

```

1  #include <stdDHT.h> //библиотека для работы с датчиком серии DHT
2  #include <Wire.h> //библиотека для работы с интерфейсов I2C
3  #include <LiquidCrystal_I2C.h> //библиотека подключения дисплея через I2C
4  #include <TroykaMQ.h> //библиотека для работы с датчиками серии MQ
5  #include <EEPROM.h> //библиотека для работы с памятью
6
7  //номер вывода на плате для подключения кнопки перехода в меню
8  #define MENU_BUTTON_PIN 2
9  //номер вывода на плате для подключения синего светодиода
10 #define BLUE_LED_PIN 4
11 //номер вывода на плате для подключения желтого светодиода
12 #define YELLOW_LED_PIN 5
13 //номер вывода на плате для подключения красного светодиода
14 #define RED_LED_PIN 6
15 //номер вывода на плате для подключения зуммера
16 #define BUZZER_PIN 7
17 //номер вывода на плате для подключения датчика температуры и влажности
18 #define DHT22_SENSOR_PIN 8
19 //номер вывода на плате для подключения реле
20 #define RELAY_PIN 9
21 //номер вывода на плате для подключения резистивной клавиатуры
22 #define ANALOG_KEYBOARD_PIN A0
23 //номер вывода на плате для подключения датчика углекислого газа
24 #define MQ135_SENSOR_PIN A1
25 //номер вывода на плате для подключения датчика метана
26 #define MQ4_SENSOR_PIN A2
27 //время прогрева датчиков газа, равное 60с
28 #define WARMING_UP_TIME 60000
29 //временной интервал для вывода прогресса прогрева, равный 3с
30 #define WARMING_UP_INTERVAL 3000
31 //временной интервал для опроса датчиков, равный 2с
32 #define SENSOR_POLLING_INTERVAL 2000
33 //допустимое отклонение от значения АЦП при анализе нажатой кнопки
34 #define ERROR 20
35 //количество аналоговых кнопок
36 #define ANALOG_BUTTONS_NUM 4
37
38 //объявляем объект библиотеки, указывая параметры дисплея
39 LiquidCrystal_I2C display(0x27, 20,4);
40 //объявляем объект библиотеки, указывая модель датчика температуры
41 DHT sensorDHT22(DHT22);
42 //объявляем объект библиотеки, указывая пин подключения к arduino
43 MQ4 sensorMQ4(MQ4_SENSOR_PIN);
44 //объявляем объект библиотеки, указывая пин подключения к arduino
45 MQ135 sensorMQ135(MQ135_SENSOR_PIN);
46 //переменная-флаг входа в меню
47 volatile int menuFlag = 0;
48 //переменная, запоминающая время (в мс) с момента запуска системы
49 unsigned long time = 0;
50 //переменная, хранящая температуру воздуха
51 int temperature = 0;
52 //переменная, хранящая влажность воздуха
53 int humidity = 0;
54 //переменная, хранящая концентрации CH4 в ppm
55 unsigned long concentrationCH4 = 0;
56 //переменная, хранящая концентрации CO2 в ppm

```

```

57 unsigned long concentrationCO2 = 0;
58 //максимальное допустимое значение температуры в градусах Цельсия
59 int maxTemperature = 25;
60 //максимальное допустимое значение влажности воздуха в процентах
61 int maxHumidity = 60;
62 //максимальное допустимое значение CH4 в ppm
63 unsigned long maxCH4Concentration = 4000;
64 //максимальное допустимое значение CO2 в ppm
65 unsigned long maxCO2Concentration = 2000;
66 //режим запускаемой сигнализации
67 byte signalingFlags = 0;
68 //значения АЦП для при нажатии на 1 из 4 кнопок
69 int values[4] = {254, 498, 774, 1023};
70
71 //Обработчик 0-ого прерывания. Изменяет значение флага меню
72 int changeMenuFlag(){
73     menuFlag = (menuFlag == 1) ? 0 : 1;
74 }
75
76 //определение номера кнопки по значению АЦП
77 int buttonNumber(int value){
78     int i = 0; //переменная цикла, прохода по массиву допустимых АЦП
79     //цикл прохода по массиву возможных значений АЦП
80     for(; i < ANALOG_BUTTONS_NUM; i++){
81         //проверка значения из массива АЦП с учетом погрешности
82         if((value <= (values[i] + ERROR)) && (value >= (values[i] - ERROR))){
83             return i + 1; //вернуть номер кнопки
84         }
85     }
86     return -1; //нет нажатых кнопок, вернуть -1
87 }
88
89 //функция проверки нажатия кнопок на клавиатуре
90 int analogKeyBoard(){
91     //счетчик для определения достоверности нажатия кнопки
92     static int count;
93     //старое значение аналоговой клавиатуры
94     static int oldKeyValue;
95     //для хранения нажатой клавиши на протяжении 10 проходов
96     static int innerKeyValue;
97     //получить текущую нажатую кнопку
98     int actualKeyValue = buttonNumber(analogRead(ANALOG_KEYBOARD_PIN));
99     //проверяем действительно ли нажата кнопка путем 10 проходов по алгоритму
100     if(innerKeyValue != actualKeyValue){
101         count = 0; //обнуление счетчика(все сначала)
102         //обновить текущее значения полученное с обработчика клавиатуры
103         innerKeyValue = actualKeyValue;
104     }
105     else{
106         count++; //увеличить счетчик проходов
107     }
108     //кнопка нажата и старое значение не совпадает с актуальным
109     if((count >= 10) && (oldKeyValue != actualKeyValue)){
110         oldKeyValue = actualKeyValue; //запоминаем новое значение
111     }
112     return oldKeyValue;

```

```

113 }
114
115 //функция прогрева и калибровки датчиков
116 void gasSensorsCalibration(){
117     unsigned long progressTime = 0; //для получения промежутков по 3с в цикле
118     int position = 0; //указатель позиции на дисплее (степень прогрева)
119     //подать напряжение на синий светодиод (начало прогрева датчиков)
120     digitalWrite(BLUE_LED_PIN, HIGH);
121     display.clear(); //очистить дисплей от символов
122     display.setCursor(1,0); //установить курсор на позицию (0, 0)
123     display.print("WARMING UP SENSORS"); //вывести надпись "warming up sensors"
124     display.setCursor(6, 2); //установить курсор на позицию (6, 2)
125     display.print("PROGRESS"); //вывести надпись "progress"
126     time = millis(); //получить текущее время работы
127     //цикл задержки на 60с, время прогрева датчиков газа
128     while((millis() - time) < WARMING_UP_TIME){
129         //каждые три секунды обновлять поле progress
130         if((millis() - progressTime) > WARMING_UP_INTERVAL){
131             progressTime = millis(); //получить текущее время работы
132             //установить курсор на позицию (position, 3)
133             display.setCursor(position++, 3);
134             display.print("#"); //вывести символ '#'
135         }
136     }
137     sensorMQ4.calibrate(); //выполнить калибровку датчика MQ4 на чистом воздухе
138     //выполнить калибровку датчика MQ135 на чистом воздухе
139     sensorMQ135.calibrate();
140     digitalWrite(BLUE_LED_PIN, LOW); //снять напряжения с синего светодиода
141     display.clear(); //очистить дисплей от символов
142 }
143
144 //функция чтения показаний с датчиков
145 void readDataFromSensors(){
146     //снять значение температуры с датчика
147     temperature = sensorDHT22.readTemperature(DHT22_SENSOR_PIN);
148     //снять значение влажности с датчика
149     humidity = sensorDHT22.readHumidity(DHT22_SENSOR_PIN);
150     //снять значение концентрации метана
151     concentrationCH4 = sensorMQ4.readMethane();
152     //снять значение концентрации углекислого газа
153     concentrationCO2 = sensorMQ135.readCO2();
154 }
155
156 //вывод информации на дисплей
157 void printInfo(unsigned long valueCH4, unsigned long valueCO2, int valueTemp,
158               int valueHum, int linePointer){
159     //если не в меню, то вывод с нулевой позиции, иначе с первой
160     int startPosition = (linePointer == -1) ? 0 : 1;
161     display.clear(); //очистить дисплей
162     //установить курсор на позицию (startPosition, 0)
163     display.setCursor(startPosition, 0);
164     display.print("CH4: "); //вывести название параметра
165     display.print(valueCH4); //вывести значение
166     display.print("PPM"); //вывести величину измерения
167     //установить курсор на позицию (startPosition, 1)
168     display.setCursor(startPosition, 1);

```

```

169 display.print("CO2: "); //вывести название параметра
170 display.print(valueCO2); //вывести значение
171 display.print("PPM"); //вывести величину измерения
172 //установить курсор на позицию (startPosition, 2)
173 display.setCursor(startPosition, 2);
174 display.print("TEMPERATURE: "); //вывести название параметра
175 display.print(valueTemp); //вывести значение
176 display.print("C"); //вывести величину измерения
177 //установить курсор на позицию (startPosition, 3)
178 display.setCursor(startPosition, 3);
179 display.print("HUMIDITY: "); //вывести название параметра
180 display.print(valueHum); //вывести значение
181 display.print("%"); //вывести величину измерения
182 //если в режиме меню, то выводить указатель '>' и '<'
183 if(linePointer != -1){
184     //установить курсор на позицию (0, linePointer)
185     display.setCursor(0, linePointer);
186     display.print(">"); //вывести указатель меню '>'
187     //установить курсор на позицию (19, linePointer)
188     display.setCursor(19, linePointer);
189     display.print("<"); //вывести указатель меню '<'
190 }
191 }
192
193 //функция анализа считанных с датчиков данных
194 void dataAnalysis(){
195     signalingFlags = 0; //сбрасывание всех флагов
196     //проверка на превышение CH4 и CO2
197     if((concentrationCH4 > maxCH4Concentration) ||
198         (concentrationCO2 > maxCO2Concentration)){
199         signalingFlags = signalingFlags | 1; //установить флаг 1
200     }
201     //проверка на превышения температуры и влажности
202     if((temperature > maxTemperature) || (humidity > maxHumidity)){
203         signalingFlags = signalingFlags | (1 << 1); //установить флаг 2
204     }
205     digitalWrite(REDA_LED_PIN, LOW); //снять питание с красного светодиода
206     digitalWrite(YELLOW_LED_PIN, LOW); //снять питание с желтого светодиода
207     digitalWrite(RELAY_PIN, LOW); //закрыть реле
208     noTone(BUZZER_PIN); //отключить пьезодинамик
209     //определение типа сигнализирования по установленным флагам
210     switch(signalingFlags){
211         case 1: //превышают только газовые параметры
212             //подать напряжение на красный светодиод
213             digitalWrite(REDA_LED_PIN, HIGH);
214             digitalWrite(RELAY_PIN, HIGH); //открыть реле
215             tone(BUZZER_PIN, 1500); //включить пьезодинамик
216             break;
217         case 2: //превышает только температура или влажность
218             //подать напряжение на желтый светодиод
219             digitalWrite(YELLOW_LED_PIN, HIGH);
220             tone(BUZZER_PIN, 1700); //включить пьезодинамик
221             break;
222         case 3: //превышает температура или влажность, а также газовые параметры
223             //подать напряжение на красный светодиод
224             digitalWrite(REDA_LED_PIN, HIGH);

```

```

225     //подать напряжение на желтый светодиод
226     digitalWrite(YELLOW_LED_PIN, HIGH);
227     digitalWrite(RELAY_PIN, HIGH); //открыть реле
228     tone(BUZZER_PIN, 1900); //включить пьезодинамик
229     break;
230 }
231 }
232
233 //основное меню
234 void menuFunc(){
235     int menuPointer = 0; //указатель на пункт меню
236     int currentKeyValue = 0; //значение нажатой кнопки
237     int flag = 0; //флаг для определения момента отпускания кнопки
238     //вывести меню с указателем '> ... <' на дисплей
239     printInfo(maxCH4Concentration, maxCO2Concentration, maxTemperature,
240             maxHumidity, menuPointer);
241     while(menuFlag == 1){ //если была нажата кнопка входа в меню
242         currentKeyValue = analogKeyBoard(); //чтение значения с клавиатуры
243         if((currentKeyValue != -1) && (flag == 0)){ //обработчик нажатия кнопки
244             flag = 1; //кнопка нажата
245             if(currentKeyValue == 1){ //если нажата 1 кнопка
246                 menuPointer++; //переход к следующему пункту меню
247                 //когда дошло до конца меню сброс в начало
248                 menuPointer = (menuPointer == 4) ? 0 : menuPointer;
249             }
250             else if(currentKeyValue == 2){ //если нажата 2 кнопка
251                 menuPointer--; //переход к предыдущему пункту меню
252                 //когда дошло до начала меню сброс в конец
253                 menuPointer = (menuPointer == -1) ? 3 : menuPointer;
254             }
255             else if(currentKeyValue == 3){ //если нажата 3 кнопка
256                 //в зависимости от пункта меню уменьшаем максимальное значение
257                 switch(menuPointer){
258                     case 0: //menuPointer указывает на максимальную концентрацию CH4
259                         maxCH4Concentration -= 25;
260                         break;
261                     case 1: //menuPointer указывает на максимальную концентрацию CO2
262                         maxCO2Concentration -= 25;
263                         break;
264                     case 2: //menuPointer указывает на максимальную температуру
265                         maxTemperature -= 1;
266                         break;
267                     case 3: //menuPointer указывает на максимальную влажность
268                         maxHumidity -= 1;
269                         break;
270                 }
271             }
272             else if(currentKeyValue == 4){ //если нажата 4 кнопка
273                 //в зависимости от пункта меню увеличиваем максимальное значение
274                 switch(menuPointer){
275                     case 0: //menuPointer указывает на максимальную концентрацию CH4
276                         maxCH4Concentration += 25;
277                         break;
278                     case 1: //menuPointer указывает на максимальную концентрацию CO2
279                         maxCO2Concentration += 25;
280                         break;

```

```

281
282         case 2: //menuPointer указывает на максимальную температуру
283             maxTemperature += 1;
284             break;
285         case 3: //menuPointer указывает на максимальную влажность
286             maxHumidity += 1;
287             break;
288     }
289 }
290
291 //обработчик отпускания кнопки
292 if((currentKeyValue == -1) && (flag == 1)){
293     flag = 0; //кнопки не нажаты
294     //обновить меню на дисплее
295     printInfo(maxCH4Concentration, maxCO2Concentration, maxTemperature,
296             maxHumidity, menuPointer);
297 }
298 }
299 }
300
301 //чтения данных из eeprom (инициализация максимальных значений)
302 void initVariablesFromEEPROM(){
303     //считать данные в maxCH4Concentration из памяти
304     EEPROM.get(0, maxCH4Concentration);
305     //считать данные в maxCO2Concentration из памяти
306     EEPROM.get(4, maxCO2Concentration);
307     //считать данные в maxTemperature из памяти
308     EEPROM.get(8, maxTemperature);
309     //считать данные в maxHumidity из памяти
310     EEPROM.get(10, maxHumidity);
311 }
312
313 //запись данных в eeprom
314 void saveVariablesIntoEEPROM(){
315     //записать данные из maxCH4Concentration в память
316     EEPROM.put(0, maxCH4Concentration);
317     //записать данные из maxCO2Concentration в память
318     EEPROM.put(4, maxCO2Concentration);
319     //записать данные из maxTemperature в память
320     EEPROM.put(8, maxTemperature);
321     //записать данные из maxHumidity в память
322     EEPROM.put(10, maxHumidity);
323 }
324
325 void setup() { //задание начальных параметров
326     //установить режим вывода на прием сигнала
327     pinMode(MENU_BUTTON_PIN, INPUT);
328     //установить режим вывода на выдачу сигнала
329     pinMode(BLUE_LED_PIN, OUTPUT);
330     //установить режим вывода на выдачу сигнала
331     pinMode(YELLOW_LED_PIN, OUTPUT);
332     //установить режим вывода на выдачу сигнала
333     pinMode(RED_LED_PIN, OUTPUT);
334     //установить режим вывода на выдачу сигнала
335     pinMode(BUZZER_PIN, OUTPUT);
336     //установить режим вывода на прием сигнала

```

```

337     pinMode(DHT22_SENSOR_PIN, INPUT);
338     //установить режим вывода на выдачу сигнала
339     pinMode(RELAY_PIN, OUTPUT);
340     //связываем прерывание с обработчиком
341     attachInterrupt(0, changeMenuFlag, RISING);
342     //инициализация дисплея
343     display.init();
344     //включить подсветку дисплея
345     display.backlight();
346     //инициализировать максимальные значения
347     initVariablesFromEEPROM();
348     //прогрев и калибровка датчиков
349     gasSensorsCalibration();
350 }
351
352 void loop() { //основной цикл программы
353     //если прошло 2 секунды опрос датчиков
354     if((millis() - time) > SENSOR_POLLING_INTERVAL){
355         time = millis(); //обновить время работы контроллера
356         readDataFromSensors(); //получить данные с датчиков
357         //вывести информацию на дисплей с обновленными параметрами
358         printInfo(concentrationCH4, concentrationCO2, temperature, humidity, -1);
359         dataAnalysis(); //анализ полученных данных
360     }
361     if(menuFlag == 1){ //зайти в меню
362         menuFunc();
363         saveVariablesIntoEEPROM(); //записать значения в память
364     }
365 }

```