25.11.2023

Простой детектор загазованности помещения на Arduino UNO и MQ-7

Оглавление

1 Функционал 2
2 Составные части (компоненты) устройства 2
3 Схема Устройства 5
4 Контейнер (коробка) для размещения составных частей Устройства 7
5 Элементы интерактивного управления Устройством 9
6 Порядок работы с Устройством 10
6.1 Начало работы 10
6.2 Выбор режима работы Устройства 11
6.2.1 Режим настройки (установка верхней границы допустимых значений датчика MQ7) 11
6.2.2 Режим измерений 11
7 Пояснение к схеме и скетчу 12
7.1 Схема устройства 12
7.2 Скетч − основные нюансы 12
8 Скетч 13

1 Функционал

Простой детектор загазованности помещения на Arduino UNO и MQ-7 (далее – Устройство) «умеет»:

1. Дискретно (16 значений) регулировать верхнюю границу допустимых значений датчика MQ7 в интерактивном режиме (с помощью микропереключателей);
2. Визуализировать заданную верхнюю границу допустимых значений датчика MQ7 (на блоке 7-ми сегментных индикаторов);
3. Периодически (с заданным в скетче интервалом времени) считывать значения датчика MQ7;
4. Визуализировать считанные значения датчика MQ7 (на блоке 7-ми сегментных индикаторов);
5. Определять превышение значений датчика MQ7 относительно заданной границы допустимых значений;
6. Выдавать звуковой (зуммер) и световой (красный светодиод) сигнал в случае, если значения датчика MQ7 превысили заданную границу;
7. Питание Устройства – один (любой) из вариантов:
   7.1 Батарейка «Крона» (9В);
   7.2 Стандартная телефонная USB-зарядка (5В);
   7.3 Соответствующий аккумулятор.

2 Составные части (компоненты) устройства

В устройстве используются следующие составные части (компоненты):

1. Arduino UNO (1 шт);
2. MQ7 − датчик угарного газа (1 шт);
3. 3642BH − блок 7-ми сегментных индикаторов (1 шт);
4. Зуммер/buzzer (1 шт);
5. SS12F15VG6 − микропереключатели/микротумблеры (5 шт) ;
6. Красный светодиод (1 шт);
7. Зеленый светодиод (1 шт);
8. Соединительные провода.

На рисунках ниже представлены фотографии основных составных частей Устройства (кроме, Arduino UNO и светодиодов).

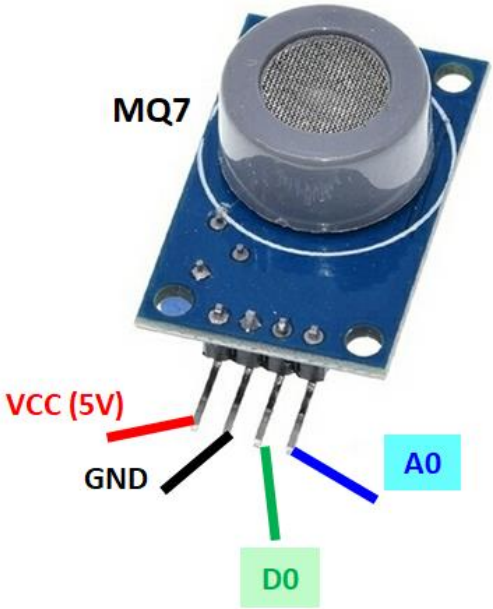

Рисунок 1 − Датчик угарного газа MQ7

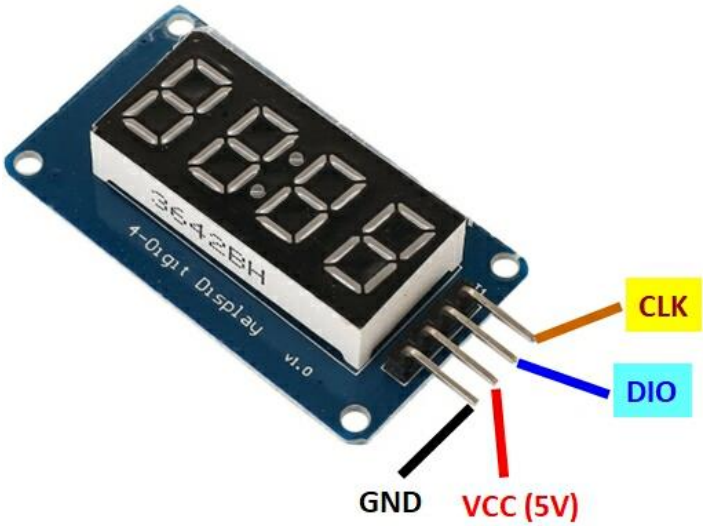

Рисунок 2 – Контакты блока 7-ми сегментных индикаторов 3642BH

Рисунок 3 – Микропереключатель SS12F15VG6

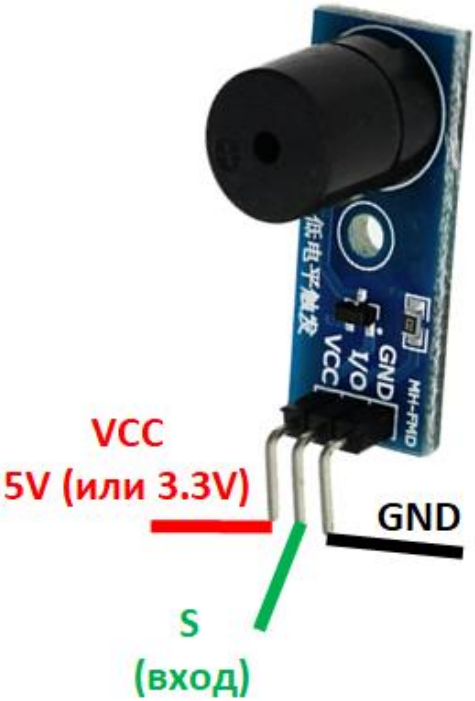

Рисунок 4 – Зуммер (buzzer)

3 Схема Устройства

Рисунок, иллюстрирующий схему подключения составных частей Устройства, приведен ниже (см., также, рисунок 8).

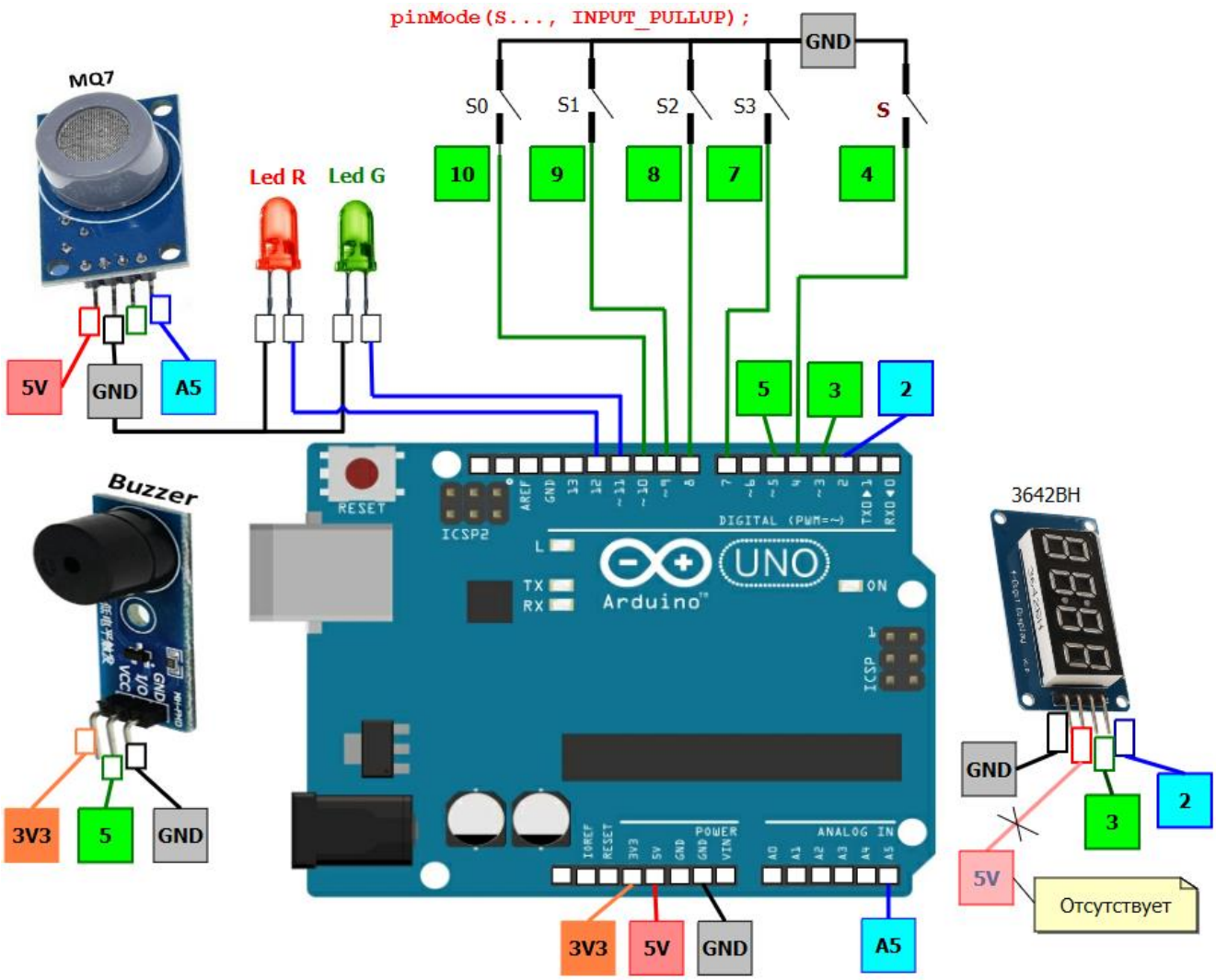

Рисунок 5 – Иллюстрация схемы подключения составных частей Устройства

В таблице ниже продублированы соединительные контакты составных частей Устройства.

Таблица 1 – Соединительные контакты составных частей Устройства

Составная часть		Куда	Примечание
Наименование	Контакт		
1	2	3	4
MQ7	VCC	Arduino, 5V	
	GND	Arduino, GND	
	D0	–	Не используется
	A0	Arduino, A5	
3642BH	GND	Arduino, GND	
	VCC	–	Не используется (и без него работает)
	DIO	Arduino, 3	

Составная часть		Куда	Примечание
Наименование	Контакт		
1	2	3	4
	CLK	Arduino, 2	
Зуммер	VCC	Arduino, 3V3	
	S	Arduino, 5	
	GND	Arduino, GND	
LED R	Катод	Arduino, GND	Красный светодиод
	Анод	Arduino, 12	
LED G	Катод	Arduino, GND	Зеленый светодиод
	Анод	Arduino, 11	
S (SS12F15VG6)	Контакт-1	Arduino, GND	Переключатель режимов работы схемы (регулировка верхней границы допустимых значений датчика MQ7)
	Контакт-2	Arduino, 4	
S0 (SS12F15VG6)	Контакт-1	Arduino, GND	0-й разряд значения верхней границы допустимых значений датчика MQ7 (в двоичном исчислении)
	Контакт-2	Arduino, 10	
S1 (SS12F15VG6)	Контакт-1	Arduino, GND	1-й разряд значения верхней границы допустимых значений датчика MQ7 (в двоичном исчислении)
	Контакт-2	Arduino, 9	
S2 (SS12F15VG6)	Контакт-1	Arduino, GND	2-й разряд значения верхней границы допустимых значений датчика MQ7 (в двоичном исчислении)
	Контакт-2	Arduino, 8	
S3 (SS12F15VG6)	Контакт-1	Arduino, GND	3-й разряд значения верхней границы допустимых значений датчика MQ7 (в двоичном исчислении)
	Контакт-2	Arduino, 7	

4 Контейнер (коробка) для размещения составных частей Устройства

В качестве коробки – вполне подошел вот такой «пищевой» контейнер, купленный в магазине по цене – что-то в пределах 50-70 рублей (см. рисунок ниже).

Рисунок 6 − Контейнер для размещения составных частей Устройства

«Изяществом форм и оригинальностью сборки» − это не блещет, но и не для ВДНХ оно предназначено… ☺

Но зато «контейнер»:

- достаточно прочен и пластичен (в том числе и не «колется» при «обработке» шилом, сверлом, надфилем и т.д.);
- имеет удобную крышку (надежно закрывается и легко открывается);
- по габаритам вполне подходит.

На рисунке ниже представлены габаритные размеры контейнера.

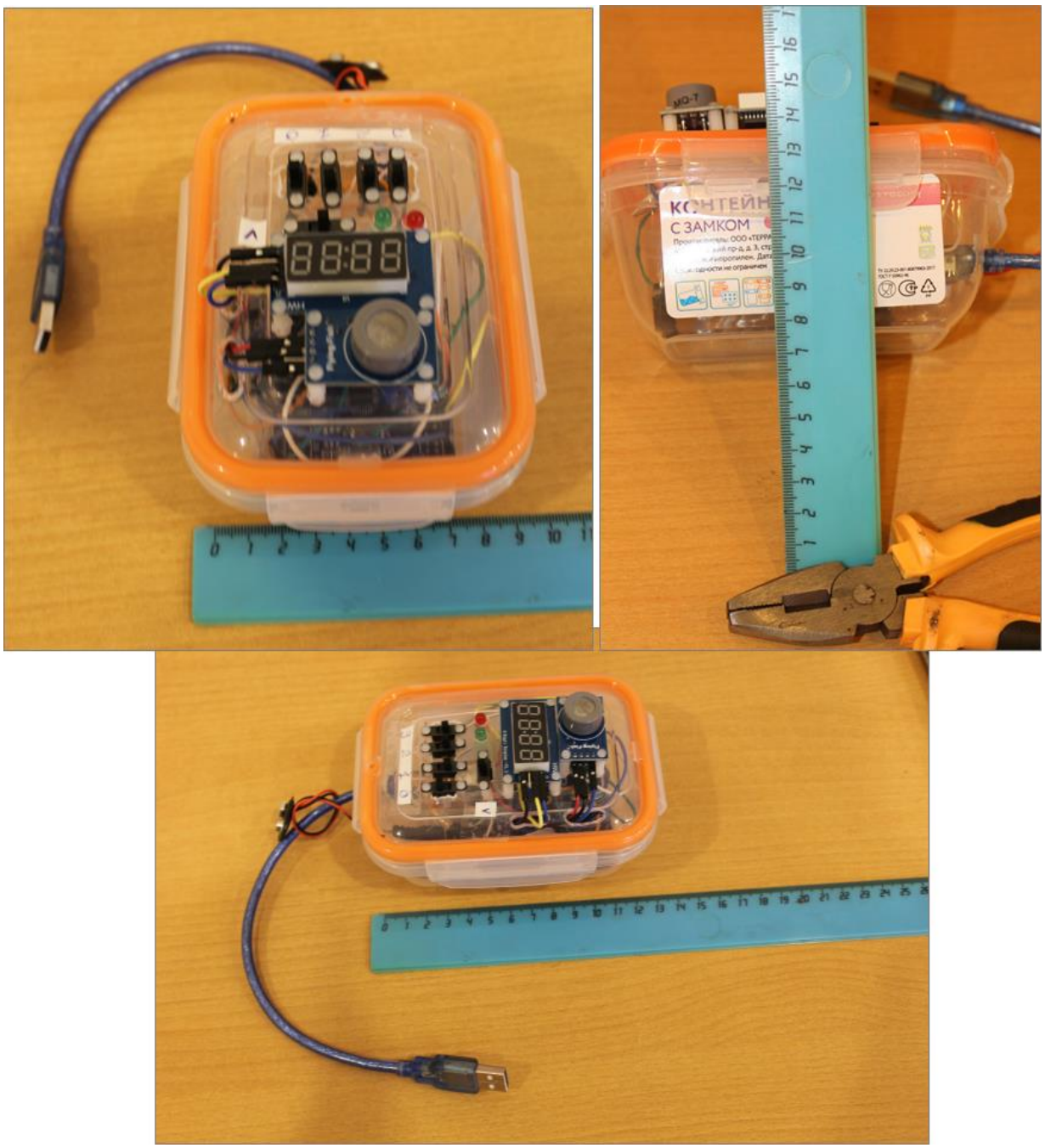

Рисунок 7 – Габариты «контейнера»

5 Элементы интерактивного управления Устройством

Элементы интерактивного управления Устройством приведены на рисунке ниже.

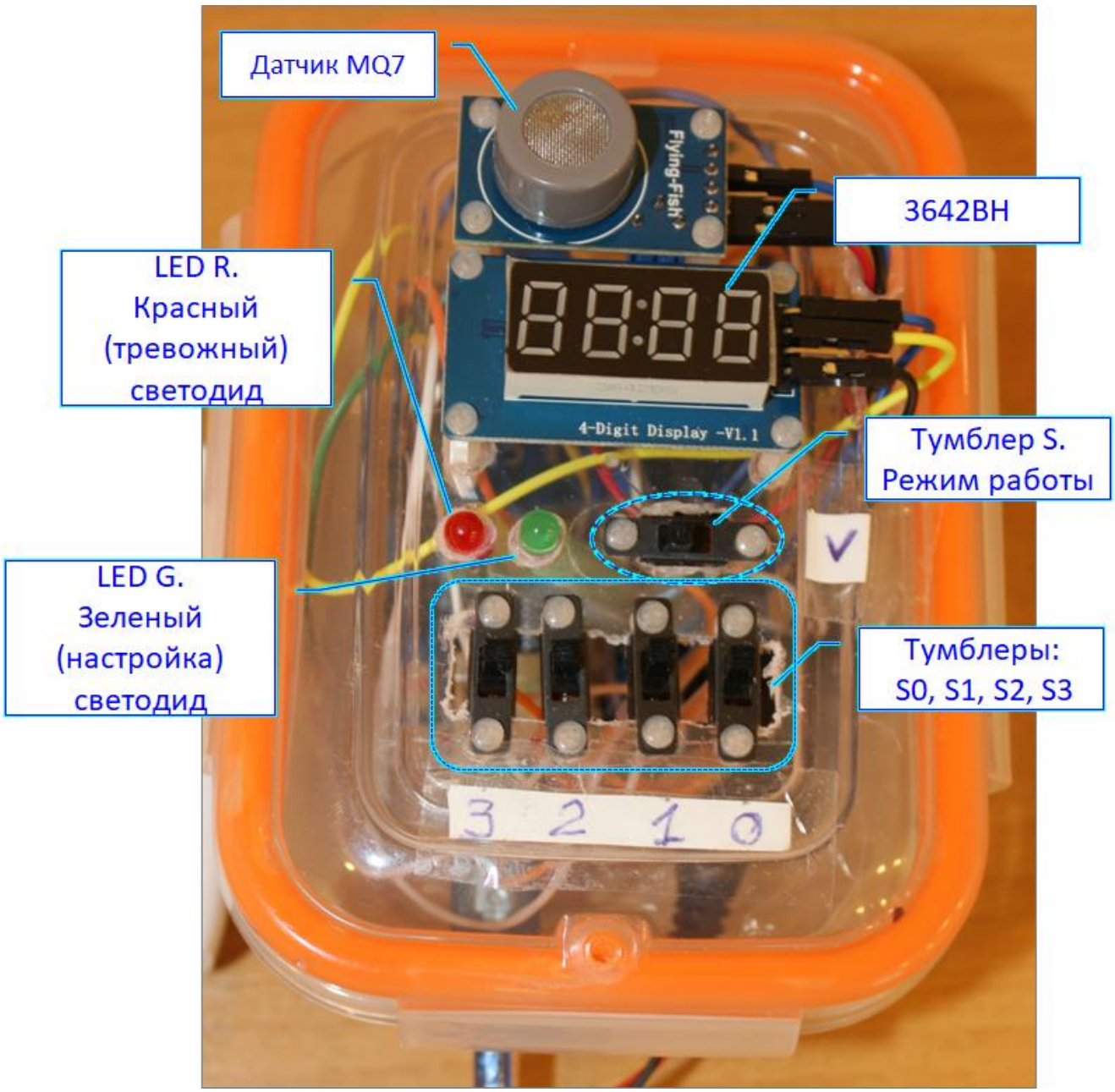

Рисунок 8 − Элементы интерактивного управления Устройством

См., также, рисунок 5.

6 Порядок работы с Устройством

Основные операция приведены ниже.

6.1 Начало работы

Для начала работы устройства необходимо подать напряжение питания.

В качестве источника питания Устройства может быть выбран один (любой) из вариантов (см., также, рисунок ниже):

1.1 Батарейка «Крона» (9В);

1.2 Стандартная телефонная USB-зарядка (5В);

1.3 Соответствующий аккумулятор.

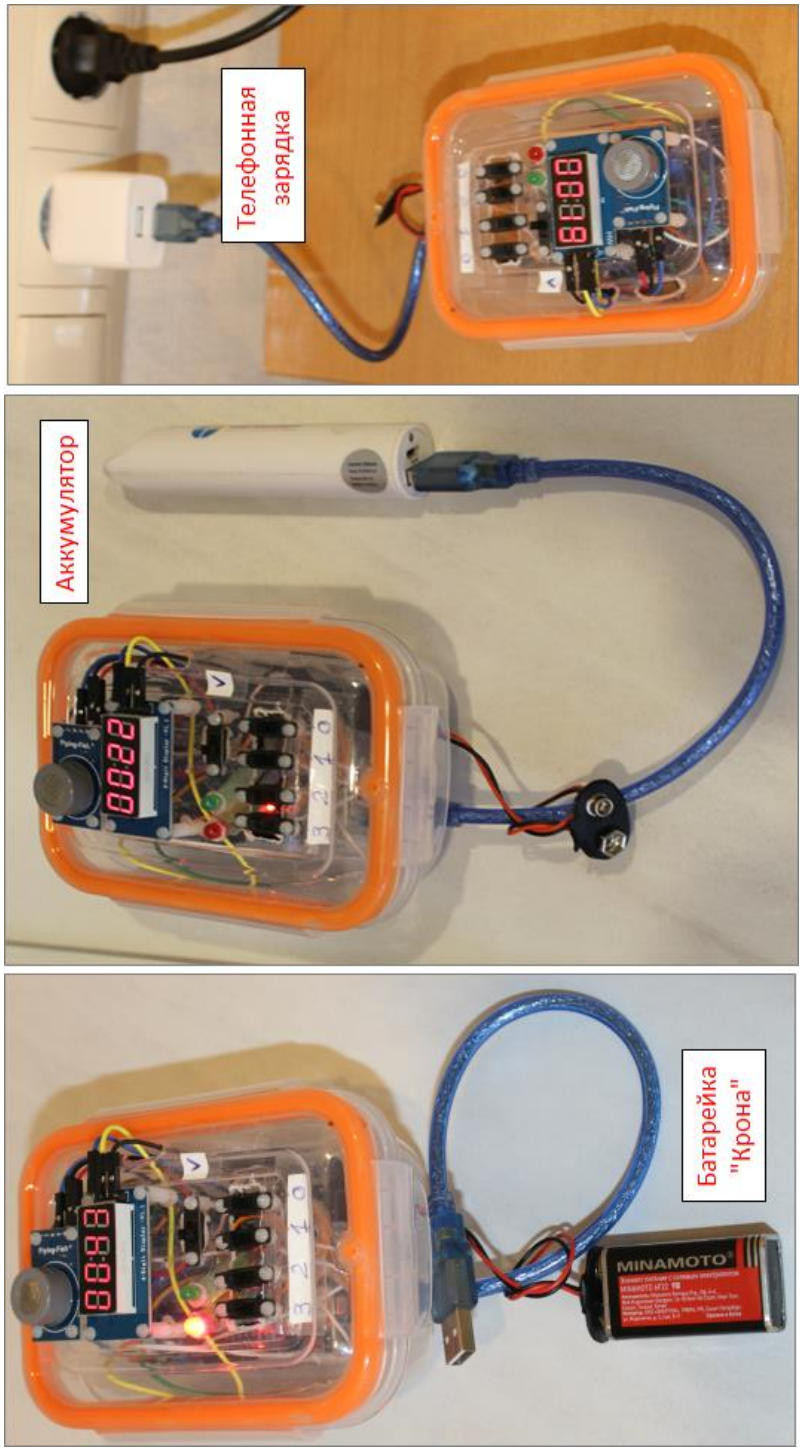

Рисунок – Возможные варианты подключения питания к Устройству

6.2 Выбор режима работы Устройства

Для выбора режима работы Устройства используется тумблер S (см. рисунок 8 и рисунок 5).

Изменяя его состояние (переключая из одного положения в другое) можно выбрать соответствующий режим.

6.2.1 Режим настройки (установка верхней границы допустимых значений датчика MQ7)

В режиме настройки горит LED G (зеленый светодиод).

Этот режим предназначен для интерактивного ввода верхней границы допустимых значений датчика MQ7.

Переключая тумблеры S0, S1, S2, S3 (с учетом формул, приведенных в разделе «7.2 Скетч − основные нюансы»), необходимо установить соответствующее значение.

Значение визуализируется на дисплее 3642ВН.

ВАЖНО! Понятное дело, что «панацеи от всего и на все времена» не существует, поэтому, есть дополнительные настроечные параметры, которые жестко зашиты в скетч:

`v_crit_min` − минимально возможное значение границы (допустимых значений датчика MQ7);

`v_crit_delta` − множитель к настроечному значению границы (допустимых значений датчика MQ7);

`DelayBetweenCycle` − задержка между циклами опроса датчика MQ7 (миллисек);

`CountSecForWaitReady` − заданное кол-во СЕКУНД ожидания стабилизации показаний MQ7 при включении питания.

6.2.2 Режим измерений

В режиме измерений зеленый светодиод LED G – погашен.

А красный светодиод LED R загорается (а также, включается звуковой сигнал), если значения от датчика MQ7 превышают установленную границу допустимых значений.

На дисплее 3642ВН визуализируются значения датчика MQ7.

7 Пояснение к схеме и скетчу

7.1 Схема устройства

Схема устройства (см. рисунок 5) – достаточно проста (никаких «наворотов» там нет).

Единственное, что следует отметить: при подключении 3642ВН контакт «VCC» (см. рисунок 2) был сознательно проигнорирован (т.е., **не** был присоединен к контакту «5V» Arduino UNO). Как оказалось, 3642ВН и так работает вполне удовлетворительно.

7.2 Скетч – основные нюансы

Все пояснения включены текст скетча (см. раздел «8 Скетч»).

Единственное, что следует отметить – верхняя граница допустимых значений датчика MQ7 рассчитывается по формуле:

```
v_crit = v_crit_delta * PinsToInt(BTN_3, BTN_2, BTN_1, BTN_0) + v_crit_min;
```

где

v_crit – верхняя граница допустимых значений датчика MQ7 (расчетное значение);

v_crit_delta – множитель к значению, интерактивно установленному с помощью тумблеров S0, S1, S2, S3 (см. рисунок 5 и рисунок 8) – задается в тексте скетча;

v_crit_min – минимально возможная граница допустимых значений датчика MQ7 (задается в тексте скетча);

PinsToInt – функция, преобразующая состояния тумблеров S0, S1, S2, S3 в десятичное число.

Формула вычисления (PinsToInt) следующая:

```
v_crit = v0 + 2*v1 + 4*v2 + 8*v3;
```

где

v0, v1, v2, v3 – значения, соответствующие состояниям (1/0) тумблеров S0, S1, S2, S3.

В скетче (см. в разделе ниже) используется, также, дополнительная библиотека **TM1637**.

Ее можно найти и скачать в Инете (в том числе и на сайте https://roamer55.ru/).

8 Скетч

Все пояснения включены в текст скетча.

```
#include "TM1637.h"   //Подключение библиотеки TM1637.h

//-----------------------------------
bool its_debug = false;  //Режим отладки (если =TRUE, то используется Serial)
//-----------------------------------
//-----------------------------------
//Датчик MQ7
#define PIN_MQ7 A5  //пин для подключения MQ-7
int v_mq7 = 0;      //значение, "пришедшее" от датчика MQ-7
//-----------------------------------
//-----------------------------------
//Зуммер (buzzer, пищалка)
#define PIN_Buzzer 5     //пин для подключения "пищалки"
int Buzzer_Freq = 1000;  //частота звучания "пищалки"
int Buzzer_Delay = 500;  //длительность звучания "пищалки"
//-----------------------------------
//-----------------------------------
//Светодиоды
#define PIN_LED_R  12   //пин для подключения "красного светодиода"
#define PIN_LED_G  11   //пин для подключения "зеленого светодиода"
#define PIN_LED_IN 13   //встроенный светодиод
//-----------------------------------
//-----------------------------------
//Дисплей 3642BH
#define CLK 2                //вывод CLK
#define DIO 3                //вывод DIO
#define BRIGHT_TYPICAL  7    //яркость дисплея от 0 до 7
TM1637 tm1637(CLK,DIO);      //Создается объект tm1637
int8_t NumTab[] = {0,0,0,0}; //Создается массив данных, 0-9 это 0-9, от 10 - 15 это A-F.
//-----------------------------------
//-----------------------------------
//Тумблеры (граница допустимых значений)
#define BTN_0 10     //тумблер S0; разряд-0 числа (в двоичном исчислении)
#define BTN_1 9      //тумблер S1; разряд-1 числа (в двоичном исчислении)
#define BTN_2 8      //тумблер S2; разряд-2 числа (в двоичном исчислении)
#define BTN_3 7      //тумблер S3; разряд-3 числа (в двоичном исчислении)
#define BTN_CTRL 4   //тумблер S; "управляющий" тумблер
                     //(HIGH - режим настройки границы; LOW - режим измерения)
int v_ctrl = 0;      //состояние тумблера S; управляющий "сигнал"
                     //(HIGH - режим настройки границы; LOW - режим измерения)
int v_crit = 0;        //значение границы (допустимых значений датчика MQ7)
int v_crit_min = 25;   //минимально возможное значение границы (допустимых значений датчика
MQ7),
                       //если все тумблеры в положении "0"
int v_crit_delta = 10; //множитель к настроечному значению границы (допустимых значений
датчика MQ7)
//-----------------------------------
//-----------------------------------
int DelayBetweenCycle    = 1000; //задержка между циклами опроса датчика MQ7 (миллисек)
int CountSecForWaitReady = 10;  //Заданное кол-во СЕКУНД ожидания стабилизации показаний
MQ7 при включении питания
int iSecForWaitReady    = 0;     //Счетчик СЕКУНД ожидания стабилизации показаний MQ7
bool DevReady           = false; //Готовность датчика MQ7 к работе
//-----------------------------------

//##################################################
void setup()
{
  //----------------------------------
  //Режим пинов тумблеров
  pinMode(BTN_0, INPUT_PULLUP);
  pinMode(BTN_1, INPUT_PULLUP);
  pinMode(BTN_2, INPUT_PULLUP);
  pinMode(BTN_3, INPUT_PULLUP);
  pinMode(BTN_CTRL, INPUT_PULLUP);
  //----------------------------------
```

```
  //---------------------------------
  //Режим пина зуммера
  pinMode(PIN_Buzzer, OUTPUT);
  //---------------------------------
  //---------------------------------
  //Светодиоды
  pinMode(PIN_LED_IN, OUTPUT);  // Встроенный светодиод. Установка режима - выход
  pinMode(PIN_LED_R, OUTPUT);   // Красный светодиод. Установка режима - выход
  pinMode(PIN_LED_G, OUTPUT);   // Зеленый светодиод. Установка режима - выход
  //...
  //"Мигнуть" светодиодами при включении
  digitalWrite(PIN_LED_IN, LOW);
  digitalWrite(PIN_LED_R, HIGH);
  digitalWrite(PIN_LED_G, HIGH);
  delay(500);
  digitalWrite(PIN_LED_R, LOW);
  digitalWrite(PIN_LED_G, LOW);
  //...
  //---------------------------------
  //---------------------------------
  //Режим пина MQ7
  pinMode(PIN_MQ7, INPUT);
  //---------------------------------
  //---------------------------------
  //Дисплей 3642ВН
  tm1637.init();                    // Инициализируется дисплей
  tm1637.set(BRIGHT_TYPICAL);       // Устанавливается яркость дисплея
  NumTab[3]=0;
  NumTab[2]=0;
  NumTab[1]=0;
  NumTab[0]=0;
  tm1637.point(false);    // Выключить двоеточие
  tm1637.display(NumTab); //Вывод массива на дисплей
  //---------------------------------
  //---------------------------------
  if (its_debug == true) //режим отладки включен
  {
     Serial.begin(9600);
     Serial.println("================");
  }
  //---------------------------------
}
//#################################################

//#################################################
void loop()
{
  //=========================
  if (DevReady == false)
  {
     //Ожидание стабилизации показаний MQ7 заданное кол-во секунд
     iSecForWaitReady = iSecForWaitReady + 1; //Счетчик СЕКУНД
     //....................
     IntConvertForDisplay(iSecForWaitReady); //Конвертация v_crit в массив NumTab[]
                                             //для визуализации на дисплее 3642ВН
     tm1637.point(true); //Включить двоеточие на 3642ВН
     tm1637.display(NumTab); //Вывод массива на дисплей
     digitalWrite(PIN_LED_R, HIGH); //Зажечь красный светодиод
     digitalWrite(PIN_LED_G, HIGH); //Зажечь зеленый светодиод
     delay(500);
     //....................

     if (iSecForWaitReady >= CountSecForWaitReady) //Время ожидания исчерпано
     {
        DevReady = true; //считается, что датчик MQ7 готов к работе
        //...................
        //"обнуляем" дисплей 3642ВН
        NumTab[3]=0;
        NumTab[2]=0;
        NumTab[1]=0;
        NumTab[0]=0;
```

```
      //....................
   }

   //....................
   tm1637.point(false); //Выключить двоеточие на 3642ВН
   tm1637.display(NumTab); //Вывод массива на дисплей
   digitalWrite(PIN_LED_R, LOW); //Погасить красный светодиод
   digitalWrite(PIN_LED_G, LOW); //Погасить зеленый светодиод
   delay(500);
   //....................

   //.....................
   if (its_debug == true)  //режим отладки включен
   {
      Serial.println("................");
      Serial.print("Wait: iSecForWaitReady=");
      Serial.print(iSecForWaitReady);
      Serial.print(" ; CountSecForWaitReady=");
      Serial.print(CountSecForWaitReady);
      Serial.print("  ;  DevReady=");
      Serial.print(DevReady);
      Serial.println("");
   }
   //......................
}
//=========================
//=========================
if (DevReady == true)
{
   //Работа Устройства
   //....................
   //Считывание состояния тумблера S (режим работы Устройства)
   v_ctrl = digitalRead(BTN_CTRL);
   //....................
   //....................
   //Считывание состояния тумблеров S0..S3
   //и вычисление границы допустимых значений датчика MQ7
   v_crit = v_crit_delta * PinsToInt(BTN_3, BTN_2, BTN_1, BTN_0) + v_crit_min;
   //....................
   //....................
   //Считывание значений датчика MQ7
   v_mq7 = analogRead(PIN_MQ7);
   //....................

   if (v_ctrl == LOW)  //Режим измерения
   {
      //....................
      digitalWrite(PIN_LED_G, LOW); //Погасить зеленый светодиод
      IntConvertForDisplay(v_mq7);  //Конвертация v_crit в массив NumTab[]
                                    //для визуализации на дисплее 3642ВН
      tm1637.point(false);          //Выключить двоеточие
      tm1637.display(NumTab);       //Вывод массива на дисплей
      //....................

      if (v_mq7 >= v_crit) //Значения от MQ7 превысили границу допустимости
      {
         //.....................
         //Включить "сигнализацию"
         digitalWrite(PIN_LED_IN, HIGH); //Зажечь встроенный светодиод
         digitalWrite(PIN_LED_R, HIGH);  //Зажечь красный светодиод
         tone(PIN_Buzzer, Buzzer_Freq);  //Включить звуковой сигнал
         //.....................
      }
      else //Значения от MQ7 в пределах нормы
      {
         //.....................
         //Отключить "сигнализацию"
         digitalWrite(PIN_LED_IN, LOW); //Погасить встроенный светодиод
         digitalWrite(PIN_LED_R, LOW);  //Погасить красный светодиод
         noTone(PIN_Buzzer);            //Отключить звуковой сигнал
         //.....................
```

```
        }
     }
     else //Режим настройки границы допустимых значений датчика MQ7
     {
        //.......................
        digitalWrite(PIN_LED_IN, LOW); //Погасить встроенный светодиод
        digitalWrite(PIN_LED_R, LOW);  //Погасить красный светодиод
        digitalWrite(PIN_LED_G, HIGH); //Зажечь зеленый светодиод
        //.......................
        //.......................
        if (its_debug == true)  //режим отладки включен
        {
           Serial.println("-----------------------------");
           Serial.print("Sets. v_crit = ");
           Serial.print(v_crit);
           Serial.println("");
        }
        //.......................
        //.......................
        IntConvertForDisplay(v_crit); //Конвертация v_crit в массив NumTab[]
                                      //для визуализации на дисплее 3642ВН
        tm1637.point(false);          //Выключение двоеточия
        tm1637.display(NumTab);       //Вывод массива NumTab[] на дисплей 3642ВН
        //.......................
        //.......................
        if (its_debug == true) //режим отладки включен
        {
           Serial.println("-----------------------------");
        }
        //.......................
     }
     //.......................
     //задержка между циклами опроса датчика MQ7
     if (DelayBetweenCycle>0) { delay(DelayBetweenCycle); }
     //.......................
  }
}
//#####################################################

//#####################################################
//Конвертация целого числа (входной параметр) в массив NumTab[]
//для визуализации на дисплее 3642ВН
bool IntConvertForDisplay(int v)
{
  bool res = false;
  //...........................
  //защита от некорректных значений входного параметра
  if (v < 0) { v = 0; }
  if (v > 9999) { v = 9999; }
  //...........................
  //...........................
  int n1 = 0;
  int n2 = 0;
  int n3 = 0;
  int n4 = 0;
  double x = 0.0;
  //...........................
  //========================
  //Конвертация
  //...........................
  //"тысячи"
  x = v/1000;
  n4 = floor(x);
  //...........................
  //...........................
  // "сотни"
  x = (v - n4*1000);
  x = x/100;
  n3 = floor(x);
  //...........................
  //...........................
```

```
  //"десятки"
  x = (v - n4*1000 - n3*100);
  x = x/10;
  n2 = floor(x);
  //.........................
  //.........................
  //"единицы"
  n1 = (v - n4*1000 - n3*100 - n2*10);
  //.........................
  //.........................
  if (its_debug == true)  //режим отладки включен
  {
  Serial.print("Display. v= ");
  Serial.print(v);
  Serial.print(" ->  ");
  Serial.print(n4);
  Serial.print(" ");
  Serial.print(n3);
  Serial.print(" ");
  Serial.print(n2);
  Serial.print(" ");
  Serial.print(n1);
  Serial.print(" ");
  Serial.println("");
  }
  //.........................
  //.........................
  //Заполнение массива NumTab[]
  //для визуализации на дисплее 3642ВН
  NumTab[3]=n1;
  NumTab[2]=n2;
  NumTab[1]=n3;
  NumTab[0]=n4;
  //........................
  res = true;
  return res;
}
//#################################################

//#################################################
//Конвертация входных значений v0..v3 (двоичное представление числа)
//в десятичное
int PinsToInt0(int v3=0, int v2=0, int v1=0, int v0=0)
{
  return v0 + 2*v1 + 4*v2 + 8*v3;
}
//#################################################

//#################################################
//Считывание значений пинов (на которых "висят" тумблеры S0..S3)
//и конвертация их (из двоичного представления числа в десятичное)
int PinsToInt(int nPin3, int nPin2, int nPin1, int nPin0)
{
  int res = 0;
  int v0 = 0;
  int v1 = 0;
  int v2 = 0;
  int v3 = 0;
  if (nPin0>1) { v0 = digitalRead(nPin0); }
  if (nPin1>1) { v1 = digitalRead(nPin1); }
  if (nPin2>1) { v2 = digitalRead(nPin2); }
  if (nPin3>1) { v3 = digitalRead(nPin3); }
  res = PinsToInt0(v3, v2, v1, v0);
  return res;
}
//#################################################
```