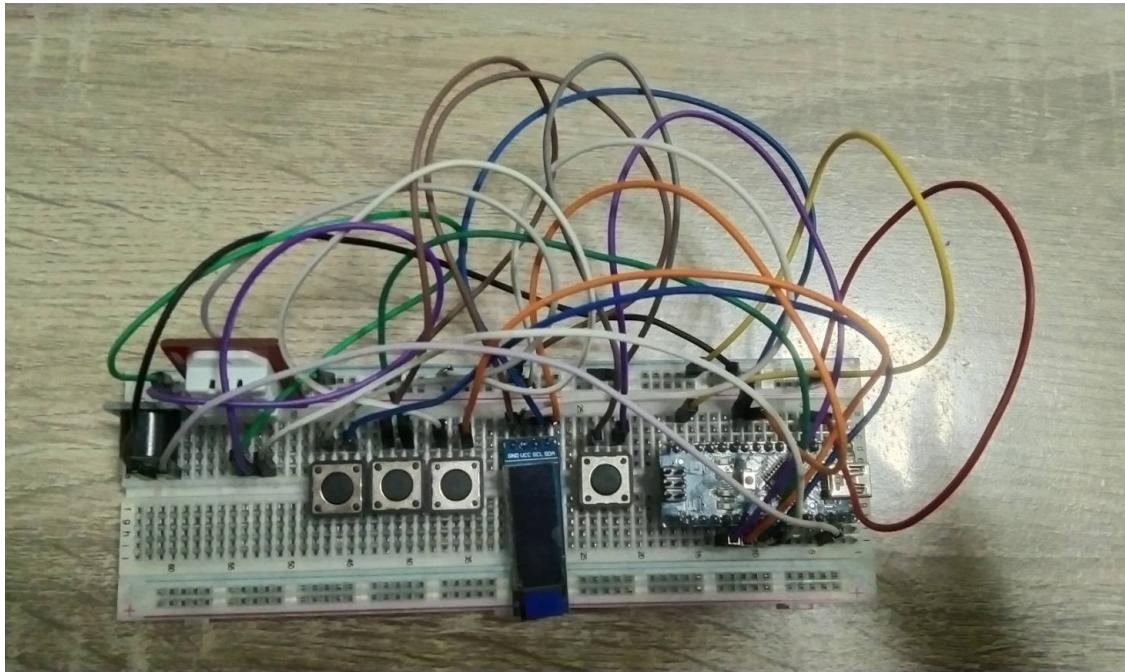


Assignment 7 mini clock



กลุ่ม: ขอบเขต ไปก่อนละกัน

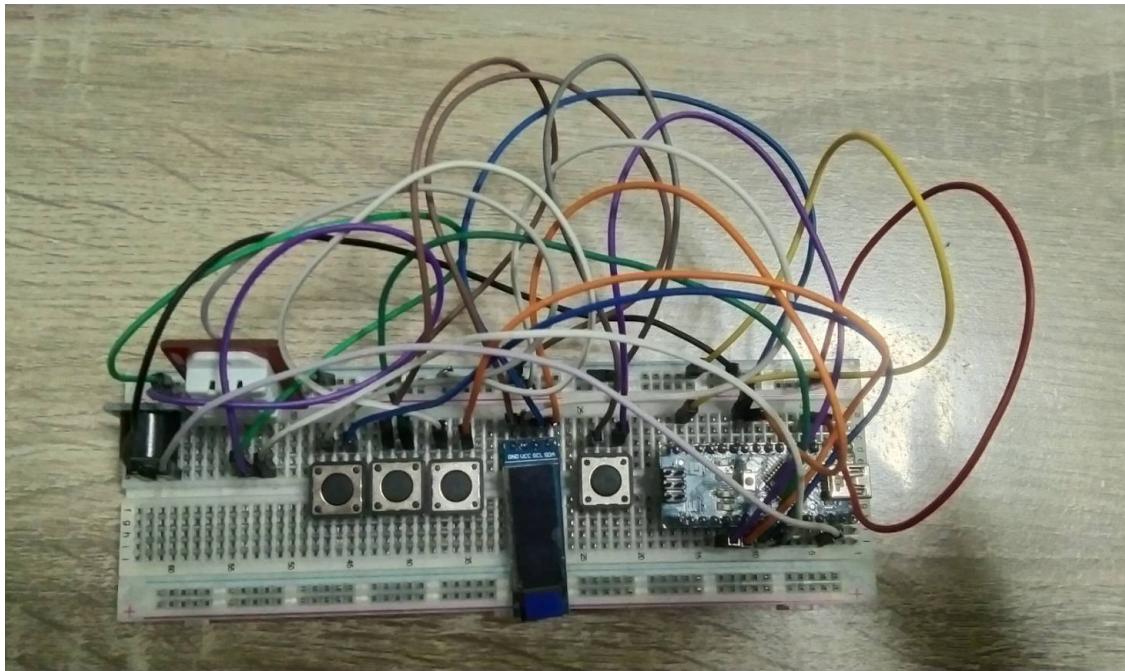
รายชื่อสมาชิก

1. นางสาว รัชฎา จรุงพัฒนาณนท์ 64010359
2. นาย ธนาทัต จงกิจติสกุล 64010312

แนวคิดการออกแบบ

กลุ่มของพวกร่างทำ Mini Clock โดยแนวคิดก็คือถ่ายทอดน้ำพิกาทั่วไป ซึ่งจะมีหมุดปักติกคือใช้เวลาปักติได้โดยสามารถบอกรูปเป็น AM/PM ได้ด้วย และยังมีหมุดตั้งเวลาเริ่มต้น หมุดน้ำพิกาปลูก หมุดจับเวลา แต่เรามีฟังก์ชันเสริม ก็คือ Mini Clock ของพวกร่างสามารถวัดอุณหภูมิและความชื้นได้ เอาไว้ดูเล่นๆได้ด้วย

วงจร



การใช้งานโดยย่อ



ปุ่มในการกดมีทั้งหมด 4 ปุ่ม ได้แก่

[mode]

[save]

[up]

[down]

Mode ทั้งหมด

- mode แสดงผลปกติ(HOME)
- mode setup เวลา
- mode Alarm
- mode stopwatch
- mode show temperature and humidity

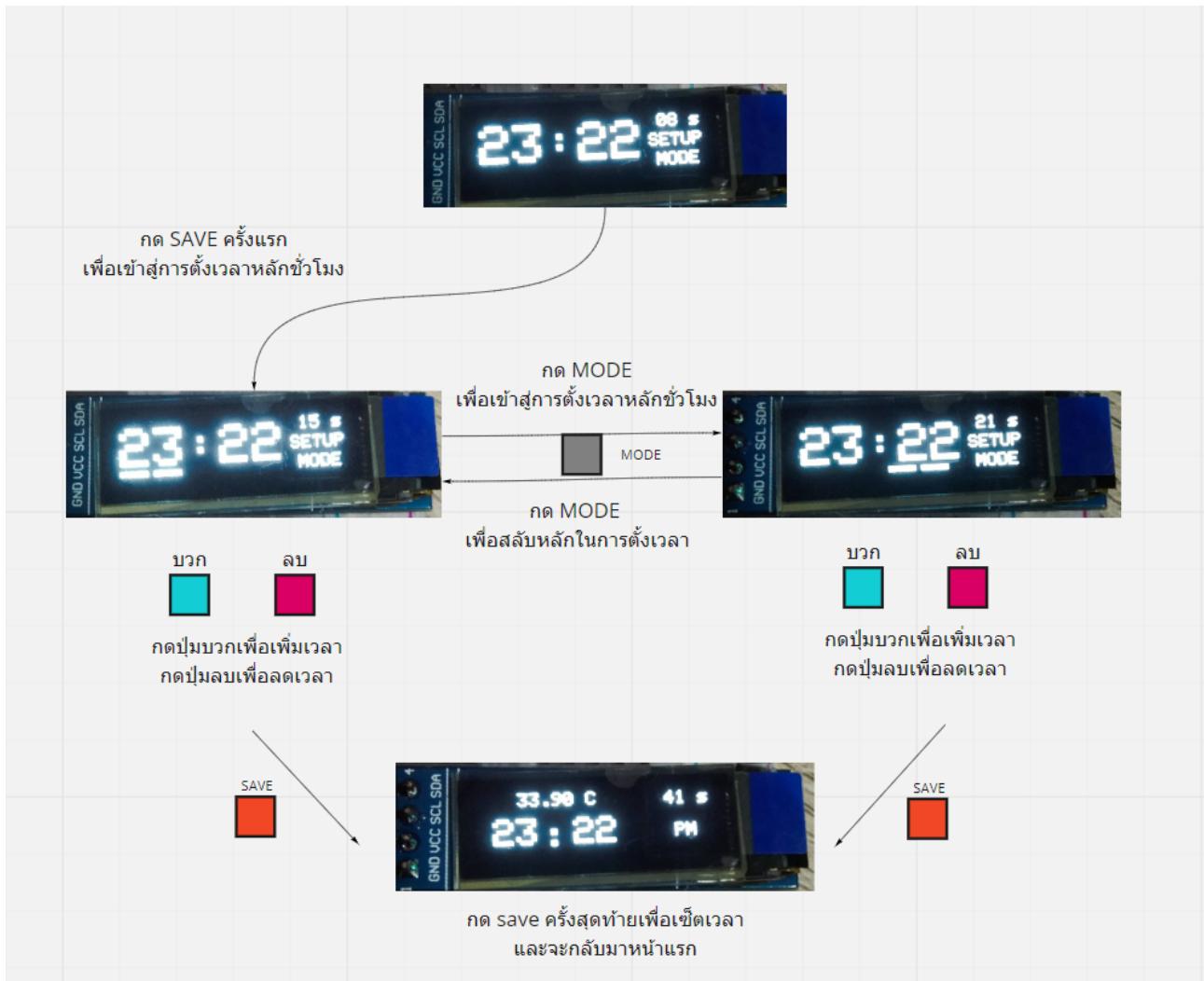
Mode 1 HOME



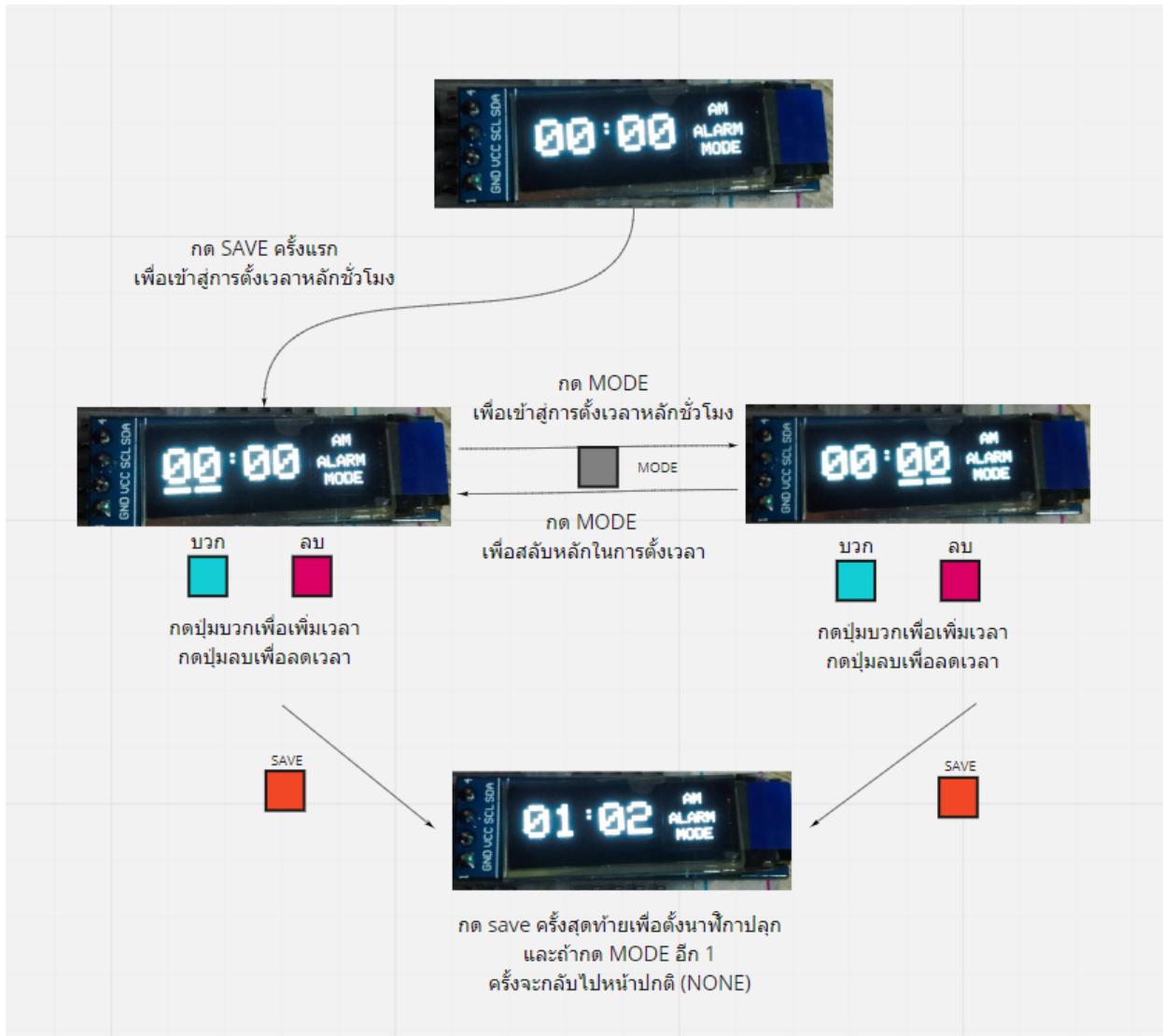
ช่วง 00 - 12 แสดงผล AM ช่วง 13-23 แสดงผล PM

ถ้ากด SAVE 1 ที จะเซฟเวลาลง EEPROM

Mode 2 setup เวลา

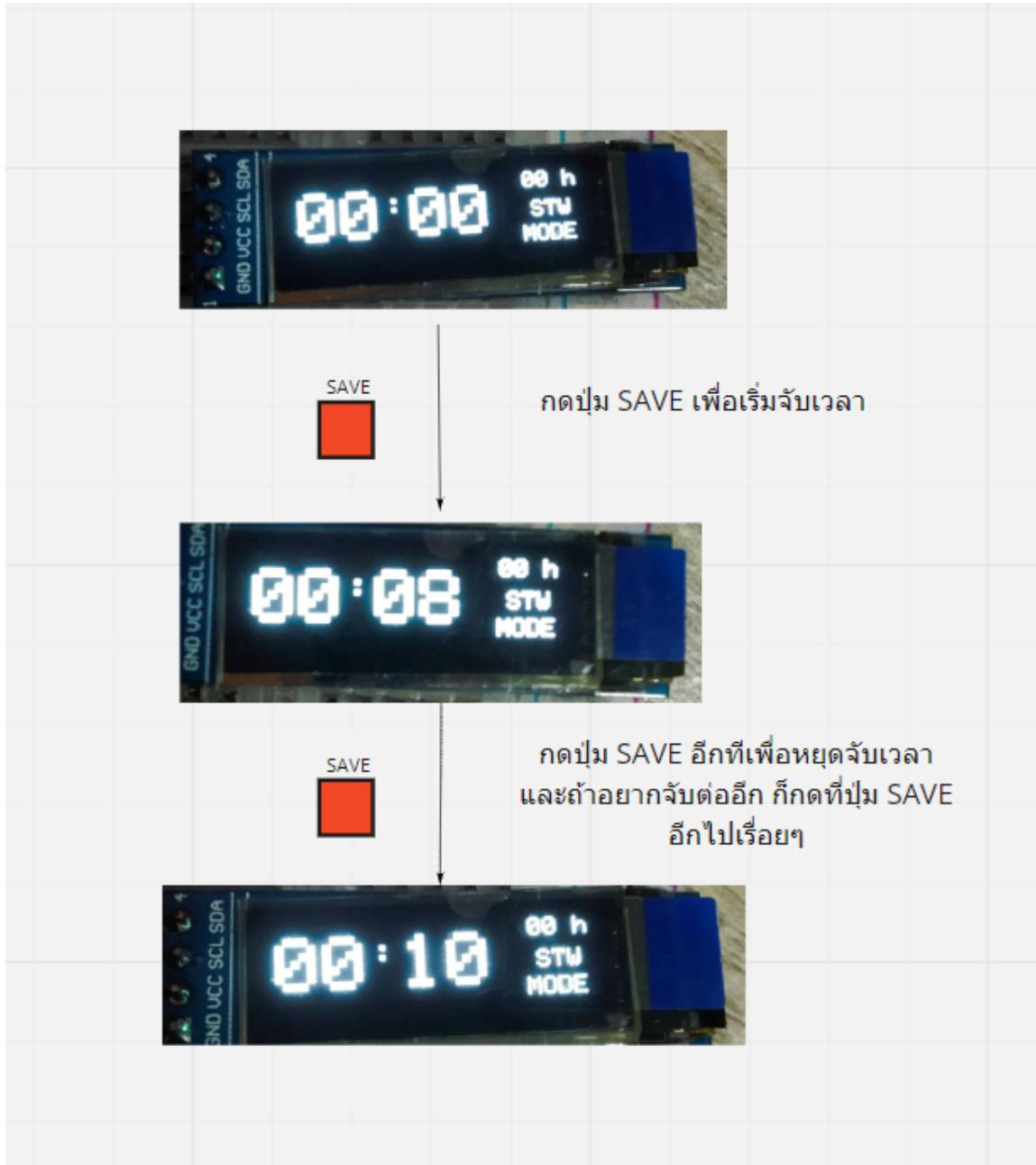


Mode 3 Alarm



ถ้า ถึงเวลาปลุกเสียง ลำโพงจะดัง กดปุ่ม [save] เพื่อ ปิดนาฬิกาปลุก

Mode 4 Stop Watch



Mode 5 show temperature and humidity



โปรแกรมและอธิบายโปรแกรมโดยย่อ

หลักการทำงาน :

- มีการ save เวลาลง EEPROM อยู่ 3 แบบ คือ ทุก 1นาที , กด save ที่หน้า home และ กด save ตอน setting mode Alarm เรียบร้อยแล้ว
- โปรแกรม การทำงาน จะ ໄล่ ตาม state ต่างๆ ถ้าเรากดปุ่ม mode เริ่มจาก state แรก ที่ Done และก็จะวน กลับมาที่ Done

ฟังก์ชัน :

function	start (Line)	end (Line)	การทำงาน
Speaker()	56	62	ฟังก์ชันเก็บเสียงนาฬิกาปลุก
sendSensor()	63	67	วัด อุณหภูมิ และ ความชื้น โดย DHT 22
plusOneSec (Time &time)	68	76	ใช้ในการตั้งค่าเริ่มต้นจาก วินาที ชั่วโมง เพื่อใช้ในการนับเวลา
TimeRuning()	77	84	นับเวลา
showTime()	85	119	แสดงผลเวลา
showTimeSetup()	120	148	แสดงผลหน้า setup เวลา
AlarmWarning()	149	159	แจ้งเตือนนาฬิกาปลุก
showMode (String str)	159	168	แสดงผลชื่อ mode
showAlarm()	169	200	แสดงผล Alrm mode
showSTW()	201	236	แสดงผล stop watch mode
dayMode()	237.	251	แสดง PM AM
showTemp()	252	267	แสดงผล อุณหภูมิและความชื้น
timeTemp()	268	275	แสดงผลอุณหภูมิหน้าแรก(Home)



function	start (Line)	end (Line)	การทำงาน
settingSTU(int x)	276	281	แสดงผลตัว __ ให้ตัวเลขใน setting
pressButton(int pin, void (*callback)())	282	292	ฟังก์ชันในการ debounce
modeSwPressed()	320	353	ฟังก์ชันในการกด mode มีฟังก์ชันของแต่ละ state อุปกรณ์ใน
saveSwPressed()	354	413	ฟังก์ชันในการกด save มีฟังก์ชันของแต่ละ state อุปกรณ์ใน
upSwPressed()	414	439	ฟังก์ชันในการกด เพิ่ม เลขในหน้า setting
downSwPressed()	440	477	ฟังก์ชันในการกด ลด เลขในหน้า setting

Code :

//copy ໄຊ້ອັນປ

	code
	<pre> 1 #include <EEPROM.h> 2 #include <Wire.h> 3 #include <Adafruit_GFX.h> 4 #include <Adafruit_SSD1306.h> 5 #include <TimerOne.h> 6 #include <DHT.h> 7 #define OLED_RESET -1 8 Adafruit_SSD1306 OLED(OLED_RESET); 9 //DHT 10 11 #define DHTPIN 6 12 #define DHTTYPE DHT22 13 DHT dht(DHTPIN, DHTTYPE); 14 //WARNINGS 15 #define buzzer 11 16 #define DO 261.6 17 #define ME 270 18 //SWITCH 19 #define modeSw 2 20 #define saveSw 3 21 #define upSw 4 22 #define downSw 5 23 int32_t bounce; 24 //STATE 25 #define None 0 26 #define Setup 1 27 #define Alarm 2 28 #define Stopwatch 3 29 #define TempMode 4 30 #define SetupMin 5 31 #define AlarmMin 6 32 int state = 0; 33 //BOOL CLICK 34 bool save = false; 35 bool mode = false; 36 bool up = false; 37 bool down = false; 38 bool setting = false; 39 bool warning = false; 40 bool isSetAlarm = false; 41 42 float temp = dht.readTemperature(); 43 float hum = dht.readHumidity(); 44 45 46 struct Time 47 { 48 int hour; 49 int minute; </pre>

```
50     int sec;
51 };
52 Time currentTime = {0, 0, 0};
53 Time loadTime = {0, 0, 0};
54 Time isAlarm = {0, 0, 0};
55 Time isStw = {0, 0, 0};
56 void Speaker()
57 {
58     tone(buzzer, DO);
59     delay(200);
60     tone(buzzer, ME);
61     delay(200);
62 }
63 void sendSensor()
64 {
65     float temp = dht.readTemperature();
66     float hum = dht.readHumidity();
67 }
68 void plusOneSec(Time &time)
69 {
70     time.sec++;
71     time.minute += time.sec / 60;
72     time.sec %= 60;
73     time.hour += time.minute / 60;
74     time.minute %= 60;
75     time.hour %= 24;
76 }
77 void TimeRuning()
78 {
79     plusOneSec(currentTime);
80     if (state == Stopwatch && setting)
81     {
82         plusOneSec(isStw);
83     }
84 }
85 void showTime()
86 {
87     // HOUR
88     OLED.setCursor(9.5, 15);
89     OLED.setTextSize(2.9999);
90     if (currentTime.hour < 10)
91     {
92         OLED.print("0");
93     }
94     OLED.print(currentTime.hour);
95     // MINUTE
96     OLED.setTextSize(3.00);
97     OLED.print(":");
98     OLED.setTextSize(2.9999);
99     if (currentTime.minute < 10)
100    {
101        OLED.print("0");
102    }
103    OLED.print(currentTime.minute);
104    // SECOND
105    OLED.setCursor(100, 2);
106    OLED.setTextSize(1.00);
```

```
107     if (currentTime.sec < 10)
108     {
109         OLED.print("0");
110     }
111     OLED.print(currentTime.sec);
112     OLED.print(" s");
113
114     //TEMPERATURE
115     OLED.setCursor(22.5, 1);
116     OLED.setTextSize(0.01);
117     OLED.print(dht.readTemperature());
118     OLED.print(" C");
119 }
120 void showTimeSetup()
121 {
122     // HOUR
123     OLED.setCursor(2, 5);
124     OLED.setTextSize(3);
125     if (currentTime.hour < 10)
126     {
127         OLED.print("0");
128     }
129     OLED.print(currentTime.hour);
130     // MINUTE
131     OLED.setTextSize(3.00);
132     OLED.print(":");
133     OLED.setTextSize(3);
134     if (currentTime.minute < 10)
135     {
136         OLED.print("0");
137     }
138     OLED.print(currentTime.minute);
139     // SECOND
140     OLED.setCursor(100, 2);
141     OLED.setTextSize(1.00);
142     if (currentTime.sec < 10)
143     {
144         OLED.print("0");
145     }
146     OLED.print(currentTime.sec);
147     OLED.print(" s");
148 }
149 void AlarmWarning()
150 {
151     if (isAlarm.hour == currentTime.hour && isAlarm.minute == currentTime.minute)
152     {
153         if (state != Alarm || state != AlarmMin )
154         {
155             Speaker();
156         }
157     }
158 }
159 void showMode(String str)
160 {
161     // MODE
162     OLED.setCursor(95, 13);
163     OLED.setTextSize(0.40);
```

```
164     OLED.print(str);
165     OLED.setCursor(99, 24);
166     OLED.setTextSize(0.40);
167     OLED.print("MODE");
168 }
169 void showAlarm()
170 {
171     // HOUR
172     OLED.setCursor(2, 5);
173     OLED.setTextSize(3.00);
174     if (isAlarm.hour < 10)
175     {
176         OLED.print("0");
177     }
178     OLED.print(isAlarm.hour);
179     // MINUTE
180     OLED.setTextSize(2.5);
181     OLED.print(":");
182     OLED.setTextSize(3.00);
183     if (isAlarm.minute < 10)
184     {
185         OLED.print("0");
186     }
187     OLED.print(isAlarm.minute);
188     // AM
189     OLED.setCursor(105, 0.5);
190     OLED.setTextSize(1.00);
191     if (isAlarm.hour >= 0 && isAlarm.hour <= 12)
192     {
193         OLED.print("AM");
194     }
195     // PM
196     if (isAlarm.hour >= 13 && isAlarm.hour <= 23)
197     {
198         OLED.print("PM");
199     }
200 }
201 void showSTW()
202 {
203     // MINUTE
204     OLED.setCursor(2, 5);
205     OLED.setTextSize(3.00);
206     if (isStw.minute < 10)
207     {
208         OLED.print("0");
209     }
210     OLED.print(isStw.minute);
211     // SECOND
212     OLED.setTextSize(2.5);
213     OLED.print(":");
214     OLED.setTextSize(3.00);
215     if (isStw.sec < 10)
216     {
217         OLED.print("0");
218     }
219     OLED.print(isStw.sec);
220     // HOUR
```

```
221     OLED.setCursor(100, 0.5);
222     OLED.setTextSize(1.00);
223     if (isStw.hour < 10)
224     {
225         OLED.print("0");
226     }
227     OLED.print(isStw.hour);
228     OLED.print(" h");
229     // MODE
230     OLED.setCursor(103, 13);
231     OLED.setTextSize(0.40);
232     OLED.print("STW");
233     OLED.setCursor(99, 24);
234     OLED.setTextSize(0.40);
235     OLED.print("MODE");
236 }
237 void dayMode()
238 {
239     // AM
240     OLED.setCursor(105, 20);
241     OLED.setTextSize(1.00);
242     if (currentTime.hour >= 0 && currentTime.hour <= 12)
243     {
244         OLED.print("AM");
245     }
246     // PM
247     if (currentTime.hour >= 13 && currentTime.hour <= 23)
248     {
249         OLED.print("PM");
250     }
251 }
252 void showTemp()
253 {
254     //TEMP
255     OLED.setCursor(22, 2);
256     OLED.setTextSize(2);
257     //OLED.print("Temp :");
258     OLED.print(dht.readTemperature());
259     OLED.print(" C");
260
261     //HUMIDITY
262     OLED.setCursor(24, 25);
263     OLED.setTextSize(1);
264     OLED.print("Hum : ");
265     OLED.print(dht.readHumidity());
266     OLED.print(" %");
267 }
268 void timeTemp()
269 {
270     //TEMPERATURE
271     OLED.setCursor(22.5, 1);
272     OLED.setTextSize(0.01);
273     OLED.print(dht.readTemperature());
274     OLED.print(" C");
275 }
276 void settingSTU(int x)
277 {
```

```
278     OLED.setCursor(x, 11);
279     OLED.setTextSize(3.00);
280     OLED.print("___");
281 }
282 void pressButton(int pin, void (*callback)())
283 {
284     if (!digitalRead(pin))
285     {
286         if (millis() - bounce > 300)
287         {
288             callback();
289         }
290         bounce = millis();
291     }
292 }
293 void setup()
294 {
295     Serial.begin(9600);
296     dht.begin();
297     OLED.begin(SSD1306_SWITCHCAPVCC, 0x3C);
298     OLED.setTextColor(WHITE);
299
300     pinMode(modeSw, INPUT_PULLUP);
301     pinMode(saveSw, INPUT_PULLUP);
302     pinMode(upSw, INPUT_PULLUP);
303     pinMode(downSw, INPUT_PULLUP);
304
305     Timer1.initialize(1000000); // 1s
306     Timer1.attachInterrupt(TimeRunning);
307
308     EEPROM.get(0, loadTime);
309     currentTime.sec = loadTime.sec;
310     currentTime.minute = loadTime.minute;
311     currentTime.hour = loadTime.hour;
312     Serial.println(currentTime.hour);
313     Serial.println(currentTime.minute);
314     Serial.println(currentTime.sec);
315     if (currentTime.hour < 0)
316     {
317         currentTime = {0, 0, 0};
318     }
319 }
320 void modeSwPressed()
321 {
322     if (state == Stopwatch)
323     {
324         setting = false;
325         isStw = {0, 0, 0};
326     }
327
328     if (setting)
329     {
330         if (state == Setup)
331         {
332             state = SetupMin;
333         }
334         else if (state == SetupMin)
```

```

335     {
336         state = Setup;
337     }
338     else if (state == Alarm)
339     {
340         state = AlarmMin;
341     }
342     else if (state == AlarmMin)
343     {
344         state = Alarm;
345     }
346 }
347 else
348 {
349     state++;
350     state %= 5;
351     //สลับ ชม.นาฬิกา
352 }
353 }
354 void saveSwPressed()
355 {
356     if (state == None)
357     {
358         EEPROM.put(0, currentTime);
359         Serial.println(currentTime.hour);
360         Serial.println(currentTime.minute);
361         Serial.println(currentTime.sec);
362         if (isSetAlarm)
363         {
364             noTone(buzzer);
365             isSetAlarm = false;
366         }
367     }
368     else if (state == Setup || state == SetupMin)
369     {
370         if (isSetAlarm)
371         {
372             noTone(buzzer);
373             isSetAlarm = false;
374         }
375         if (setting)
376         {
377             //กลับไป state 0
378             state = None;
379             setting = false;
380         }
381         //เข้าสู่ setup
382     else
383     {
384         setting = true;
385     }
386 }
387 else if (state == Alarm || state == AlarmMin)
388 {
389     if (setting)
390     {
// saveการตั้งค่า

```

```
392         EEPROM.put(0, isAlarm);
393         isSetAlarm = true;
394         setting = false;
395     }
396     //ເພົ່າສູ່ alarm
397     else
398     {
399         setting = true;
400     }
401 }
402 else if (state == Stopwatch)
403 {
404     if (isSetAlarm)
405     {
406         noTone(buzzer);
407         isSetAlarm = false;
408     }
409     //ເພົ່າສູ່ STW
410     setting = !setting;
411     // stop play
412 }
413 }
414 void upSwPressed()
415 {
416     if (setting)
417     {
418         if (state == Setup)
419         {
420             currentTime.hour;
421             currentTime.hour %= 24;
422         }
423         else if (state == SetupMin)
424         {
425             currentTime.minute;
426             currentTime.minute %= 60;
427         }
428         else if (state == Alarm)
429         {
430             isAlarm.hour++;
431             isAlarm.hour %= 24;
432         }
433         else if (state == AlarmMin)
434         {
435             isAlarm.minute++;
436             isAlarm.minute %= 60;
437         }
438     }
439 }
440 void downSwPressed()
441 {
442     if (setting)
443     {
444         if (state == Setup)
445         {
446             currentTime.hour--;
447             if (currentTime.hour < 0)
448             {
```

```
449         currentTime.hour = 23;
450     }
451 }
452 else if (state == SetupMin)
453 {
454     currentTime.minute--;
455     if (currentTime.minute < 0)
456     {
457         currentTime.minute = 59;
458     }
459 }
460 else if (state == Alarm)
461 {
462     isAlarm.hour--;
463     if (isAlarm.hour < 0)
464     {
465         isAlarm.hour = 23;
466     }
467 }
468 else if (state == AlarmMin)
469 {
470     isAlarm.minute--;
471     if (isAlarm.minute < 0)
472     {
473         isAlarm.minute = 59;
474     }
475 }
476 }
477 }
478 void loop()
479 {
480     Serial.println(state);
481     OLED.clearDisplay();
482     //กดปุ่ม
483     pressButton(modeSw, modeSwPressed);
484     pressButton(saveSw, saveSwPressed);
485     pressButton(upSw, upSwPressed);
486     pressButton(downSw, downSwPressed);
487     // saveEEPROM
488     if (currentTime.sec == 0)
489     {
490         EEPROM.put(0, currentTime);
491     }
492     // TimeRun
493     if (state == None)
494     {
495         showTime();
496         dayMode();
497         timeTemp();
498     }
499     // STU
500     else if (state == Setup || state == SetupMin)
501     {
502         showTimeSetup();
503         showMode("SETUP");
504         if (setting)
505         {
```

```
506         settingSTU(2 + (48 * (state == SetupMin)));
507     }
508 }
509 // ALARM
510 else if (state == Alarm || state == AlarmMin)
511 {
512     showAlarm();
513     showMode("ALARM");
514     if (setting)
515     {
516         settingSTU(2 + (48 * (state == AlarmMin)));
517     }
518 }
519 // STW
520 else if (state == Stopwatch)
521 {
522     showSTW();
523 }
524 else if (state == TempMode)
525 {
526     showTemp();
527 }
528
529 if (isSetAlarm)
530 {
531     AlarmWarning();
532 }
533
534 OLED.display();
535 }
```