

```

01.// группа 150501, Божко И.И.
02.// Микропроцессорное устройство контроля параметров супермаркета
03.// Подключение библиотек
04.#include <Wire.h>
05.#include <LiquidCrystal_I2C.h>
06.#include <DHT.h>
07.#include <AmperkaKB.h>
08.
09.#define DHTPIN 5 // вывод для подключения датчика DHT22
10.#define LIGHT_SENSOR A0 // вывод для подключения датчика LM393
11.#define RAIN_SENSOR A2 // вывод для подключения датчика MH-RD
11.#define PIN_RELAY 6 // вывод для подключения реле
12.// присваиваем имя dht для датчика DHT22
13.DHT dht(DHTPIN, DHT22);
14.// присваиваем имя KB для клавиатуры
15.AmperkaKB KB(13, 12, 11, 10, 9, 8, 7);
16.// присваиваем имя LCD для дисплея
17.LiquidCrystal_I2C LCD(0x27,16,2);
18.// переменная для выбора режима работы
19.int printingNow = 1;
20.// минимальное допустимое значение температуры
21.int limitT = 15;
22.// максимальное допустимое значение уровня утечки
23.int limitR = 50;
24.// минимальное допустимое значение освещённости
25.int limitL = 200;
26.// включение/выключение звукового сигнала
27.int siren = 1;
28.// функция мигания светодиодом
29.void blinkLed(int a, int delayed, int enddelay) {
30.  digitalWrite(a, HIGH); // включение светодиода
31.  delay(delayed); // задержка
32.  digitalWrite(a, LOW); // выключение светодиода
33.  if(enddelay) delay(delayed); // задержка
34.}
35.// функция инициализации переменных
36.void setup() {
37.  // инициализация вывода для подключения реле
38.  pinMode(PIN_RELAY, OUTPUT);
39.  dht.begin(); // инициализация датчика DHT22
40.  KB.begin(KB4x3); // инициализация клавиатуры
41.  LCD.init(); // инициализация LCD дисплея
42.  LCD.backlight(); // включение подсветки дисплея
43.  // инициализация выводов для подключения светодиодов
44.  pinMode(2, OUTPUT);
45.  pinMode(3, OUTPUT);
46.  pinMode(4, OUTPUT);
47.  // мигание светодиодами для индикации включения

```

```

48.  blinkLed(4, 100, 1);
49.  blinkLed(2, 100, 1);
50.  blinkLed(3, 100, 1);
51.  blinkLed(2, 100, 1);
52.  blinkLed(4, 100, 1);
53.  blinkLed(2, 100, 1);
54.  blinkLed(3, 100, 1);
55.}
56.// основной цикл программы
57.void loop() {
58.  // вспомогательный цикл для регулировки частоты снятия показаний
59.  for(int i = 0; i < 20; i++) {
60.    delay(100); // задержка 100мс
61.    // чтение показаний с датчиков
62.    if(i == 10) {
63.      // переменные для отслеживания превышения допустимых значений
64.      int alertT = 0;
65.      int alertR = 0;
66.      int alertL = 0;
67.      // чтение показаний с датчиков
68.      float temp = dht.readTemperature();
69.      float fleak = analogRead(RAIN_SENSOR);
70.      int leak = round(100.0 - (fleak * (100.0 / 1023.0)));
71.      float flight = analogRead(LIGHT_SENSOR);
72.      int light = round(1000 - (flight * 1000.0 / 1023.0));
73.      // вывод значения температуры на экран
74.      if(printingNow == 1) {
75.        LCD.clear();
76.        LCD.setCursor(0, 0);
77.        LCD.print("TEMPERATURE:");
78.        LCD.setCursor(14, 0);
79.        LCD.print(limitT);
80.        LCD.setCursor(0, 1);
81.        LCD.print(temp);
82.        LCD.setCursor(6, 1);
83.        LCD.print((char)223);
84.        LCD.print("C");
85.      };
86.      // вывод значения освещённости на экран
87.      if(printingNow == 2) {
88.        LCD.clear();
89.        LCD.setCursor(0, 0);
90.        LCD.print("LIGHT:");
91.        LCD.setCursor(13, 0);
92.        LCD.print(limitL);
93.        LCD.setCursor(0, 1);
94.        LCD.print(light);
95.        LCD.setCursor(6, 1);

```

```

96.         LCD.print("lx");
97.     };
98.     // вывод значения уровня утечки на экран
99.     if(printingNow == 3) {
100.         LCD.clear();
101.         LCD.setCursor(0, 0);
102.         LCD.print("LEAK:");
103.         LCD.setCursor(13, 0);
104.         LCD.print(limitR);
105.         LCD.setCursor(0, 1);
106.         LCD.print(leak);
107.     }
108.     // проверка значения температуры
109.     if(temp < limitT) alertT = 1;
110.     // проверка значения уровня утечки
111.     if(leak > limitR) alertR = 1;
112.     // проверка значения освещённости
113.     if(light < limitL) alertL = 1;
114.     // включение органов индикации при отклонении от нормы
115.     if(alertT || alertR || alertL) {
116.         // включение динамика
117.         if(siren == 1) digitalWrite(PIN_RELAY, HIGH);
118.         // включение зелёного светодиода
119.         if(alertT) digitalWrite(4, HIGH);
120.         // включение белого светодиода
121.         if(alertR) digitalWrite(3, HIGH);
122.         // включение оранжевого светодиода
123.         if(alertL) digitalWrite(2, HIGH);
124.         delay(300); // задержка 300мс
125.         digitalWrite(PIN_RELAY, LOW); // выключение динамика
126.         // выключение зелёного светодиода
127.         digitalWrite(4, LOW);
128.         // выключение белого светодиода
129.         digitalWrite(3, LOW);
130.         // выключение оранжевого светодиода
131.         digitalWrite(2, LOW);
132.     };
133. }
134. // чтение состояния клавиатуры
135. KB.read();
136. if (KB.isHold()) {
137.     if(KB.getNum == 0) {
138.         if(siren == 1) {
139.             siren = 0;
140.             LCD.clear();
141.             LCD.setCursor(0, 0);
142.             LCD.print("SOUND OFF");
143.             delay(1000);

```

```

144.     }
145.     else {
146.         siren = 1;
147.         LCD.clear();
148.         LCD.setCursor(0, 0);
149.         LCD.print("SOUND ON");
150.         delay(1000);
151.     }
152. }
153. }
154. if (KB.justPressed()) { // нажата кнопка
155.     if(KB.getNum == 15) { // нажата кнопка #
156.         // переключение в режим вывода освещённости
157.         if(printingNow == 1) {
158.             printingNow = 2;
159.             for(int k = 0; k < 3; k++) { // индикация переключения
160.                 blinkLed(2, 100, 1); // мигание оранжевым светодиодом
161.             };
162.         }
163.         // переключение в режим вывода уровня утечки
164.         else if(printingNow == 2) {
165.             printingNow = 3;
166.             for(int k = 0; k < 3; k++) { // индикация переключения
167.                 blinkLed(3, 100, 1); // мигание белым светодиодом
168.             };
169.         }
170.         // переключение в режим вывода температуры
171.         else {
172.             printingNow = 1;
173.             for(int k = 0; k < 3; k++) { // индикация переключения
174.                 blinkLed(4, 100, 1); // мигание зелёным светодиодом
175.             };
176.         };
177.     }
178.     if(KB.getNum == 14) { // нажата кнопка *
179.         // новое значение допустимого значения
180.         int newNum = 0;
181.         // 10с на ввод нового значения
182.         for(int j = 0; j < 100; j++) {
183.             delay(100); // задержка 100мс
184.             KB.read(); // чтение значения с клавиатуры
185.             if (KB.justPressed()) { // нажата кнопка
186.                 // нажата кнопка * - подтверждение ввода
187.                 if(KB.getNum == 14) break;
188.                 // добавление новой цифры
189.                 if(KB.getNum != 14 && KB.getNum != 15) {
190.                     newNum = newNum * 10 + KB.getNum;
191.                 }

```

```

192.         };
193.     };
194.     // изменение допустимых значений температуры
195.     if(printingNow == 1) {
196.         // проверка правильности ввода
197.         if(newNum >= 0 && newNum <= 80) {
198.             limitT = newNum; // установка нового значения
199.             LCD.clear();
200.             LCD.setCursor(0, 0);
201.             LCD.print("NEW TEMP:");
202.             LCD.setCursor(0, 1);
203.             LCD.print(newNum);
204.             for(int k = 0; k < 3; k++) { // индикация изменения
205.                 // мигание зелёным светодиодом
206.                 blinkLed(4, 100, 1);
207.             };
208.         } else {
209.             LCD.clear();
210.             LCD.setCursor(0, 0);
211.             LCD.print("INVALID");
212.             LCD.setCursor(0, 1);
213.             LCD.print("INPUT");
214.             delay(1000);
215.         };
216.         // изменение допустимых значений освещённости
217.     } else if(printingNow == 2) {
218.         // проверка правильности ввода
219.         if(newNum > 0 && newNum < 5000) {
220.             limitL = newNum; // установка нового значения
221.             LCD.clear();
222.             LCD.setCursor(0, 0);
223.             LCD.print("NEW LIGHT:");
224.             LCD.setCursor(0, 1);
225.             LCD.print(newNum);
226.             for(int k = 0; k < 3; k++) { // индикация изменения
227.                 // мигание оранжевым светодиодом
228.                 blinkLed(2, 100, 1);
229.             };
230.         } else {
231.             LCD.clear();
232.             LCD.setCursor(0, 0);
233.             LCD.print("INVALID");
234.             LCD.setCursor(0, 1);
235.             LCD.print("INPUT");
236.             delay(1000);
237.         };
238.         // изменение допустимых значений уровня утечки
239.     } else if(printingNow == 3) {

```

```

240.          // проверка правильности ввода
241.          if(newNum > 0 && newNum < 100) {
242.              limitR = newNum; // установка нового значения
243.              LCD.clear();
244.              LCD.setCursor(0, 0);
245.              LCD.print("NEW LEAK:");
246.              LCD.setCursor(0, 1);
247.              LCD.print(newNum);
248.              for(int k = 0; k < 3; k++) { // индикация изменения
249.                  // мигание белым светодиодом
250.                  blinkLed(3, 100, 1);
251.              };
252.          } else {
253.              LCD.clear();
254.              LCD.setCursor(0, 0);
255.              LCD.print("INVALID");
256.              LCD.setCursor(0, 1);
257.              LCD.print("INPUT");
258.              delay(1000);
259.          };
260.      };
261.  }
262.  }
263.  }
264.  }

```