**СИСТЕМА АВТОМАТИЗАЦИИ И КОНТРОЛЯ КОНЦЕНТРАЦИИ**

В качестве элементной базы для реализации автоматизации и контроля системы используются микроконтроллеры – программируемые микросхемы для управления электронными устройствами; применяются датчики газа – модуль, работающий на основе сопротивления резистора, позволяющий выявить различные газа в воздухе, а также магнитный клапан на баллоне с углекислым газом, позволяющий ограничивать и контролировать подачу CO2 в системе.

Целью автоматизированной системы управления является разработка системы контроля и автоматизации уровня углекислого газа, а также производить контроль над уровнем содержания СО2 в и при необходимости увеличивать или понижать, производить процесс мониторинга, сбора, обработки, оценки данных окружающей среды и автоматически управлять системой с использованием удалённого доступа и мобильных технологий.

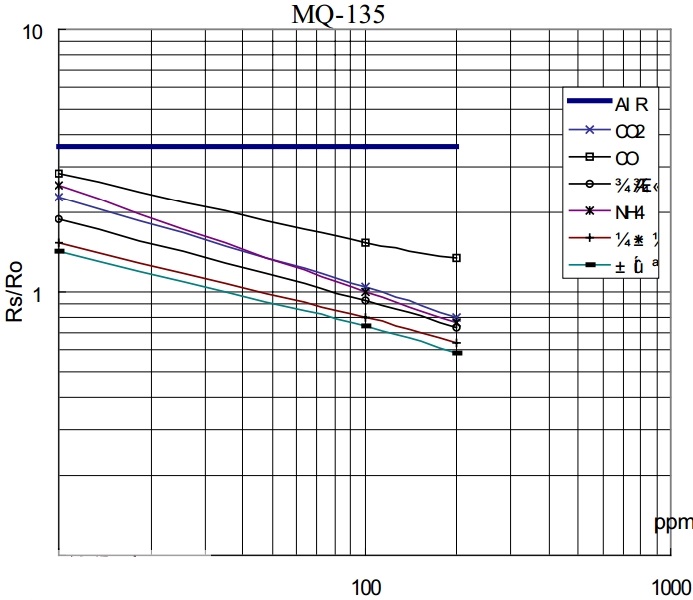
Совместно с платой Arduino будут использоваться подходящие для работы с ней датчики газа MQ-135, реле для контроля магнитного клапана, контролирующего выпуск углекислого газа, а также модуль часов реального времени DS1307 для контроля времени выпуска CO2.

Датчик MQ-135 (Рисунок 2) относиться к полупроводниковым приборам. Принцип работы датчика основан на изменении сопротивления тонкопленочного слоя диоксида олова SnO2 при контакте с молекулами определяемого газа. Чувствительный элемент датчика состоит из керамической трубки с покрытием Al2O3 и нанесенного на неё чувствительного слоя диоксида олова. Внутри трубки проходит нагревательный элемент, который нагревает чувствительный слой до температуры, при которой он начинает реагировать на определяемый газ. Чувствительность к разным газам достигается варьированием состава примесей в чувствительном слое.

Помимо углекислого газа, датчик также реагирует на присутствие других газов: угарного газа, аммиака, бензола, оксидов азота и паров спирта. На рисунке 1 приведена зависимость относительного сопротивления датчика от парциального давления разных газов — таким образом, из сопротивления можно вычислить концентрацию газа в воздухе.

Детектируемый газ: NH3, бензол, спирт, дым

Диапазон чувствительности: 10-300ppm NH3 10-1000ppm бензол 10-600ppm спирт 1%/м-10%/м дым



*Рисунок 1– Зависимость относительного сопротивления датчика от парциального давления разных газов*

**Особенности:**

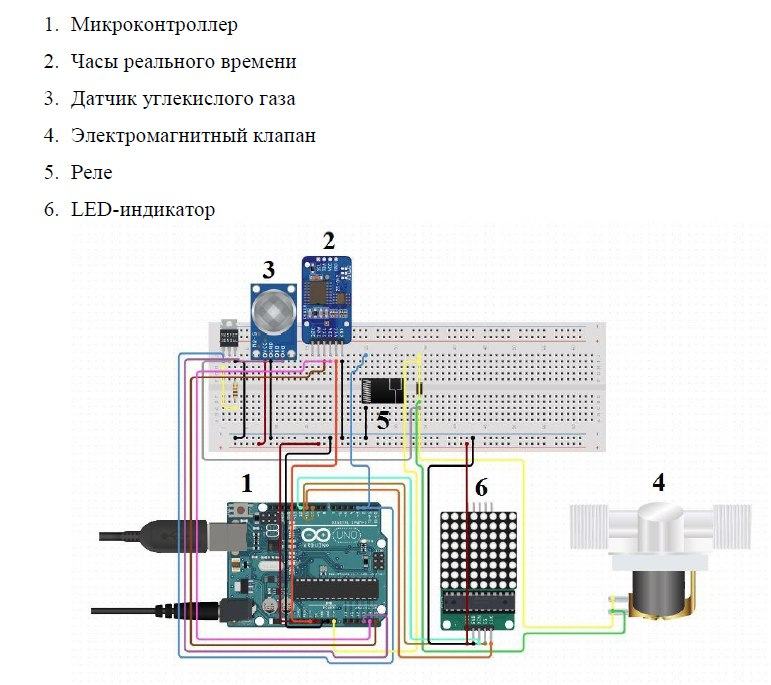
* **Чувствителен к бензолу, спирту и дыму;**
* **Выходное напряжение зависит от концентрации измеряемых газов;**
* **Быстрая реакция и восстановление;**
* **Регулируемая чувствительность.**



*Рисунок 2 – Датчик газа MQ-135*

Принцип работы системы:

Система, проводящая анализ воздуха на концентрацию газа получает информацию от датчиков размещенных по периметру помещения, после получения данных получает среднее значение концентрации, проводит проверку, достаточна ли имеющаяся концентрация для поддержания благоприятного для растений уровня CO2. Также проводится проверка, соблюдены ли условия временного режима день/ночь, при котором выпуск в среду углекислого газа возможен лишь днем. После прохождения двух этих проверок, если уровень CO2 недостаточно высок, магнитный клапан переключается для выпуска газа в среду. Выпуск углекислого газа происходит поэтапно, чтобы он мог распространиться по всему объему помещения.



*Рисунок 2 – Общий вид схемы системы*

Информационные ресурсы системы автоматизации и контроля

* Веб-ресурс (информационно-управляющий сайт);
* Система мониторинга;
* Набор, управляющих скриптов.

Основные функции информационно-управляющего сайта

* Производить процесс мониторинга, сбора, обработки, оценки данных окружающей среды, поступающих с датчиков;
* Автоматически управлять АСУ с использованием удалённого доступа и мобильных технологий.

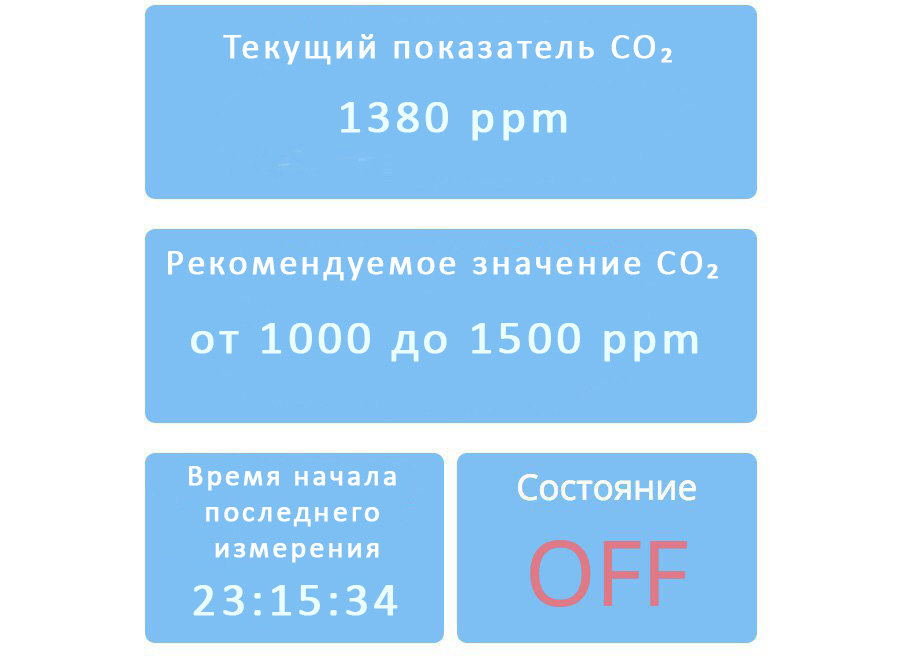
Принцип работы управляющего веб-сайта

Arduino подключается к серверу и отправляет GET запрос, где содержатся значения датчиков температуры. Сервер принимает запрос, и записывает значения уровня углекислого газа в текстовые файлы. При этом читает из текстового файла значение установленного выхода для arduino и отправляет в ответ на запрос контроллера. Arduino принимает ответ от сервера и согласно ему устанавливает состояние своего выхода. Панель управления, используя Ajax, считывает значение температуры из текстовых файлов и обновляет показания датчиков. А также считывает их текстового файла состояние выхода и обновляет его на странице. С помощью того же Ajax через форму в текстовый файл записывается значение выхода контроллера, откуда потом будет брать значение сервер и отправлять контроллеру.

Визуальное представление веб-сайта

На главной странице находится основные управляющие и информирующие компоненты, такие как:

* Индикатор показания текущего состояния датчика углекислого газа;
* Таблица с рекомендуемыми показателями состояния датчика;
* Текущий промежуток времени, за которое производится измерение;
* Состояние работы системы (on/off);



*Рисунок 3. Внешний вид веб-ресурса*

В дальнейшем планируется добавить систему индикации количества газа внутри баллона, что поможет узнать о количестве оставшегося газа, чтобы при нехватке его можно было заменить баллон подачи CO2, из-за чего можно будет без потерь во времени поддерживать жизненный цикл растений без малейших перебоев.

// библиотека для работы с датчиками MQ (Troyka-модуль)

#include <TroykaMQ.h>

#include "RTClib.h"

RTC\_DS1307 rtc;

// имя для пина, к которому подключен датчик

#define PIN\_MQ135\_1 A0 //1-4 датчики со2

#define PIN\_MQ135\_2 A1

#define PIN\_MQ135\_3 A2

#define PIN\_MQ135\_4 A3

#define PIN\_RELAY 5 //реле

float spr;

// создаём объект для работы с датчиком и передаём ему номер пина

MQ135 mq135\_1(PIN\_MQ135\_1);

MQ135 mq135\_2(PIN\_MQ135\_2);

MQ135 mq135\_3(PIN\_MQ135\_3);

MQ135 mq135\_4(PIN\_MQ135\_4);

void setup()

{

pinMode(PIN\_RELAY, OUTPUT); // Объявляем пин реле как выход

digitalWrite(PIN\_RELAY, HIGH); // Выключаем реле - посылаем высокий сигнал

//следующие 14 строк - вбивание времени

Serial.begin(57600);

if (! rtc.begin()) {

Serial.println("Couldn't find RTC");

while (1);

}

if (! rtc.isrunning()) {

Serial.println("RTC is NOT running!");

// following line sets the RTC to the date & time this sketch was compiled

rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

// This line sets the RTC with an explicit date & time, for example to set

// January 21, 2014 at 3am you would call:

// rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));

}

// открываем последовательный порт

// перед калибровкой датчика прогрейте его 60 секунд

// выполняем калибровку датчика на чистом воздухе

mq135\_1.calibrate();

mq135\_2.calibrate();

mq135\_3.calibrate();

mq135\_4.calibrate();

// при знании сопративления датчика на чистом воздухе

// можно его указать вручную, допустим 160

// mq135.calibrate(160);

// выводим сопротивление датчика в чистом воздухе (Ro) в serial-порт

Serial.print("Ro1 = ");

Serial.println(mq135\_1.getRo());

Serial.print("Ro2 = ");

Serial.println(mq135\_2.getRo());

Serial.print("Ro3 = ");

Serial.println(mq135\_3.getRo());

Serial.print("Ro4 = ");

Serial.println(mq135\_4.getRo());

}

void loop()

{

DateTime now = rtc.now();

Serial.print(now.year(), DEC);

Serial.print('/');

Serial.print(now.month(), DEC);

Serial.print('/');

Serial.print(now.day(), DEC);

Serial.print('\_');

Serial.print('\_');

Serial.print('\_');

Serial.print('\_');

Serial.print(now.hour(), DEC);

Serial.print(':');

Serial.print(now.minute(), DEC);

Serial.print(':');

Serial.print(now.second(), DEC);

Serial.println();

// выводим отношения текущего сопротивление датчика

// к сопротивлению датчика в чистом воздухе (Rs/Ro)

Serial.print("Ratio\_1: ");

Serial.print(mq135\_1.readRatio());

// выводим значения газов в ppm

Serial.print("\tCO2\_1: ");

Serial.print(mq135\_1.readCO2());

Serial.println(" ppm");

delay(100);

Serial.print("Ratio\_2: ");

Serial.print(mq135\_2.readRatio());

// выводим значения газов в ppm

Serial.print("\tCO2\_2: ");

Serial.print(mq135\_2.readCO2());

Serial.println(" ppm");

Serial.print("Ratio\_3: ");

Serial.print(mq135\_3.readRatio());

// выводим значения газов в ppm

Serial.print("\tCO2\_3: ");

Serial.print(mq135\_3.readCO2());

Serial.println(" ppm");

Serial.print("Ratio\_4: ");

Serial.print(mq135\_4.readRatio());

// выводим значения газов в ppm

Serial.print("\tCO2\_4: ");

Serial.print(mq135\_4.readCO2());

Serial.println(" ppm");

//делей, чтобы не выбивать слишком часто значения co2

delay(1000);

spr= ((mq135\_2.readCO2()+mq135\_1.readCO2()+mq135\_3.readCO2()+mq135\_4.readCO2())/4); //среднее

Serial.println(spr);

//проверка состояния со2, с учетом времени

if (now.hour()>6 && now.hour()<20) {

if (spr < 1000) {

digitalWrite(PIN\_RELAY, LOW); // Включаем реле - посылаем низкий уровень сигнала

delay(5000);

digitalWrite(PIN\_RELAY, HIGH); // Отключаем реле - посылаем высокий уровень сигнала

delay(1000);

}

}

}