**Система автоматического управления контроля и регулирования микроклимата в помещениях**

[1. КОМПОНЕНТНАЯ СОСТАВЛЯЮЩАЯ ДЛЯ РАЗРАБОТКИ СИСТЕМЫ 2](#_Toc35554628)

[1.1. Arduino UNO 2](#_Toc35554629)

[1.2. Электромагнитное реле 3](#_Toc35554630)

[1.3. Датчик температуры DS18B20 4](#_Toc35554631)

[1.4. Постоянный резистор 5](#_Toc35554632)

[1.5. Часы реального времени RTC на модуле DS1307 6](#_Toc35554633)

[1.6. Макетная плата 6](#_Toc35554634)

[2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ РЕГУЛИРОВАНИЯ МИКРОКЛИМАТА 8](#_Toc35554635)

[2.1. Подключение компонентов в системе 8](#_Toc35554636)

[ПРИЛОЖЕНИЕ 11](#_Toc35554637)

1. **КОМПОНЕНТНАЯ СОСТАВЛЯЮЩАЯ ДЛЯ РАЗРАБОТКИ СИСТЕМЫ**

Большое количество автоматизированных систем основываются на одноплатных компьютерах и микроконтроллерах. Сбор, обработка, дальнейшее использование данных осуществляется с помощью этих электронных компонентов.

Для большого объема данных и сложных операций используются одноплатные компьютеры, например, Raspberry PI. Задачи, не требующие выполнения очень сложных операций и не требующие выполнения объемной обработки данных на устройстве, могут быть реализованы при помощи микроконтроллеров.

Микроконтроллер – микросхема, предназначенная для управления электронными устройствами. В данной работе разработка системы выполняется на базе микроконтроллера Arduino UNO (рисунок 2.1).

## 1.1. Arduino UNO

Arduino UNO – плата микроконтроллера с открытым кодом, основанная на микроконтроллере Microchip ATmega328P [2].

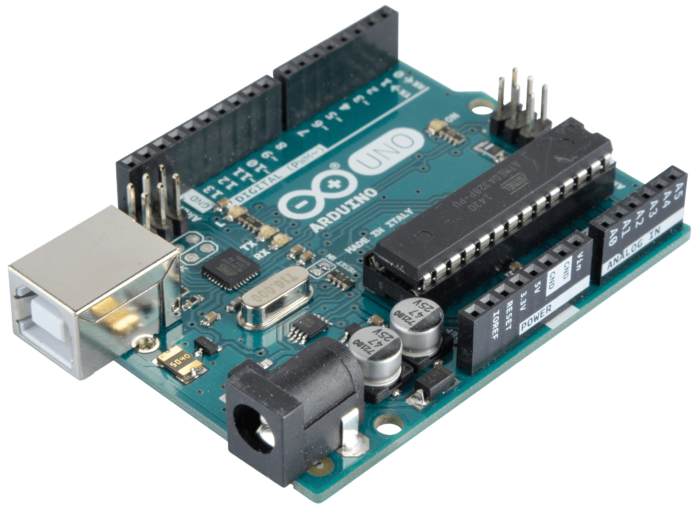


Рисунок 2.1 – Arduino UNO

Технические характеристики Arduino UNO представлены в таблице 1.

Таблица 1. Технические характеристики Arduino UNO

|  |  |
| --- | --- |
| Микроконтроллер | ATmega328 |
| Рабочее напряжение | 5В |
| Напряжение питания (рекомендуемое) | 7-12В |
| Напряжение питания (предельное) | 6-20В |
| Цифровые входы/выходы | 14 (из них 6 могут использоваться в качестве ШИМ-выходов) |
| Аналоговые входы | 6 |
| Максимальный ток одного вывода | 40 мА |
| Максимальный выходной ток вывода 3.3V | 50 мА |
| Flash-память | 32 КБ (ATmega328) из которых 0.5 КБ используются загрузчиком |
| SRAM | 2 КБ (ATmega328) |
| EEPROM | 1 КБ (ATmega328) |
| Тактовая частота | 16 МГц |

## 1.2. Электромагнитное реле

К Arduino UNO можно подключать устройства, которые работают с относительно большими токами или напряжением. Подключать напрямую двигатели, лампы накаливания к плате нельзя, так как она не предназначена для таких нагрузок. Сделать такое подключение возможно с помощью электромагнитного реле (рисунок 2.2).



Рисунок 2.2 – Электромагнитное реле

Электромагнитное реле – это электрическое устройство, замыкающее и размыкающее механическим путем цепь нагрузки при помощи магнита. Реле состоит из электромагнита, подвижного якоря и переключателя [3].

Схема реле представлена на рисунке 2.3.

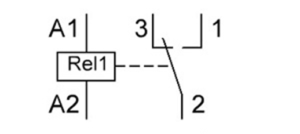


Рисунок 2.3 – Схема модуля реле

Реле характеризуется следующими параметрами:

* Напряжение или ток срабатывания.
* Напряжение или ток отпускания.
* Время срабатывания и отпускания.
* Рабочие ток и напряжение.
* Внутреннее сопротивление.

## 1.3. Датчик температуры DS18B20

Для измерения температуры используется герметичный датчик DS18B20 (рисунок 2.5). Датчик представляет собой кабель определенной длины, на конце которого располагается герметичный терморезистор [4]. Принцип работы таких датчиков заключается в следующем: в зависимости от температуры окружения, терморезистор меняет свое сопротивление.



Рисунок 2.5 – Датчик измерения температуры DS18B20

Датчик DS18B20 работает в среде с диапазоном температур от -55 ˚С до +125 ˚С. Подключение датчика измерения температуры осуществляется с помощью резистора, сопротивление которого равняется 4,7 кОм. На один цифровой вывод платы микроконтроллера можно присоединить большое множество датчиков DS18B20. У каждого датчика есть свой физический номер, по которому можно определить, к какому датчику обращаться.

## 1.4. Постоянный резистор

Резисторы – пассивные элементы электрических цепей, которые обладают постоянным или переменным электрическим сопротивлением (рисунок 2.6) [5].



Рисунок 2.6 – Резисторы с постоянным сопротивлением

## 1.5. Часы реального времени RTC на модуле DS1307

Модуль часов реального времени – плата, содержащая микросхему DS1307 (рисунок 2.7) [6]. На плате располагается разъем для батарейки. Модуль применяется для точного контроля времени, даты. Из-за автономного питания от батарейки модуль часов можно отключать от Arduino, дата и время не будут сбиваться.



Рисунок 2.7 – Модуль часов реального времени DS1307

Технические параметры модуля часов реального времени:

* Питание – 5В;
* Диапазон рабочих температур от -40С до 85С;
* 56 байт памяти;
* Литиевая батарейка LIR2032;
* Реализует 12-ти и 24-х часовые режимы;
* Поддержка интерфейса I2C.

**1.6. Макетная плата**

Макетная плата – печатная плата для моделирования и сборки прототипов электронных устройств [7]. Существует два вида макетных плат:

* Для монтажа посредством пайкой;
* Для монтажа без пайки.

Макетные платы, позволяющие соединять электронные компоненты между собой без пайки, удобны в использовании по причине того, что в случае неправильного подключения какого-либо модуля, можно без труда произвести изменения в подключении (рисунок 2.8)

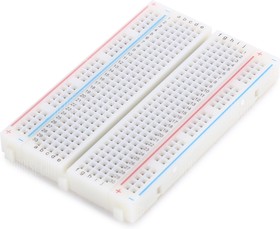


Рисунок 2.8 – Макетная плата

1. **ПРОЕКТИРОВАНИЕ СИСТЕМЫ РЕГУЛИРОВАНИЯ МИКРОКЛИМАТА**

Для проектирования системы необходимы такие компоненты как: микроконтроллер Arduino UNO, модуль электромагнитного реле, резистор, датчик измерения температуры DS18B20. Программная часть разрабатывается с помощью среды разработки Arduino IDE.

Arduino IDE – среда для разработки и загрузки микропрограмм в память контроллера.

## Подключение компонентов в системе

Для того, чтобы система работала корректно, необходимо произвести правильное подключение всех компонентов системы так, чтобы соблюсти полярность.

Датчики измерения температуры DS18B20 подключаются к микроконтроллеру Arduino UNO через резистор с номинальным сопротивлением 4,7 кОм.

Резисторы, которые выдают (ВЫДАЮТ) постоянное сопротивление, подключаются к положительным контактам модуля реле, которые предназначены для подключения устройств (рисунок 3.7).

Электромагнитный модуль реле подключается к плате микроконтроллера Arduino UNO с помощью трех контактов:

* Контакт VСС модуля реле подключается к контакту 5V на плате микроконтроллера;
* Контакт GND модуля реле подключается к контакту GND на плате микроконтроллера;
* Контакт IN модуля реле подключается к одному из цифровых выводов на плате микроконтроллера для обмена данными.

Для охлаждения и обогрева воздуха используется модуль реле с двумя каналами подключения. К одному из каналов подключен резистор для охлаждения воздуха, к другому каналу подключен резистор для обогрева.

Система в данный момент тестируется в контейнере, в котором растет салат. В ночное время для салата оптимальной температурой является +16 ˚C, днем температура должна соответствовать значению +22 ˚С.

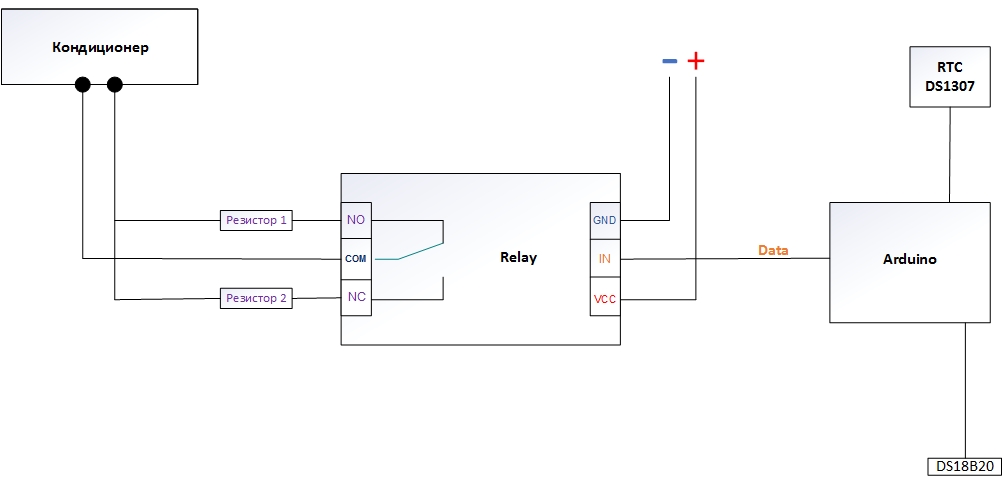


Рисунок 3.7 – Подключение компонентов в системе

Модель, спроектированная с помощью online-сервиса TinkerCAD, изображена на рисунке 3.8.

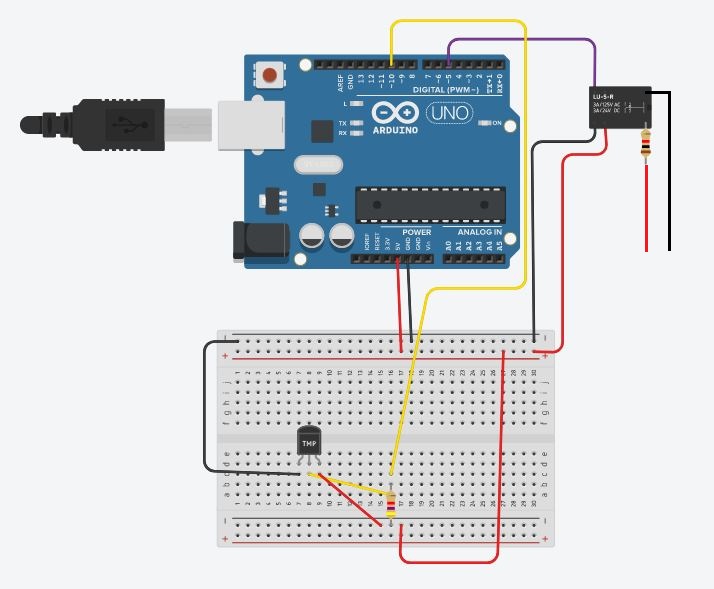


Рисунок 3.8 – Модель, спроектированная в TinkerCAD

Исходный код для управления модулем реле в режиме охлаждения:

if (time.Hours >= 19 && time.minutes >= 00 && time.Hours <= 19 && time.minutes <= 30)

{

if (sr > 14)

{

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

//digitalWrite(PIN\_RELAY, HIGH);

}

if (sr < 14)

{

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

//digitalWrite(PIN\_RELAY, LOW);

}

Исходный код для установки времени и даты для модуля RTC DS1307:

time.begin();

time.settime(0,00,19,01,12,19,1);

**ПРИЛОЖЕНИЕ**

#include <TroykaDHT.h> //DHT 11

#include "OneWire.h" //DS18B20

#include "DallasTemperature.h" //DS18B20

#include <iarduino\_RTC.h> //Часы

iarduino\_RTC time(RTC\_DS1307);

#define ONE\_WIRE\_BUS 10 //DS18B20

//#define PIN\_RELAY 5

int in1 = 9;

int in2 = 11;

DHT dht1(2, DHT11);

DHT dht2(3, DHT11);

DHT dht3(4, DHT11);

DHT dht4(6, DHT11);

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

float sr;

DeviceAddress sensor1 = { 0x28, 0xA6, 0x1E, 0x79, 0x97, 0x20, 0x3, 0x64 };

DeviceAddress sensor2 = { 0x28, 0xCD, 0xD, 0x79, 0x97, 0x20, 0x3, 0x6D };

DeviceAddress sensor3= { 0x28, 0x12, 0x4F, 0x79, 0x97, 0x19, 0x3, 0x78 };

DeviceAddress sensor4= { 0x28, 0x22, 0x6E, 0x79, 0x97, 0x20, 0x3, 0x3F };

void setup()

{

Serial.begin(9600);

time.begin();

time.settime(0,00,19,01,12,19,1);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

//pinMode(PIN\_RELAY, OUTPUT);

//digitalWrite(PIN\_RELAY, HIGH);

dht1.begin();

dht2.begin();

dht3.begin();

dht4.begin();

sensors.begin();

}

void loop()

{

dht1.read();

dht2.read();

dht3.read();

dht4.read();

Serial.println("Датчики DHT11");

Serial.println("------------------------");

Serial.println("------------------------");

Serial.print("Температура\_1 = ");

Serial.print(dht1.getTemperatureC());

Serial.println(" C \t");

Serial.print("Температура\_2 = ");

Serial.print(dht2.getTemperatureC());

Serial.println(" C \t");

Serial.print("Температура\_3 = ");

Serial.print(dht3.getTemperatureC());

Serial.println(" C \t");

Serial.print("Температура\_4 = ");

Serial.print(dht4.getTemperatureC());

Serial.println(" C \t");

Serial.println("------------------------");

Serial.print("Влажность\_1 = ");

Serial.print(dht1.getHumidity());

Serial.println(" %");

Serial.print("Влажность\_2 = ");

Serial.print(dht2.getHumidity());

Serial.println(" %");

Serial.print("Влажность\_3 = ");

Serial.print(dht3.getHumidity());

Serial.println(" %");

Serial.print("Влажность\_4 = ");

Serial.print(dht4.getHumidity());

Serial.println(" %");

Serial.println("------------------------");

Serial.println("------------------------");

Serial.println("Датчики DS18B20");

Serial.println("------------------------");

Serial.println("------------------------");

sensors.requestTemperatures();

Serial.print("Sensor 1(\*C): ");

Serial.println(sensors.getTempC(sensor1));

Serial.print("Sensor 2(\*C): ");

Serial.println(sensors.getTempC(sensor2));

Serial.print("Sensor 3(\*C): ");

Serial.println(sensors.getTempC(sensor3));

Serial.print("Sensor 4(\*C): ");

Serial.println(sensors.getTempC(sensor4));

Serial.println("------------------------");

Serial.println("------------------------");

sr = (sensors.getTempC(sensor1) + sensors.getTempC(sensor2) + sensors.getTempC(sensor3) + sensors.getTempC(sensor4)) / 4;

Serial.println(sr);

Serial.println("------------------------");

time.gettime();

Serial.println(time.gettime("d-m-Y, H:i:s, D"));

if (time.Hours >= 19 && time.minutes >= 00 && time.Hours <= 19 && time.minutes <= 30)

{

if (sr > 14)

{

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

//digitalWrite(PIN\_RELAY, HIGH);

}

if (sr < 14)

{

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

//digitalWrite(PIN\_RELAY, LOW);

}

Serial.println(time.gettime("d-m-Y, H:i:s, D"));

}

if (time.Hours >= 19 && time.minutes >= 31 && time.Hours <= 19 && time.minutes <= 59)

{

if (sr < 21)

{

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

//digitalWrite(PIN\_RELAY, LOW);

}

if (sr > 21)

{

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

//digitalWrite(PIN\_RELAY, HIGH);

}

Serial.println(time.gettime("d-m-Y, H:i:s, D"));

}

delay(30000);

}