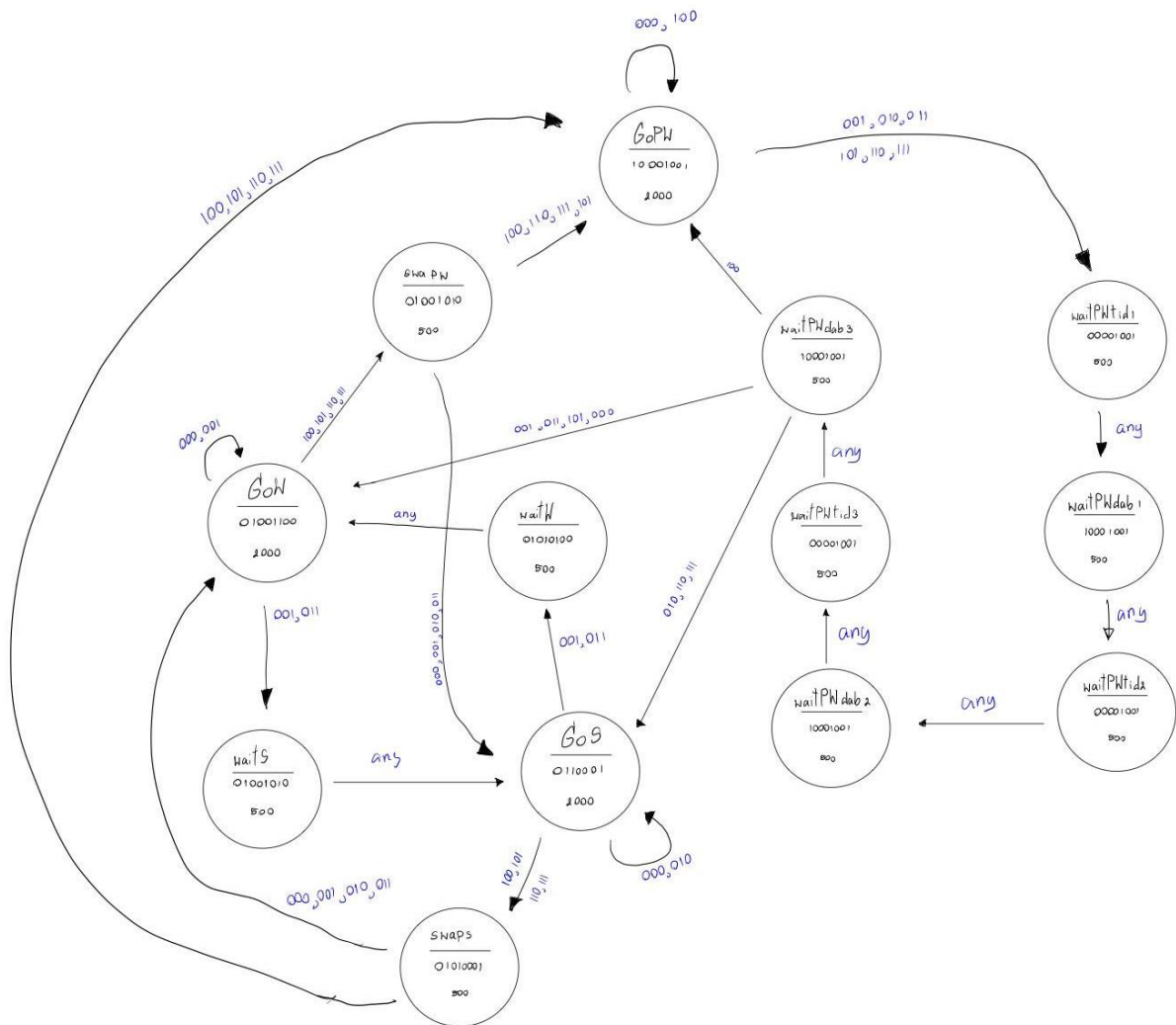


State Transition Graph

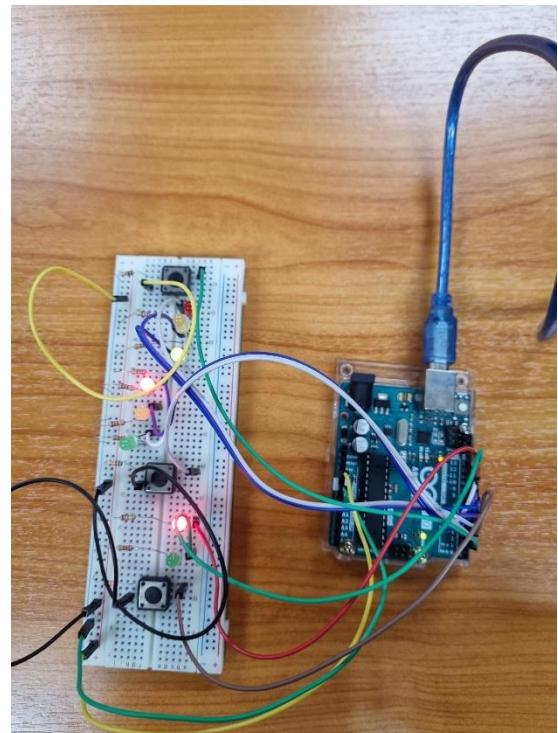
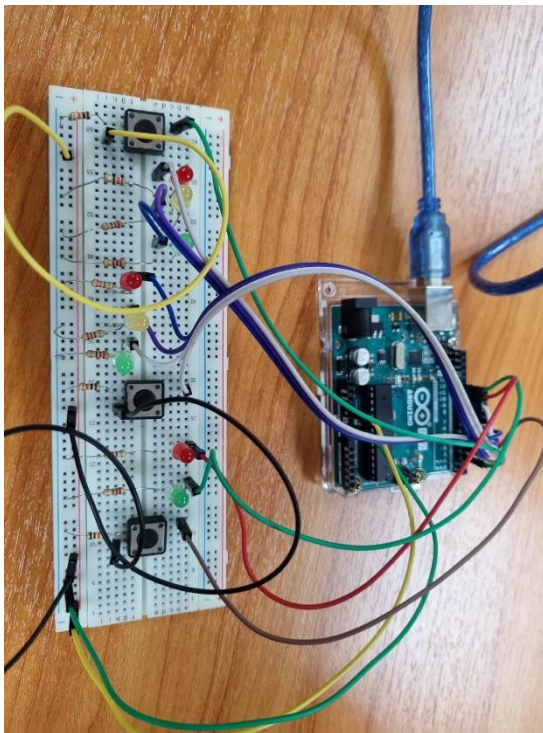
A 21
PSW



State Transition Table

num	name	light	time	input							
				000	001	010	011	100	101	110	111
0	goW	01001100	2000	goW	goW	waitS	waitS	swapw	swapw	swapw	swapw
1	waitW	01010100	500	goW	goW	goW	goW	goW	goW	goW	goW
2	goS	01100001	2000	goS	waitW	goS	waitW	swaps	swaps	swaps	waitW
3	waitS	01100010	500	goS	goS	goS	goS	goS	goS	goS	goS
4	GoPW	10001001	2000	GoPW	waitPWtid1	waitPWtid1	waitPWtid1	GoPW	waitPWtid1	waitPWtid1	waitPWtid1
5	waitPWtid1	00001001	500	waitPWdab1	waitPWdab1	waitPWdab1	waitPWdab1	waitPWdab1	waitPWdab1	waitPWdab1	waitPWdab1
6	waitPWdab1	10001001	500	waitPWtid2	waitPWtid2	waitPWtid2	waitPWtid2	waitPWtid2	waitPWtid2	waitPWtid2	waitPWtid2
7	waitPWtid2	00001001	500	waitPWdab2	waitPWdab2	waitPWdab2	waitPWdab2	waitPWdab2	waitPWdab2	waitPWdab2	waitPWdab2
8	waitPWdab2	10001001	500	waitPWtid3	waitPWtid3	waitPWtid3	waitPWtid3	waitPWtid3	waitPWtid3	waitPWtid3	waitPWtid3
9	waitPWtid3	00001001	500	waitPWdab3	waitPWdab3	waitPWdab3	waitPWdab3	waitPWdab3	waitPWdab3	waitPWdab3	waitPWdab3
10	waitPWdab3	10001001	500	goW	goW	goS	goW	GoPW	goW	goS	goS
11	swaps	01010001	500	goW	goW	goW	goW	GoPW	GoPW	GoPW	GoPW
12	swapw	01001010	500	goS	goS	goS	goS	GoPW	GoPW	GoPW	GoPW

รูปประกอบ



Source code

```
1 #define LED_W_R 5
2 #define LED_W_Y 6
3 #define LED_W_G 7
4 #define WEST_BUTTON_PIN 2
5 #define LED_S_R 8
6 #define LED_S_Y 9
7 #define LED_S_G 10
8 #define SOUTH_BUTTON_PIN 3
9 #define LED_P_R 11
10 #define LED_P_G 12
11 #define P_BUTTON_PIN 4
12
13 #define goW 0
14 #define waitW 1
15 #define goS 2
16 #define waitS 3
17 #define GoPW 4
18 #define waitPWtid1 5
19 #define waitPWdab1 6
20 #define waitPWtid2 7
21 #define waitPWdab2 8
22 #define waitPWtid3 9
23 #define waitPWdab3 10
24 #define swaps 11
25 #define swapw 12
26
27 int input, input1, input2, input3;
28
29 struct State {
30     unsigned long ST_Out;
31     unsigned long Time;
32     unsigned long Next[8];
33 };
34 typedef const struct State SType;
35 SType FSM[13] = {
36     {B01001100, 2000, {goW, goW, waitS, waitS, swapw, swapw, swapw, swapw}},
37     {B01010001, 500, {goW, goW, goW, goW, goW, goW, goW, goW}},
38     {B01100001, 2000, {goS, waitW, goS, waitW, swaps, swaps, swaps, waitW}},
39     {B01001010, 500, {goS, goS, goS, goS, goS, goS, goS, goS}},
40     {B10001001, 2000, {GoPW, waitPWtid1, waitPWtid1, waitPWtid1, GoPW, waitPWtid1, waitPWtid1, waitPWtid1}},
41     {B00001001, 500, {waitPWdab1, waitPWdab1, waitPWdab1, waitPWdab1, waitPWdab1, waitPWdab1, waitPWdab1, waitPWdab1}},
42     {B10001001, 500, {waitPWtid2, waitPWtid2, waitPWtid2, waitPWtid2, waitPWtid2, waitPWtid2, waitPWtid2, waitPWtid2}},
43     {B00001001, 500, {waitPWdab2, waitPWdab2, waitPWdab2, waitPWdab2, waitPWdab2, waitPWdab2, waitPWdab2, waitPWdab2}},
44     {B10001001, 500, {waitPWtid3, waitPWtid3, waitPWtid3, waitPWtid3, waitPWtid3, waitPWtid3, waitPWtid3, waitPWtid3}},
45     {B00001001, 500, {waitPWdab3, waitPWdab3, waitPWdab3, waitPWdab3, waitPWdab3, waitPWdab3, waitPWdab3, waitPWdab3}},
46     {B10001001, 500, {goW, goW, goS, goW, GoPW, goW, goS, goS}},
47     {B01010001, 500, {goW, goW, goW, goW, GoPW, GoPW, GoPW, GoPW}},
48     {B01001010, 500, {goS, goS, goS, goS, GoPW, GoPW, GoPW, GoPW}},
49 };
50 unsigned long s = 0; // index to the current state
51
```

```

52 void setup() {
53     pinMode(LED_W_R, OUTPUT);
54     pinMode(LED_W_Y, OUTPUT);
55     pinMode(LED_W_G, OUTPUT);
56     pinMode(WEST_BUTTON_PIN, INPUT);
57     pinMode(LED_S_R, OUTPUT);
58     pinMode(LED_S_Y, OUTPUT);
59     pinMode(LED_S_G, OUTPUT);
60     pinMode(SOUTH_BUTTON_PIN, INPUT);
61     pinMode(LED_P_R, OUTPUT);
62     pinMode(LED_P_G, OUTPUT);
63     pinMode(P_BUTTON_PIN, INPUT);
64     Serial.begin(9600);
65 }
67 void loop() {
68     //Serial.println(String("BW = ") + input1 + String("BS = ") + input2 + String("BP = ") + input3);
69     //Serial.println(input);
70     Serial.println(S);
71
72     digitalWrite(LED_W_R, bitRead(FSM[S].ST_Out, 0));
73     digitalWrite(LED_W_Y, bitRead(FSM[S].ST_Out, 1));
74     digitalWrite(LED_W_G, bitRead(FSM[S].ST_Out, 2));
75     digitalWrite(LED_S_R, bitRead(FSM[S].ST_Out, 3));
76     digitalWrite(LED_S_Y, bitRead(FSM[S].ST_Out, 4));
77     digitalWrite(LED_S_G, bitRead(FSM[S].ST_Out, 5));
78     digitalWrite(LED_P_R, bitRead(FSM[S].ST_Out, 6));
79     digitalWrite(LED_P_G, bitRead(FSM[S].ST_Out, 7));
80
81     delay(FSM[S].Time);
82
83     input1 = digitalRead(WEST_BUTTON_PIN);
84     input2 = digitalRead(SOUTH_BUTTON_PIN);
85     input3 = digitalRead(P_BUTTON_PIN);
86     input = input2*2 + input1 + input3*4;
87     S = FSM[S].Next[input];
88 }

```

คำอธิบาย

บรรทัดที่ 1-11 เป็นการกำหนดเลขpinของตัวLEDและButton

บรรทัดที่ 13-25 เป็นการกำหนดค่าstate

บรรทัดที่ 27,83-86 ให้ตัวแปร input1อ่านค่าbuttonรอกึงทิศwest ตัวแปร input2 อ่านค่าbuttonรอกึงทางทิศsouth และinput3 ให้อ่านค่าของbuttonคนเดิน ตัวแปร input คือ ตัวแปรที่เอาไว้แปลงค่าเลข3bit ให้เป็นเลขฐาน10

บรรทัดที่ 29-34 เป็น **struct** ของแต่ละstate

บรรทัดที่ 35-49 เป็น **struct** ของ **state** แต่ละ state ประกอบด้วย ลักษณะการแสดงผลของLED, ค่า **delay** ของLED, stateตัวถัดไป

บรรทัดที่50,87 ให้ตัวแปรS ในการเปลี่ยนเป็นstateถัดไป ดูจากตัวแปรinput

บรรทัดที่52-65 กำหนดค่า **pinMode** ของLEDและ button

บรรทัดที่ 67-81 เป็นloopการทำงานของแสดงผลLEDตามstateที่รับมาจากตัวแปรinput