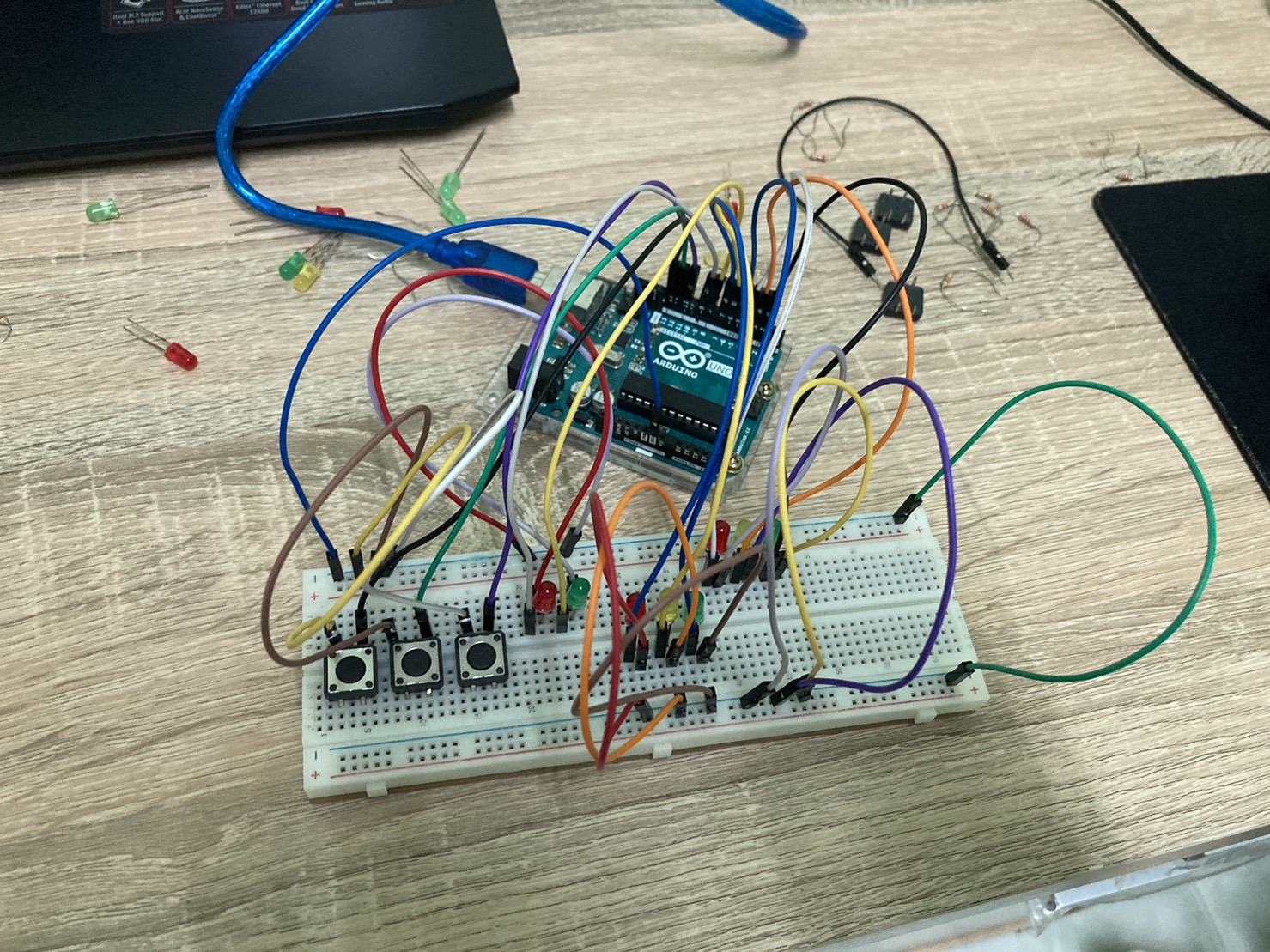
Assignment 5 : Finite State Machine

**State Transition Table**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Num** | **Name** | **Lights** | **Input** | | | | | | | |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **0** | **waitWest** | **B10010100** | **goWest** | **goWest** | **goWest** | **goWest** | **goWEst** | **goWest** | **goWest** | **goWest** |
| **1** | **goWest** | **B10100001** | **goWest** | **goWest** | **waitSouth** | **waitSouth** | **wait\_Walk\_West** | **wait\_Walk\_West** | **waitSouth** | **waitSouth** |
| **2** | **waitSouth** | **B10100010** | **goSouth** | **goSouth** | **goSouth** | **goSouth** | **goSouth** | **goSouth** | **goSouth** | **goSouth** |
| **4** | **wait\_Walk\_South** | **B10010100** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** |
| **5** | **wait\_Walk\_West** | **B10100010** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** | **goWalk** |
| **6** | **goWalk** | **B01100100** | **goWalk** | **button1\_OFF** | **button1\_OFF** | **button1\_OFF** | **goWalk** | **button1\_OFF** | **button1\_OFF** | **button1\_OFF** |
| **7** | **button1\_OFF** | **B00100100** | **button1\_ON** | **button1\_ON** | **button1\_ON** | **button1\_ON** | **button1\_ON** | **button1\_ON** | **button1\_ON** | **button1\_ON** |
| **8** | **button1\_ON** | **B01100100** | **button2\_OFF** | **button2\_OFF** | **button2\_OFF** | **button2\_OFF** | **button2\_OFF** | **button2\_OFF** | **button2\_OFF** | **button2\_OFF** |
| **9** | **button2\_OFF** | **B00100100** | **button2\_ON** | **button2\_ON** | **button2\_ON** | **button2\_ON** | **button2\_ON** | **button2\_ON** | **button2\_ON** | **button2\_ON** |
| **10** | **button2\_ON** | **B01100100** | button3\_OFF | button3\_OFF | button3\_OFF | button3\_OFF | button3\_OFF | button3\_OFF | button3\_OFF | button3\_OFF |
| **11** | **button3\_OFF** | **B00100100** | **button3\_ON** | **button3\_ON** | **button3\_ON** | **button3\_ON** | **button3\_ON** | **button3\_ON** | **button3\_ON** | **button3\_ON** |
| **12** | **button3\_ON** | **B01100100** | **goWalk** | **goWest** | **goSouth** | **goWest** | **goWalk** | **goWest** | **goSouth** | **goWest** |

**Diagram

Description automatically generated**



#define LED\_W\_R 4

#define LED\_W\_Y 3

#define LED\_W\_G 2

#define WEST\_BUTTON\_PIN 11

#define LED\_S\_R 7

#define LED\_S\_Y 6

#define LED\_S\_G 5

#define SOUTH\_BUTTON\_PIN 12

#define LED\_WALK\_G 8

#define LED\_WALK\_R 9

#define WALK\_BUTTON\_PIN 13

#define waitWest 0

#define goWest 1

#define waitSouth 2

#define goSouth 3

#define wait\_WALK\_SOUTH 4

#define wait\_WALK\_WEST 5

#define goWalk 6

#define b1\_OFF 7

#define b1\_ON 8

#define b2\_OFF 9

#define b2\_ON 10

#define b3\_OFF 11

#define b3\_ON 12

struct State

{

  unsigned long ST\_Out;

  unsigned long Time;

  unsigned long Next[8];

};

typedef const struct State SType;

SType FSM[13] = {

    {B10010100, 400, {goWest, goWest, goWest, goWest, goWest, goWest, goWest, goWest}},                                            // waitWest

    {B10100001, 2000, {goWest, goWest, waitSouth, waitSouth, wait\_WALK\_WEST, wait\_WALK\_WEST, waitSouth, waitSouth}},               // goWest

    {B10100010, 400, {goSouth, goSouth, goSouth, goSouth, goSouth, goSouth, goSouth, goSouth}},                                    // waitSouth

    {B10001100, 2000, {goSouth, waitWest, goSouth, waitWest, wait\_WALK\_SOUTH, wait\_WALK\_SOUTH, wait\_WALK\_SOUTH, wait\_WALK\_SOUTH}}, // goSouth

    {B10010100, 400, {goWalk, goWalk, goWalk, goWalk, goWalk, goWalk, goWalk, goWalk}},                                            // wait\_WALK\_SOUTH

    {B10100010, 400, {goWalk, goWalk, goWalk, goWalk, goWalk, goWalk, goWalk, goWalk}},                                            // wait\_WALK\_WEST

    {B01100100, 2000, {goWalk, b1\_OFF, b1\_OFF, b1\_OFF, goWalk, b1\_OFF, b1\_OFF, b1\_OFF}},                                           // goWalk

    {B00100100, 300, {b1\_ON, b1\_ON, b1\_ON, b1\_ON, b1\_ON, b1\_ON, b1\_ON, b1\_ON}},                                                    // b1\_OFF

    {B01100100, 300, {b2\_OFF, b2\_OFF, b2\_OFF, b2\_OFF, b2\_OFF, b2\_OFF, b2\_OFF, b2\_OFF}},                                            // b1\_ON

    {B00100100, 300, {b2\_ON, b2\_ON, b2\_ON, b2\_ON, b2\_ON, b2\_ON, b2\_ON, b2\_ON}},                                                    // b2\_OFF

    {B01100100, 300, {b3\_OFF, b3\_OFF, b3\_OFF, b3\_OFF, b3\_OFF, b3\_OFF, b3\_OFF, b3\_OFF}},                                            // b2\_ON

    {B00100100, 300, {b3\_ON, b3\_ON, b3\_ON, b3\_ON, b3\_ON, b3\_ON, b3\_ON, b3\_ON}},                                                    // b3\_OFF

    {B01100100, 300, {goWalk, goWest, goSouth, goWest, goWalk, goWest, goSouth, goWest}},                                          // b3\_ON

};

unsigned long S = 0;

void setup()

{

  Serial.begin(9600);

  pinMode(LED\_W\_G, OUTPUT);

  pinMode(LED\_W\_Y, OUTPUT);

  pinMode(LED\_W\_R, OUTPUT);

  pinMode(WEST\_BUTTON\_PIN, INPUT);

  pinMode(LED\_S\_G, OUTPUT);

  pinMode(LED\_S\_Y, OUTPUT);

  pinMode(LED\_S\_R, OUTPUT);

  pinMode(SOUTH\_BUTTON\_PIN, INPUT);

  pinMode(LED\_WALK\_G, OUTPUT);

  pinMode(LED\_WALK\_R, OUTPUT);

  pinMode(WALK\_BUTTON\_PIN, INPUT);

}

int West, South, Walk, input;

void loop()

{

  digitalWrite(LED\_W\_G, FSM[S].ST\_Out & B00000001);

  digitalWrite(LED\_W\_Y, FSM[S].ST\_Out & B00000010);

  digitalWrite(LED\_W\_R, FSM[S].ST\_Out & B00000100);

  digitalWrite(LED\_S\_G, FSM[S].ST\_Out & B00001000);

  digitalWrite(LED\_S\_Y, FSM[S].ST\_Out & B00010000);

  digitalWrite(LED\_S\_R, FSM[S].ST\_Out & B00100000);

  digitalWrite(LED\_WALK\_G, FSM[S].ST\_Out & B01000000);

  digitalWrite(LED\_WALK\_R, FSM[S].ST\_Out & B10000000);

  delay(FSM[S].Time);

  West  = !digitalRead(WEST\_BUTTON\_PIN);

  South = !digitalRead(SOUTH\_BUTTON\_PIN);

  Walk  = !digitalRead(WALK\_BUTTON\_PIN);

  input = Walk \* 4 + South \* 2 + West;

  Serial.print(Walk);Serial.print(South);Serial.println(West);

  Serial.println(input);Serial.println("-------------");

  S = FSM[S].Next[input];

}