



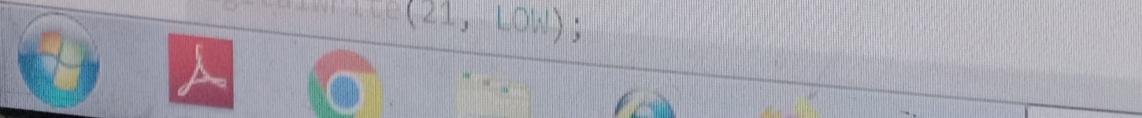
```
30
31 #include <WiFi.h>
32 #include <WiFiClient.h>
33 #include <BlynkSimpleEsp32.h>
34 #include "DHTesp.h"
35
36
37 BlynkTimer timer;
38
39
40 char auth[] = BLYNK_AUTH_TOKEN;
41
42 char ssid[] = "Wokwi-GUEST";
43 char pass[] = "";
44 int val = 0, va1,va2,va3,va4,va5,ge, t =15 ;           I
45 float tmp,hum = 0;
46
47 int ledPin = 33;
48 int inputPin = 27;
49 int pirState,k;
50 int v = 0;
51
52
53 //temp symbol
54 byte t1[8]={B00000, B00001, B00010, B00100, B00100, B00100, B00100, B00111,};
55 byte t2[8]={B00111, B00111, B00111, B01111,B11111, B11111, B01111, B00011,};
56 byte t3[8]={B00000, B10000, B01011, B00100, B00111, B00100, B00111, B11100,};
57 byte t4[8]={B11111, B11100, B11100,B11111, B11111, B11110, B11000,};
58
59 //humidity symbol
```

[sketch.ino](#)[diagram.json](#)[libraries.txt](#)[Library Manager](#) ▾

```
59 //humidity symbol
60 byte hum1[8]={B00000, B00001, B00011, B00011,B00111, B01111, B01111, B11111,};
61 byte hum2[8]={B11111, B11111, B11111, B01111,B00011, B00000, B00000, B00000,};
62 byte hum3[8]={B00000, B10000, B11000, B11000, B11100, B11110, B11110, B11111,};
63 byte hum4[8]={B11111, B11111, B11111, B11110, B11100, B00000, B00000, B00000,};
64
65 //Home Symbol
66 byte house1[8]={B00000, B00001, B00011, B00011, B00111, B01111, B01111, B11111,};
67 byte house2[8]={B11111, B11111, B11100, B11100, B11100, B11100, B11100, B11100,};
68 byte house3[8]={B00000, B10010, B11010, B11010, B11110, B11110, B11110, B11111,};
69 byte house4[8]={B11111, B11111, B11111, B10001, B10001, B10001, B11111, B11111,};
70
71 byte d[8] = { 0b00011,0b00011,0b00000,0b00000,0b00000,0b00000,0b00000,0b00000
72
73 byte Lck[] = { B01110, B10001, B10001, B11111, B11011, B11011, B11111, B00000
74
75
76
77 DHTesp temps;
78
79 BLYNK_WRITE(V0){
80     va1 = param.asInt();
81     digitalWrite(5, va1);
82
83 }
84 BLYNK_WRITE(V1){
85     va2 = param.asInt();
86     digitalWrite(18, va2);
87 }
88 }
```

```
89     BLYNK_WRITE(V2){
90         va3 = param.asInt();
91         digitalWrite(19, va3);
92     }
93     BLYNK_WRITE(V3){
94         va4 = param.asInt();
95         digitalWrite(4, va4);
96     }
97
98     BLYNK_WRITE(V4){
99         va5 = param.asInt();
100        digitalWrite(2, va5);
101    }
102
103    BLYNK_WRITE(V7) {
104        pirState = param.asInt();
105        if(pirState == 0){
106            digitalWrite(ledPin, LOW);
107            k = 1;
108            ge = 0;
109        }
110        else {
111            digitalWrite(ledPin, HIGH);
112            k= 0;
113            ge = 1;
114        }
115    }
116
117    void myTimer()
```

```
118 {
119     Blynk.virtualWrite(V5,tmp);
120     Blynk.virtualWrite(V6,hum);
121 }
122
123
124 void setup()
125 {
126
127     Serial.begin(115200);
128     Blynk.begin(auth, ssid, pass);
129
130     pinMode(5, OUTPUT);
131     pinMode(18, OUTPUT);
132     pinMode(19, OUTPUT);
133     pinMode(4, OUTPUT);
134     pinMode(23, INPUT);
135     pinMode(2,OUTPUT);
136     temps.setup(t, DHTesp::DHT22);
137     pinMode(ledPin, OUTPUT);
138     pinMode(inputPin, INPUT_PULLUP);
139
140     lcd.init();
141     lcd.backlight();
142
143     digitalWrite(5, LOW);
144     digitalWrite(18, LOW);
145     digitalWrite(19, LOW);
146     digitalWrite(21, LOW);
```



```
148
149 lcd.setCursor(0,0);
150 lcd.print("CircuitDesignContest");
151 lcd.setCursor(8,1);
152 lcd.print("2022");
153 lcd.setCursor(0,2);
154 lcd.print("-----");
155 lcd.setCursor(9,3);
156 lcd.print("- eDiYLaBs");
157 delay(3000);
158 lcd.clear();
159 lcd.createChar(6, Lck);
160 lcd.createChar(1,house1);
161 lcd.createChar(2,house2);
162 lcd.createChar(3,house3);
163 lcd.createChar(4,house4);
164 lcd.setCursor(1,2);
165 lcd.write(1);
166 lcd.setCursor(1,3);
167 lcd.write(2);
168 lcd.setCursor(2,2);
169 lcd.write(3);
170 lcd.setCursor(2,3);
171 lcd.write(4);
172
173 lcd.setCursor(17,2);
174 lcd.write(1);
175 lcd.setCursor(17,3);
176 lcd.write(2);
177 lcd.setCursor(18,2);
178 lcd.write(2);
```

```
180     lcd.write(4);
181
182
183     lcd.setCursor(19,0);
184     lcd.write(6);
185     lcd.setCursor(9,0);
186     lcd.print("connected-");
187     lcd.setCursor(2,1);
188     lcd.print("HOME AUTOMATION");
189     lcd.setCursor(6,2);
190     lcd.print("USING IOT");
191     delay(3000);
192
193     Blynk.virtualWrite(V7, pirState);
194     timer.setInterval(1000L, myTimer);
195
196 }
197
198
199 void loop()
200 {
201     Blynk.run();
202     timer.run();
203     val = digitalRead(23);
204     if(val == 1)
205     {
206         digitalWrite(2,va5);
207     }
208
209 else{
```

```
210     |     |     digitalWrite(2,LOW);
211 }
212
213 TempAndHumidity x = temps.getTempAndHumidity();
214 tmp = x.temperature ;
215 hum = x.humidity ;
216
217 v = digitalRead(inputPin);
218 if (v == HIGH) {
219     if (k == 1)  {
220         digitalWrite(ledPin, LOW);
221         k = 0 ;
222         ge = 0;
223     }
224     else if (k == 0)  {
225         digitalWrite(ledPin, HIGH);
226         k = 1;
227         ge = 1;
228     }
229 }
230
231
232 if (val == 1){
233     lcd.clear();
234     lcd.setCursor(19,0);
235     lcd.write(6);
236     lcd.setCursor(0, 1);
237     lcd.print("SW_1= ");
238     lcd.print("ON ");
```

```
238     lcd.print("ON ");
239 }
240 else{
241     lcd.clear();
242     lcd.setCursor(19,0);
243     lcd.write(6);
244     lcd.setCursor(0, 1);
245     lcd.print("SW_1= ");
246     lcd.print("OFF");
247 }
248 if (va2 == 1){
249
250
251     lcd.setCursor(11, 1);
252     lcd.print("SW_2= ");
253     lcd.print("ON ");
254 }
255 else{
256     lcd.setCursor(11, 1);
257     lcd.print("SW_2= ");
258     lcd.print("OFF");
259 }
260 if (va3 == 1){
261
262     lcd.setCursor(0, 2);
263     lcd.print("SW_3= ");
264     lcd.print("ON ");
265 }
266 else{
267 }
```

```
267     |     lcd.setCursor(0, 2);
268     |     lcd.print("SW_3= ");
269     |     lcd.print("OFF");
270     |
271     }
272     if (va4 == 1){
273
274     lcd.setCursor(11, 2);
275     lcd.print("SW_4= ");
276     lcd.print("ON ");
277     }
278     else{
279
280     lcd.setCursor(11, 2);
281     lcd.print("SW_4= ");
282     lcd.print("OFF");
283     }
284     if (va5 == 1){
285
286     lcd.setCursor(0, 3);
287     lcd.print("OD_L= ");
288     lcd.print("ON ");
289     }
290     else{
291
292     lcd.setCursor(0, 3);
293     lcd.print("OD_L= ");
294     lcd.print("OFF");
295     }
296     if (ge == 1){
```

```
296     if (ge == 1){  
297  
298         lcd.setCursor(11, 3);  
299         lcd.print("WR_L= ");  
300         lcd.print("ON ");  
301     }  
302     else{  
303         lcd.setCursor(11, 3);  
304         lcd.print("WR_L= ");  
305         lcd.print("OFF");  
306     }  
307     delay(1500);  
308  
309     lcd.clear();  
310     lcd.createChar(1,t1);  
311     lcd.createChar(2,t2);  
312     lcd.createChar(3,t3);  
313     lcd.createChar(4,t4);  
314     lcd.createChar(5, d);  
315     lcd.createChar(6, Lck);  
316  
317     lcd.setCursor(19,0);  
318     lcd.write(6);  
319     lcd.setCursor(1,1);  
320     lcd.write(1);  
321     lcd.setCursor(1,2);  
322     lcd.write(2);  
323     lcd.setCursor(2,1);  
324     lcd.write(3);  
325
```

```
325     lcd.write(3);
326     lcd.setCursor(2,2);
327     lcd.write(4);
328     lcd.setCursor(4,1);
329     lcd.print("Temperature :");
330     lcd.setCursor(7,2);
331     lcd.print(tmp);
332     lcd.setCursor(11,2);
333     lcd.write(5);
334     lcd.setCursor(12,2);
335     lcd.print("C");
336
337     delay(750);
338     lcd.clear();
339
340     lcd.createChar(1,hum1);
341     lcd.createChar(2,hum2);
342     lcd.createChar(3,hum3);
343     lcd.createChar(4,hum4);
344
345     lcd.setCursor(19,0);
346     lcd.write(6);
347     lcd.setCursor(3,1);
348     lcd.write(1);
349     lcd.setCursor(3,2);
350     lcd.write(2);
351     lcd.setCursor(4,1);
352     lcd.write(3);
353     lcd.setCursor(4,2);
354     lcd.write(4);
```



sketch.ino diagram.json libraries.txt Library Manager ▾

```
348     lcd.write(1);
349     lcd.setCursor(3,2);
350     lcd.write(2);
351     lcd.setCursor(4,1);
352     lcd.write(3);
353     lcd.setCursor(4,2);
354     lcd.write(4);
355     lcd.setCursor(6,1);
356     lcd.print("Humidity :");
357     lcd.setCursor(7,2);
358     lcd.print(hum);
359     lcd.setCursor(12,2);
360     lcd.print("%");
361     delay(750);
362
363
364 }
```

Simulation

Code

