SBTLS – CODE

int Lane1[] = {4,3,2}; // Lane 1 Red, Yellow and Green

int Lane2[] = {5,6,7};// Lane 2 Red, Yellow and Green

int Lane3[] = {8,9,10};// Lane 3 Red, Yellow and Green

int Lane4[] = {11,12,13};// Lane 4 Red, Yellow and Green

void setup()

{

pinMode(14, INPUT);

pinMode(15, INPUT);

for (int i = 0; i < 3; i++)

{

pinMode(Lane1[i], OUTPUT);

pinMode(Lane2[i], OUTPUT);

pinMode(Lane3[i], OUTPUT);

pinMode(Lane4[i], OUTPUT);

}

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

}

void loop()

{

digitalWrite(Lane1[2], HIGH);

digitalWrite(Lane3[0], HIGH);

digitalWrite(Lane4[0], HIGH);

digitalWrite(Lane2[0], HIGH);

delay(7000);

if (digitalRead(14) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else if(digitalRead(15) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else{

}

digitalWrite(Lane1[2], LOW);

digitalWrite(Lane3[0], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane3[1], HIGH);

delay(3000);

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(7000);

if (digitalRead(14) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else if(digitalRead(15) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else{

}

digitalWrite(Lane3[2], LOW);

digitalWrite(Lane4[0], LOW);

digitalWrite(Lane3[1], HIGH);

digitalWrite(Lane4[1], HIGH);

delay(3000);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane4[1], LOW);

digitalWrite(Lane3[0], HIGH);

digitalWrite(Lane4[2], HIGH);

delay(7000);

if (digitalRead(14) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else if(digitalRead(15) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else{

}

digitalWrite(Lane4[2], LOW);

digitalWrite(Lane2[0], LOW);

digitalWrite(Lane4[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(3000);

digitalWrite(Lane4[1], LOW);

digitalWrite(Lane2[1], LOW);

digitalWrite(Lane4[0], HIGH);

digitalWrite(Lane2[2], HIGH);

delay(7000);

if (digitalRead(14) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else if(digitalRead(15) == LOW) {

for (int i = 0; i < 3; i++)

{

digitalWrite(Lane1[i], LOW);

digitalWrite(Lane2[i], LOW);

digitalWrite(Lane3[i], LOW);

digitalWrite(Lane4[i], LOW);

}

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(7000);

void(\* resetFunc) (void) = 0; //declare reset function @ address 0

}

else{

}

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(3000);

digitalWrite(Lane2[1], LOW);

digitalWrite(Lane1[1], LOW);

}

digitalWrite(Lane1[2], LOW);

digitalWrite(Lane3[0], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane3[1], HIGH);

delay(2000);

digitalWrite(Lane1[1], LOW);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane1[0], HIGH);

digitalWrite(Lane3[2], HIGH);

delay(2000);

digitalWrite(Lane3[2], LOW);

digitalWrite(Lane3[1], HIGH);

delay(2000);

digitalWrite(Lane3[1], HIGH);

digitalWrite(Lane3[0], LOW );

delay(2000);

digitalWrite(Lane3[1], LOW);

digitalWrite(Lane3[0], HIGH);

digitalWrite(Lane2[0], LOW);

digitalWrite(Lane2[1], HIGH);

digitalWrite(Lane2[2], HIGH);

delay(2000);

digitalWrite(Lane2[1], LOW);

delay(2000);

digitalWrite(Lane1[0], LOW);

digitalWrite(Lane2[2], LOW);

digitalWrite(Lane1[1], HIGH);

digitalWrite(Lane2[1], HIGH);

delay(2000);

digitalWrite(Lane2[1], LOW);

digitalWrite(Lane1[1], LOW);