Four ways of Traffic Light Controller System Using Arduino Uno

Description:

"Four ways of Traffic Light Controller System Using Arduino Uno", in this project we have used Arduino uno board, 100 ohms Resistors, red LEDs, yellow LEDs and green LEDs as per in the traffic signal. In this system, instead of a manual traffic signal, we could go for an autonomous traffic light simulator, where the controller on each crossway will determine the duration of signal lights for the traffic of that particular road. The controller, on detecting the traffic amount will trigger the control and ensure that the green signal is given for the required duration. A simulation-based traffic light controller system is good for analyzing and making necessary modifications. Further development would be done on the traffic control system to become a more suitable and less costly model.

Components:

Arduino UNO
Resistor X 12 (100 ohms)
Red LEDs X 4
Yellow LEDs X 4
Green LEDs X 4
Ground

Block diagram:

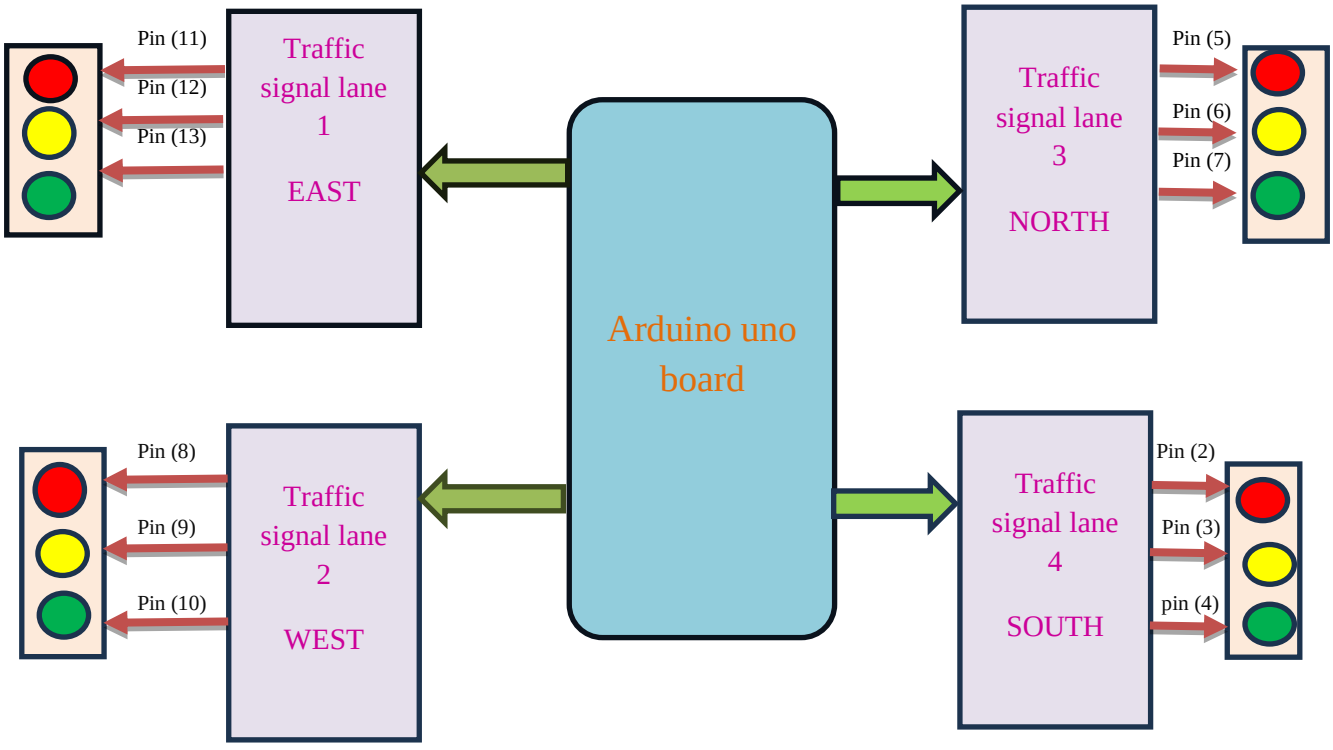
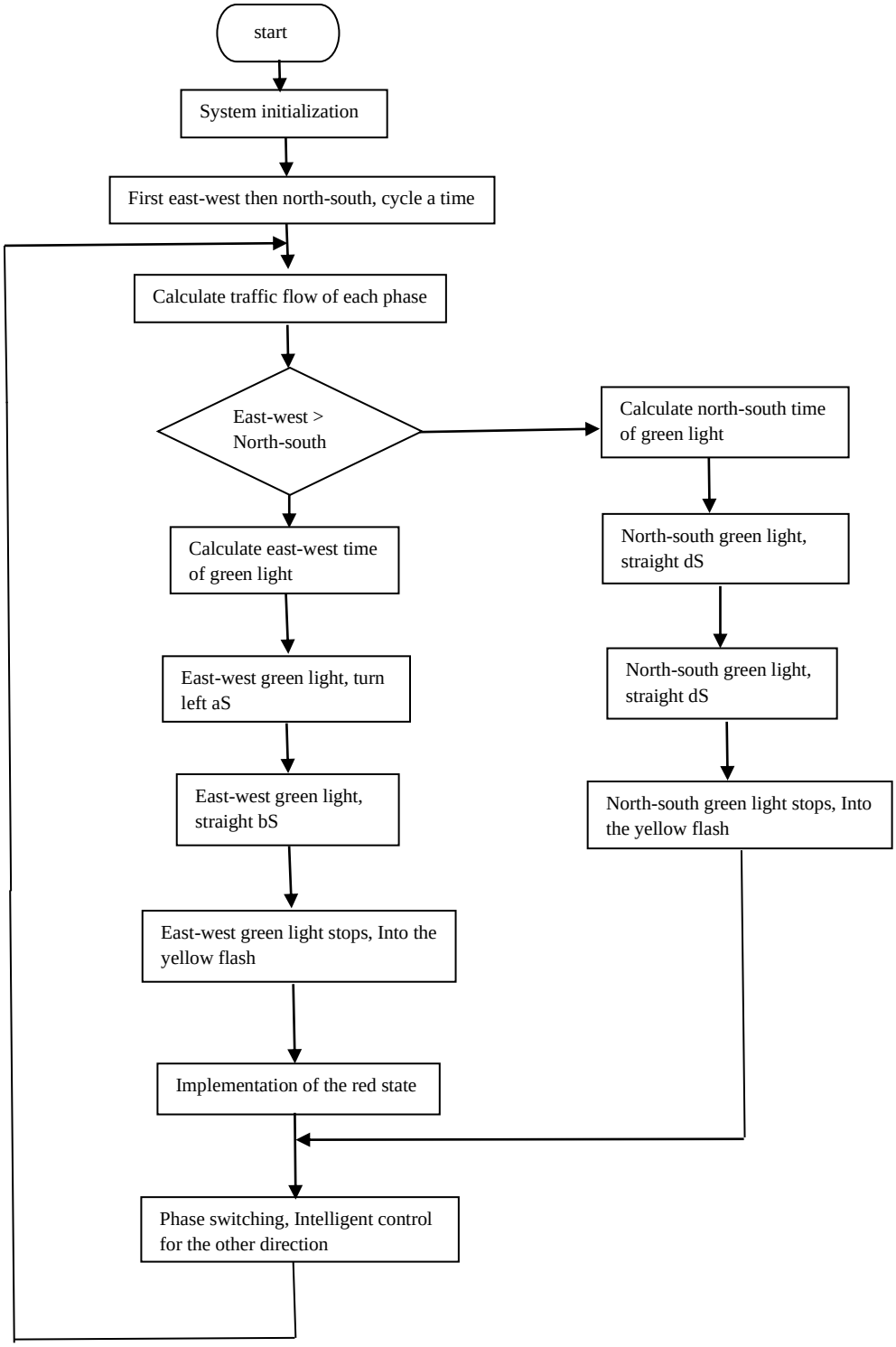


Table:

S, No	Description	Name	Type	Data Direction	Spectification	Remarks
01	Red LED(D1)	PD11	OUT	D0	Digital	Active High
02	Yellow LED (D1)	PD12	OUT	D0	Digital	Active High
03	Green LED (D1)	PD13	OUT	D0	Digital	Active High
04	Red LED(D2)	PD8	OUT	D0	Digital	Active High
05	Yellow LED (D2)	PD9	OUT	D0	Digital	Active High
06	Green LED (D2)	PD10	OUT	D0	Digital	Active High
07	Red LED(D3)	PD5	OUT	D0	Digital	Active High
08	Yellow LED (D3)	PD6	OUT	D0	Digital	Active High
09	Green LED (D3)	PD7	OUT	D0	Digital	Active High
10	Red LED(D4)	PD2	OUT	D0	Digital	Active High
11	Yellow LED (D4)	PD3	OUT	D0	Digital	Active High
12	Green LED (D4)	PD4	OUT	D0	Digital	Active High

Flowchart:

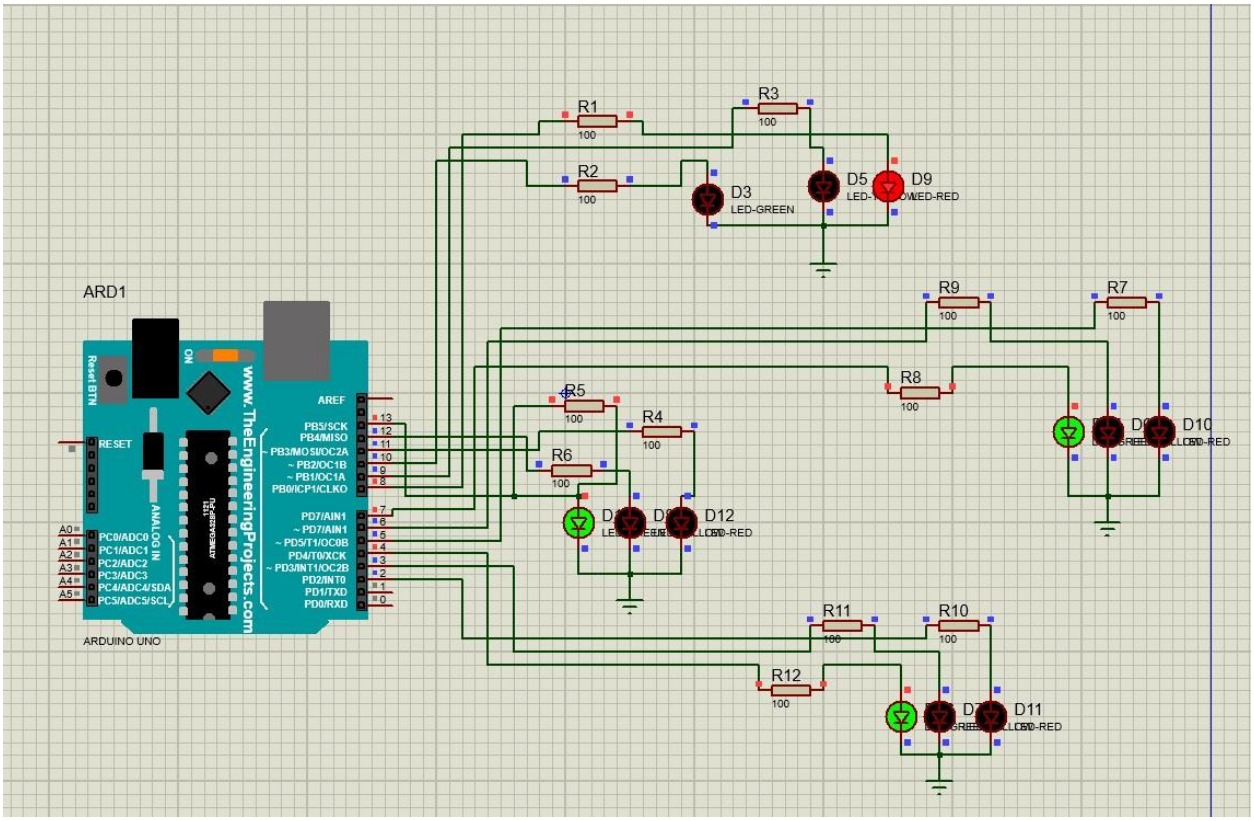

start
System initialization
First east-west then north-south, cycle a time
Calculate traffic flow of each phase
East-west > North-south
Calculate north-south time of green light
North-south green light, straight dS
North-south green light, straight dS
North-south green light stops, Into the yellow flash
Calculate east-west time of green light
East-west green light, turn left aS
East-west green light, straight bS
East-west green light stops, Into the yellow flash
Implementation of the red state
Phase switching, Intelligent control for the other direction

CODE:

```
int Lane1[] = {13,12,11}; // Lane 1 Red, Yellow and Green
int Lane2[] = {10,9,8};// Lane 2 Red, Yellow and Green
int Lane3[] = {7,6,5};// Lane 3 Red, Yellow and Green
int Lane4[] = {4,3,2};// Lane 4 Red, Yellow and Green

void setup()
{
  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(700);
  digitalWrite(Lane1[2], LOW);
  digitalWrite(Lane3[0], LOW);
  digitalWrite(Lane1[1], HIGH);
  digitalWrite(Lane3[1], HIGH);
  delay(700);
  digitalWrite(Lane1[1], LOW);
  digitalWrite(Lane3[1], LOW);
  digitalWrite(Lane1[0], HIGH);
  digitalWrite(Lane3[2], HIGH);
  delay(700);
```

```
 digitalWrite(Lane3[2], LOW);
 digitalWrite(Lane4[0], LOW);
 digitalWrite(Lane3[1], HIGH);
 digitalWrite(Lane4[1], HIGH);
 delay(700);
 digitalWrite(Lane3[1], LOW);
 digitalWrite(Lane4[1], LOW);
 digitalWrite(Lane3[0], HIGH);
 digitalWrite(Lane4[2], HIGH);
 delay(700);
 digitalWrite(Lane4[2], LOW);
 digitalWrite(Lane2[0], LOW);
 digitalWrite(Lane4[1], HIGH);
 digitalWrite(Lane2[1], HIGH);
 delay(700);
 digitalWrite(Lane4[1], LOW);
 digitalWrite(Lane2[1], LOW);
 digitalWrite(Lane4[0], HIGH);
 digitalWrite(Lane2[2], HIGH);
 delay(700);
 digitalWrite(Lane1[0], LOW);
 digitalWrite(Lane2[2], LOW);
 digitalWrite(Lane1[1], HIGH);
 digitalWrite(Lane2[1], HIGH);
 delay(700);
 digitalWrite(Lane2[1], LOW);
 digitalWrite(Lane1[1], LOW);
}
```

SCHEMATIC:



ANOTHER DESIGN: