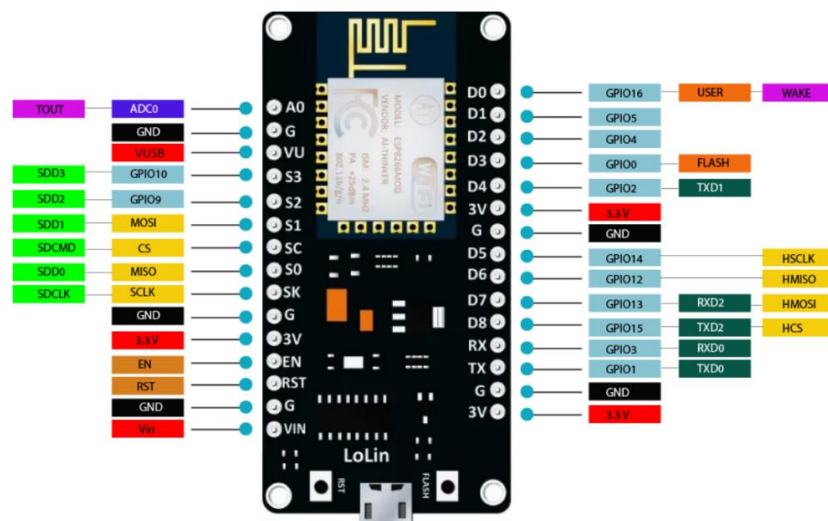


Рисунок 4 – Распиновка платы ESP 8266

Модуль 16-битного АЦП ADS1115

Плата ESP 8266 имеет один аналоговый выход. Поэтому, для реализации всех датчиков, требуемых подключения к аналоговым портам, было принято решение использовать АЦП ADS1115 в размере двух штук.

Модуль 16-битного АЦП ADS1115 (Рисунки 5 - 6) имеет внутренний источник опорного напряжения, программируемую частоту дискретизации, настраиваемый коэффициент усиления. [4]

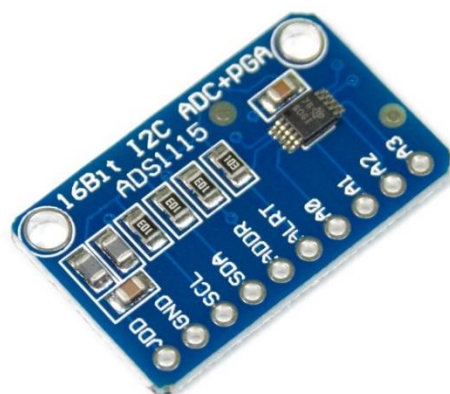


Рисунок 5 – АЦП ADS1115

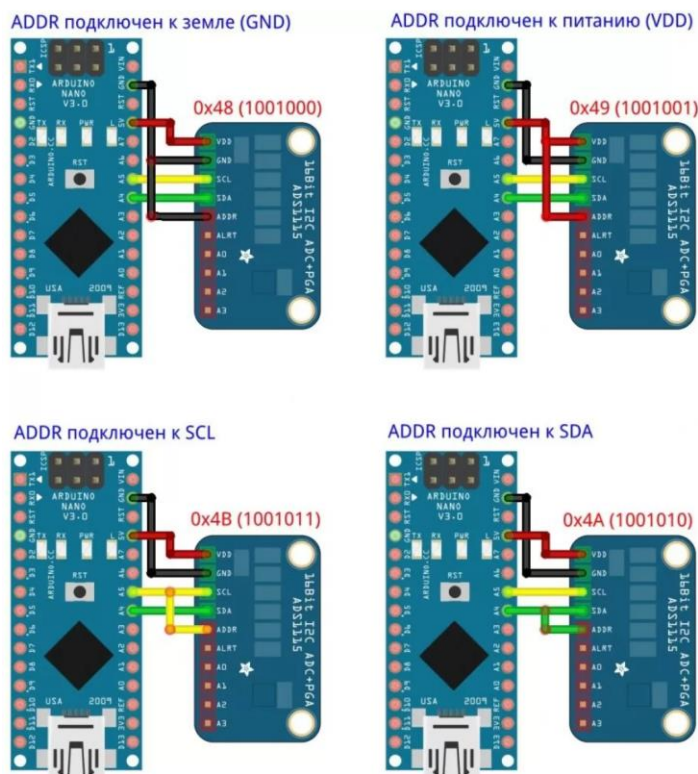


Рисунок 6 – Схемы подключений ADS 1115

Фоторезисторы MLG5516B

Фоторезистор – фотоэлектрический полупроводниковый приемник излучения, принцип действия которого основан на эффекте фотопроводимости. Эффект фотопроводимости заключается в уменьшении электросопротивления полупроводникового материала при освещении (Рисунок 7).



Рисунок 7 – Фоторезистор MLG5516B

Фоторезистор подсоединяется одним выводом к земле, а вторым к АЦП (аналого – цифровому преобразователю). Ко второму выводу также

подключается резистор с сопротивлением 10 кОм, через резистор происходит подача питания на плату (в данном случае 5V). Подключение фоторезистора к аналоговому порту позволит получать сопротивление в диапазоне 0 - 4096.[5]

Наличие данных датчиков предоставляет пользователю полный контроль над степенью освещенности прототипа. В свою очередь система производит автоповорот, опираясь на полученные данные о освещенности. При разработке прототипа было принято решение использовать 4 фоторезистора по следующим причинам:

1. удобная работа с шаговым двигателем (поворот на 90/180/-90 градусов);
2. оптимальный способ контролировать секторы работы системы (каждый фоторезистор контролирует четверть).

Шаговый двигатель 28BYJ-48

Шаговый двигатель 28BYJ-48 (Рисунок 8) – это бесколлекторный двигатель, вращение вала осуществляется шагами. На роторе (валу), расположен магнит, а вокруг него расположены катушки, если поочередно подавать ток на эти катушки, создается магнитное поле, которое отталкивает или притягивает магнитный вал, тем самым заставляя двигатель вращаться.

Так как двигатель потребляет значительный ток, мы не можем подключить его напрямую к выводам ESP 8266, для этого воспользуемся так называемым «Драйвером двигателя» основанном на микросхеме ULN2003.

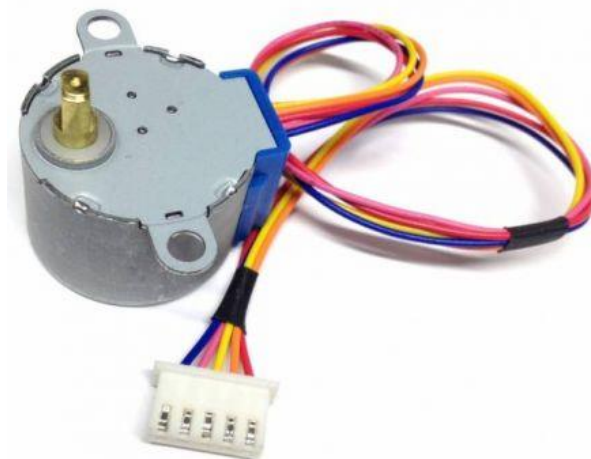


Рисунок 8 – Шаговый двигатель 28BYJ-48

Так как двигатель потребляет значительный ток, мы не можем подключить его напрямую к выводам ESP 8266, для этого воспользуемся так называемым «Драйвером двигателя» основанном на микросхеме ULN2003(Рисунок 9).[6]

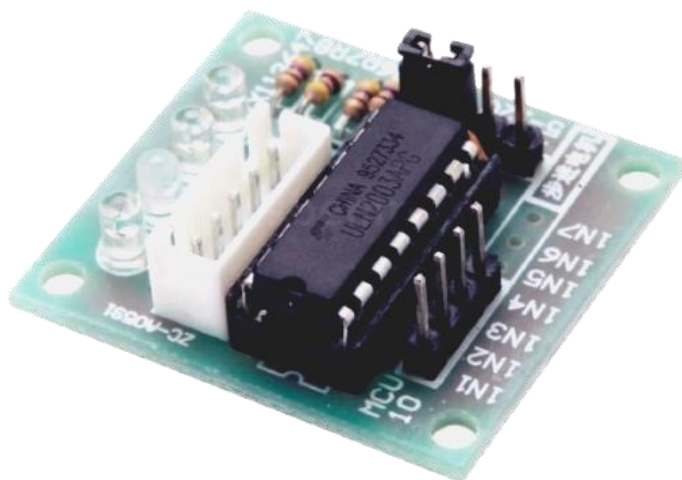


Рисунок 9 – Драйвер ULN2003

Модуль влажности почвы

Работа модуля влажности почвы (Рисунок 10) заключается в определении содержания воды в почве.

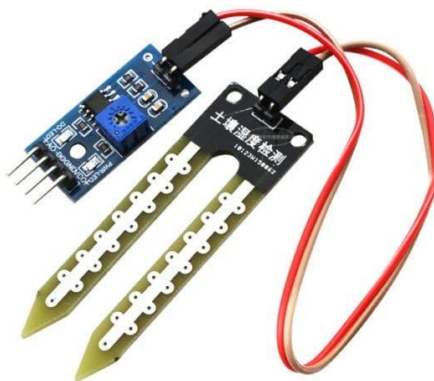


Рисунок 10 – Модуль влажности почвы

Считывать показания влажности почвы могут такие датчики, построенные на ёмкостном и резистивном физических принципах. В работе будет применяться резистивный датчик из-за его доступности, в будущем его можно заменить на ёмкостный, чтобы не навредить почве в долгосрочной перспективе и получать наиболее точные значения.

Принцип подключения модуля: вывод VCC подключается к цифровому выводу, затем подключаем GND к GND на плате и вывод A0 модуля к выводу A0 аналого-цифрового преобразователя.[7]

Наличие рассматриваемого датчика в системе позволит пользователю постоянно отслеживать параметры почвы с целью своевременного увлажнения (полива).

Датчик температуры и влажности внешней среды DHT22

DHT22 – это цифровой датчик, предназначенный для измерения температуры и влажности окружающей среды с высокой точностью (Рисунок 11).

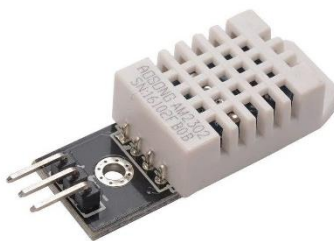


Рисунок 11 – Датчик DHT22

Обычно в проектах используется дешёвый DHT11 низкой точности, который корректно измеряет только температуру, а влажность лишь косвенно. Поэтому было принято решение использовать датчик DHT22, который дает действительно точные измерения.

Принцип подключения датчика: VCC подключаем к питанию 3,3 - 5В, черный провод к GND, вывод данных к цифровому пину. Чтобы обеспечить надлежащую связь между датчиком и микроконтроллером, добавляем подтягивающий резистор на 10 кОм между линией передачи данных и VCC (чтобы сигнал оставался на высоком уровне). [8]

Система полива

Водяной насос – это устройство, способное генерировать поток жидкости с использованием кинетической энергии (Рисунок 12).



Рисунок 12 – Водяной погружной насос DC 3-5V

Нельзя напрямую подключить мотор-насос к микроконтроллеру, потому что он будет потреблять больше тока, что может привести к выходу микроконтроллера из строя, поэтому для управления мотором используется электромеханическое реле (модуль, который включает или выключает внешние устройства, определенным образом замыкая или размыкая отдельную электрическую цепь, к которой они подключены)(Рисунок 13).[9]

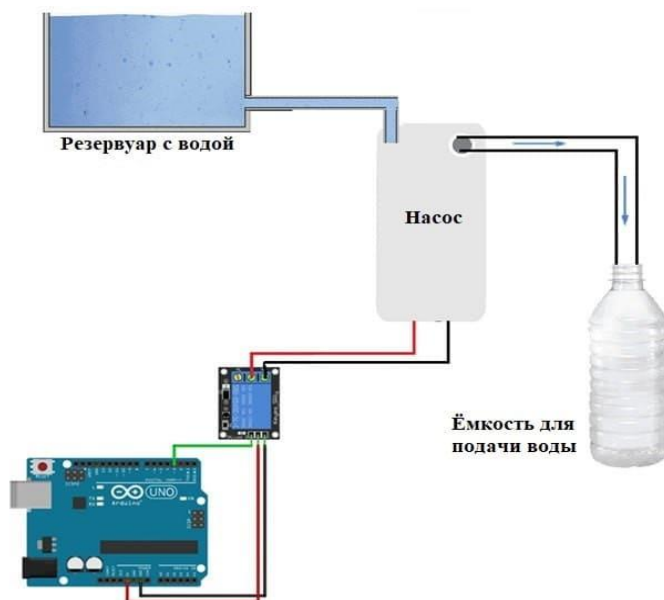


Рисунок 13 – Подключение насоса

3.2 Создание монтажной электрической схемы

Общая монтажная схема отладочного прототипа из перечисленных выше компонентов представлена на рисунке 14.

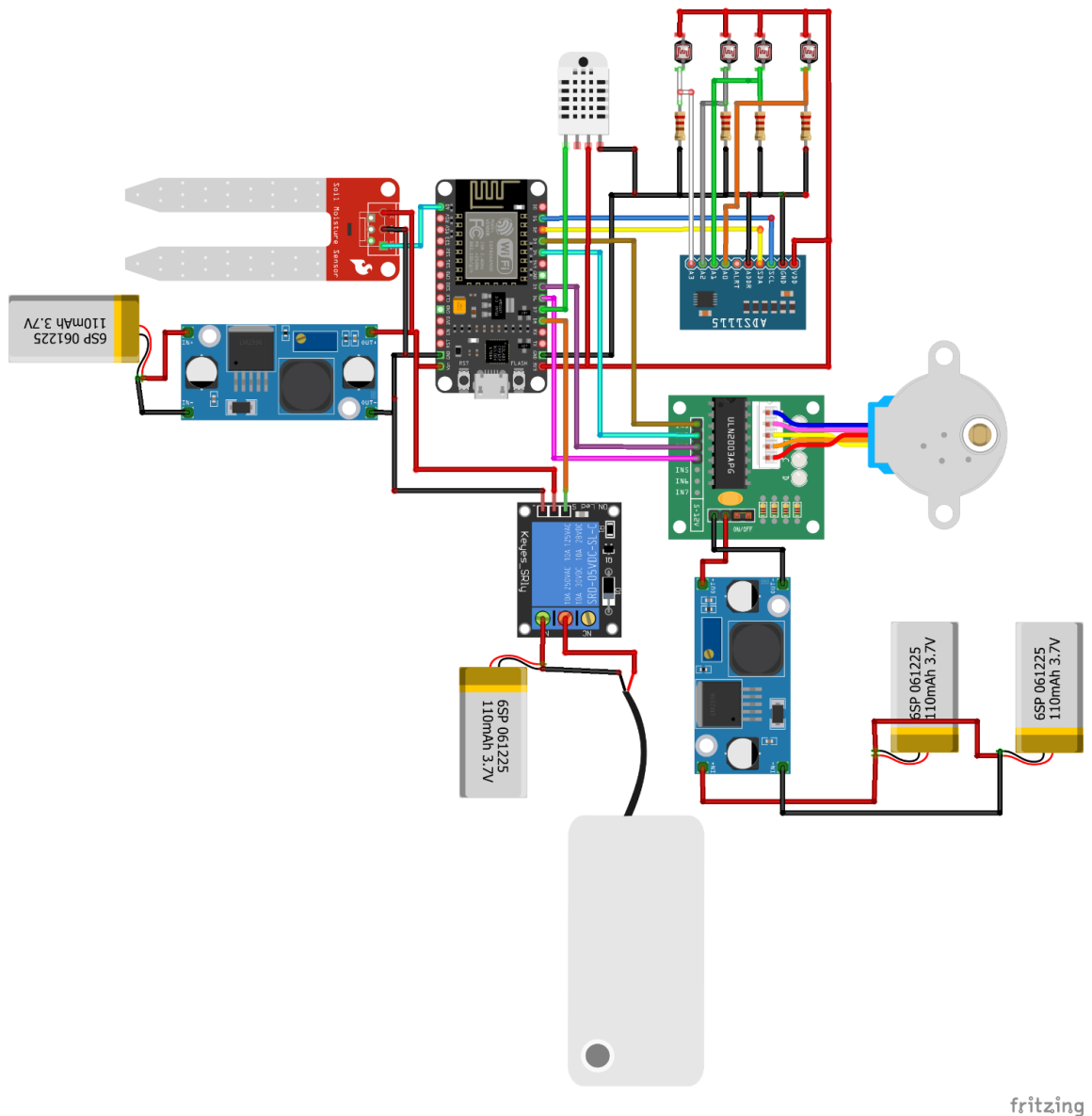


Рисунок 14 – Монтажная электрическая схема отладочного прототипа

3.3 Сборка макета прототипа

Объединив все перечисленные выше компоненты, был создан отладочный прототип системы (Рисунок 15). Макет был создан с целью апробации работы фоторезисторов, шагового двигателя, датчика влажности почвы, датчика температуры внешней среды.

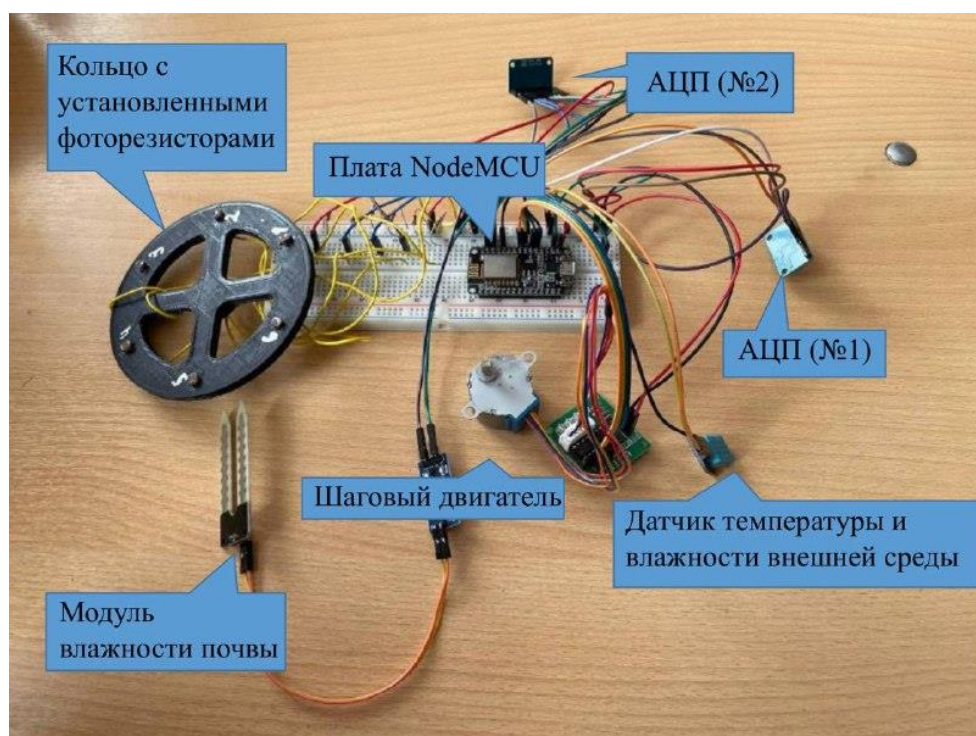


Рисунок 15 – Сборка первого прототипа

Макет состоит из технологического диска, созданного с помощью аддитивных технологий, на котором размещается массив фоторезисторов (6шт)

В ходе разработки этого макета были выполнены следующие задачи:

1. тестирование работы алгоритма;
2. обработка, получение и отправка данных с датчиков на сервер mqtt;
3. поворот шагового двигателя в зависимости от полученных значений;
4. определение точных параметров печати;
5. апробация правильной электрической коммутации на вращающемся основании.

4. Реализация схемотехнической части проекта

4.1 Реализация основных функций

Написанный код позволит удостовериться в работоспособности всех подключенных к плате компонентов. Он должен содержать в себе такие функции, как:

1. многократное получение данных с фоторезисторов;

2. высчитывание среднего арифметического значения каждого датчика, для сглаживания флуктуаций в показаниях датчиков;
3. получение значения максимально и минимально освещенных фоторезисторов;
4. получение значений модуля влажности почвы и значений температуры и влажности внешней среды;
5. получение значения угла поворота шагового двигателя;
6. поворот шагового двигателя от максимально освещенного фоторезистора к минимально освещенному;

Код включает в себя такие методы, как:

1. считывание аналоговых и цифровых значений с датчиков, вращение шагового двигателя;
2. расчет периодичности выполнения считывания.[10]

Считывание аналоговых и цифровых значений с датчиков, вращение шагового двигателя.

Данный код позволяет считать значения фоторезистора и модуля влажности почвы. (Рисунок 16)

```
#define FIRST_SENSOR 36 //Присвоение константе значения аналогового пина на плате(36)
#define SECOND_SENSOR 14 //Присвоение константе значения цифрового пина на плате(14)

void setup() {
  Serial.begin(115200) //плата посылает по последовательному порту данные со скоростью 115200 бит в секунду
  Serial.print(analogRead(FIRST_SENSOR)); //Вывод считанных данных с аналогового порта
  Serial.print(digitalRead(SECOND_SENSOR)); //Вывод считанных данных с аналогового порта
}
```

Рисунок 16 – Считывание данных с аналоговых и цифровых портов

Код, представленный ниже, позволяет считать значение датчика температуры внешней среды. (Рисунок 17)

```
#include "DHT.h" //Библиотека для считывания температуры
#define DHTTYPE DHT22 //Тип датчика(DHT22)
#define DHTPIN 2 //Номер цифрового порта(2)

DHT dht(DHTPIN, DHTTYPE); //Функция считывания и вывода показателя температуры

void setup(){
  Serial.begin(115200);
  dht.begin(); //Запуск функции (строка 5)
}
```

Рисунок 17 – считывание данных температуры внешней среды

Код, написанный ниже, позволяет управлять шаговым двигателем.
(Рисунок 18)

```
#include <AccelStepper.h> // подключение библиотеки
#define STEP 512 // Указание шага вращения шагового двигателя (512 = 90°)
#define MOTORPIN1 5 //Указание пинов подключения
#define MOTORPIN2 9
#define MOTORPIN3 10
#define MOTORPIN4 11
#define HALFSTEP 4
AccelStepper stepper(HALFSTEP, MOTORPIN1, MOTORPIN3, MOTORPIN2, MOTORPIN4); // Создание экземпляра AccelStepper

void setup(){
  stepper.setMaxSpeed(900.0); // установка максимальной скорости шагового двигателя
  stepper.setSpeed(200); // Установка скорости шагового двигателя
  stepper.run(); // Запуск шагового двигателя
  stepper.moveTo(STEP); // Движение шагового двигателя
  stepper.disableOutputs(); // Отключение шагового двигателя для предотвращения нагревания
  stepper.distanceToGo()==0; // Проверка того, достиг ли шаговый двигатель заданной координаты
}
```

Рисунок 18 – Функции для работы с шаговым двигателем

Ниже представленный код позволит считывать данные с аналоговых портов АЦП ADS1115(Рисунок 19)

```
#include <Adafruit_ADS1X15.h> // Подключение библиотек для работы с ADS1115
#include <Wire.h>
Adafruit_ADS1115 ads1; // Определение ADS1115
void setup()
{
  Serial.begin(9600);
  ads1.setGain(GAIN_TWOTHIRDS);
  ads1.begin(0x48); // Вид подключения (рисунок 6)
}

void loop()
{
  int16_t ads10, ads11, ads12, ads13; // Задание переменных для считывания данных с АЦП
  ads10 = ads1.readADC_SingleEnded(0); // Считывание аналоговых значений с портов 0,1,2,3
  ads11 = ads1.readADC_SingleEnded(1);
  ads12 = ads1.readADC_SingleEnded(2);
  ads13 = ads1.readADC_SingleEnded(3);
}
```

Рисунок 19 – Функции для работы с АЦП ADS1115

Расчет периодичности выполнения считывания

Код, представленный ниже, позволит считывать данные с используемых датчиков через определенный период. (Рисунок 20)

```

unsigned long cycle_lasttime; //unsigned число в диапазоне от 0-4,294,967,295.
//Чаще всего этот тип данных используется для хранения результатов функции millis(),
// которая возвращает количество миллисекунд, на протяжении которого работал ваш код.
int cycle_period = 100000; //Установка периода считывания данных в мС
void setup() {
    Serial.begin(115200); //плата посылает по последовательному порту данные со скоростью 115200 бит в секунду
    cycle_lasttime = millis(); //функция millis() позволяет считывать время в данную секунду
}
void loop(){
    if (millis() - cycle_lasttime >= cycle_period){
        main_loop(); //запуск главной функции
        cycle_lasttime = millis(); //Присвоение переменной значения времени
    }
}

```

Рисунок 20 – Метод считывания данных через определенный промежуток времени

4.2 Составление плана программного обеспечения, адаптированного под пользовательский интерфейс

Составление плана можно разбить на такие подтемы, как:

1. изучение протокола mqtt;
2. создание бизнес-логики полного программно-аппаратного кода;
 - 2.1 определение задач, выполняемых программой;
 - 2.2 составление блок-схем функций;
3. написание программно-аппаратного кода, используя созданную схему бизнес-логики.

4.3 Изучение протокола MQTT

MQTT (Message Queue Telemetry Transport) – это легкий, компактный и открытый протокол обмена данными, созданный для передачи данных на удаленных локациях, где требуется небольшой размер кода и есть ограничения по пропускной способности канала. [11]

Обмен сообщениями в протоколе MQTT осуществляется между клиентом (client), который может быть издателем или подписчиком (publisher/subscriber) сообщений, и брокером (broker) сообщений.

Данный протокол был выбран, исходя из следующих особенностей:

1. компактный и легковесный — минимальные накладные расходы на пересылку данных, для экономии трафика;

2. устойчивость к потерям — гарантированная доставка в условиях нестабильных сетевых подключений;
3. асинхронный протокол — позволяет обслуживать большое количество устройств, и не зависит от сетевых задержек;
4. поддержка qos — возможность управлять приоритетом сообщений и гарантировать доставку сообщения адресату;
5. динамическая конфигурация — не требует предварительного согласования полей и форматов данных, может конфигурироваться «на лету»;
6. удобная адресация — поля данных имеют текстовые названия, понятные для человека. не нужно запоминать цифровые адреса и битовые смещения.

Устройства MQTT используют определенные типы сообщений для взаимодействия с брокером, ниже представлены основные:

1. Connect – установить соединение с брокером;
2. Disconnect – разорвать соединение с брокером;
3. Publish – опубликовать данные в топик на брокере;
4. Subscribe – подписаться на топик на брокере;
5. Unsubscribe – отписаться от топика.

Ниже представлена схема взаимодействия между подписчиком, издателем и брокером (Рисунок 21)



Рисунок 21 – Схема простого взаимодействия между подписчиком, издателем и брокером

4.4 Разработка бизнес-логики системы

Определение воспроизводимых программой функций.

Прежде чем начать писать бизнес логику кода, основанного на изученном протоколе MQTT, необходимо произвести анализ функций, воспроизводимых программой.

Ниже представлен список воспроизводимых функций:

1. функции считывания параметров всего комплекта используемых датчиков прототипа;
2. функции автоматической отправки данных на сервер с определенным интервалом времени;
3. функции автоматического и ручного поворота;
4. функции приема данных от пользователя;
5. функции замены данных по умолчанию данными, отправленными пользователем;
6. функции логирования.

Функция автоматического поворота:

Автоматический поворот осуществляется шаговым двигателем после считывания данных с фоторезисторов по установленной по умолчанию или измененной пользователем скорости.

Функция ручного поворота:

При ручном повороте происходит непрерывное вращение прототипа со скоростью, заданной пользователем.

Составление блок-схем функций.

Составление блок-схем можно разбить на такие пункты, как:

1. Составление блоков главных функций программы;
2. Составление блоков функций приема данных от пользователя;
3. Составление блоков функций отправки данных на сервер.

Составление блоков главных функций программы

Главная функция №1 (Рисунок 22) отвечает за вызов функций, которые:

1. Выполняют считывание и отправку данных всех датчиков на сервер;

2. Выполняют автоповорот системы.

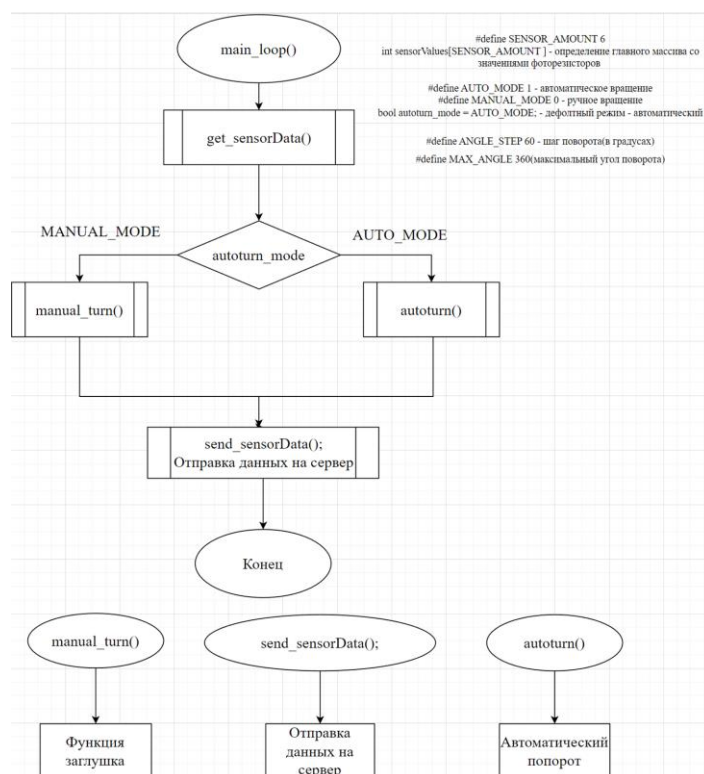


Рисунок 22 – Главная функция №1

Главная функция №2 (Рисунок 23) проверяет, прошел ли определенный промежуток времени и вызывает код работы библиотеки, отключающую за всю работу библиотеки MQTT.

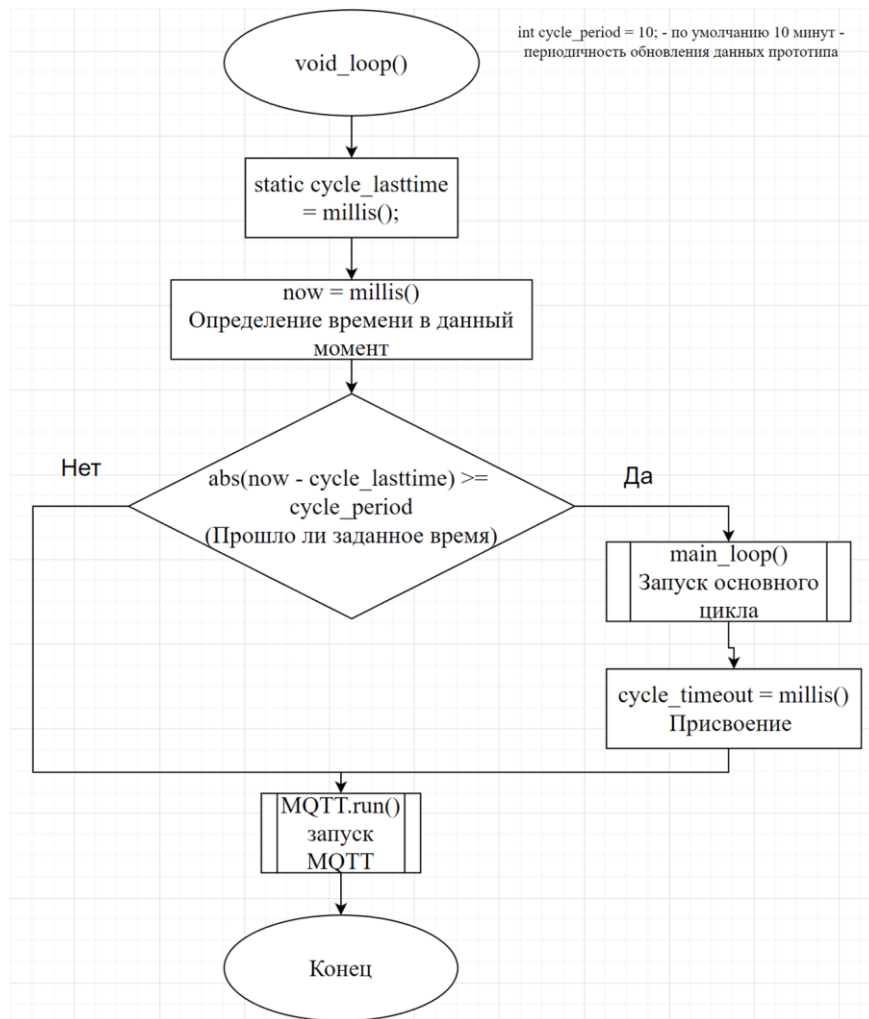


Рисунок 23 – Главная функция №2

Функция считывания данных с фоторезистора (Рисунок 24) является одной из основных функций и вызывается в Главной функции №1.

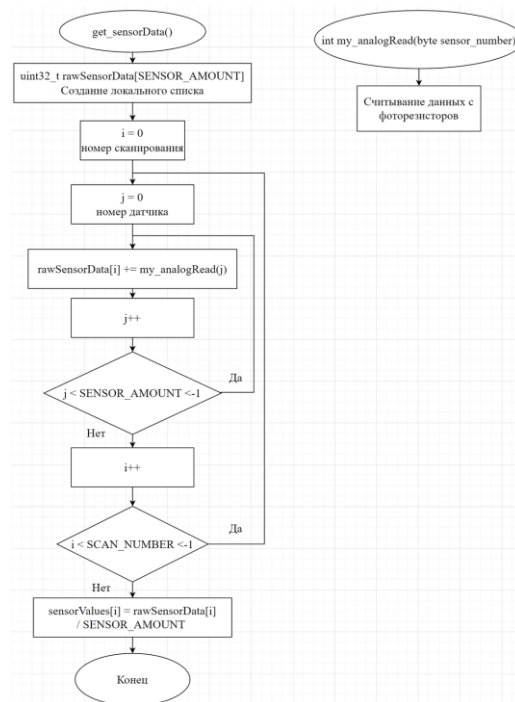


Рисунок 24 – Функция считывания данных с фоторезисторов

Составление блоков функций приема данных от пользователя

Функция смены режима поворота (Рисунок 25) отвечает за установку автоматического/ручного вращения прототипа.

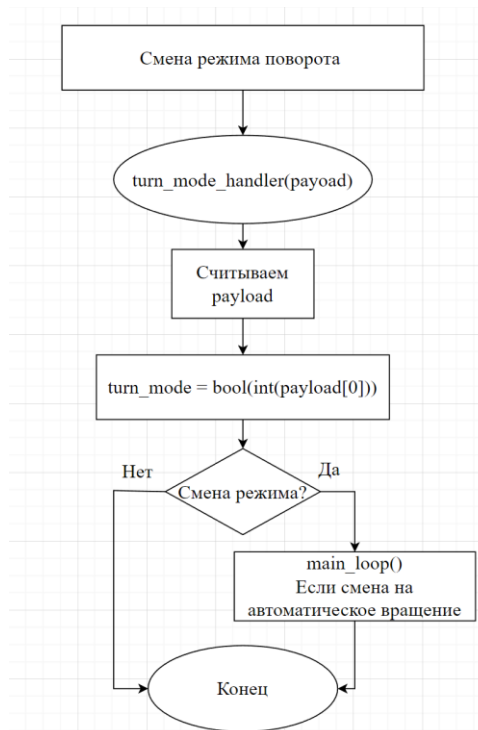


Рисунок 25 – Функция смены режима поворота

Функция смены скорости вращения (Рисунок 26) заменяет скорость на отправленную пользователем, вызывает функцию отправки состояния системы.

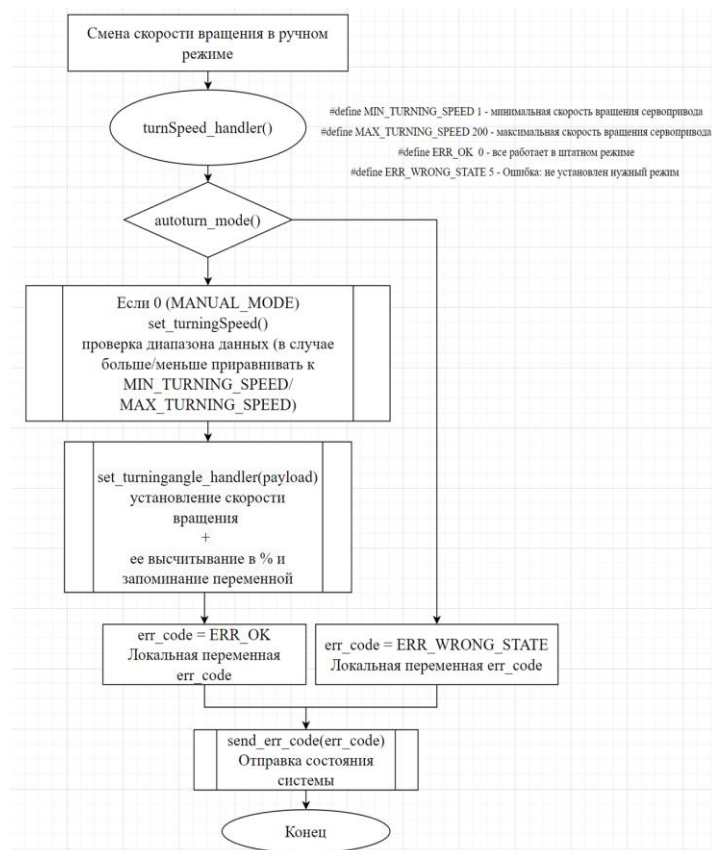


Рисунок 26 – Функция смены скорости вращения прототипа

Функция отправки данных пользователю (Рисунок 27) отвечает за считывание и отправку данных с каждого схемотехнического момента в режиме реального времени.

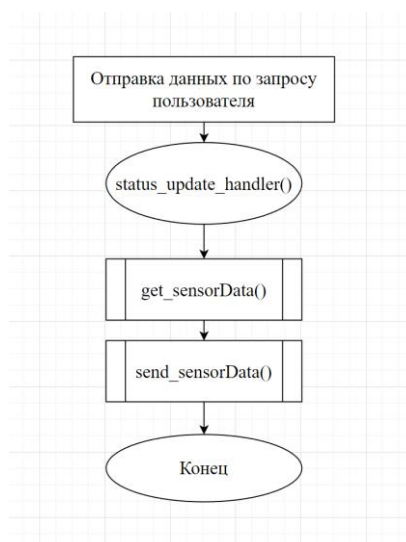


Рисунок 27 – Функция отправки данных пользователю

Функция изменения периода считывания данных (Рисунок 28) отвечает за изменение переменной, отвечающий за частоту выполнения Главного цикла №1.

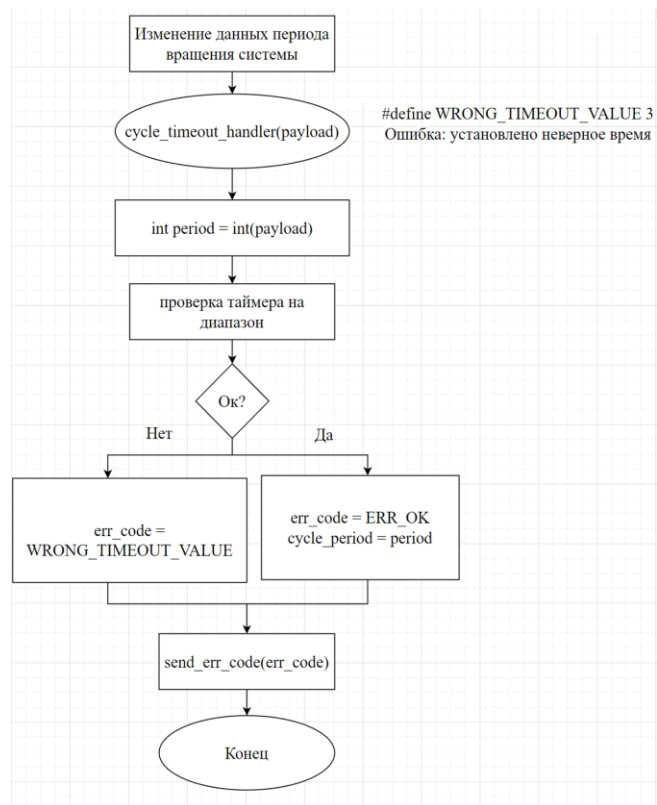


Рисунок 28 – Функция изменения данных периода считывания данных
Составление блоков функций отправки данных на сервер

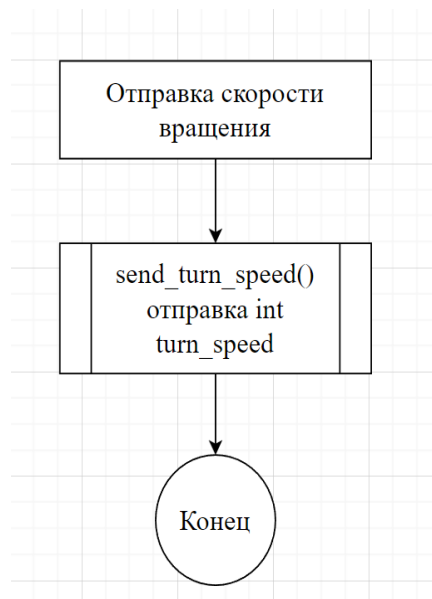


Рисунок 29 – Функция отправки скорости вращения прототипа

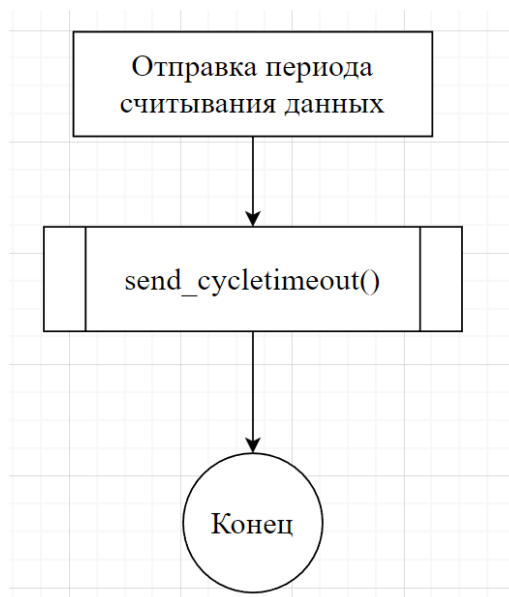


Рисунок 30 – Функция отправки периода считывания данных

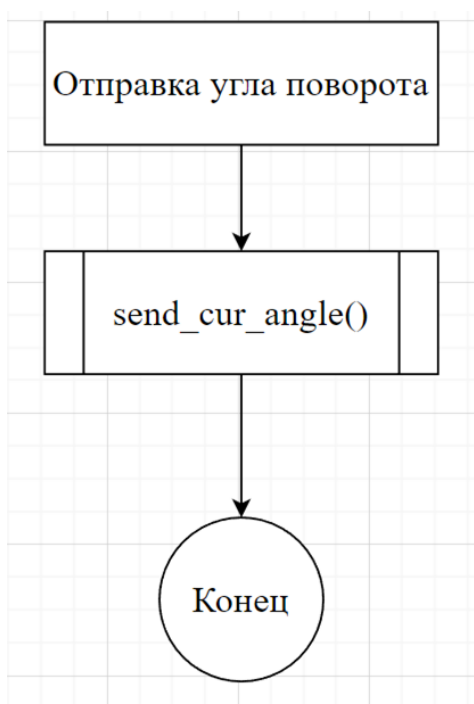


Рисунок 31 – Функция отправки угла поворота

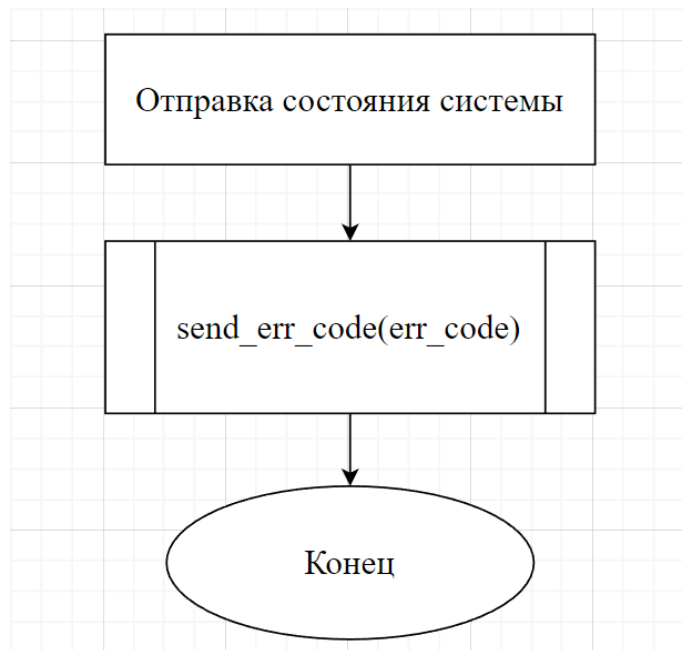


Рисунок 32 – Функция отправки состояния системы

Разработанные блок-схемы позволяют без труда написать исходный код.

5. Реализация проектно – конструкторской части проекта

5.1 Разработка трехмерной модели прототипа

Данная задача включает в себя выполнение таких подзадач, как:

1. определение внешнего вида и компоновки изделия;
2. разработка корпуса блока электроники и внешнего корпуса аппарата;
3. разработка отсека для воды;
4. разработка емкости для посева растений;
5. сборка аппарата.

5.2 Определение внешнего вида и компоновки изделия

К разрабатываемому изделию предъявляются следующие технические требования:

1. размещение резервуара с водой для осуществления полива;
2. поворот изделия для обеспечения равномерности освещенности;
3. осуществление полива растения в зависимости от влажности почвы;
4. измерение температуры окружающей среды.

На основе этих требований были выделены основные конструктивные задачи:

1. разработать общую компоновку изделия;
2. разработать компоновку отдельных узлов системы;
3. создать трехмерные твердотельные модели в виде деталей и сборочных единиц в САПР;
4. изготовить и протестировать образец.

По результатам конструирования был разработан прототип системы, состоящий из следующих сборочных узлов (Рисунок 33):

1. блок электроники, совмещенный с внешним корпусом системы;
2. резервуар для воды;
3. внешняя емкость для посева растения.

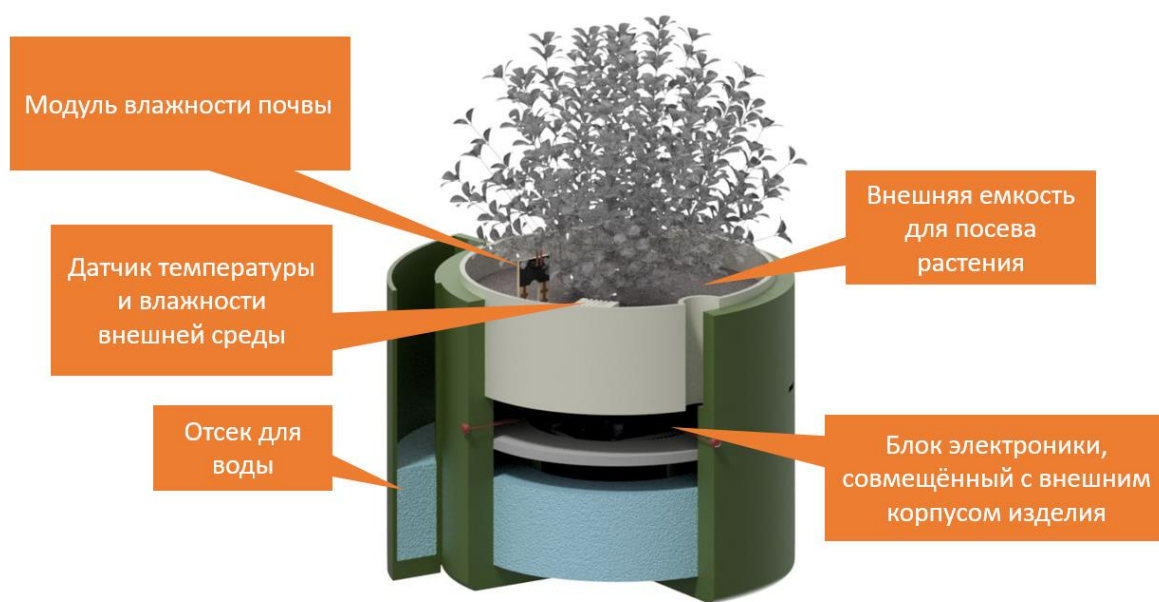


Рисунок 33 – Компоновка прототипа системы

5.3 Разработка корпуса блока электроники и внешнего корпуса аппарата

Корпус блока электроники - один из основных узлов изделия (Рисунок 34). Он представляет собой емкость, позволяющую разместить плату управления и подключить к ней шаговый двигатель, датчик влажности почвы, датчик температуры и фоторезисторы. Для разработки внешнего корпуса использовалась программа Autodesk Inventor 2022. Корпус прототипа представляет собой цилиндр диаметром 177 мм и высотой 145 мм. Также в нем находится опорный кронштейн для емкости с растением. Кронштейн находится на высоте 100 мм от основания внешнего корпуса. Электроника аппарата расположена в центральном отсеке внешнего корпуса, представляющего собой цилиндрическую емкость диаметром 90 мм и высотой 60 мм, защищенную от воды, находящуюся во внешнем корпусе вне центрального отсека. Отсек, в котором находится вода отделен крышкой, внутренним диаметром 90 мм и внешним 147 мм. В центре основания находится отверстие под шаговый двигатель.



Рисунок 34– Внешний корпус аппарата и блок электроники

5.4 Разработка резервуара для воды

Резервуар для воды - это емкость, в которой будет находиться вода для осуществления полива растения (Рисунок 35). Вода в него заливается сверху и поступает через трубку во внутреннюю часть корпуса аппарата. На отсеке находится крепление, с помощью которого он присоединяется к внешнему корпусу. Высота 150 мм, толщина стенки 2.5 мм и 5.3 мм.

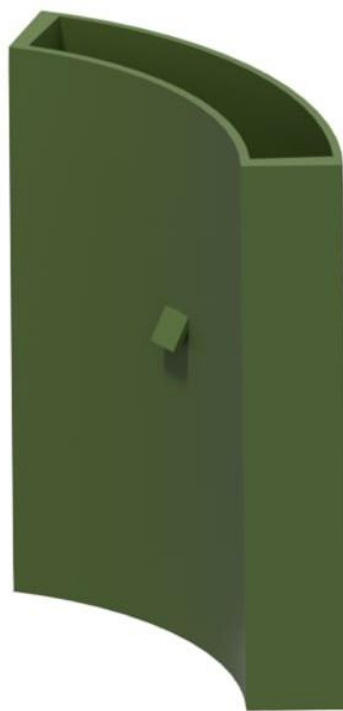


Рисунок 35 – Резервуар для воды

5.5 Разработка емкости для посева растения

Емкость для посева растения - это верхний отсек аппарата, в котором будет находиться почва (Рисунок 36). В нем предусмотрено отверстие под вывод датчиков влажности и температуры. Емкость имеет цилиндрическую форму диаметром 157 мм и толщиной стенки 5 мм. Она будет находиться на ступени основного корпуса, не задевая блок электроники.

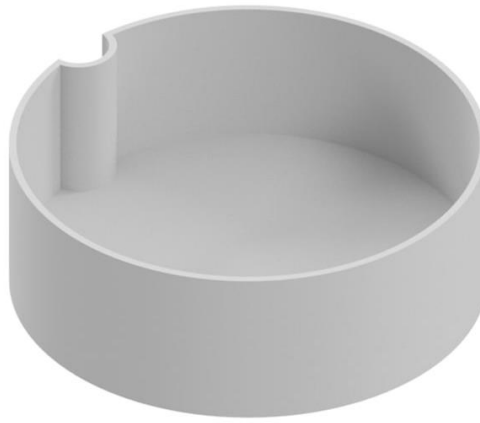


Рисунок 36– Емкость для посева растений

5.6 Алгоритм сборки изделия

Рассмотрим сборку поэтапно:

1. блок электроники помещаем в специальный отсек, находящийся во внешнем корпусе изделия;
2. помещаем насос в отсек для воды во внешнем корпусе изделия;
3. закрываем отделение с водой специальной крышкой для герметизации;
4. помещаем емкость для посева растения во внешний корпус изделия;
5. присоединяем отсек для воды;
6. сборка готова (Рисунок 37).



Рисунок 37 – Сборка изделия

ЗАКЛЮЧЕНИЕ

В рамках работы был проведен поиск и анализ аналогов устройства, определены их основные характеристики и особенности. Также был разработан и создан прототип устройства, обладающий следующими преимуществами по сравнению с аналогами:

1. большое количество разнородных датчиков, позволяющих контролировать множество параметров, важных для выращивания растений;
2. реализация системы автоматического контроля и поддержания уровня освещенности;
3. наличие протокола mqtt, позволяющего совершать обмен данными о состоянии датчиков с пользователем и получать данные от пользователя.

Для апробации работоспособности и функциональности всех датчиков был разработан прототип, позволяющий выполнять такие функции, как считывание параметров всех датчиков, автоповорот изделия, отправка данных на сервер. Был разработан работоспособный программный код, выполняющий все вышеперечисленные задачи.

После успешной апробации отладочного прототипа, был создан усовершенствованный прототип, дополненный функционалом получения данных от пользователя при помощи обмена данными через сервер, опираясь на которые происходит изменение определенных параметров работы прототипа.

В ходе исследования и разработки проекта были получены следующие навыки:

1. освоены следующие программные комплексы и продукты:
 - a. arduino ide, отвечающая за программную часть прототипа;
 - b. inventor, отвечающий за аппаратную часть прототипа;
2. Изучены онлайн сервисы:
 - a. fritzing, позволяющий собрать электрическую модель прототипа;
 - b. diagrams, позволяющий составить удобную схему бизнес логики всех используемых функций;
3. приобретены навыки:

- a. сборки, подключения и отладки датчиков и микроконтроллеров;
- b. программирования в среде arduino ide;
- c. 3d моделирования в среде inventor.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Умные горшки // OZON
URL: <https://www.ozon.ru/category/umnye-gorshki/> (Дата обращения: 03.11.2022).
2. Arduino и Raspberry Pi в проектах Internet of Things/Петин В.А. — СПб. БХВ-Петербург, 2016. — 320 с.: ил. — (Электроника) (дата обращения: 10.11.2022).
3. Установка платы ESP8266 // ARDUINOMASTER
URL: <https://arduino-master.ru/platy-arduino/esp8266-nodemcu-v3-lua/> (Дата обращения: 15.11.2022).
4. Модуль 16-битного АЦП ADS1115 // 3DiY
URL: <https://3d-diy.ru/wiki/arduino-moduli/modul-16-bitnogo-at-s-p-ads1115/>
(Дата обращения: 17.11.2022)
5. Подключение фоторезистора к ардуино и работа с датчиком освещенности // ARDUINOMASTER
URL: <https://arduino-master.ru/datchiki-arduino/photorezistor-arduino-datchik-sveta/> (Дата обращения: 20.11.2022).
6. Обзор шагового двигателя 28BYJ-48 с драйвером ULN2003// RobotChip
URL: <https://robotchip.ru/obzor-28byj-48-s-drayverom-uln2003/> (Дата обращения: 25.09.2022).
7. Модуль влажности почвы // 3DIY SHOP
URL: <https://3d-diy.ru/wiki/arduino-datchiki/datchik-vlazhnosti-pochvy-arduino/> (Дата обращения: 30.11.2022).
8. DHT22 и Arduino – схема подключения // Вольтик
URL: <https://voltiq.ru/dht22-and-arduino/> (Дата обращения: 05.12.2022).

9. Управление водяным насосом с помощью Arduino // ROBOTIQUE
URL: <https://www.robotique.tech/robotics/control-a-water-pump-by-arduino/> (Дата обращения: 15.12.2022).
10. Что такое MQTT и для чего он нужен в IoT? Описание протокола MQTT // IPC2U
URL: <https://ipc2u.ru/articles/prostye-resheniya/chto-takoe-mqtt/> (Дата обращения: 20.12.2022).
11. Основы программирования микроконтроллеров: Учебно-методическое пособие к образовательному набору по микроэлектронике “Амперка”: образовательный робототехнический модуль (базовый уровень): 12 – 15 лет/Артем Бачинин, Василий Понкратов, Виктор Накоряков, под редакцией Сергея Косаченко - М. : Издательство “Экзамен”, 2017. – 184с.(дата обращения: 29.12.2022)