```
001 //Гр. 050541, Каранкевич В.Г.
002 //Курсовой проект "Устройство отображения информации спортивного комплекса"
003
004 #include <string.h>
005 #include <stdio.h>
006
007 #include <Key.h> // Подключаем библиотеку
008 #include <Keypad.h> // Подключаем библиотеку
009
010 #include <SPI.h> // Подключаем библиотеку для работы SPI протоколом
011 #include <Adafruit_GFX.h> // Подключаем библиотеку с функция для работы со
светодиодной матрицей
012 #include <Max72xxPanel.h> // Подключаем библиотеку для работы для управления
матричными светодиодоми
013
014 #include "DHT.h" //Подключаем библиотеку для работы с датчиком температуры и
влажности
015 #include <BH1750.h> //Подключаем библиотеку для работы с датчиком освещенности
016
017 #include <Key.h> //Подключаем библиотеку для работы с клавиатурой
018 #include <Keypad.h> //Подключаем библиотеку для работы с матричной клавиатурой
019
020 #define DHTPIN 3 //Разьем к которому подключен датчик температуры и влажности
021 #define LEDPIN 5 //Разьем к которому подключен светодиод
022 #define SPEAKERPIN 2 //Разьем к которому подключен датчик температуры и влажности
023 #define CSPIN  4 // Указываем к какому выводу подключен контакт CS
024
025 DHT dht(DHTPIN, DHT11); //Инициация датчика
026 BH1750 lightMeter; //Переменная для освещенности
027
028 //Настройка параметров для звукового оповещения
029 int notes[] = {   //Проигрывание нужной частоты
030  850,  750, 900,
031 };
032 int times[] = {   //Длительность проигрывания частоты
033  4000, 1000, 3000,
034 };
035 int delays[] = {  //Задержка между проигрыванием частоты
036  1000, 1000, 1000,
037 };
038
039 //Параметры для клавиатуры
040 const byte ROWS = 3; // число строк клавиатуры
041 const byte COLS = 1; // число столбцов клавиатуры
042 char hexaKeys[ROWS][COLS] = {
043       {'1'},
044       {'2'},
045       {'3'}
046 };
047
048 byte rowPins[ROWS] = {10,9,8};// к каким выводам подключаем управление строками
049 byte colPins[COLS] = {7}; // к каким выводам подключаем управление столбцами
050 //Инициализация customKeypad (клавиатуры) значениями
051 Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
052
053 int numberOfHorizontalDisplays = 1; // Количество матриц по горизонтали
054 int numberOfVerticalDisplays = 4; // Количество матриц по-вертикали
055
056 //Инициализация светодиодной матрицы
057 Max72xxPanel matrix = Max72xxPanel(CSPIN, numberOfHorizontalDisplays,
numberOfVerticalDisplays);
058
```

```
059 //Строка для вывода текста на дисплей в функции outPutText
060 String runningLineString = "";
061 int wait = 100; // интервал, чем меньше тем бытрее бежит строка
062 int spacer = 1; // Промежуток между символами (кол-во точек)
063 int width = 5 + spacer; // Ширина шрифта составляет 5 пикселей
064
065 //Переменные для работы с температурой
066 int limitTemperature = 23;
067 int temperature = 0;
068
069 //Функция, которая запускается в первый раз при включении устройства
070 void setup() {
071
072   //Инициализация датчиков
073   initIllumination();
074   initHumidityAndTemerature();
075
076   outPutText("Запуск"); //Вывод текста на дисплей
077   ledIndicatorOn(); //Мигание светодиодом
078   dinamikOn(); //Издание звукового сигнала
079
080 }
081 //Конец функции
082
083 int key = NO_KEY; //Инициализация нажатой клавиши
084 //Режим для работы с выводом текста на дисплей
085 int modeForOutPutTextSensor = 1;
086
087 //Основной цикл программы
088 void loop() {
089
090    do{
091     //Условие для нажатой клавиши для вывода на дисплей параметров
092     modeWorkDevices();
093     //Условие температуры для срабатывания звукового сигнала
094     if(temperature > limitTemperature) {
095        dinamikOn();
096        outPutText("Превышение!");
097     }
098
099     //Получение номера клавиши нажатой пользователем
100     delay(3000);
101     key = customKeypad.getKey();
102
103   }while(key == NO_KEY);
104
105   //Преключение режимов в зависимости от нажатой клавиши
106   if(key == '1'){
107     modeForOutPutTextSensor = 1;
108   }else if(key == '2'){
109     modeForOutPutTextSensor = 2;
110   }else if(key == '3'){
111     modeForOutPutTextSensor = 3;
112   }
113
114   key = NO_KEY; //Обнуление нажатой клавиши
115 }
116 //Конец функции для Основного цикла программы
117
118 //Функция проверка условия для нажатой клавиши для вывода
119 //на дисплей параметров датчиков
120 void modeWorkDevices(){
```

```
121     //Условие для нажатой клавиши для вывода на дисплей параметров
122     if(modeForOutPutTextSensor == 1){
123       //Вывод на диспелей освещенности, температуры и влажности
124       outPutText(getParametrHumidityAndTemerature());
125       outPutText(getParametrIllumination());
126     }else if(modeForOutPutTextSensor == 2){
127       //Вывод на диспелей освещенность
128       outPutText(getParametrIllumination());
129     }else if(modeForOutPutTextSensor == 3){
130       //Вывод на диспелей температуру и влажность
131       outPutText(getParametrHumidityAndTemerature());
132     }
133 }
134 //Конец функции для вывыода параметров
135 
136 //Включение динамика и его выключение
137 void dinamikOn(){
138     for (int i = 0; i < 3; i++){
139       //Подаем на вывод звука нужные частоты и долготу звучания (определенные в
notes и times)
140       tone(SPEAKERPIN, notes[i], times[i]);
141       delay(delays[i]);
142     }
143       noTone(SPEAKERPIN); //Выключаем динамик
144 }
145 //Конец функции работы с динамиком
146 
147 //Включение светодиода (при запуске программы)
148 void ledIndicatorOn(){
149   pinMode(LEDPIN, OUTPUT);        // назначаем наш пин "выходом"
150   digitalWrite(LEDPIN, HIGH);    // включаем светодиод
151    delay(1000);                   // ждем 1000 миллисекунд (1 секунда)
152    digitalWrite(LEDPIN, LOW);     // выключаем светодиод
153    delay(1000);                   // ждем еще 1 секунду
154   digitalWrite(LEDPIN, HIGH);
155 }
156 //Конец функции работы со светодиодом
157 
158 //Инициализация и запуск работы датчика освещенности
159 void initIllumination(){
160   lightMeter.begin();
161 }
162 //Конец функции инициализации
163 
164 char bufferForLux[10]; //Буфер для записи значения люкс
165 //Значение люкс, которое будет выводится из функции getParametrIllumination()
166 char * outLuxText = new char[15];
167 
168 //Функция получения строки для вывода с датчика ОСВЕЩЕННОСТИ
169 char* getParametrIllumination(){
170   //Получение значения освещенности в люксах
171   float lux = lightMeter.readLightLevel();
172 
173   memset(bufferForLux, 0, 10); //Очищение буфера
174   memset(outLuxText, 0, 15); //Очищение буфера
175   itoa(lux,bufferForLux,10); //Запись значения в буфер
176 
177   strcat(outLuxText, bufferForLux);
178   strcat(outLuxText, " lx");
179 
180   return outLuxText;
181 }
```

```
182 //Конец функции получения строки для вывода с датчика ОСВЕЩЕННОСТИ
183
184 //Инициализация и запуск работы датчика освещенности и влажности
185 void initHumidityAndTemerature() {
186   dht.begin();
187 }
188
189 char bufferForTemperature[10]; //Буфер для параметра температуры
190 char bufferForHumidity[10]; //Буфер для параметра влажности
191 //Строка хранящая значения для вывода температуры и влажности
192 char * outHumidityAndTemperatureText = new char[60];
193
194 //Функция получения строки для вывода с датчика ТЕМПЕРАТУРЫ И ВЛАГИ
195 char* getParametrHumidityAndTemerature(){
196   float humidity = dht.readHumidity(); //Измеряем влажность
197   temperature = dht.readTemperature(); //Измеряем температуру
198   if (isnan(humidity) || isnan(temperature)) {  // Проверка. Если не удается
считать показания, выводится «Ошибка считывания», и программа завершает работу
199     Serial.println("Ошибка считывания");
200     return;
201     }
202
203   memset(bufferForTemperature, 0, 10);
204   memset(bufferForHumidity, 0, 10);
205   memset(outHumidityAndTemperatureText, 0, 60);
206   //char buffer[10];
207   itoa(humidity,bufferForHumidity,10); //Заполнение буффера строковым значение для
температуры
208   //char buffer2[10];
209   itoa(temperature,bufferForTemperature,10); //Заполнение буффера строковым
значение для влажности
210
211
212   strcat(outHumidityAndTemperatureText, bufferForHumidity);
213   strcat(outHumidityAndTemperatureText, "% ");
214   strcat(outHumidityAndTemperatureText, bufferForTemperature);
215   strcat(outHumidityAndTemperatureText, "C");
216
217   return outHumidityAndTemperatureText;
218 }
219 //Конец функции получения строки для вывода с датчика ТЕМПЕРАТУРЫ И ВЛАГИ
220
221 //Функция для вывода информации на дисплей
222 void outPutText(char* str){ //Атрибут строки который будет выводиться на дисплей
223   runningLineString = utf8rus(str); //Передаем атрибут строки для вывода в
переменную
224
225   matrix.setIntensity(0); // Задаем яркость от 0 до 15
226   matrix.setRotation(3); // Направление текста 1,2,3,4
227
228   if (Serial.available()) { // получили данные
229     runningLineString = utf8rus(Serial_Read()); // Считываем и сохроняем в
переменную
230   }
231   for ( int i = 0 ; i < width * runningLineString.length() + matrix.width() -
spacer; i++ )
232   {
233     matrix.fillScreen(LOW);
234     int letter = i / width; // номер символа выводимого на матрицу
235     int x = (matrix.width() - 1) - i % width;
236     int y = (matrix.height() - 8) / 2; // отцентрировать текст по вертикали
237     while ( x + width - spacer >= 0 && letter >= 0 ) {
```

```
      if ( letter < runningLineString.length() ) {
        matrix.drawChar(x, y, runningLineString[letter], HIGH, LOW, 1);
      }
      letter--;
      x -= width;
    }
    matrix.write(); // выведим значения на матрицу

    delay(wait);
  }
}
//Конец функции для вывода информации на дисплей

//Функция для считывания строковой переменной
String Serial_Read() {
  unsigned char c;                        // переменная для чтения сериал порта
  String Serial_string = "";              // Формируемая из символов строка
  while (Serial.available() > 0) {        // Если в сериал порту есть символы
    c = Serial.read();                    // Читаем символ
    if (c == '\n') {                      // Если это конец строки
      return Serial_string;               // Возвращаем строку
    }
    Serial_string = Serial_string + String(char(c)); //Добавить символ в строку
  }
  return Serial_string;
}
//Конец функции для считывания строковой переменной

// Функция перекодировки русских букв из UTF-8 в Win-1251
String utf8rus(String source)
{
  int i, k;
  String target;
  unsigned char n;
  char m[2] = { '0', '\0' };
  k = source.length(); i = 0;
  while (i < k) {
    n = source[i]; i++;
    if (n >= 0xC0) {
      switch (n) {
        case 0xD0: {
            n = source[i]; i++;
            if (n == 0x81) {
              n = 0xA8;
              break;
            }
            if (n >= 0x90 && n <= 0xBF) n = n + 0x2F;
            break;
          }
        case 0xD1: {
            n = source[i]; i++;
            if (n == 0x91) {
              n = 0xB7;
              break;
            }
            if (n >= 0x80 && n <= 0x8F) n = n + 0x6F;
            break;
          }
      }
    }
    m[0] = n; target = target + String(m);
  }
```

```
300   return target;
301 }
302 //Конец функции перекодировки русских букв
```