

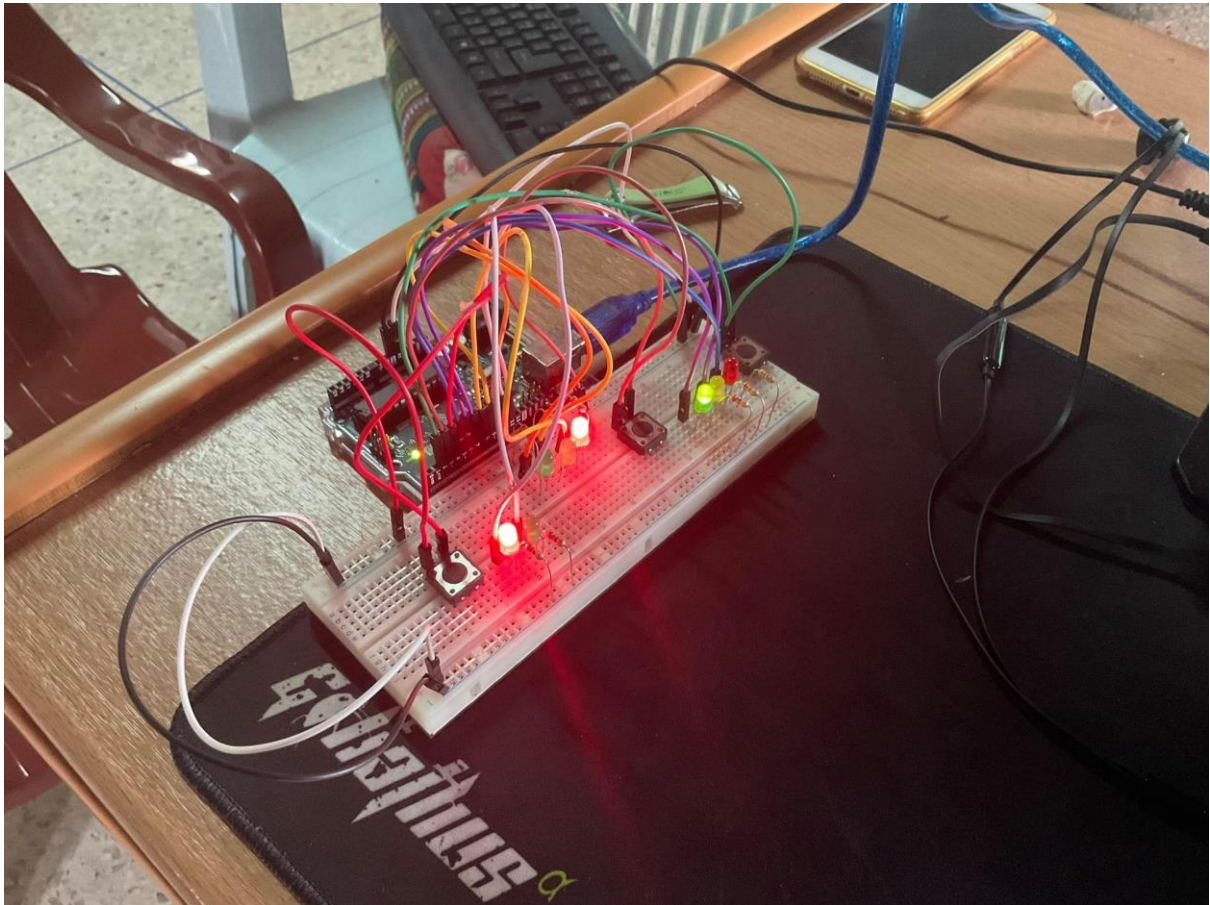
Assignment 5

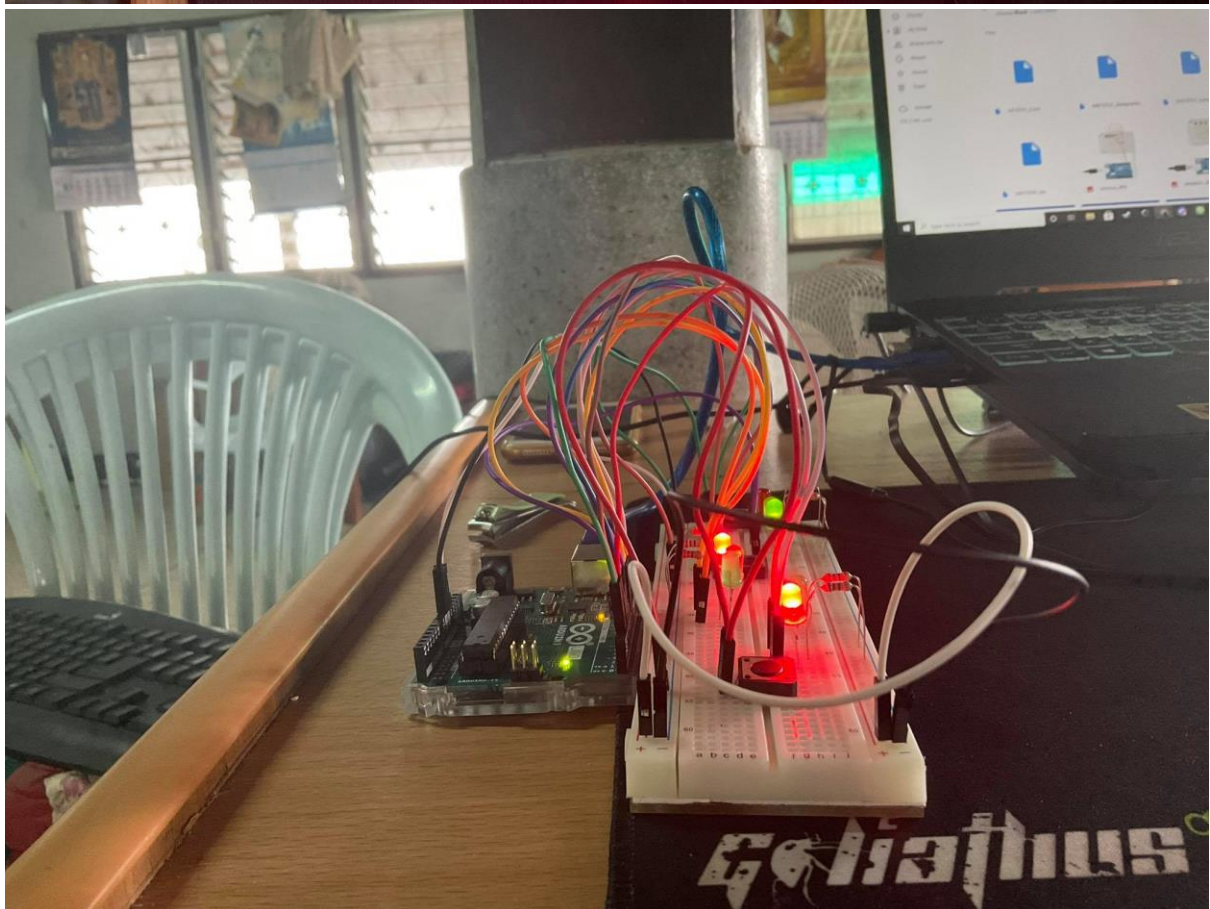
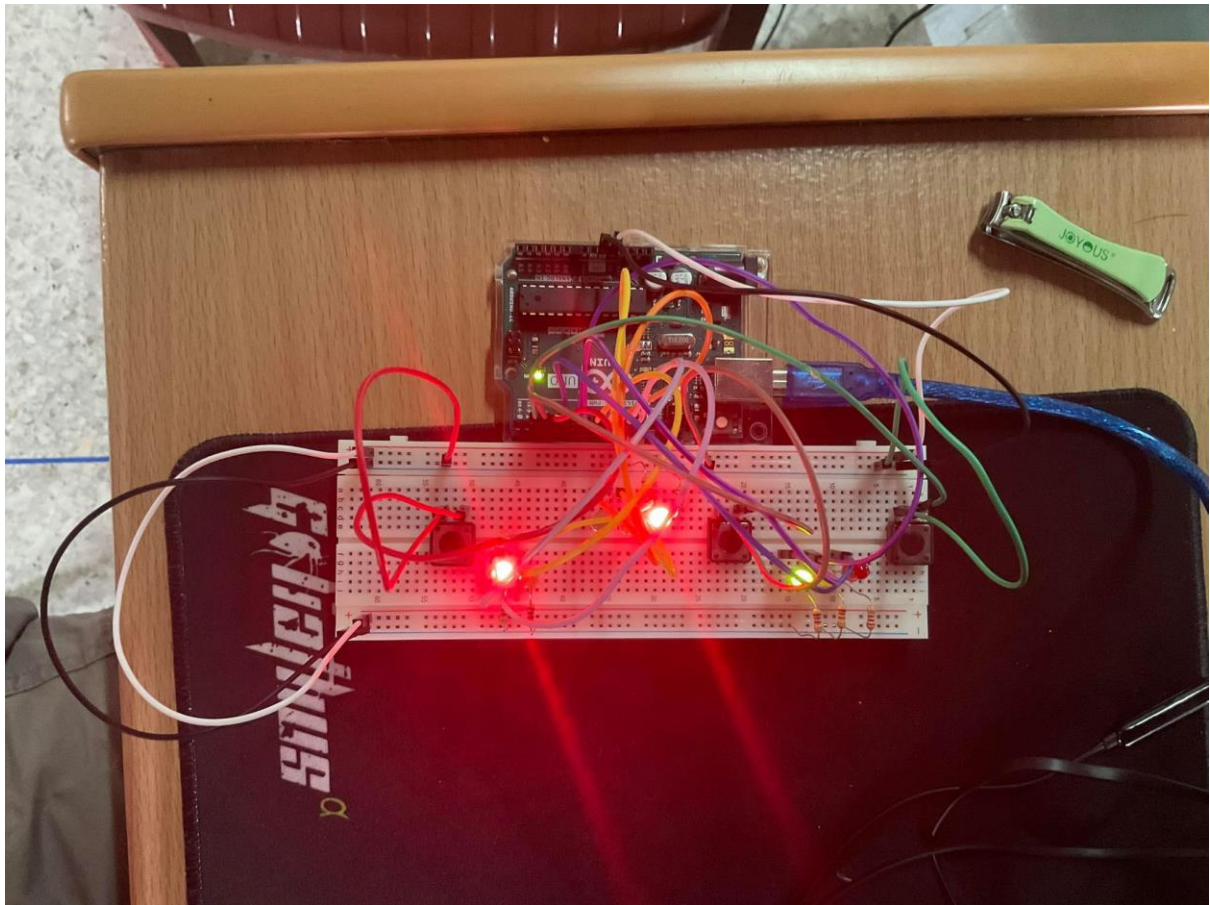
ชื่อกลุ่ม : 9 A.M.

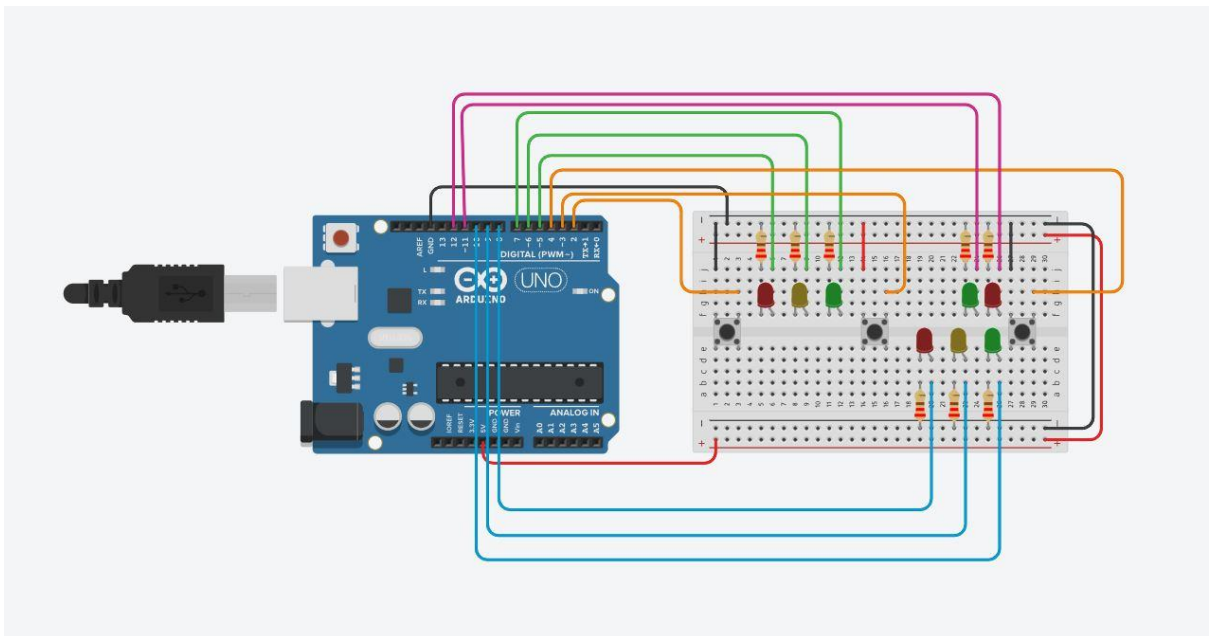
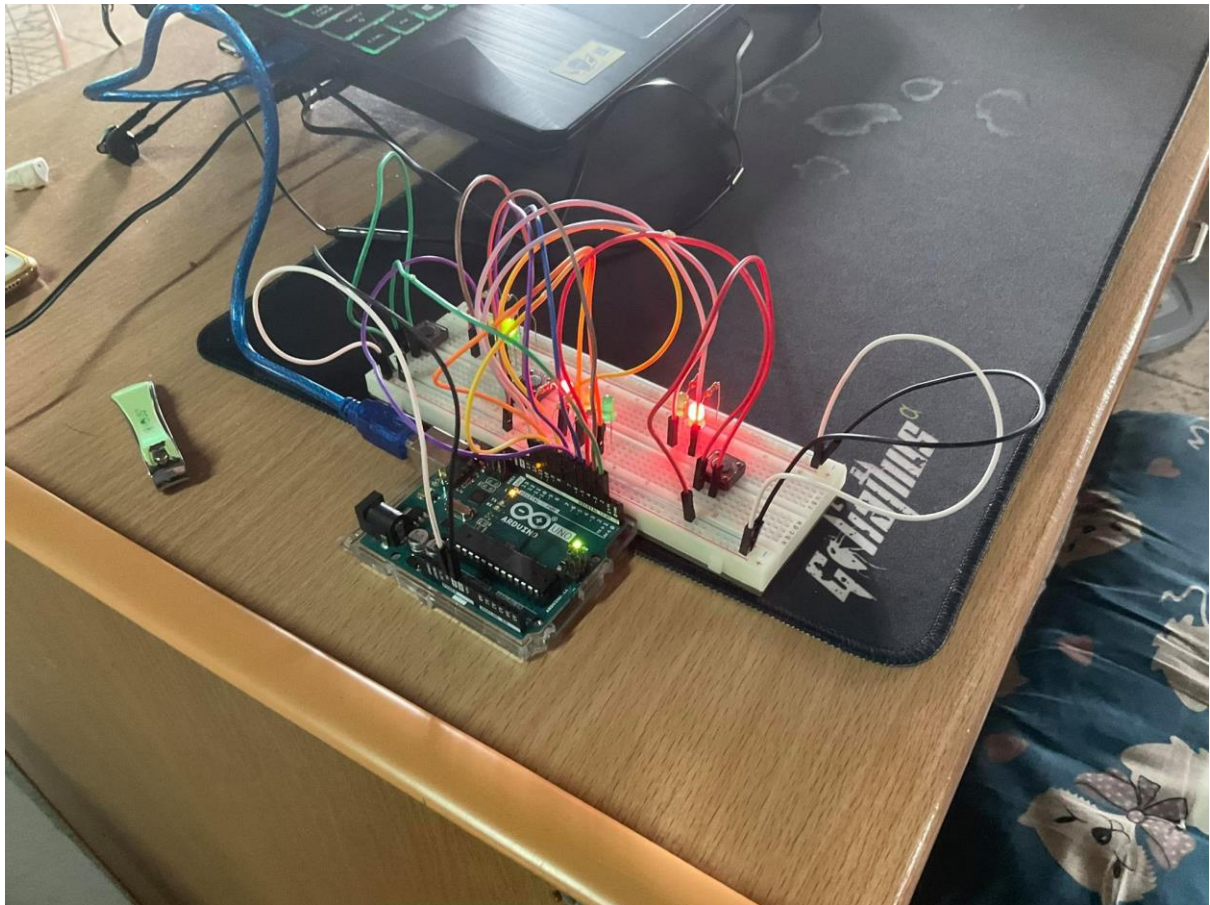
สมาชิกกลุ่ม : 64010761 นายวรพล รังษี

64010757 นายวรโชติ ใจเร็ว

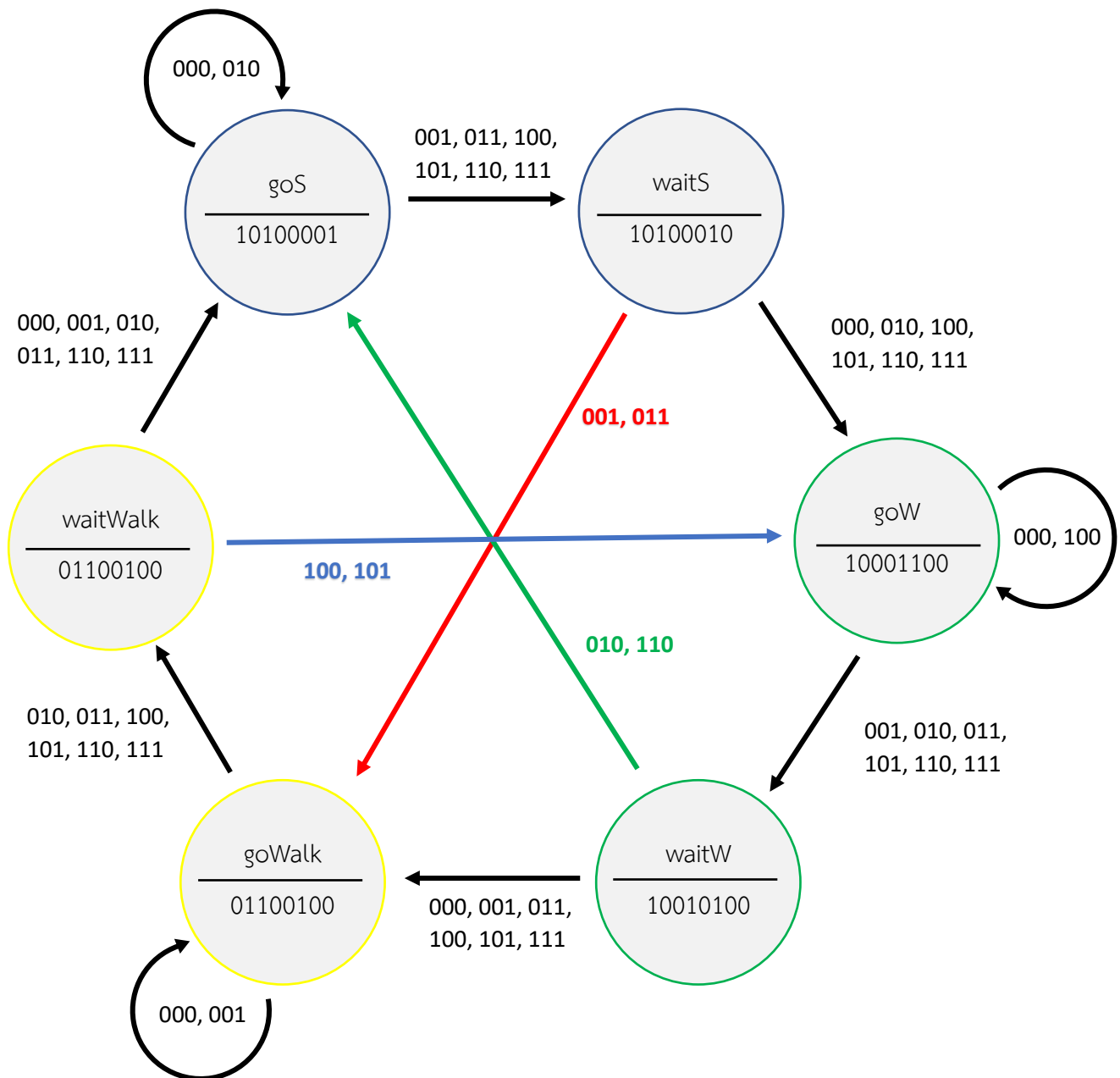
รูปถ่ายชิ้นงาน :







State transition graph :



PIN			สถานการณ์
2	3	4	
0	0	0	ไม่มีรถหรือคนอยู่เลย
0	0	1	มีคนมารอข้ามถนน และไม่มีรถเลย
0	1	0	มีแค่รถฝั่ง South เท่านั้น
0	1	1	มีรถฝั่ง South และมีคนรอข้ามถนน แต่ไม่มีรถฝั่ง West
1	0	0	มีแค่รถฝั่ง West เท่านั้น
1	0	1	มีรถฝั่ง West และมีคนรอข้ามถนน แต่ไม่มีรถฝั่ง South
1	1	0	มีรถทั้งฝั่ง West และ South แต่ไม่มีคนรอข้ามถนน
1	1	1	มีรถทั้ง 2 ฝั่ง และมีคนรอข้ามถนน

State transition table :

num	name	lights	time	ln=0	ln=1	ln=2	ln=3	ln=4	ln=5	ln=6	ln=7
0	goS	10100001	3000	goS	waitS	goS	waitS	waitS	waitS	waitS	waitS
1	waitS	10100010	1000	goW	goWalk	goW	goWalk	goW	goW	goW	goW
2	goW	10001100	3000	goW	waitW	waitW	waitW	goW	waitW	waitW	waitW
3	waitW	10010100	1000	goWalk	goWalk	goS	goWalk	goWalk	goWalk	goS	goWalk
4	goWalk	01100100	3000	goWalk	goWalk	waitWalk	waitWalk	waitWalk	waitWalk	waitWalk	waitWalk
5	waitWalk	01100100	900	goS	goS	goS	goS	goW	goW	goS	goS

Source Code : https://github.com/NineWP/My-arduino-ITC/blob/main/64010761_Assignment_5/64010761_Assignment_5.ino

ส่วนประกอบหลักๆของ โปรแกรม :

1. struct State โดยจะใช้เก็บข้อมูลของแต่ละ State ได้แก่ ST_Out (Output ของ State) , Time(delay) , Next[8](State ถัดไปที่สามารถไปได้) โดย State ทั้งหมด 6 State จะถูกใส่ข้อมูลตาม State transition table
2. void setup กำหนด pinMode ของ Switch และ LED
3. void loop จะทำหน้าที่อ่าน bit จาก ST_Out และ เปิด - ปิด LED แต่ละดวง รวมถึงคอยรับค่า input จาก switch ทั้ง 3 เพื่อจะเปลี่ยน State ตาม input

```
#define LED_W_R 5
#define LED_W_Y 6
#define LED_W_G 7
#define WEST_BUTTON_PIN 2
#define LED_S_R 8
#define LED_S_Y 9
#define LED_S_G 10
#define SOUTH_BUTTON_PIN 3
#define LED_WALK_R 12
#define LED_WALK_G 11
#define WALK_BUTTON_PIN 4
#define goS 0
#define waitS 1
#define goW 2
#define waitW 3
#define goWalk 4
#define waitWalk 5

struct State {
    unsigned long ST_Out; // 6-bit pattern to street output
    unsigned long Time; // delay in ms units
    unsigned long Next[8];
```

```

}; // next state for inputs 0,1,2,3
typedef const struct State SType;
SType FSM[6] = {
    {B10100001, 3000, {goS, waitS, goS, waitS, waitS, waitS, waitS, waitS}},
    {B10100010, 1000, {goW, goWalk, goW, goWalk, goW, goW, goW, goW}},
    {B10001100, 3000, {goW, waitW, waitW, waitW, goW, waitW, waitW, waitW}},
    {B10010100, 1000, {goWalk, goWalk, goS, goWalk, goWalk, goWalk, goS, goWalk}},
    {B01100100, 3000, {goWalk, goWalk, waitWalk, waitWalk, waitWalk, waitWalk, waitWalk, waitWalk}},
    {B01100100, 250, {goS, goS, goS, goS, goW, goW, goS, goS}}
};

unsigned long S = 0, C = 7, B = 0; // index to the current state

void setup() {
    pinMode(LED_W_R, OUTPUT);
    pinMode(LED_W_Y, OUTPUT);
    pinMode(LED_W_G, OUTPUT);
    pinMode(WEST_BUTTON_PIN, INPUT_PULLUP);
    pinMode(LED_S_R, OUTPUT);
    pinMode(LED_S_Y, OUTPUT);
    pinMode(LED_S_G, OUTPUT);
    pinMode(SOUTH_BUTTON_PIN, INPUT_PULLUP);
    pinMode(LED_WALK_R, OUTPUT);
    pinMode(LED_WALK_G, OUTPUT);
    pinMode(WALK_BUTTON_PIN, INPUT_PULLUP);
    Serial.begin(9600);
}

int input, input1, input2, input3;

void loop() {
    if (S != 5) {
        digitalWrite(LED_W_R, FSM[S].ST_Out & B00100000);
        digitalWrite(LED_W_Y, FSM[S].ST_Out & B00010000);
        digitalWrite(LED_W_G, FSM[S].ST_Out & B00001000);
        digitalWrite(LED_S_R, FSM[S].ST_Out & B00000100);
        digitalWrite(LED_S_Y, FSM[S].ST_Out & B00000010);
        digitalWrite(LED_S_G, FSM[S].ST_Out & B00000001);
        digitalWrite(LED_WALK_G, FSM[S].ST_Out & B01000000);
        digitalWrite(LED_WALK_R, FSM[S].ST_Out & B10000000);
    }

    delay(FSM[S].Time);

    if (S == 5 && C >= 1)
    {
        Serial.print(S);
        if (B == 1)
            digitalWrite(LED_WALK_G, FSM[S].ST_Out & B01000000);
    }
}

```

```
else
    digitalWrite(LED_WALK_G, FSM[S].ST_Out & B00000000);
B = !(B);
C--; Serial.println(C);
return;
}

input1 = digitalRead(WEST_BUTTON_PIN);
input2 = digitalRead(SOUTH_BUTTON_PIN);
input3 = digitalRead(WALK_BUTTON_PIN);
input1 = !(input1);
input2 = !(input2);
input3 = !(input3);
input = input1 * 4 + input2 * 2 + input3;
S = FSM[S].Next[input];
C = 7; B = 0;
}
```