```
001. /*
002.  Дашкевич Андрей Петрович, гр. 050541
003. "Микропроцессорное устройство контроля параметров элеватора"
004. */
005.
006. #include <Arduino.h>
007. #include <DHTStable.h>        //библиотека для датчика DHT11
008. #include <Wire.h>        //библиотеки для дисплея LCD1602
009. #include <LiquidCrystal_I2C.h>
010. #include <TroykaMQ.h>        //библиотека для датчика MQ2
011.
012. #include <Adafruit_NeoPixel.h>    //библиотеки для установки
013. #include <avr/sleep.h>     // спящего режима микроконтроллера
014.
015. #define DHT11_PIN 2 //подкл DHT11 ко входу D2
016. #define MQ2_PIN A0 //подкл MQ2 ко входу A0
017. #define MQ2_D 13
018. #define PIEZO_PIN 12    //подкл пьезодинамика ко входу D12
019. #define PIN_BUTTON_POWER 3    //кнопка вкл/выкл ко входу D3
020. #define PIN_BUTTON_SCREEN 5   //кнопка перекл экрана к  D5
021. #define RED 11           //красный светодиод к D11
022. #define YELLOW 9         //желтый светодиод к D9
023. #define GREEN 10         //зеленый светодиод к D10
024.
025. DHTStable dht;           //инициализация DHT11
026. MQ2 mq2(MQ2_PIN);        //инициализация MQ2
027. LiquidCrystal_I2C lcd(0x27, 16, 2); //инициализация дисплея
028.
029. bool is_startUp = false;           //вкл устройство
030. bool is_shutDown = false;     //устройство в спящий режим
031. bool is_deviceOn = true;
032. bool first_init = true;       //первое включение
033.
034. bool screen_num = false; //переменная для переключения экрана
035.
036. float temp = 0;          //тем-ра воздуха
037. float hum = 0;           //влажность воздуха
038. float smoke = 0;         //содержание дыма
039.
040. void initialization() {  //мигание светодиодами при включении
041.
042. digitalWrite(YELLOW, HIGH);
043. delay(500);
044. digitalWrite(YELLOW, LOW);
045. digitalWrite(RED, HIGH);
046. delay(500);
047. digitalWrite(RED, LOW);
048. digitalWrite(GREEN, HIGH);
049. delay(500);
050. digitalWrite(GREEN, LOW);
051. digitalWrite(YELLOW, HIGH);
052. delay(500);
053. digitalWrite(YELLOW, LOW);
```

```
digitalWrite(RED, HIGH);
delay(500);
digitalWrite(RED, LOW);
digitalWrite(GREEN, HIGH);
delay(500);
digitalWrite(GREEN, LOW);

digitalWrite(YELLOW, HIGH);
}
void make_sound(int mode) {   //воспроизведение мелодии
switch(mode) {
case 1:                       //включение
tone(PIEZO_PIN, 433, 100);
delay(500);
tone(PIEZO_PIN, 578, 200);
delay(500);
break;
case 2:                       //выключение
tone(PIEZO_PIN, 438, 250);
delay(500);
tone(PIEZO_PIN, 578, 200);
delay(500);
break;
case 3:                       //превышение значений
tone(PIEZO_PIN, 800);
delay(1500);
noTone(PIEZO_PIN);
break;
}
}

void sleep() {           //переход в спящий режим
lcd.clear();
lcd.print("SLEEP");

digitalWrite(RED,LOW);        //выкл светодиодов
digitalWrite(GREEN, LOW);
digitalWrite(YELLOW, LOW);

make_sound(2);      //соответствующее звуковое сопровождение
delay(3000);

lcd.clear();
lcd.noBacklight();       //отключение подсветки дисплея

is_shutDown = false;
set_sleep_mode(SLEEP_MODE_PWR_DOWN);
sleep_mode();            //переход в СР
}

void wake_up() {         //выход из спящего режима
lcd.backlight();         //включение подсветки дисплея
lcd.clear();
```

```
lcd.setCursor(5, 0);
lcd.print("WAKE UP");

initialization();        //мигание светодиодами
make_sound(1);      //соответствующее звуковое сопровождение
delay(3000);

is_startUp = false;
lcd.clear();
}

void power_button_handler() {      //обработка прерывания при
if (is_startUp || is_shutDown) return; //нажатии кнопки 1
static unsigned long millisPrev;

if(millis() - 100 > millisPrev) {
is_deviceOn = !is_deviceOn;
Serial.println("function");
if (is_deviceOn) is_startUp = true;  //установка флага выбора
else is_shutDown = true;           //дальнейшего режима
}

millisPrev = millis();

}

void setup() { //инициализация

Serial.begin(9600); //открытие порта

mq2.calibrate(); //калибровка датчика MQ2

pinMode(MQ2_PIN, INPUT);
pinMode(MQ2_D, OUTPUT);
pinMode(DHT11_PIN, INPUT);
pinMode(PIEZO_PIN, OUTPUT);
pinMode(ledPower, OUTPUT);

pinMode(YELLOW, OUTPUT);
pinMode(RED, OUTPUT);
pinMode(GREEN, OUTPUT);

lcd.init();
lcd.backlight();

    attachInterrupt(digitalPinToInterrupt(PIN_BUTTON_POWER),
power_button_handler, RISING); //прерывание при LOW->HIGHT
}

void dht11_parametrs() {      //считывание параметров dht11
int chk = dht.read11(DHT11_PIN);
switch (chk)        //проверка на возникновение ошибок
{
```

```
160. case DHTLIB_OK:
161. Serial.print("OK,\t");
162. break;
163. case DHTLIB_ERROR_CHECKSUM:
164. Serial.print("Checksum error,\t");
165. break;
166. case DHTLIB_ERROR_TIMEOUT:
167. Serial.print("Time out error,\t");
168. break;
169. default:
170. Serial.print("Unknown error,\t");
171. break;
172. }
173.
174. temp = dht.getTemperature();
175. hum = dht.getHumidity();
176. delay(1000);
177. }
178.
179. void mq2_parametrs() {        //получение значений mq2
180. Serial.print("Smoke: ");
181. smoke = mq2.readSmoke();
182. Serial.print(smoke);
183.
184. Serial.println("ppm");
185.
186. }
187.
188. void read_parametrs() {       //чтение всех параметров
189. dht11_parametrs();
190. mq2_parametrs();
191. }
192.
193. void check_parametrs(int value) {  //проверка полученных
194. if (temp > 35 || hum > 35 || smoke > 350){
195. digitalWrite(GREEN, LOW);  //значений на соответствие норме
196. digitalWrite(RED, HIGH);
197. }
198. else {
199. digitalWrite(RED, LOW);
200. digitalWrite(GREEN, HIGH);
201. }
202. if (smoke > 350) make_sound(3); //превышение дыма -> вкл
203. if (value == 0) {                              //пьезодинамика
204. lcd.clear();
205. lcd.setCursor(0, 0);
206. lcd.print("Temp:");
207. lcd.print(temp);
208. if (temp > 35) lcd.print("!");
209. lcd.setCursor(0, 1);
210. lcd.print("Hum:");
211. lcd.print(hum);
212. if (hum > 35) lcd.print("!");
```

```
213. }
214. if (value == 1) {
215. lcd.clear();
216. lcd.setCursor(0, 0);
217. lcd.print("Smoke:");
218. lcd.print(smoke);
219. lcd.print("ppm");
220. if (smoke > 350) lcd.print("!");
221. }
222. }
223.
224. void screen_switching(int state) {//вывод значений на экран
225. if (state == true) {
226. screen_num = !screen_num;
227. digitalWrite(RED, LOW);
228. digitalWrite(GREEN, LOW);
229. }
230. if (screen_num == 0) check_parametrs(0);
231. if (screen_num == 1) check_parametrs(1);
232. }
233.
234. void screen_button_handler(){//переключение экрана по нажатию
235. int button_state = digitalRead(PIN_BUTTON_SCREEN);//кнопки 2
236. screen_switching(button_state);
237. }
238.
239. void loop() {
240.
241. if (first_init) {        //если включено в первый раз
242. first_init = !first_init;
243. lcd.clear();
244. lcd.setCursor(0,0);
245. lcd.print("INITIALIZATION");      //мигание светодиодами
246. initialization();
247. make_sound(1);      //звуковая индикация
248. }
249.
250. if (is_startUp) wake_up();    //проверки на возможные
251. if (is_shutDown) sleep();     // режимы работы
252.
253. read_parametrs();
254. screen_button_handler();
255.
256. delay(1000);
257. }
```