ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное учреждение Высшего образования

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

**Кафедра математического и программного обеспечения**

**информационных систем**

**УМНЫЙ ДОМ**

Проект

по дисциплине «Программирование встраиваемых систем»

БЕЛГОРОД 2022

**Задача:** разработать систему “умный дом”, на базе платформы Arduino UNO

**Функции “умного дома”:**

* Автоматическое включение света при приближении к дому
* Измерение уровня температуры в комнате
* Измерение уровня газа в воздухе
* Отправка по смс команд системе

**Актуальность темы:** Система «Умный дом» на Arduino пользуется большим спросом у людей, которые стремятся создать максимальный комфорт в доме и офисе.

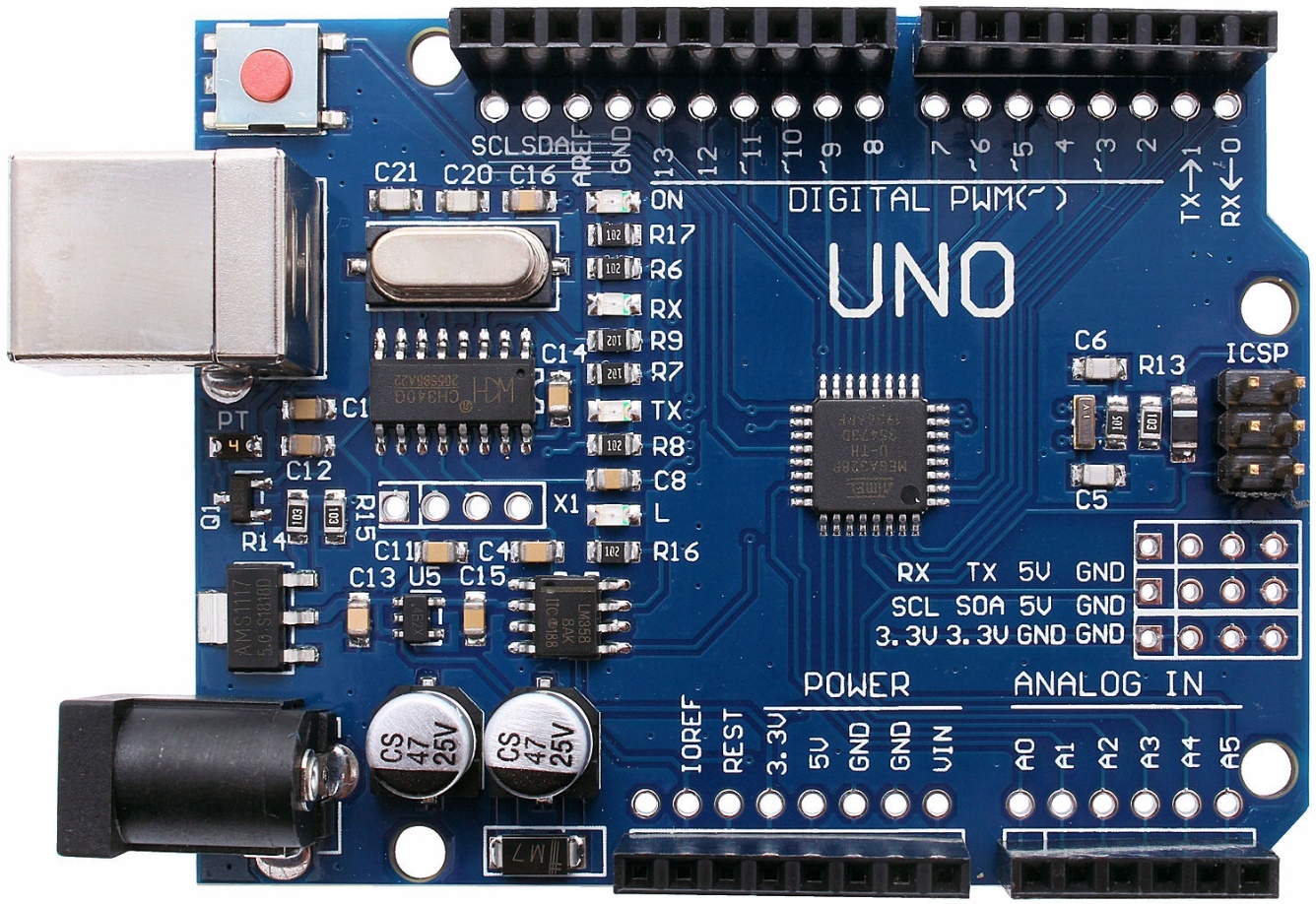
Ее особенность — в способности управлять различными системами без участия владельца, а суть заключается в объединении электронных устройств в одну сеть для экономии электроэнергии, управления освещением и электроприборами, оповещения о проникновении в дом посторонних лиц и решении других задач.

**Необходимые компоненты:**

* Arduino UNO
* GSM/GPRS модуль SIM900 Shield
* PIR-датчик HC-SR501
* Датчик влажности и температуры DHT11
* LCD-дисплей
* Датчика газа MQ2
* Соединительные провода

# АНАЛИЗ КОМПОНЕНТОВ

***Arduino UNO***



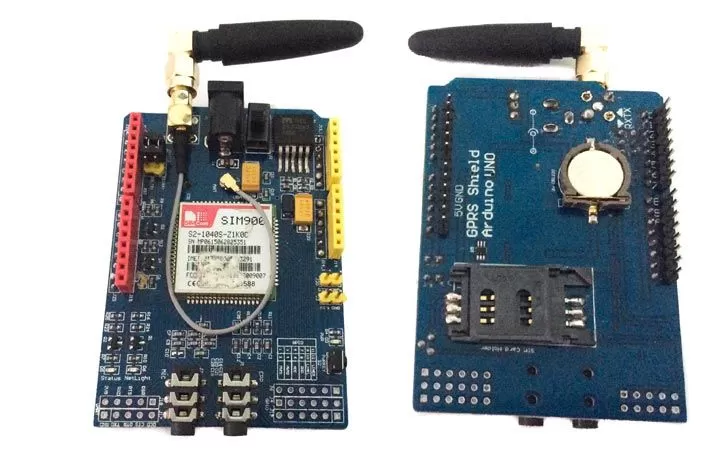
Arduino Uno – плата от компании Arduino, построенная на микроконтроллере ATmega 328.

Плата имеет на борту 6 аналоговых входов, 14 цифровых выводов общего назначения (могут являться как входами, так и выходами), кварцевый генератор на 16 МГц, два разъема: силовой и USB, разъем ISCP для внутрисхемного программирования и кнопку горячей перезагрузки устройства. Для стабильной работы плату необходимо подключить к питанию либо через встроенный USB Разъем, либо подключив разъем питания к источнику от 7 до 12В. Через переходник питания плата также может работать и от батареи формата Крона.

Основное отличие платы от предыдущих – для взаимодействия по USB Arduino Uno использует отдельный микроконтроллер ATmega8U2. Прошлые версии Arduino использовали для этого микросхему программатора FTDI.

Несложно догадаться, что благодаря своему итальянскому происхождению, слова “Arduino” и “Uno” взяты именно из этого языка. Компания назвалась “Arduino” в честь короля Италии 11 века Ардуина, а Уно переводится с итальянского как “первый”.

**GSM/GPRS модуль SIM900 Shield**

****

Плата Arduino GPRS/GSM Shield предоставляет нам возможность использовать для удаленного приема и передачи данных мобильной GSM-связи.

Осуществить это можно тремя способами:

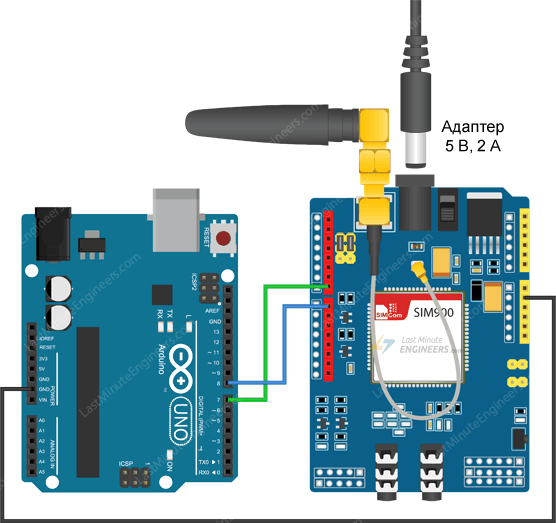
* Используя отправку/прием коротких текстовых сообщений (SMS);
* Отправкой голосовых (аудио) команд на основе технологий CSD (стандартная технология передачи данных в сети GSM) и/или DTMF (двухтональный многочастотный аналоговый сигнал, используемый для набора телефонного номера);
* Используя пакетную передачу данных на основе технологии GPRS.

Также на ней расположены:

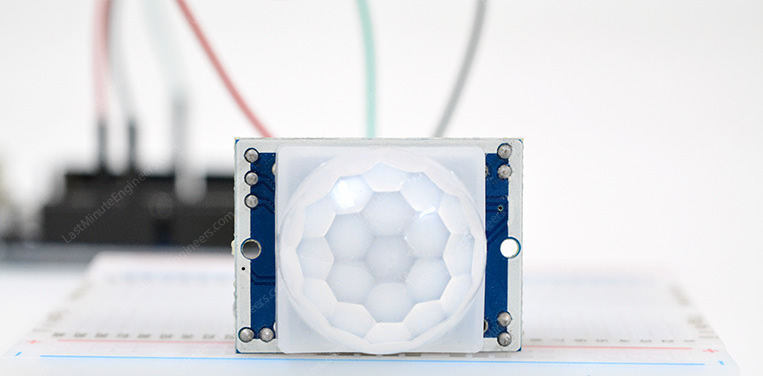
* слот для SIM-карты;
* джек 3,5 мм для аудио-входа и выхода;
* разъём для внешней антенны.

Общение с платой производится через serial-соединение с помощью набора AT-команд. С помощью перемычек на плате возможно установить используемые для коммуникации контакты: аппаратные 0- 1-й или 2-3 (на некоторых платах) 7- 8-й для работы через SoftwareSerial.

**Схема подключения SIM900 Shield к Arduino**



**PIR-датчик HC-SR501**

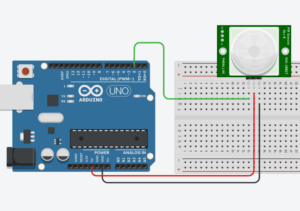


Конструкция ПИР датчика движения не очень сложна – он состоит из пироэлектрического элемента, отличающегося высокой чувствительностью (деталь цилиндрической формы, в центре которой расположен кристалл) к наличию в зоне действия определенного уровня инфракрасного излучения. Чем выше температура объекта, тем больше излучение. Сверху PIR-датчика устанавливается полусфера, разделенная на несколько участков (линз), каждый из которых обеспечивает фокусировку излучения тепловой энергии на различные сегменты датчика движения. Чаще всего в качестве линзы применяют линзу Френеля, которая за счет концентрации теплового излучения позволяет расширить диапазон чувствительности инфракрасного датчика движения Ардуино.

PIR-sensor конструктивно разделен на две половины. Это обусловлено тем, что для устройства сигнализации важно именно наличие движения в зоне чувствительности, а не сам уровень излучения. Поэтому части установлены таким способом, что при улавливании одной большего уровня излучения, на выход будет подаваться сигнал со значением high или low.



**Схема подключения PIR-датчика к Arduino**



**Пример кода**

#define PIN\_PIR 2

#define PIN\_LED 13

void setup() {

Serial.begin(9600);

pinMode(PIN\_PIR, INPUT);

pinMode(PIN\_LED, OUTPUT);

}

void loop() {

int pirVal = digitalRead(PIN\_PIR);

Serial.println(digitalRead(PIN\_PIR));

//Если обнаружили движение

if (pirVal)

{

digitalWrite(PIN\_LED, HIGH);

Serial.println("Motion detected");

delay(2000);

}

else

{

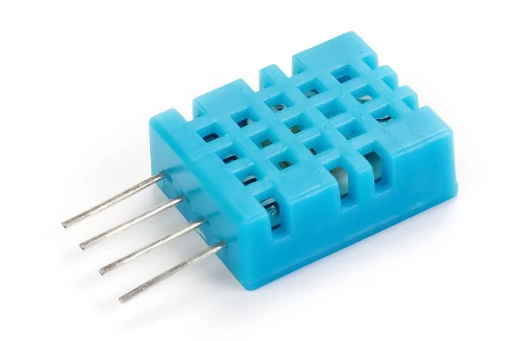
//Serial.print("No motion");

digitalWrite(PIN\_LED, LOW);

}

}

**Датчик температуры и влажности DHT11**

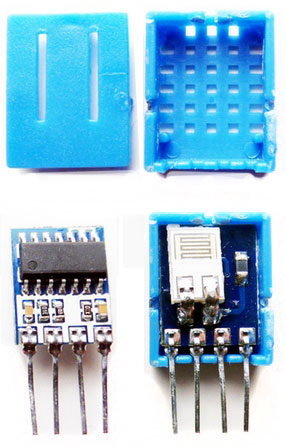
****

Существует семейство датчиков влажности и температуры DHT (DHT11, DHT22 и др.), которые используются для построения домашней метеостанции. Одновременно измерять температуру и относительную влажность выглядит оправданным, поскольку второе напрямую зависит от первого. Так, повышение температуры батарей центрального отопления приводит к уменьшению относительной влажности воздуха.

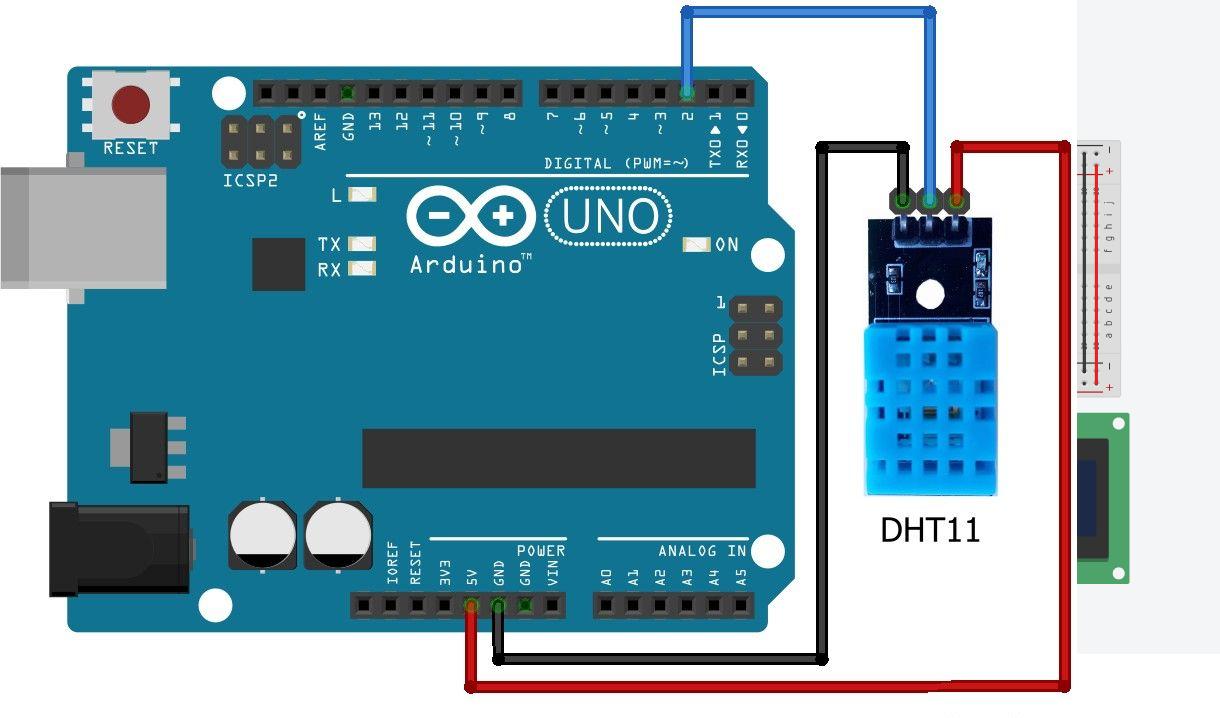
Считается, что оптимальное значение влажности около 50% — именно при такой влажности растения, люди и животные чувствуют себя комфортно.

Датчик температуры и влажности **DHT11** является дешёвым датчиком, который удобно использовать в домашних условиях в учебных целях. Он не обеспечивает точные показания и имеет ограниченный диапазон измерений. Но тем не менее, он очень популярен из-за своей простоты. Существуют более дорогие аналоги, в частности его старший брат **DHT22**.

Датчики DHT состоят из двух основных частей: ёмкостный датчик влажности и термистор. Также в корпусе установлен простенький чип для преобразования аналогового сигнала в цифровой. Считывать цифровой сигнал на выходе достаточно просто, можно использовать любой контроллер, не только Arduino.



**Схема подключения DHT11 к Arduino**

****

**Пример кода**

#include "DHT.h"

#define DHTPIN 2

DHT dht(DHTPIN, DHT11);

void setup() {

Serial.begin(9600);

dht.begin();

}

void loop() {

delay(2000); // 2 секунды задержки

float h = dht.readHumidity(); //Измеряем влажность

float t = dht.readTemperature(); //Измеряем температуру

if (isnan(h) || isnan(t)) { // Проверка. Если не удается считать показания, выводится «Ошибка считывания», и программа завершает работу

Serial.println("Ошибка считывания");

return;

}

Serial.print("Влажность: ");

Serial.print(h);

Serial.print(" %\t");

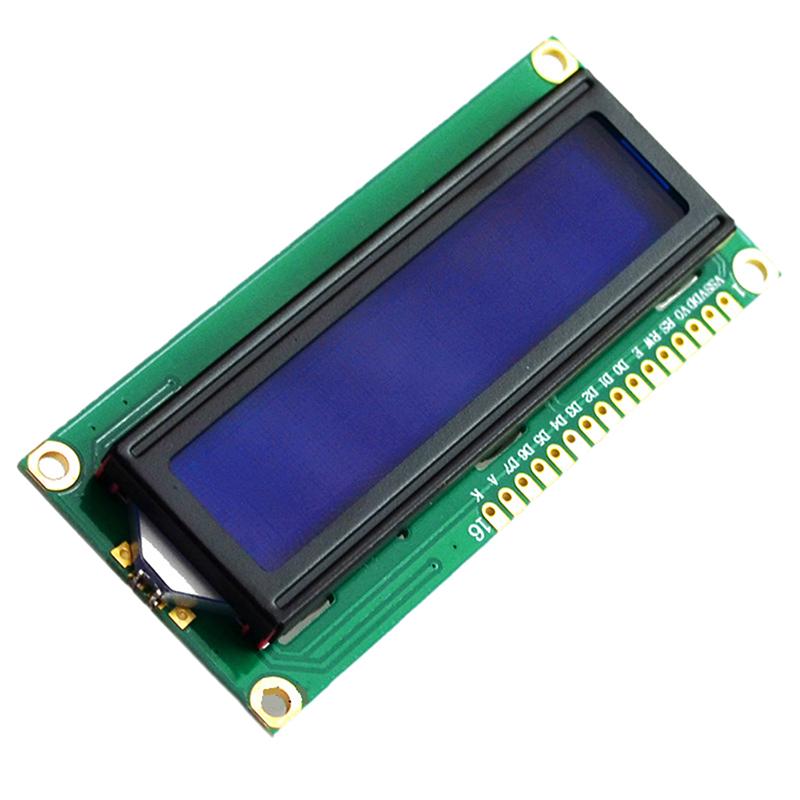
Serial.print("Температура: ");

Serial.print(t);

Serial.println(" \*C "); //Вывод показателей на экран

}

**LCD-дисплей 1602A**



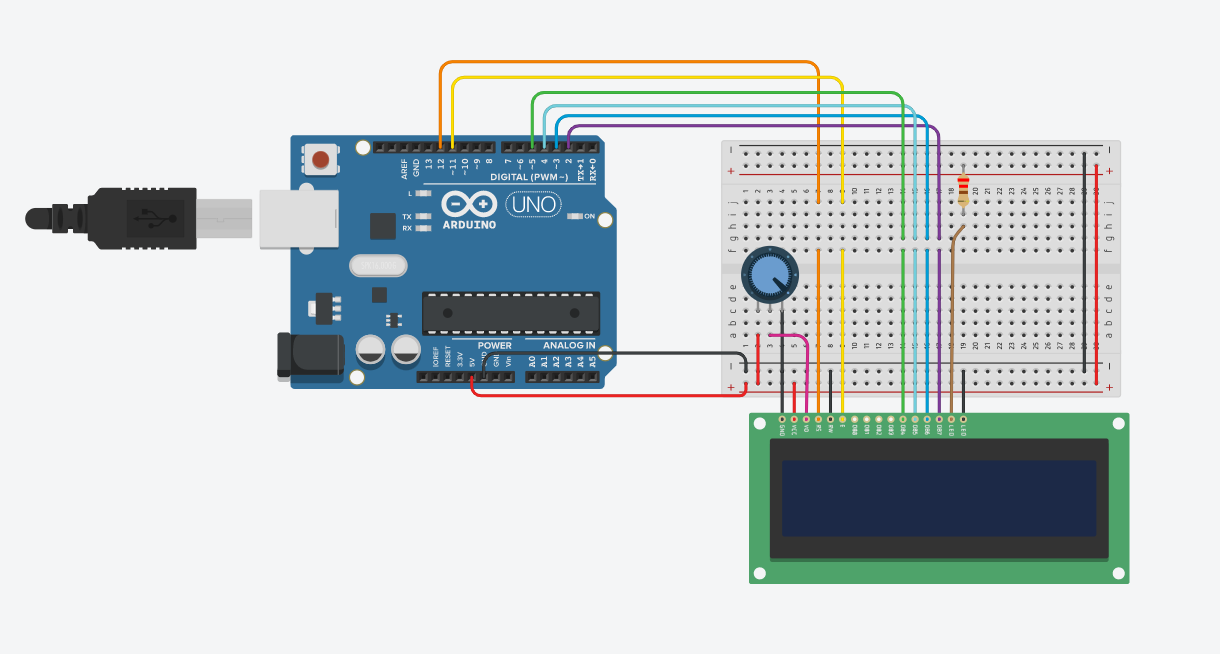
Жидкокристаллический дисплей (Liquid Crystal Display) сокращенно LCD построен на технологии жидких кристаллов. При проектировании электронные устройства, нам нужно недорогое устройство для отображения информации и второй не менее важный фактор наличии готовых библиотек для Arduino. Из всех доступных LCD дисплеев на рынке, наиболее часто используемой является LCD 1602A, который может отображать ASCII символа в 2 строки (16 знаков в 1 строке) каждый символ в виде матрицы 5х7 пикселей.

LCD 1602A представляет собой электронный модуль основанный на драйвере HD44780 от Hitachi. LCD1602 имеет 16 контактов и может работать в 4-битном режиме (с использованием только 4 линии данных) или 8-битном режиме (с использованием всех 8 строк данных), так же можно использовать интерфейс I2C.

На задней части модуля расположено два чипа в «капельном» исполнении (ST7066U и ST7065S) и электрическая обвязка, рисовать принципиальную схему не вижу смысла, только расскажу о резисторе R8 (100 Ом), который служит ограничительным резистором для светодиодной подсветки, так что можно подключить 5В напрямую к контакту A. Немного попозже напишу статью в которой расскажу как можно менять подсветку LCD дисплея с помощью ШИП и транзистора.



**Схема подключения LCD-дисплея к Arduino**



**Пример кода**

#include <LiquidCrystal.h>

int seconds = 0;

LiquidCrystal lcd\_1(12, 11, 5, 4, 3, 2);

void setup()

{

lcd\_1.begin(16, 2); // Set up the number of columns and rows on the LCD.

// Print a message to the LCD.

lcd\_1.print("hello world!");

}

void loop()

{

// set the cursor to column 0, line 1

// (note: line 1 is the second row, since counting

// begins with 0):

lcd\_1.setCursor(0, 1);

// print the number of seconds since reset:

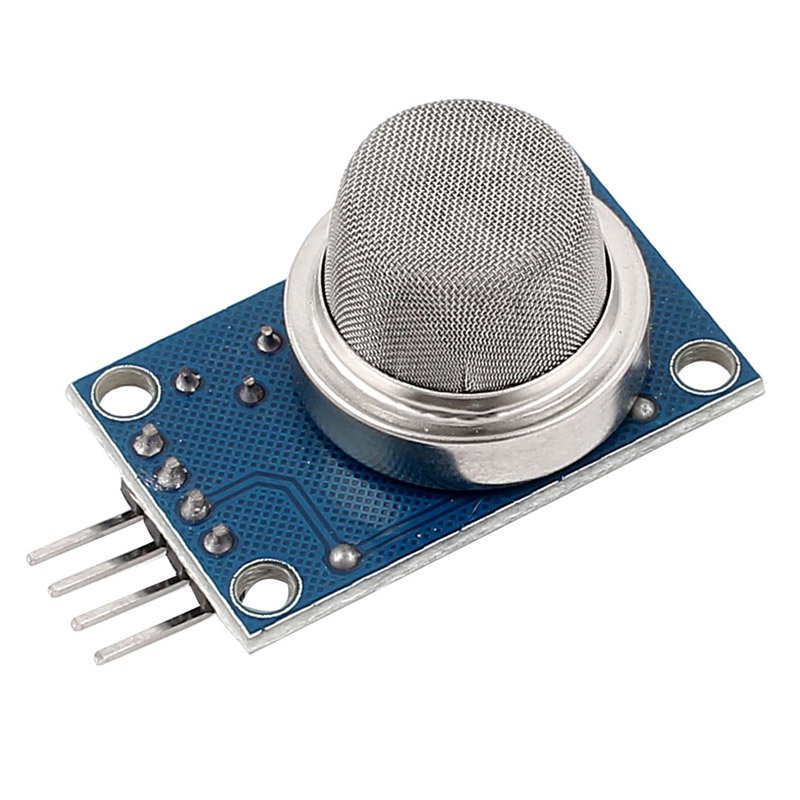
lcd\_1.print(seconds);

delay(1000); // Wait for 1000 millisecond(s)

seconds += 1;

}

**Датчика газа MQ2**

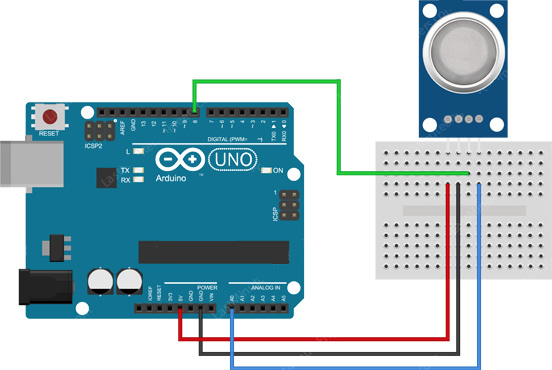


Датчик MQ-2 Ардуино позволяет выявлять в воздухе минимальную концентрацию водорода и углеводородных газов (пропан, метан, бутан). Применяют сенсоры MQ-2 в проектах умного дома для своевременного обнаружения газа или дыма. Сенсор относится к семейству датчиков MQ, которые отличаются низкой стоимостью, простотой использования и легкостью подключения к микроконтроллеру Ардуино.

Принцип сенсора основан на детекторе, изготовленного из сплава оксида олова и алюминия, который в процессе работы сенсора существенно нагревается. В результате химической реакции, происходящей при попадании молекул углеводородных газов на чувствительный элемент, изменяется сопротивление сенсора. Измеряя изменения сопротивления, можно узнать точное значение концентрации газа в воздухе.

При измерении газов, термин «концентрация» используется для описания количества газа в воздухе по объему. Наиболее распространенными единицами измерения являются доли на миллион и процентная концентрация. Доли на миллион (ppm) — это отношение одного газа к другому.

**Схема подключения MQ-2 к Arduino**



**Пример кода**

#define MQ2pin (0)

float sensorValue; // переменная для хранения значения датчика

void setup()

{

Serial.begin(9600); // настроить последовательный порт на скорость 9600

Serial.println("Gas sensor warming up!");

delay(20000); // дать MQ-2 время для прогрева

}

void loop()

{

sensorValue = analogRead(MQ2pin); // прочитать аналоговый вход 0

Serial.print("Sensor Value: ");

Serial.print(sensorValue);

if(sensorValue > 300)

{

Serial.print(" | Smoke detected!");

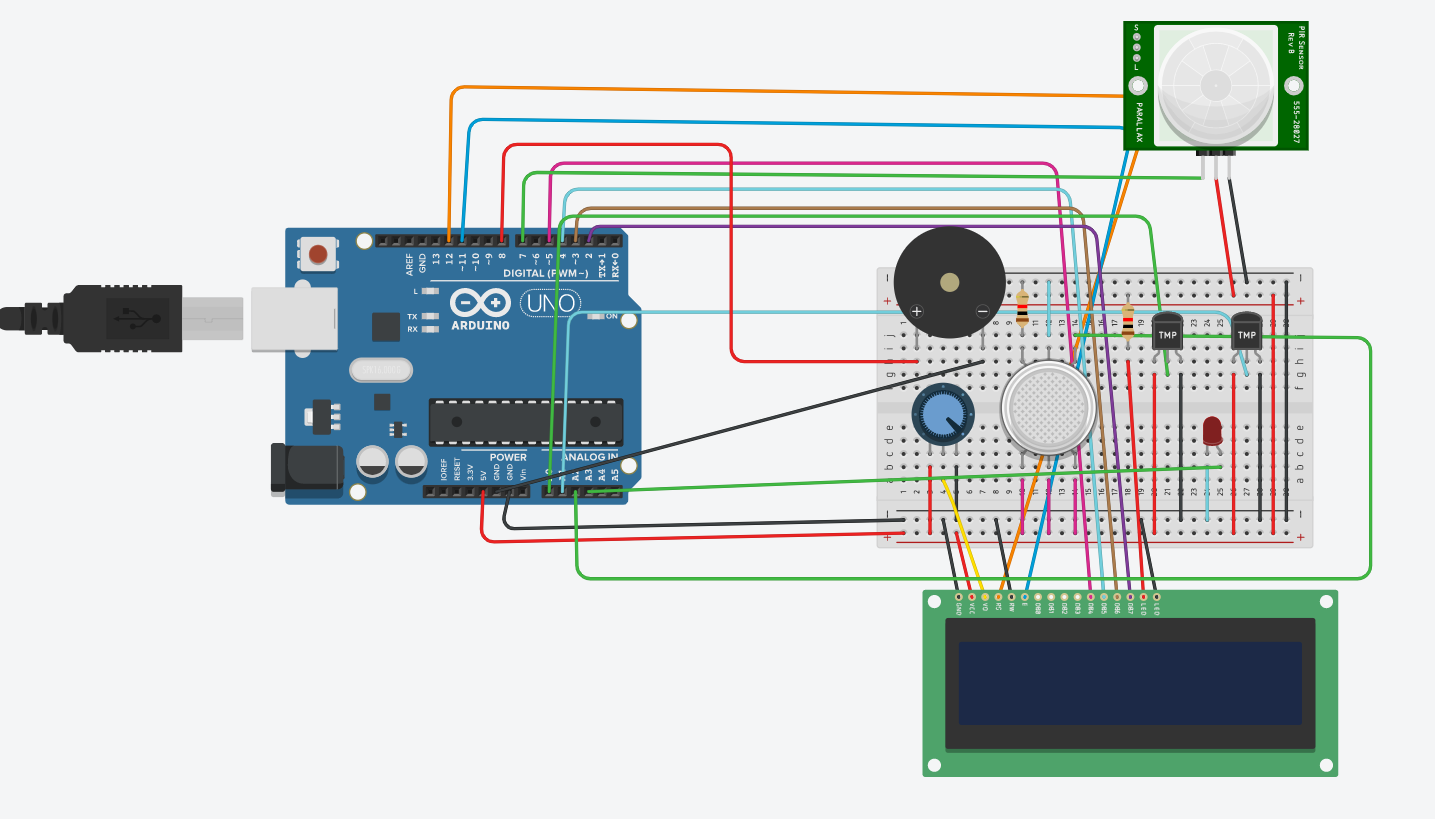
}

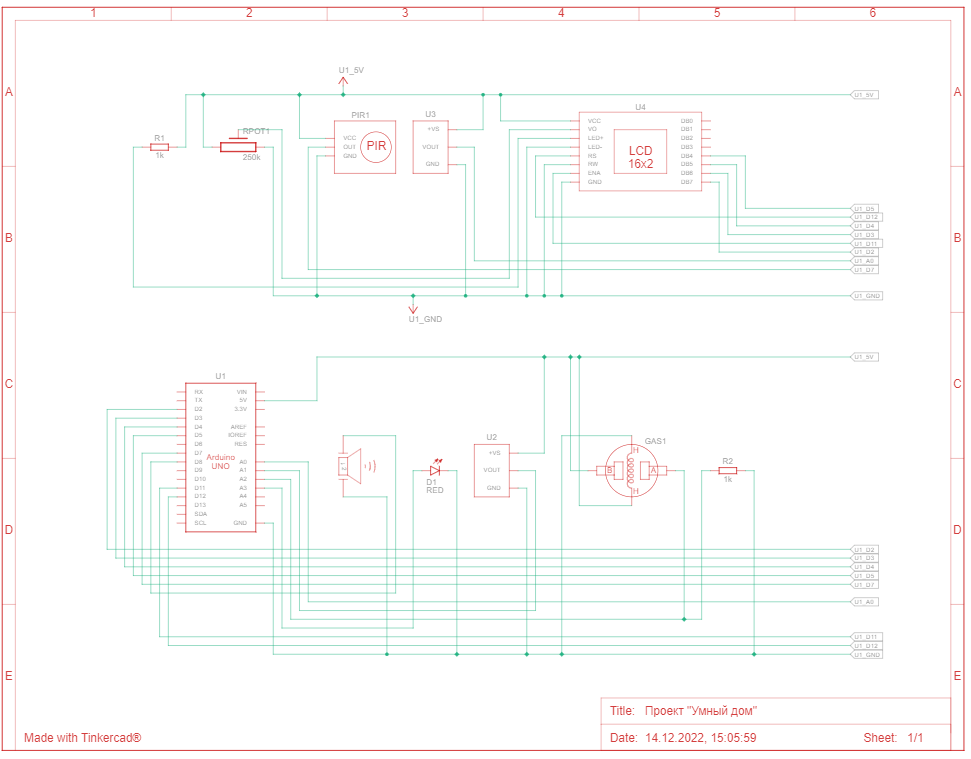
Serial.println("");

delay(200)

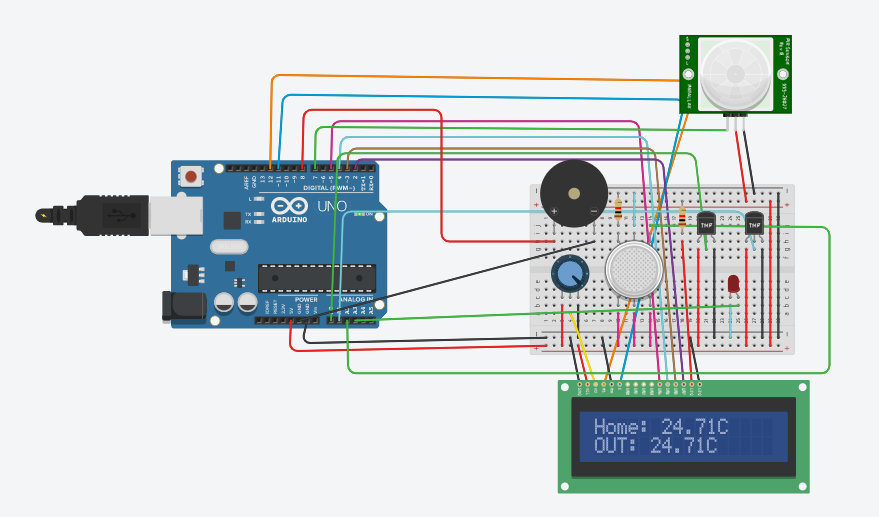
**Работа готового проекта**

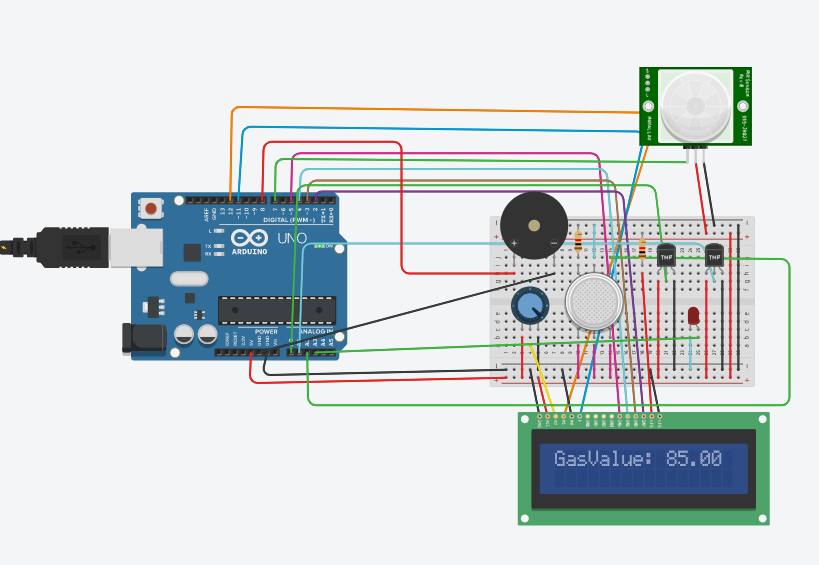
Данный проект представляет собой систему “Умный дом”, которая определяет температуру воздуха в помещении и на улице, может предупредить об опасном кол-ве газа в помещении, а также распознаёт движение с помощью PIR-датчика.

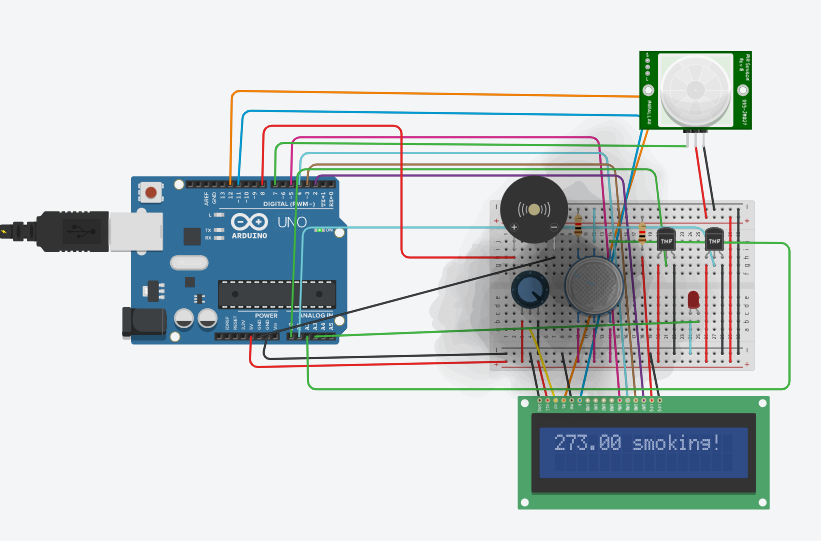
*Схема проекта*

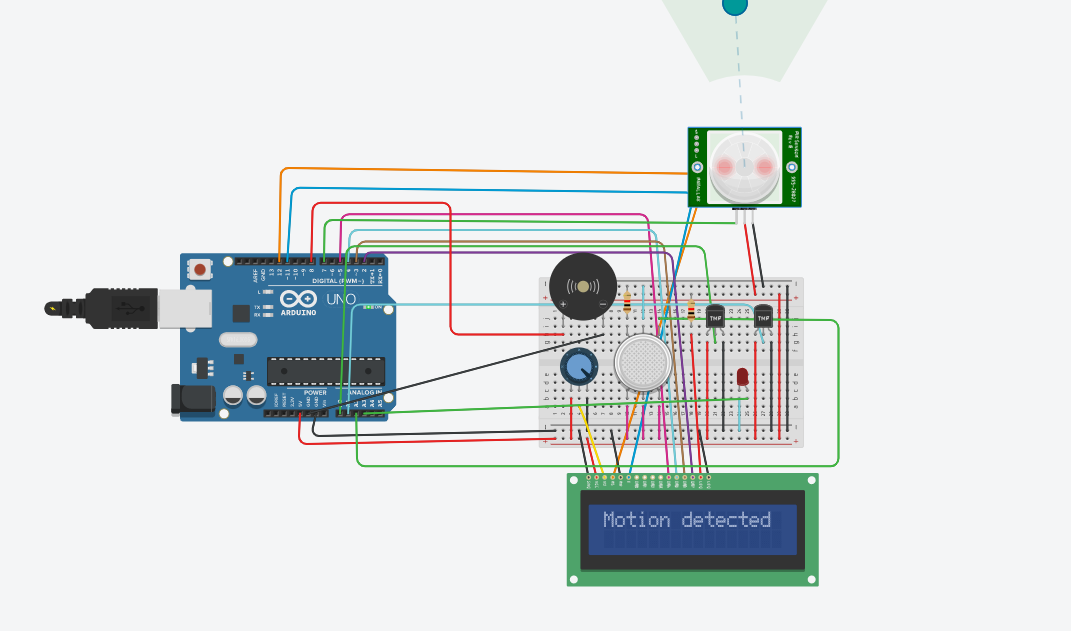


**Результат работы программы**









**ЛИСТИНГ ПРОГРАММЫ**

#include <LiquidCrystal.h>

#define Home A1

#define OUT A0

#define PIN\_PIR 7

#define PIN\_MQ2 A2

#define LED A3

#define pinSound 8

LiquidCrystal lcd(12,11,5,4,3,2);

void setup()

{

pinMode(Home,INPUT);

pinMode(OUT,INPUT);

lcd.begin(16,2);

pinMode(PIN\_PIR, INPUT);

pinMode(PIN\_MQ2, INPUT);

Serial.begin(9600);

}

void loop()

{

float Tin = temp(Home);

float Tout = temp(OUT);

float gasValue = analogRead(PIN\_MQ2);

int pirVal = digitalRead(PIN\_PIR);

if (gasValue < 200 && !pirVal)

{

lcd.setCursor(0,0);

lcd.print("Home: ");

lcd.print(Tin);

lcd.print("C");

lcd.setCursor(0,1);

lcd.print("OUT: ");

lcd.print(Tout);

lcd.print("C");

delay(1500);

lcd.clear();

}

//Если обнаружили движение

if (pirVal && gasValue < 200)

{

lcd.print("Motion detected");

delay(250);

lcd.clear();

tone(pinSound, 200);

}

else if (gasValue < 200 && pirVal)

{

noTone(pinSound);

}

if(gasValue >= 200 && !pirVal)

{

lcd.print(gasValue);

lcd.print(" smoking!");

digitalWrite(LED, LOW);

delay(500);

digitalWrite(LED,HIGH);

delay(500);

lcd.clear();

tone(pinSound, 200);

delay(500);

}

else if(!pirVal)

{

digitalWrite(LED,LOW);

lcd.print("GasValue: ");

lcd.print(gasValue);

delay(1500);

lcd.clear();

noTone(pinSound);

}

}

float temp(int pin)

{

int value = analogRead(pin);

float voltage = value \* 5.0;

voltage = voltage /= 1024;

return (voltage-0.5)\*100;

}