

## Проект 1.3

### Система детекции нарушения климатических условий

Создание системы детекции нарушения климатических условий является одной из наиболее актуальных задач в современном мире. Ардуино предоставляет отличную платформу для разработки подобных систем, благодаря своей простоте использования и гибкости. В данной лабораторной работе мы будем создавать систему климат-контроля на базе Arduino, которая будет автоматически регулировать температуру и влажность в помещении. Мы рассмотрим основные компоненты системы, такие как: датчики температуры и влажности, модуль управления нагрузкой, а также программирование Arduino для работы с этими компонентами. В результате выполнения лабораторной работы мы получим полнофункциональную систему климат-контроля, которая сможет самостоятельно поддерживать оптимальные условия в помещении.

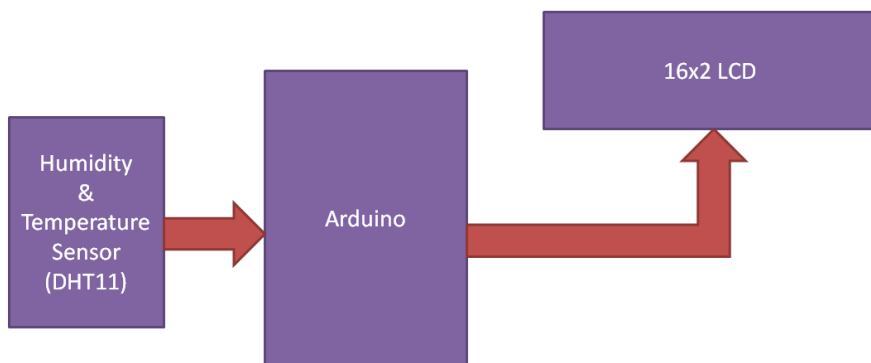
Базовая лабораторная работа будет разделена на несколько отдельных частей: подключение каждого из датчиков и модулей отдельно. В последствии вам предстоит объединить все в умную систему.

### Теоретическая часть:

#### Система контроля температуры и влажности

Температура и влажность являются одними из важнейших параметров окружающей нас среды. В этом проекте мы будем измерять температуру и влажность окружающей среды и с помощью Arduino выводить их значения на жидкокристаллический (ЖК) дисплей 16x2. Для измерения этих параметров мы будем использовать комбинированный датчик для измерения температуры и влажности, который может измерять температуру в градусах Цельсия и влажность в процентах.

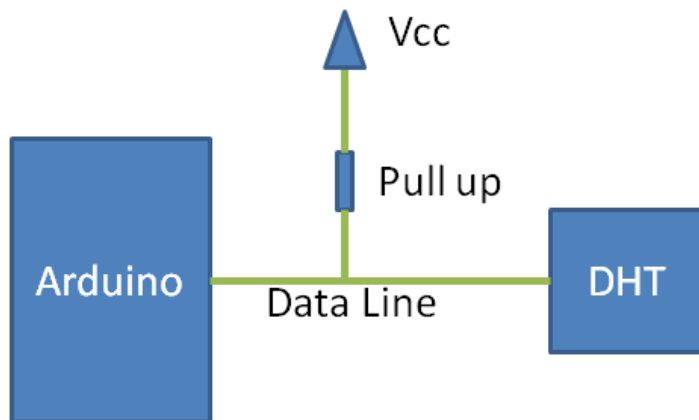
В этом проекте мы будем использовать датчик DHT11. Этот модуль способен измерять температуру и влажность и выдавать их значения на выход в виде калиброванного цифрового сигнала. Этот датчик обеспечивает высокую точность измерений, высокую надежность и стабильность измерений. Принцип действия данного датчика основан на резистивном типе измерения влажности и NTC-типе (NTC - отрицательный температурный коэффициент) измерения температуры. Данный датчик имеет встроенный 8-битный микроконтроллер, что обеспечивает ему быстрый отклик на изменения окружающей среды и приемлемую стоимость. Датчик имеет 4 выходных контакта.



Модуль DHT11 работает по принципу последовательной связи по одиночному проводнику. Этот модуль передает данные в виде последовательности импульсов с определенным периодом. Но прежде, чем он начнет передавать данные ему нужно подать команду инициализации с Arduino. Весь этот процесс занимает примерно 4 мс. Процесс передачи 40 бит данных будет включать в себя:

8-bit integral RH data + 8-bit decimal RH data + 8-bit integral T data + 8-bit decimal T data + 8-bit check sum (проверочная сумма).

Процесс передачи данных осуществляется следующим образом: вначале плата Arduino подает на датчик DHT11 сигнал перепада с высокого уровня на низкий с задержкой 18 мкс чтобы запустить датчик DHT11 в работу. Затем Arduino pull-up (подтягивает ее к Vcc как показано на рисунке) линию передачи данных ждет 20-40 мкс ответа от датчика DHT11. Как только датчик DHT11 обнаруживает сигнал, запускающий его в работу (сигнал старта), он передает ответ в виде сигнала низкого уровня длительностью около 80 мкс. А затем контроллер DHT датчика pull up линию передачи данных и удерживает ее в этом состоянии около 80 мкс пока DHT модуль готовится к передаче данных.



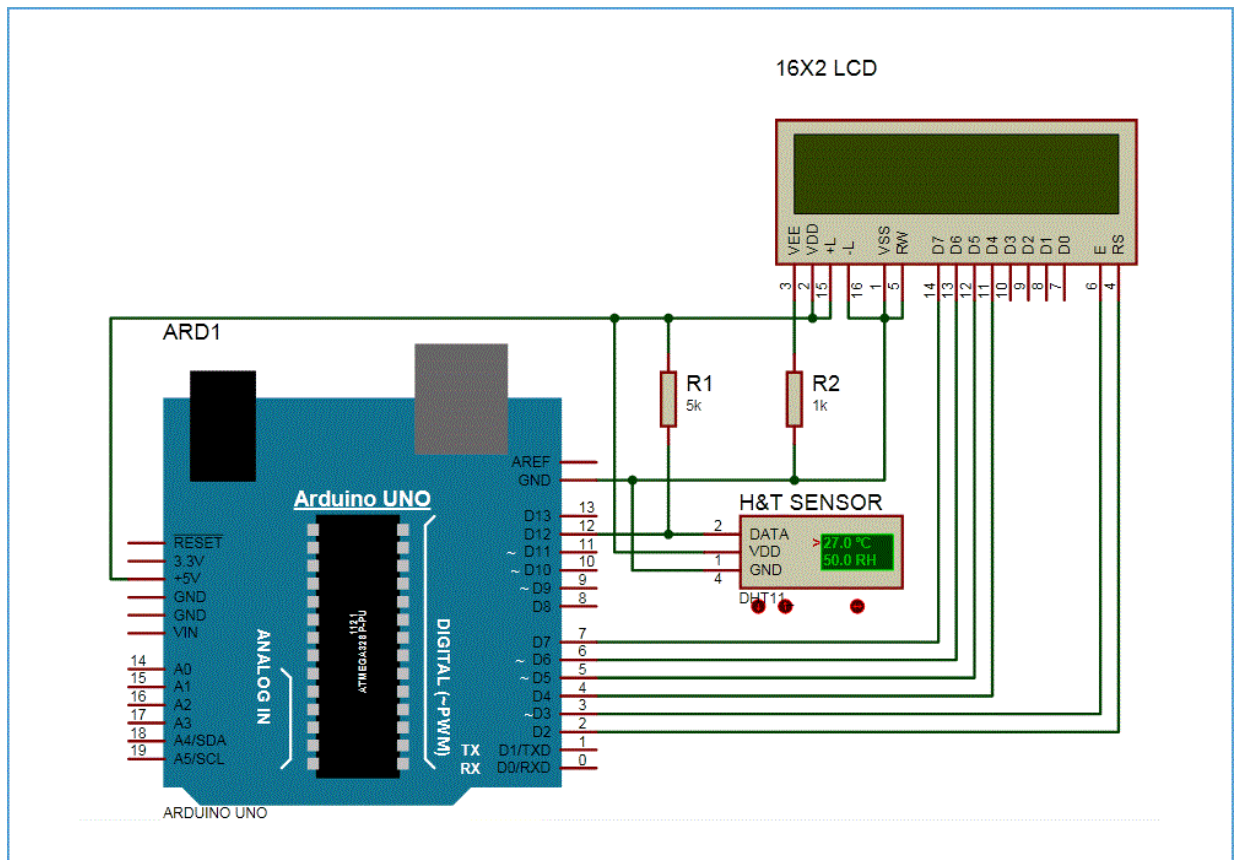
Как только на линии передачи данных появилось напряжение низкого уровня это значит, что датчик DHT11 передает ответный сигнал на Arduino. Как только этот процесс будет завершен, DHT снова "подтягивает" (pull-up) линию передачи данных на 80 мкс чтобы подготовиться к передаче данных.

Формат данных, передаваемых с DHT на Arduino, для каждого бита начинается с 50 мкс напряжения низкого уровня, а продолжительность напряжения высокого уровня после этого зависит от того какой бит передается - "0" или "1".

Но если вы мало что поняли в этих процессах, не расстраивайтесь, потому что для Arduino написана специальная библиотека для работы с модулем DHT, которая сводит все операции с этим модулем к очень простым действиям – и вы это наглядно увидите в тексте программы, представленном в конце. А вот если бы вы вместо Arduino использовали бы, к примеру, микроконтроллер AVR, то вам все описанные операции пришлось бы программировать вручную. В рассмотренных процессах весьма важным является правильный выбор значения подтягивающего (pull up) резистора. Поскольку мы располагаем датчик DHT на расстоянии менее чем 20 метров от Arduino, то

подтягивающего резистора номиналом 5 кОм будет достаточно. Если же мы будем располагать датчик DHT на расстоянии более 20 метров, то в этом случае необходимо использование соответствующего значения подтягивающего резистора.

*Принципиальная схема будет иметь следующий вид:*



## Система автоматического полива растений

В данной лабораторной работе мы соберем и запрограммируем систему на базе платы Arduino UNO и датчика влажности почвы. Она будет автоматически поддерживать оптимальный уровень влаги в почве и поливать ваши растения, позволяя вам не беспокоиться об этом во время вашего отсутствия. Эта система может быть использована как внутри помещений дома, так и в саду.

Принцип работы системы.

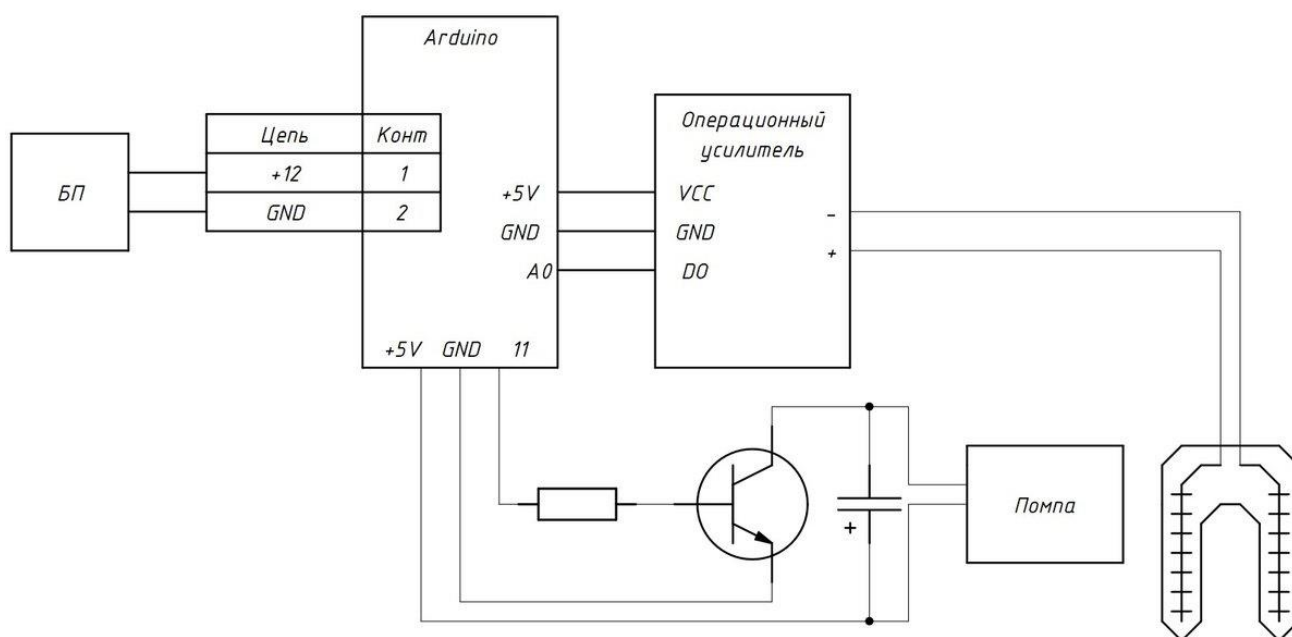
1) Датчик влажности будет использоваться для измерения уровня влаги в почве. Когда этот уровень опустится до определенного порога, плата Arduino Uno отправит сигнал на водяной насос, который начнет подачу воды в почву, обеспечивая орошение наших растений. Когда уровень влажности в почве достигнет нужного уровня, датчик влажности обнаружит это и отправит сигнал для остановки водяного насоса.

**Справка:** Принцип работы датчика влажности почвы достаточно прост – он содержит 2 зонда с оголенными контактами, которые работают как резистор с переменным сопротивлением – их сопротивление изменяется в зависимости от уровня влаги в почве. Сопротивление данных зондов обратно пропорционально влажности почвы, то есть чем выше содержание влаги почвы, тем лучше ее проводимость и тем меньше ее

сопротивление. Низкий уровень влаги в почве свидетельствует о ее плохой проводимости и, следовательно, ее большом сопротивлении. Датчик обеспечивает на своем выходе уровень аналогового напряжения, пропорциональный сопротивлению почвы.

2\*) Система сможет посылать сообщения, в которых будет содержаться информация об уровне влажности почвы, статус работы насоса (и т.д.) через Интернет в любую систему распространения информации, будь то электронный почтовый ящик владельца или какой-либо из мессенджеров. Например: система каждую секунду проверяет уровень влажности почвы и раз в минуту отправляет уведомление со средним показателем влажности почвы.

*Принципиальная схема будет иметь следующий вид:*



### Система автоматической детекции огня

Датчик огня представляет собой устройство, предназначенное для обнаружения огня или пламени. При обнаружении огня затем могут выполняться различные действия: включение звуковой сигнализации, деактивация линии с топливом, включение системы тушения огня и т.д.

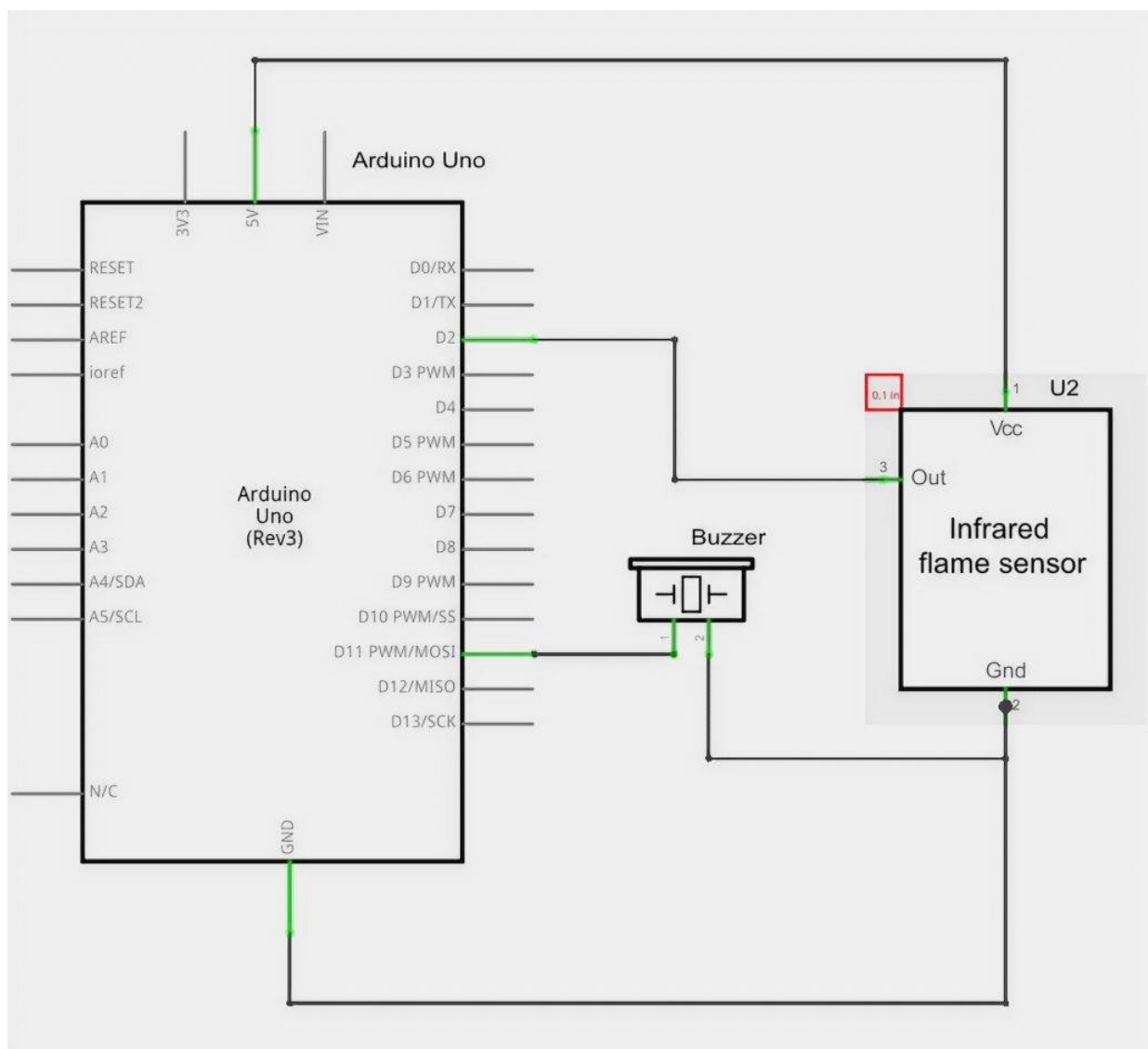
Существуют различные датчики обнаружения огня (пламени), например, ультрафиолетовый детектор, инфракрасный детектор, инфракрасные термические камеры, комбинированный инфракрасный-ультрафиолетовый детектор и т.д.

Во время горения огня выделяется небольшое количество инфракрасного света, который может быть принят (обнаружен) с помощью фотодиода (инфракрасного приемника). Затем мы будем использовать операционный усилитель чтобы проверить изменения напряжения на выходе инфракрасного приемника (IR receiver). При обнаружении огня на выходном

контакте датчика (DO) будет формироваться напряжение низкого уровня (0V), а при отсутствии огня на его выходном контакте будет напряжение высокого уровня (5V).

В этом проекте мы использовали инфракрасный датчик огня. В нем используется датчик YG1006, который имеет в своем составе высокоскоростной и высокочувствительный NPN кремниевый фототранзистор. Он может обнаруживать инфракрасный свет с длиной волны от 700 до 1000 нм, а его угол обнаружения приблизительно составляет 60°. Модуль датчика огня состоит из фотодиода (инфракрасного приемника), резистора, конденсатора, потенциометра и компаратора LM393 в виде одной интегральной схемы. Чувствительность регулируется имеющимся на плате потенциометром. Рабочее напряжение составляет от 3.3v до 5v DC (постоянного тока), с цифровым выходом.

*Принципиальная схема будет иметь следующий вид:*

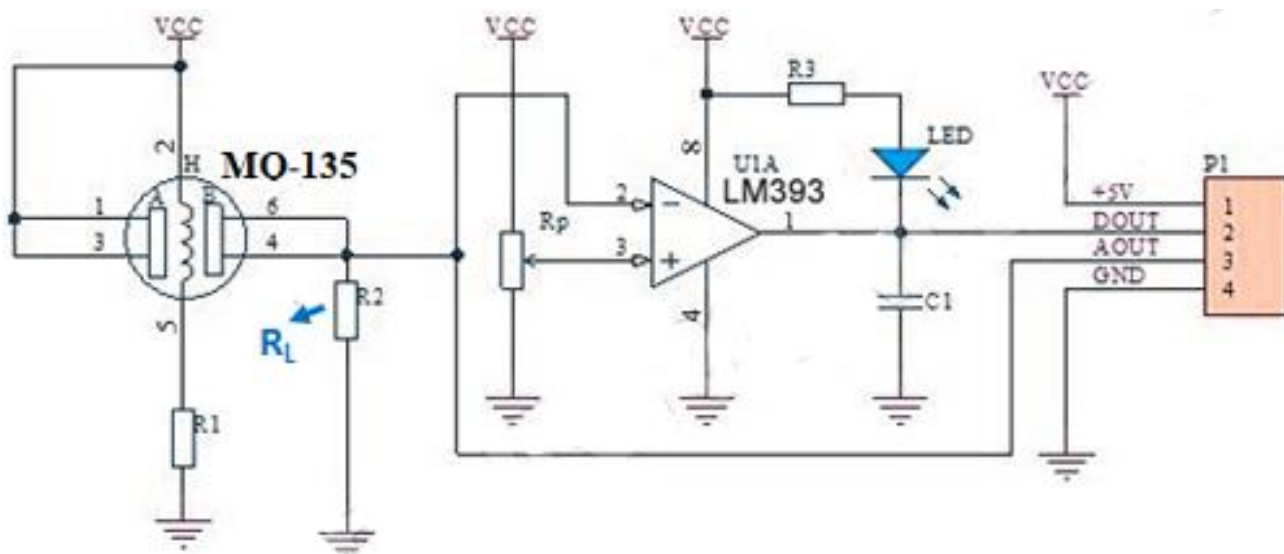


## Система контроля уровня углекислого газа

Уровень концентрации углекислого газа ( $\text{CO}_2$ ) в атмосфере Земли увеличивается с каждым годом, что косвенно приводит к проблеме глобального потепления. Также углекислый газ играет важную роль в процессах, происходящих в современном мире. Поэтому измерение его концентрации в окружающем воздухе является достаточно важной и актуальной задачей.

Мы рассмотрим подключение датчика MQ-135 к плате Arduino и измерение с его помощью концентрации углекислого газа ( $\text{CO}_2$ ).

Датчик газа MQ-135 способен обнаруживать широкий диапазон различных газов в окружающем воздухе:  $\text{NH}_3$ ,  $\text{NO}_x$ , алкоголь, бензол, дым и углекислый газ ( $\text{CO}_2$ ). Датчик MQ-135 можно купить как в виде модуля, так и в виде отдельного датчика. В нашем проекте мы будем использовать его в виде модуля для измерения концентрации  $\text{CO}_2$  в единицах PPM (parts per million – частей на миллион). Схема модуля датчика MQ-135 показана на следующем рисунке:



В этой схеме весьма важную роль играет нагрузочный резистор  $R_L$  – его сопротивление может изменяться в зависимости от концентрации газа. В соответствии с даташитом на датчик MQ-135 сопротивление нагрузочного резистора может изменяться от 10 кОм до 47 кОм. Даташит рекомендует чтобы вы калибровали датчик для 100ppm  $\text{NH}_3$  или для 50ppm концентрации алкоголя в воздухе и использовали значение нагрузочного резистора  $R_L$  примерно 20 кОм. Но если вы внимательно посмотрите на плату модуля MQ-135, то вы увидите, что значение резистора  $R_L$  на ней составляет 1 кОм (102).



Поэтому, чтобы корректно измерять концентрацию CO<sub>2</sub>, вам необходимо заменить этот резистор 1 кОм на резистор сопротивлением 22 кОм.

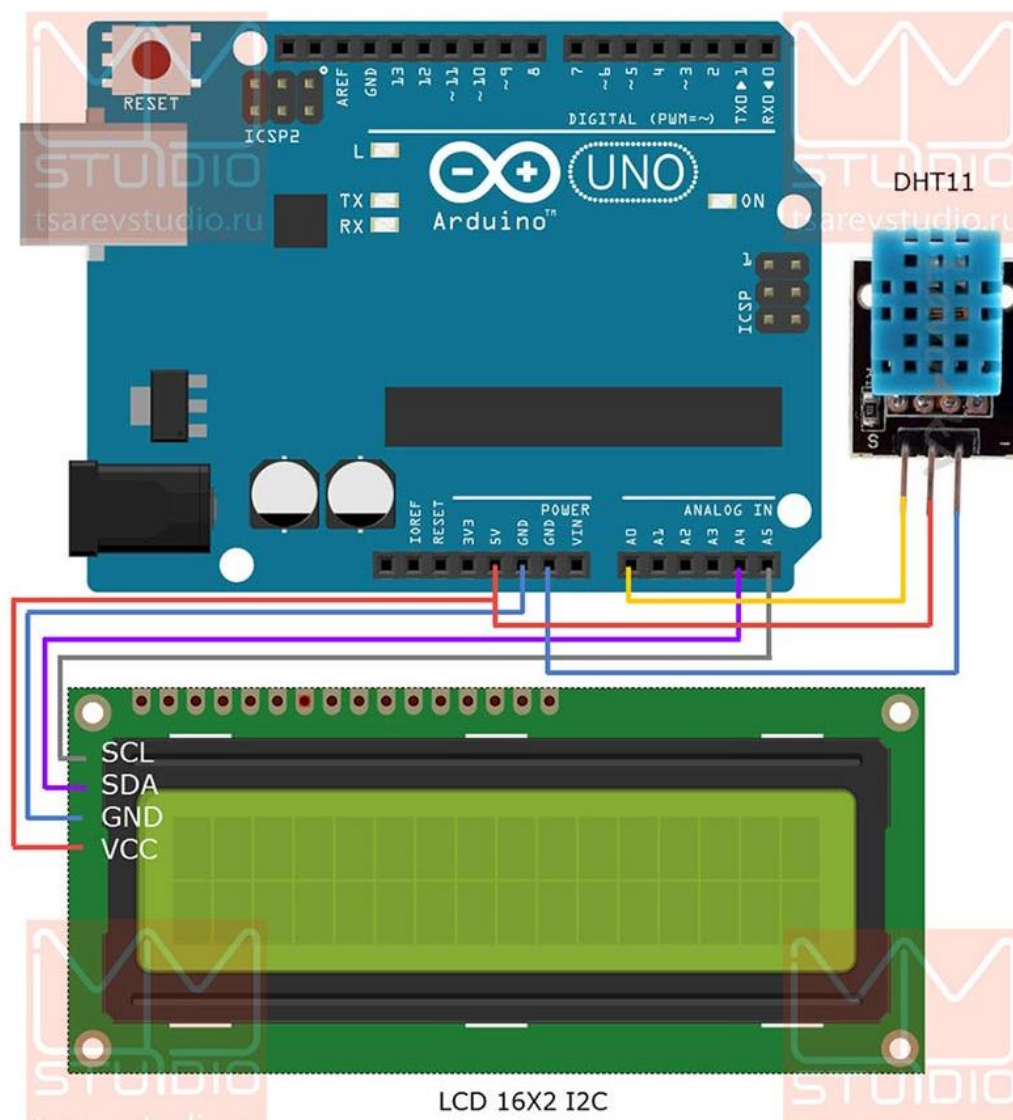
## Практическая часть:

### 1. Измерение температуры и влажности

Функционально схема устройства состоит из 3-х основных блоков: первый блок измеряет температуру и влажность с помощью датчика DHT11, второй блок считывает эти значения с выхода датчика и преобразует их в форму, удобную для восприятия, а третий блок отображает значения влажности и температуры на ЖК дисплее.

Работа нашей схемы будет основана на последовательной передаче данных по одиночному проводнику. Вначале Arduino передает стартовый сигнал (то есть чтобы DHT модуль начал работу) DHT модулю, а затем DHT модуль (то есть модуль измерения температуры и влажности) выдает Arduino ответный сигнал, содержащий данные о температуре и влажности. Arduino собирает эти данные, обрабатывает и передает на ЖК дисплей 16x2.

*Физическая схема подключения:*



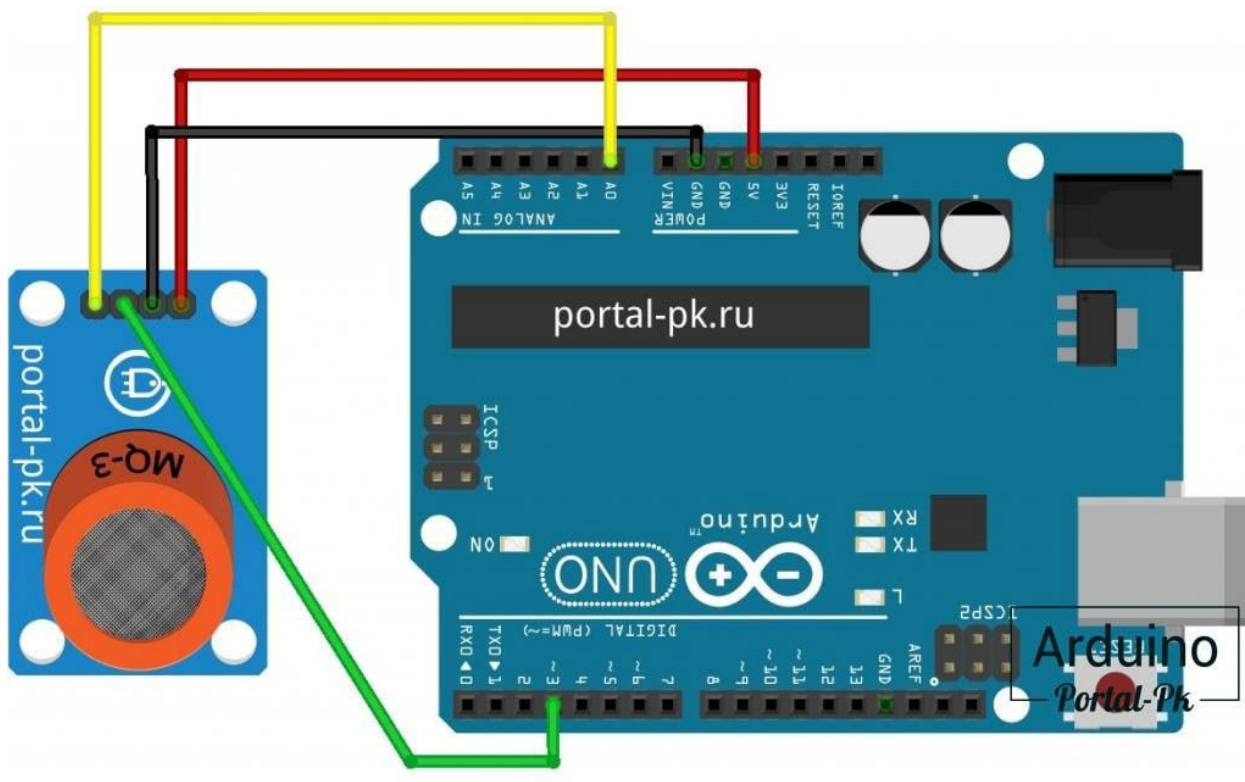
Таким образом наша схема будет иметь следующий код:

```
#include <Wire.h> // Добавляем необходимые библиотеки
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
#define DHTPIN A0 // к какому пину будет подключен сигнальный
выход датчика
#define DHTTYPE DHT11 //выбор используемого датчика DHT 11
//#define DHTTYPE DHT22 // DHT 22 (AM2302)
//#define DHTTYPE DHT21 // DHT 21 (AM2301)
//инициализация датчика
DHT dht(DHTPIN, DHTTYPE);
byte degree[8] = // кодируем символ градуса
{
  B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000,
};
LiquidCrystal_I2C lcd(0x27,16,2); // Задаем адрес и размерность
дисплея
void setup()
{
  lcd.init(); // Инициализация lcd
  lcd.backlight(); // Включаем подсветку
  lcd.createChar(1, degree); // Создаем символ под номером 1
  Serial.begin(9600);
  dht.begin();
}
void loop() {
  // Добавляем паузы в несколько секунд между измерениями
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very
  slow sensor)
  float h = dht.readHumidity();
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit
  float f = dht.readTemperature(true);
  // Выводим показания влажности и температуры
  lcd.setCursor(0, 0); // Устанавливаем курсор в начало 1 строки
  lcd.print("Hum = % "); // Выводим текст
  lcd.setCursor(7, 0); // Устанавливаем курсор на 7 символ
  lcd.print(h, 1); // Выводим на экран значение влажности
  lcd.setCursor(0, 1); // Устанавливаем курсор в начало 2 строки
  lcd.print("Temp = \1C "); // Выводим текст, \1 - значок градуса
  lcd.setCursor(7, 1); // Устанавливаем курсор на 7 символ
  lcd.print(t,1); // Выводим значение температуры
}
```



## 2. Измерение уровня углекислого газа

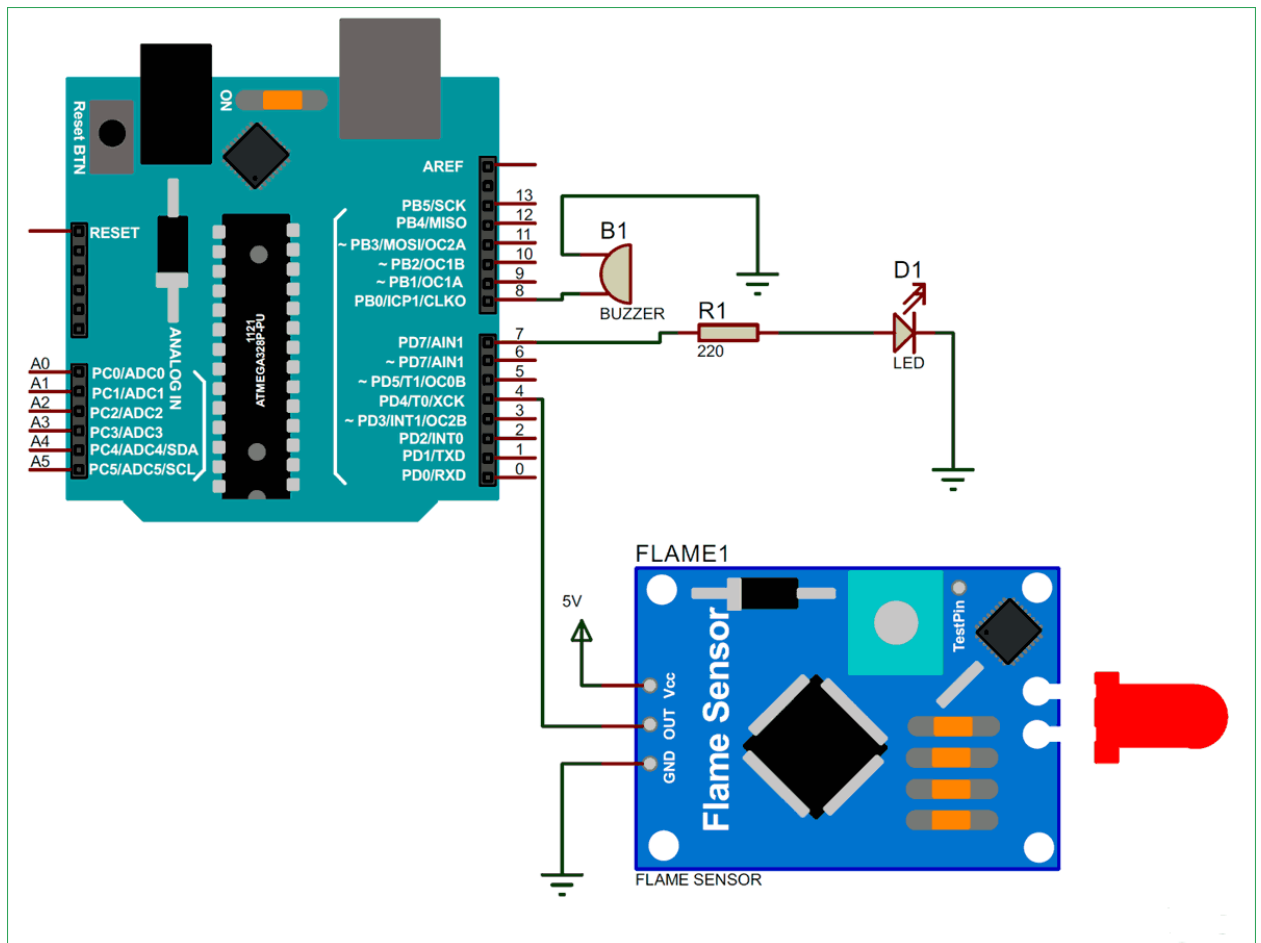
Физическая схема подключения:



Таким образом наша схема будет иметь следующий код:

```
#include "MQ135.h"
int soilMoistureValue = 0;
int percentage=0;
#define ANALOGPIN A1 //подключение аналогового сигнального пина
#define RZERO 206.85
MQ135 gasSensor = MQ135(ANALOGPIN);
boolean noGas; //переменная для хранения значения о присутствии
газа
int gasValue = 0; //переменная для хранения количества газа
void setup()
{
    pinMode(3,OUTPUT);
    float rzero = gasSensor.getRZero();
    Serial.begin(9600);
}
void loop()
{
    soilMoistureValue = analogRead(A0);
    float ppm = gasSensor.getPPM();
    Serial.println(percentage);
    Serial.print("CO2 ppm value : ");
    Serial.println(ppm);
    delay(2000); // 2 секунды задержки
}
```

### 3. Автоматическая детекция пламени



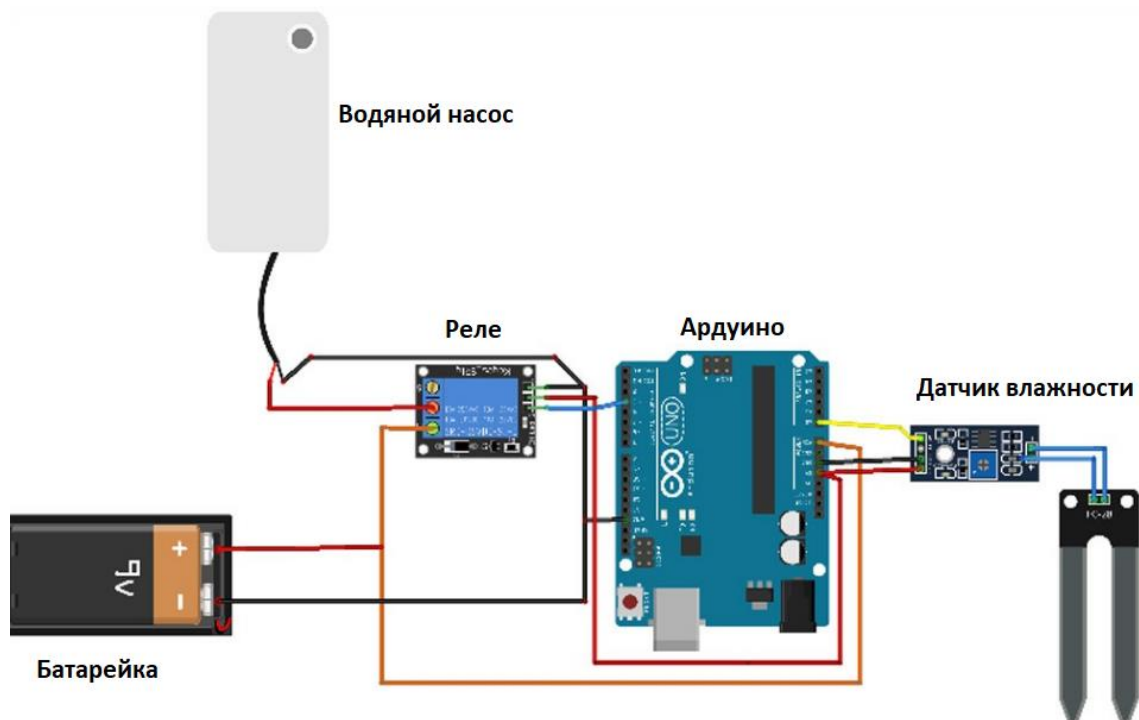
Таким образом наша схема будет иметь следующий код:

```
int sensorPin = 4;
int flame;
void setup()
{
    pinMode(sensorPin, INPUT);
    Serial.begin(9600);
}
void loop()
{
    flame = digitalRead(sensorPin);

    if (flame == 1)
    {
        Serial.println("Обнаружено пламя");
        delay(150);
    }
    else
    {
        Serial.println("Пламя не обнаружено");
        // delay(500);
    }
}
```

## 4. Автоматический полив

Физическая схема подключения



Таким образом наша схема будет иметь следующий код:

Инициализация разъемов и основных переменных:

```
#define PUMP_PIN 3
int soilMoistureValue = 0;
int percentage=0;
```

Блок кода, который срабатывает при первичном запуске системы:

```
void setup() {
  pinMode(PUMP_PIN, OUTPUT);
  Serial.begin(9600);
}
```

Блок кода, который работает циклично на протяжении работы системы:

```
void loop() {
  soilMoistureValue = analogRead(A0);
  Serial.println(percentage);
  percentage = map(soilMoistureValue, 490, 1023, 100, 0);
  if(percentage < 10)
  {
    Serial.println(" pump on");
    digitalWrite(PUMP_PIN, LOW);
  }
  if(percentage > 80)
  {
    Serial.println("pump off");
    digitalWrite(PUMP_PIN, HIGH);
  }
}
```

Далее приведен полный код скетча:

```
#define PUMP_PIN 3
int soilMoistureValue = 0;
int percentage=0;

void setup() {
  pinMode(PUMP_PIN,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  soilMoistureValue = analogRead(A0);
  Serial.println(percentage);
  percentage = map(soilMoistureValue, 490, 1023, 100, 0);
  if(percentage < 10)
  {
    Serial.println(" pump on");
    digitalWrite(PUMP_PIN,LOW);
  }
  if(percentage >80)
  {
    Serial.println("pump off");
    digitalWrite(PUMP_PIN,HIGH);
  }
}
```

На этом написание скетча можно считать завершенным. Осталось загрузить его в плату и протестировать корректность работы системы.

## Усложняем задачу:

Мы сделали 4 отдельные схемы, но было бы классно собрать их в 1 систему, чтобы они работали вместе и давали нам полные данные о показателях помещения.

- 1) Объедините все 4 схемы в 1 с помощью беспаячной платы
- 2) Добавьте к схеме детекции огня пищалку и красный светодиод. В случае обнаружения огня пищалка должна давать соответствующий сигнал, а красный диод мигать
- 3) Добавьте аналогичный системы звуко и светооповещения для других датчиков. Их триггером будет служить некоторое критическое значение, задаваемое константой.
- 4) Подключите выведение всей информации из консоли на экран в последовательном виде, с подписями что есть что.
- 5) Чтобы система детекции стала системой климат контроля, необходимо добавить устройство изменения условий. В данном случае, чтобы изменять влажность воздуха достаточно установить на конец трубки насоса ороситель и сделать так, чтобы он включался не только по датчику влажности почвы, но и по датчику влажности воздуха.

\*\*\*

- 6) Добавьте для каждого из датчиков собственную мелодию и мигание диода, чтобы можно было однозначно идентифицировать нештатную ситуацию моментально.