001 //Кардаш Степан Павлович

002 //Группа 150501

003 //Тема: микропроцессорное устройство контроля концентрации метана

004

005 #include <stDHT.h> //библиотека для работы с датчиком серии DHT

006 #include <Wire.h> //библиотека для работы с интерфейсов I2C

007 #include <LiquidCrystal\_I2C.h> //библиотека подключения дисплея через I2C

008 #include <TroykaMQ.h> //библиотека для работы с датчиками серии MQ

009 #include <EEPROM.h> //библиотека для работы с памятью

010

011 //номер вывода на плате для подключения кнопки перехода в меню

012 #define MENU\_BUTTON\_PIN 2

013 //номер вывода на плате для подключения синего светодиода

014 #define BLUE\_LED\_PIN 4

015 //номер вывода на плате для подключения желтого светодиода

016 #define YELLOW\_LED\_PIN 5

017 //номер вывода на плате для подключения красного светодиода

018 #define RED\_LED\_PIN 6

019 //номер вывода на плате для подключения зуммера

020 #define BUZZER\_PIN 7

021 //номер вывода на плате для подключения датчика температуры и влажности

022 #define DHT22\_SENSOR\_PIN 8

023 //номер вывода на плате для подключения реле

024 #define RELAY\_PIN 9

025 //номер вывода на плате для подключения резистивной клавиатуры

026 #define ANALOG\_KEYBOARD\_PIN A0

027 //номер вывода на плате для подключения датчика углекислого газа

028 #define MQ135\_SENSOR\_PIN A1

029 //номер вывода на плате для подключения датчика метана

030 #define MQ4\_SENSOR\_PIN A2

031 //время прогрева датчиков газа, равное 60с

032 #define WARMING\_UP\_TIME 60000

033 //временной интервал для вывода прогресса прогревания, равный 3с

034 #define WARMING\_UP\_INTERVAL 3000

035 //временной интервал для опроса датчиков, равный 2с

036 #define SENSOR\_POLLING\_INTERVAL 2000

037 //допустимое отклонение от значения АЦП при анализе нажатой кнопки

038 #define ERROR 20

039 //количество аналоговых кнопок

040 #define ANALOG\_BUTTONS\_NUM 4

041

042 //объявляем объект библиотеки, указывая параметры дисплея

043 LiquidCrystal\_I2C display(0x27, 20,4);

044 //объявляем объект библиотеки, указывая модель датчика температуры

045 DHT sensorDHT22(DHT22);

046 //объявляем объект библиотеки, указывая пин подключения к arduino

047 MQ4 sensorMQ4(MQ4\_SENSOR\_PIN);

048 //объявляем объект библиотеки, указывая пин подключения к arduino

049 MQ135 sensorMQ135(MQ135\_SENSOR\_PIN);

050 //переменная-флаг входа в меню

051 volatile int menuFlag = 0;

052 //переменная, запоминающая время (в мс) с момента запуска системы

053 unsigned long time = 0;

054 //переменная, храненящая температуру воздуха

055 int temperature = 0;

056 //переменная, храненящая влажность воздуха

057 int humidity = 0;

058 //переменначя, храненящая концентрации CH4 в ppm

059 unsigned long concentrationCH4 = 0;

060 //переменначя, храненящая концентрации CO2 в ppm

061 unsigned long concentrationCO2 = 0;

062 //максимальное допустимое значение температуры в градусах Цельсия

063 int maxTemperature = 25;

064 //максимальное допустимое значение влажности воздуха в процентах

065 int maxHumidity = 60;

066 //максимальное допустимое значение CH4 в ppm

067 unsigned long maxCH4Concentration = 4000;

068 //максимальное допустимое значение CO2 в ppm

069 unsigned long maxCO2Concentration = 2000;

070 //режим запускаемой сигнализации

071 byte signalingFlags = 0;

072 //значения АЦП для при нажатии на 1 из 4 кнопок

073 int values[4] = {254, 498, 774, 1023};

074

075 //Обработчик 0-ого прерывания. Изменяет значение флага меню

076 int changeMenuFlag(){

077 menuFlag = (menuFlag == 1) ? 0 : 1;

078 }

079

080 //опрделение номера кнопки по значению АЦП

081 int buttonNumber(int value){

082 int i = 0; //переменная цикла, прохода по массиву допустимых АЦП

083 //цикл прохода по массиву возможных значений АЦП

084 for(; i < ANALOG\_BUTTONS\_NUM; i++){

085 //проверка значения из массива АЦП с учетом погрешности

086 if((value <= (values[i] + ERROR)) && (value >= (values[i] - ERROR))){

087 return i + 1; //вернуть номер кнопки

088 }

089 }

090 return -1; //нет нажатых кнопок, вернуть -1

091 }

092

093 //функция проверки нажатия кнопок на клавиатуре

094 int analogKeyBoard(){

095 //счетчик для определения дастоверности нажатия кнопки

096 static int count;

097 //старое значение аналоговой клавиатуры

098 static int oldKeyValue;

099 //для хранения нажатой клавиши на протяжении 10 проходов

100 static int innerKeyValue;

101 //получить текущую нажатую кнопку

102 int actualKeyValue = buttonNumber(analogRead(ANALOG\_KEYBOARD\_PIN));

103 //проверяем действильно ли нажато кнопка путем 10 проходов по алгоритму

104 if(innerKeyValue != actualKeyValue){

105 count = 0; //обнуление счетчика(все снчала)

106 //обновить текущее значения полученное с обработчика клавитуры

107 innerKeyValue = actualKeyValue;

108 }

109 else{

110 count++; //увеличить счетчик проходов

111 }

112 //кнопка нажата и старое значение не совпадает с актуальным

113 if((count >= 10) && (oldKeyValue != actualKeyValue)){

114 oldKeyValue = actualKeyValue; //запоминаем новое значение

115 }

116 return oldKeyValue;

117 }

118

119 //фцнкция прогрева и калибровки датчиков

120 void gasSensorsCalibration(){

121 unsigned long progressTime = 0; //для получения промежутков по 3с в цикле

122 int position = 0; //указатель позиции на дисплее (степень прогрева)

123 //подать напряжение на синий светодиод (начало прогрева датчиков)

124 digitalWrite(BLUE\_LED\_PIN, HIGH);

125 display.clear(); //очистить дисплей от символов

126 display.setCursor(1,0); //установить курсор на позицию (0, 0)

127 display.print("WARMING UP SENSORS"); //вывести надпись "warming up sensors"

128 display.setCursor(6, 2); //установить курсор на позицию (6, 2)

129 display.print("PROGRESS"); //вывести надпись "progress"

130 time = millis(); //получить текущеее время работы

131 //цикл задержки на 60с, время прогрева датчиков газа

132 while((millis() - time) < WARMING\_UP\_TIME){

133 //каждые три секунды обновлять поле progress

134 if((millis() - progressTime) > WARMING\_UP\_INTERVAL){

135 progressTime = millis(); //получить текущеее время работы

136 //установить курсор на позицию (position, 3)

137 display.setCursor(position++, 3);

138 display.print("#"); //вывести символ '#'

139 }

140 }

141 sensorMQ4.calibrate(); //выполнить калибровку датчика MQ4 на чистом воздухе

142 //выполнить калибровку датчика MQ135 на чистом воздухе

143 sensorMQ135.calibrate();

144 digitalWrite(BLUE\_LED\_PIN, LOW); //снять напряжения с синего светодиода

145 display.clear(); //очистить дисплей от символов

146 }

147

148 //функция чтения показаний с датчиков

149 void readDataFromSensors(){

150 //снять значение температуры с датчика

151 temperature = sensorDHT22.readTemperature(DHT22\_SENSOR\_PIN);

152 //снять значение влажности с датчика

153 humidity = sensorDHT22.readHumidity(DHT22\_SENSOR\_PIN);

154 //снять значение концентрации метана

155 concentrationCH4 = sensorMQ4.readMethane();

156 //снять значение концентрации углекислого газа

157 concentrationCO2 = sensorMQ135.readCO2();

158 }

159

160 //вывод информации на дисплей

161 void printInfo(unsigned long valueCH4, unsigned long valueCO2, int valueTemp,

162 int valueHum, int linePointer){

163 //если не в меню, то вывод с нулевой позиции, иначе с первой

164 int startPosition = (linePointer == -1) ? 0 : 1;

165 display.clear(); //очистить дисплей

166 //установить курсор на позицию (startPosition, 0)

167 display.setCursor(startPosition, 0);

168 display.print("CH4: "); //вывести название параметра

169 display.print(valueCH4); //вывести значениие

170 display.print("PPM"); //вывести велечину измерения

171 //установить курсор на позицию (startPosition, 1)

172 display.setCursor(startPosition, 1);

173 display.print("CO2: "); //вывести название параметра

174 display.print(valueCO2); //вывести значениие

175 display.print("PPM"); //вывести велечину измерения

176 //установить курсор на позицию (startPosition, 2)

177 display.setCursor(startPosition, 2);

178 display.print("TEMPERATURE: "); //вывести название параметра

179 display.print(valueTemp); //вывести значениие

180 display.print("C"); //вывести велечину измерения

181 //установить курсор на позицию (startPosition, 3)

182 display.setCursor(startPosition, 3);

183 display.print("HUMIDITY: "); //вывести название параметра

184 display.print(valueHum); //вывести значениие

185 display.print("%"); //вывести велечину измерения

186 //если в режиме меню, то выводить указаетль '>' и '<'

187 if(linePointer != -1){

188 //установить курсор на позицию (0, linePointer)

189 display.setCursor(0, linePointer);

190 display.print(">"); //вывести указатель меню '>'

191 //установить курсор на позицию (19, linePointer)

192 display.setCursor(19, linePointer);

193 display.print("<"); //вывести указатель меню '<'

194 }

195 }

196

197 //функция анализа считанных с датчиков данных

198 void dataAnalysis(){

199 signalingFlags = 0; //сбрасывание всех флагов

200 //проверка на превышение CH4 и CO2

201 if((concentrationCH4 > maxCH4Concentration) ||

202 (concentrationCO2 > maxCO2Concentration)){

203 signalingFlags = signalingFlags | 1; //установить флаг 1

204 }

205 //проверка на превышения температуры и влажности

206 if((temperature > maxTemperature) || (humidity > maxHumidity)){

207 signalingFlags = signalingFlags | (1 << 1); //установить флаг 2

208 }

209 digitalWrite(RED\_LED\_PIN, LOW); //снять питание с красного светодиода

210 digitalWrite(YELLOW\_LED\_PIN, LOW); //снять питание с желтого светодиода

211 digitalWrite(RELAY\_PIN, LOW); //закрыть реле

212 noTone(BUZZER\_PIN); //отключить пъезодинамик

213 //определение типа сигнализирования по установленным флагам

214 switch(signalingFlags){

215 case 1: //превышают только газовые параметры

216 //подать напряжение на красный светодиод

217 digitalWrite(RED\_LED\_PIN, HIGH);

218 digitalWrite(RELAY\_PIN, HIGH); //открыть реле

219 tone(BUZZER\_PIN, 1500); //включить пъезодинамик

220 break;

221 case 2: //превышает только температура или влажность

222 //подать напряжение на желтый светодиод

223 digitalWrite(YELLOW\_LED\_PIN, HIGH);

224 tone(BUZZER\_PIN, 1700); //включить пъезодинамик

225 break;

226 case 3: //превышает температура или влажность, а также газовые параметры

227 //подать напряжение на красный светодиод

228 digitalWrite(RED\_LED\_PIN, HIGH);

229 //подать напряжение на желтый светодиод

230 digitalWrite(YELLOW\_LED\_PIN, HIGH);

231 digitalWrite(RELAY\_PIN, HIGH); //открыть реле

232 tone(BUZZER\_PIN, 1900); //включить пъезодинамик

233 break;

234 }

235 }

236

237 //основное меню

238 void menuFunc(){

239 int menuPointer = 0; //указатель на пункт меню

240 int currentKeyValue = 0; //значение нажатой кнопки

241 int flag = 0; //флаг для определения момента отпускания кнопки

242 //вывести меню с указателеме '> ... <' на дисплей

243 printInfo(maxCH4Concentration, maxCO2Concentration, maxTemperature,

244 maxHumidity, menuPointer);

245 while(menuFlag == 1){ //если была нажата кнопка входа в меню

246 currentKeyValue = analogKeyBoard(); //чтение значения с клавиатуры

247 if((currentKeyValue != -1) && (flag == 0)){ //обработчик нажатия кнопки

248 flag = 1; //кнопка нажата

249 if(currentKeyValue == 1){ //если нажата 1 кнопка

250 menuPointer++; //переход к следующему пункту меню

251 //когда дошло до конца меню сброс в начало

252 menuPointer = (menuPointer == 4) ? 0 : menuPointer;

253 }

254 else if(currentKeyValue == 2){ //если нажата 2 кнопка

255 menuPointer--; //переход к предыдущему пункту меню

256 //когда дошло до начала меню сброс в конец

257 menuPointer = (menuPointer == -1) ? 3 : menuPointer;

258 }

259 else if(currentKeyValue == 3){ //если нажата 3 кнопка

260 //в зависимости от пункта меню уменьшаем максимальное значение

261 switch(menuPointer){

262 case 0: //menuPointer указывает на максимальную концентрацию CH4

263 maxCH4Concentration -= 25;

264 break;

265 case 1: //menuPointer указывает на максимальную концентрацию CO2

266 maxCO2Concentration -= 25;

267 break;

268 case 2: //menuPointer указывает на максимальную температуру

269 maxTemperature -= 1;

270 break;

271 case 3: //menuPointer указывает на максимальную влажность

272 maxHumidity -= 1;

273 break;

274 }

275 }

276 else if(currentKeyValue == 4){ //если нажата 4 кнопка

277 //в зависимости от пункта меню увеличиваем максимальное значение

278 switch(menuPointer){

279 case 0: //menuPointer указывает на максимальную концентрацию CH4

280 maxCH4Concentration += 25;

281 break;

282 case 1: //menuPointer указывает на максимальную концентрацию CO2

283 maxCO2Concentration += 25;

284 break;

285

286 case 2: //menuPointer указывает на максимальную температуру

287 maxTemperature += 1;

288 break;

289 case 3: //menuPointer указывает на максимальную влажность

290 maxHumidity += 1;

291 break;

292 }

293 }

294 }

295 //обработчик отпускания кнопки

296 if((currentKeyValue == -1) && (flag == 1)){

297 flag = 0; //кнопки не нажаты

298 //обновить меню на дисплее

299 printInfo(maxCH4Concentration, maxCO2Concentration, maxTemperature,

300 maxHumidity, menuPointer);

301 }

302 }

303 }

304

305 //чтения данных из eeprom (инициализация максимальных значений)

306 void initVariablesFromEEPROM(){

307 //считать данные в maxCH4Concentration из памяти

308 EEPROM.get(0, maxCH4Concentration);

309 //считать данные в maxCO2Concentration из памяти

310 EEPROM.get(4, maxCO2Concentration);

311 //считать данные в maxTemperature из памяти

312 EEPROM.get(8, maxTemperature);

313 //считать данные в maxHumidity из памяти

314 EEPROM.get(10, maxHumidity);

315 }

316

317 //запись данных в eeprom

318 void saveVariablesIntoEEPROM(){

319 //записать данные из maxCH4Concentration в память

320 EEPROM.put(0, maxCH4Concentration);

321 //записать данные из maxCO2Concentration в память

322 EEPROM.put(4, maxCO2Concentration);

323 //записать данные из maxTemperature в память

324 EEPROM.put(8, maxTemperature);

325 //записать данные из maxHumidity в память

326 EEPROM.put(10, maxHumidity);

327 }

328

329 //задание начальных параметров

330 void setup() {

331 //установить режим вывода на прием сигнала

332 pinMode(MENU\_BUTTON\_PIN, INPUT);

333 //установить режим вывода на выдачу сигнала

334 pinMode(BLUE\_LED\_PIN, OUTPUT);

335 //установить режим вывода на выдачу сигнала

336 pinMode(YELLOW\_LED\_PIN, OUTPUT);

337 //установить режим вывода на выдачу сигнала

338 pinMode(RED\_LED\_PIN, OUTPUT);

339 //установить режим вывода на выдачу сигнала

340 pinMode(BUZZER\_PIN, OUTPUT);

341 //установить режим вывода на прием сигнала

342 pinMode(DHT22\_SENSOR\_PIN, INPUT);

343 //установить режим вывода на выдачу сигнала

344 pinMode(RELAY\_PIN, OUTPUT);

345 //связываем прерывание с обработчиком

346 attachInterrupt(0, changeMenuFlag, RISING);

347 //инициализация дисплея

348 display.init();

349 //включить подсветку дисплея

350 display.backlight();

351 //инициализировать максимльные значения

352 initVariablesFromEEPROM();

353 //прогрев и калибровка датчиков

354 gasSensorsCalibration();

355 }

356

357 //основной цикл программы

358 void loop() {

359 //если прошло 2 секунды опрос датчиков

360 if((millis() - time) > SENSOR\_POLLING\_INTERVAL){

361 time = millis(); //обновить время работы контроллера

362 readDataFromSensors(); //получить данные с датчиков

363 //вывести информацию на дисплей с обновленными параметрами

364 printInfo(concentrationCH4, concentrationCO2, temperature, humidity, -1);

365 dataAnalysis(); //анализ полученных данных

366 }

367 if(menuFlag == 1){ //зайти в меню

368 menuFunc();

369 saveVariablesIntoEEPROM(); //запимсать значения в память

370 }

371 }

372