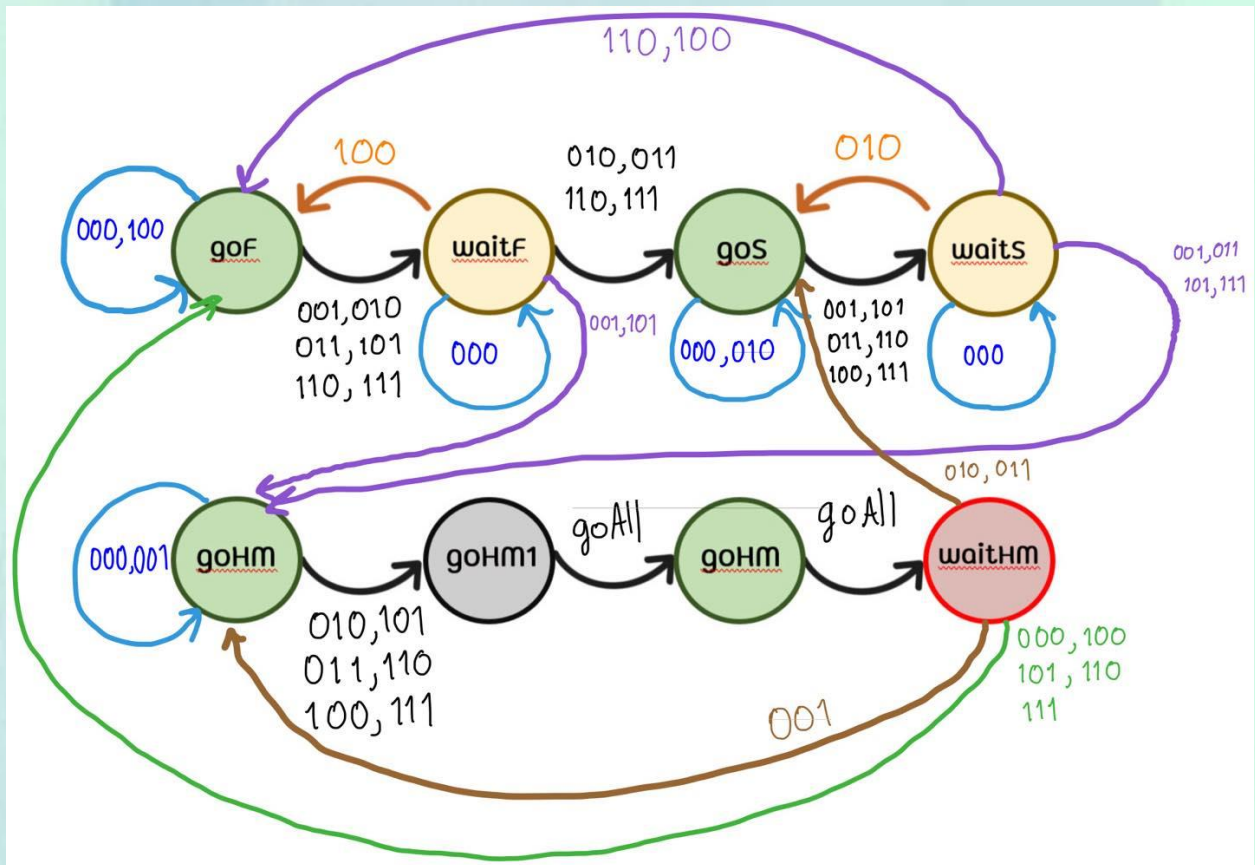


Assignment #b : FSM

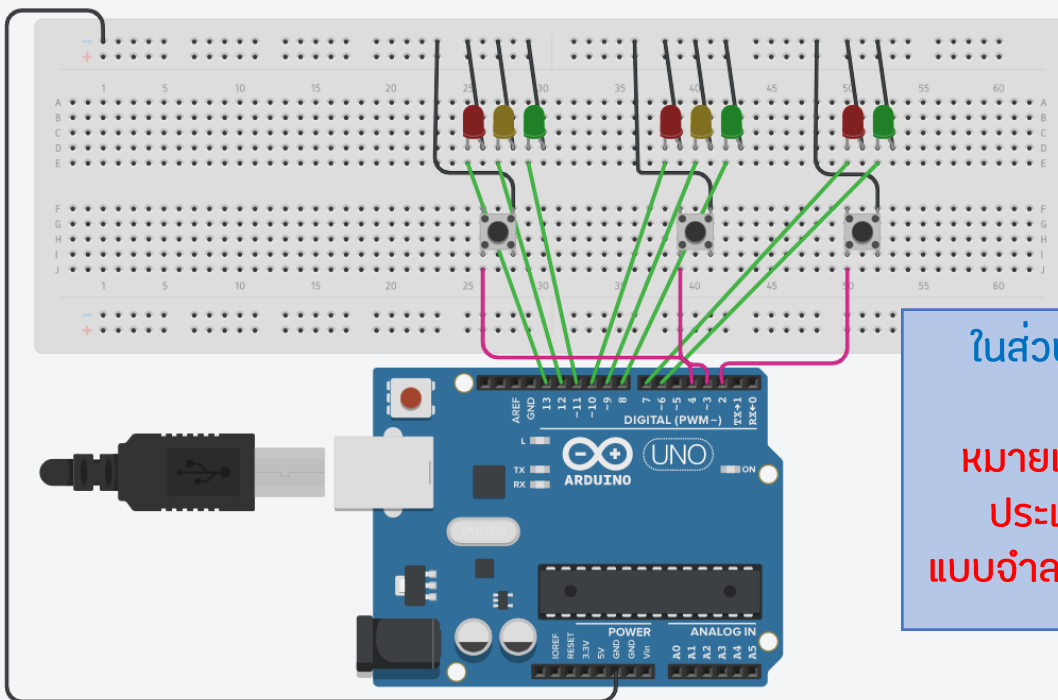
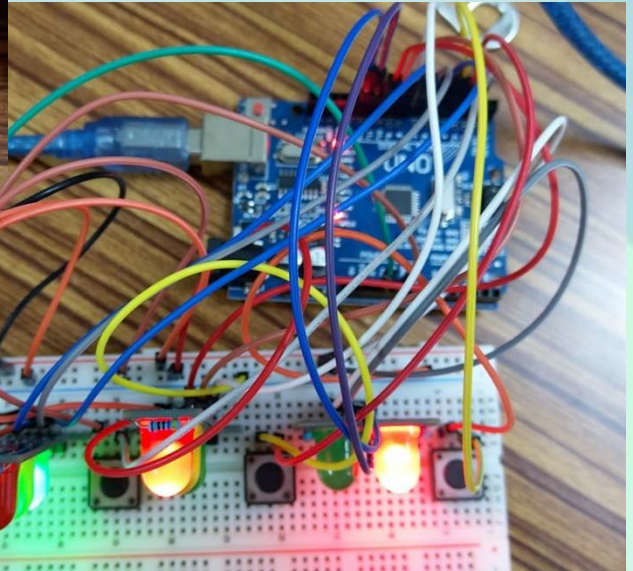
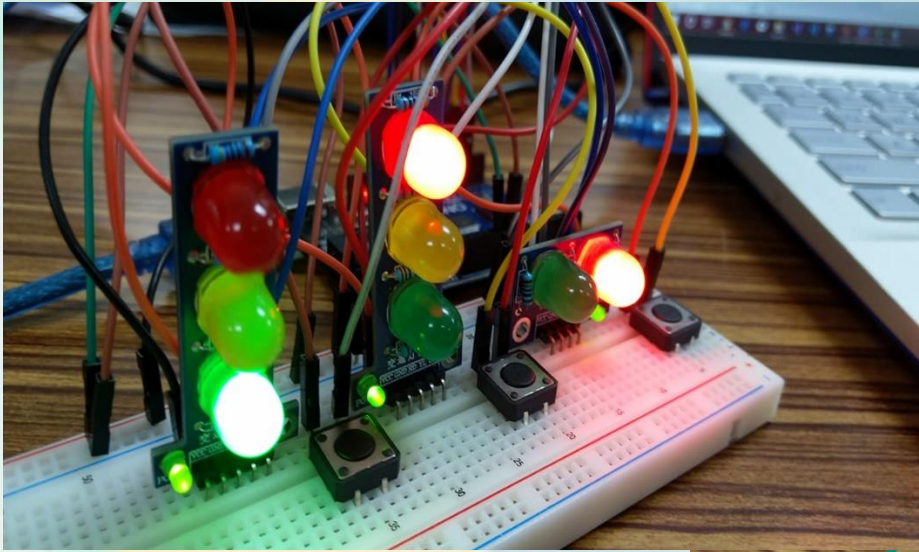
❖ State Transition Table

				000	001	010	011	100	101	110	111
num	name	status	time	0	1	2	3	4	5	6	7
0	goF	00110001	1000	goF	waitF	waitF	waitF	goF	waitF	waitF	waitF
1	waitF	11110001	500	waitF	goHM	goS	goS	goF	goHM	goS	goS
2	goS	10000101	1000	goS	waitS	goS	waitS	waitS	waitS	waitS	waitS
3	waitS	10011101	500	waitS	goHM	goS	goHM	goF	goHM	goF	goHM
4	goHM	10010010	500	goHM	goHM	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1
5	goHM1	10010000	500	goHM2	goHM2	goHM2	goHM2	goHM2	goHM2	goHM2	goHM2
6	goHM2	10010010	500	waitHM	waitHM	waitHM	waitHM	waitHM	waitHM	waitHM	waitHM
7	waitHM	10010000	500	goF	goHM	goS	goS	goF	goF	goF	goF

❖ State Transition



Basic Circuit



ในส่วนนี้จะเป็นการแสดงการต่อ
วงจรในรูปอย่างง่าย
หมายเหตุ เนื่องจากใช้ LED คนละ
ประเภทกัน ทางผู้จัดทำจึงทำ
แบบจำลองในรูปแบบ โค้ดโปรแกรม



Source Code

ในส่วนนี้จะเป็นการกำหนด pin ต่างๆของ ไฟจราจร และไฟคนเดิน

```
1 #define LED_F_R 7
2 #define LED_F_Y 6
3 #define LED_F_G 5
4
5 #define LED_S_R 10
6 #define LED_S_Y 9
7 #define LED_S_G 8
8
9 #define HM_R 12
10 #define HM_G 11
11
12 #define F_Button 2
13 #define S_Button 3
14 #define HM_Button 4
15 //-----//
16 #define goF 0
17 #define waitF 1
18
19 #define goS 2
20 #define waitS 3
21
22 #define waitHM 7
23 #define goHM 4
24 #define goHM1 5
25 #define goHM2 6
26
```

ส่วนต่อมาเป็นการกำหนด state

```
27 struct State {
28     unsigned long ST_Out; // 6-bit pattern to street output
29     unsigned long Time; // delay in ms units
30     unsigned long Next[8];
31 }; // next state for inputs 0,1,2,3
32 typedef const struct State SType;
33
34 SType FSM[8] =
35 {
36     // 000 001 010 011 100 101 110 111
37     {B00110001, 1000, {goF, waitF, waitF, waitF, goF, waitF, waitF, waitF}}, //0 F
38     {0xF1, 500, {waitF, goHM, goS, goS, goF, goHM, goS, goS}}, //1 wait F1
39     {0x85, 1000, {goS, waitS, goS, waitS, waitS, waitS, waitS, waitS}}, //2 S
40     {0x9D, 500, {waitS, goHM, goS, goHM, goF, goHM, goF, goHM}}, //3 waitS 9D
41     {0x92, 500, {goHM, goHM, goHM1, goHM1, goHM1, goHM1, goHM1, goHM1}}, // WALK
42     {B10010000, 500, {goHM2, goHM2, goHM2, goHM2, goHM2, goHM2, goHM2, goHM2}}, //5 KAPIB
43     {0x92, 500, {waitHM, waitHM, waitHM, waitHM, waitHM, waitHM, waitHM, waitHM}}, //6
44     {0x90, 500, {goF, goHM, goS, goS, goF, goF, goF, goF}} //7
45 };
```



```

void setup()
{

  pinMode(LED_F_R, OUTPUT);
  pinMode(LED_F_Y, OUTPUT);
  pinMode(LED_F_G, OUTPUT);
  pinMode(F_Button, INPUT_PULLUP);
  pinMode(LED_S_R, OUTPUT);
  pinMode(LED_S_Y, OUTPUT);
  pinMode(LED_S_G, OUTPUT);
  pinMode(S_Button, INPUT_PULLUP);
  pinMode(HM_R, OUTPUT);
  pinMode(HM_G, OUTPUT);
  pinMode(HM_Button, INPUT_PULLUP);
  Serial.begin(9600);

}

```

ส่วนต่อมาเป็นการประกาศใช้งาน
พิน ต่างๆ ทั้งหลอดไฟ และปุ่มกด

```

int input, inputF, inputs, inputW;
void loop()
{
  digitalWrite(LED_F_G, FSM[S].ST_Out & B10000000);
  digitalWrite(LED_F_Y, !(FSM[S].ST_Out & B01000000));
  digitalWrite(LED_F_R, FSM[S].ST_Out & B00100000);

  digitalWrite(LED_S_G, FSM[S].ST_Out & B00010000);
  digitalWrite(LED_S_Y, !(FSM[S].ST_Out & B00001000));
  digitalWrite(LED_S_R, FSM[S].ST_Out & B00000100);

  digitalWrite(HM_G, !(FSM[S].ST_Out & B00000010));
  digitalWrite(HM_R, !(FSM[S].ST_Out & B00000001));
  delay(FSM[S].Time);

  if(S != waitF && S != waitS && S != goHM1 && S != waitHM && S != goHM2 )
  {
    input = 0 ;
    inputF = !digitalRead(F_Button);
    inputs = !digitalRead(S_Button);
    inputW = !digitalRead(HM_Button);
    input = inputF * 4 + inputs * 2 + inputW;
  }
  Serial.println(input);
  S = FSM[S].Next[input];
}

```

ส่วนต่อมาเป็นการกำหนดสถานะ
ของหลอดไฟ และกำหนดเงื่อนไข
หาก สถานะของไฟเป็น wait อยู่จะ
ไม่มีการรับค่าที่ INPUT ใหม่