VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

L5 + L6

IoT Fundamentals – ECE3501

Allen Ben Philipose – 18BIS0043

Lab Task – 3

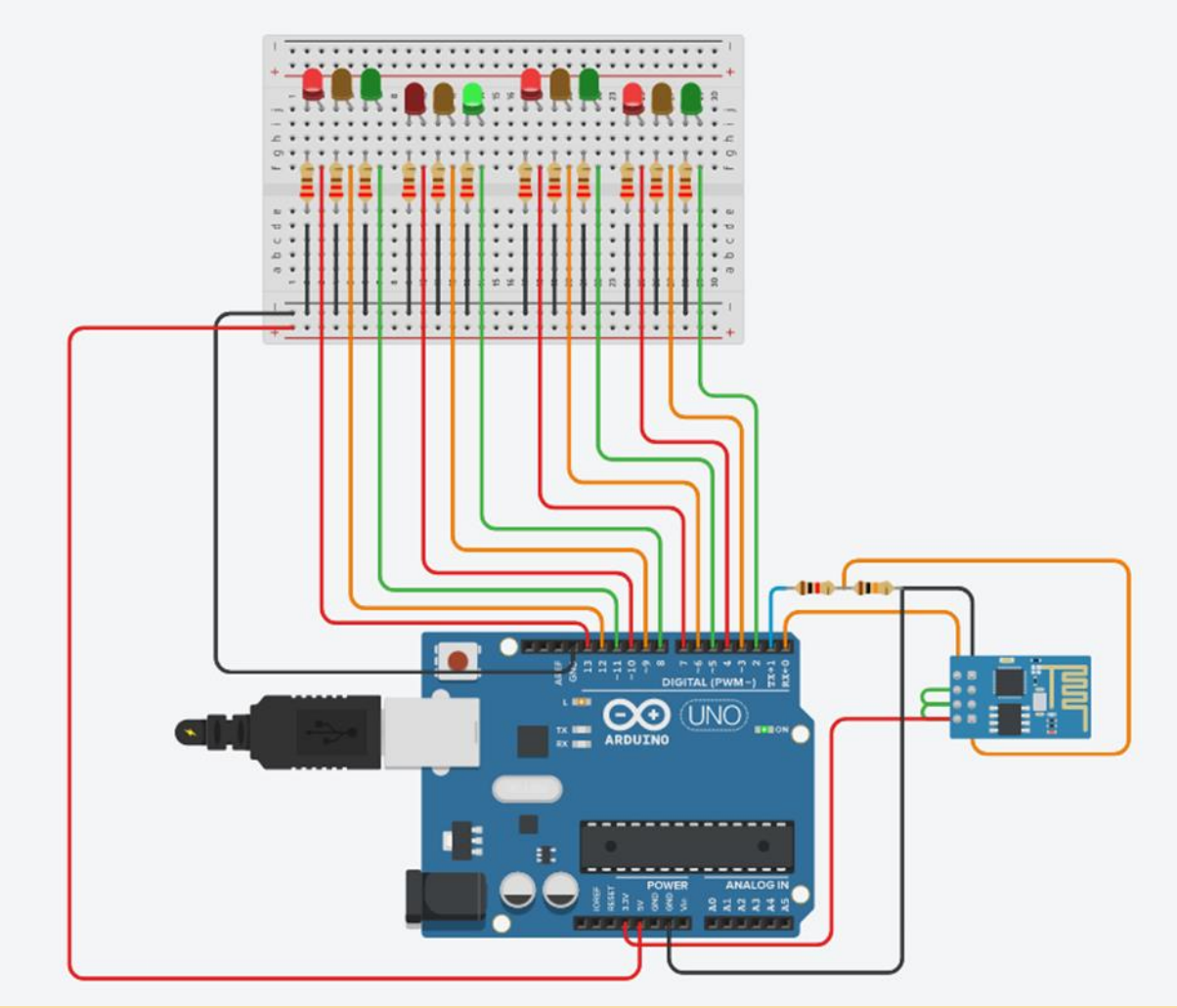
To: Prof. Suresh Chavhan

REGULAR TRAFFIC SIGNAL SYSTEM

Aim

To design a circuit using Arduino for simulating a regular traffic signal system, based on micro controller delay calculation only.

Circuit Diagram



Tools Required

Tinkercad – for simulating the connection and coding of the Arduino circuit

Output from Tinkercad

Traffic lights changing at regular intervals with effective and visible transitions, to simulate the real-life traffic system

Code

```
/*
String ssid     = "Simulator Wifi";  // SSID to connect to
String password = ""; // Our virtual wifi has no password
String host     = "api.thingspeak.com"; // Open Weather Map API
const int httpPort   = 80;
String      uri                                 = "/update?api_key=LCD1W68O1CZS8ULS&field1=";

int setupESP8266(void) {
  // Start our ESP8266 Serial Communication
```

```
  Serial.begin(115200);    // Serial connection
over USB to computer

  Serial.println("AT");    // Serial connection
on Tx / Rx port to ESP8266

  delay(10);        // Wait a little for the ESP
to respond

  if (!Serial.find("OK")) return 1;


  // Connect to 123D Circuits Simulator Wifi

  Serial.println("AT+CWJAP=\"" + ssid + "\",\""
+ password + "\"");

  delay(10);        // Wait a little for the ESP
to respond

  if (!Serial.find("OK")) return 2;


  // Open TCP connection to the host:

  Serial.println("AT+CIPSTART=\"TCP\",\""     +
host + "\"," + httpPort);

  delay(50);        // Wait a little for the ESP
to respond

  if (!Serial.find("OK")) return 3;

  return 0;

}
```

```
void anydata(void) {


  int   temp   =   map(analogRead(A0),20,358,-
40,125);


  // Construct our HTTP call
  String httpPacket = "GET " + uri + String(temp)
+ " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
  int length = httpPacket.length();


  // Send our message length
  Serial.print("AT+CIPSEND=");
  Serial.println(length);
  delay(10); // Wait a little for the ESP to
respond if (!Serial.find(">")) return -1;


  // Send our http request
  Serial.print(httpPacket);
  delay(10); // Wait a little for the ESP to
respond
  if (!Serial.find("SEND OK\r\n")) return;
}
```

```
*/
void setup() {
  for(int i=2;i<=13;i++)
  {pinMode(i, OUTPUT);}
  // setupESP8266();
}
void signal1() {
  digitalWrite(12, LOW);
  digitalWrite(4, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(11, HIGH);
  delay(4000);
  digitalWrite(12, HIGH);
  digitalWrite(11, LOW);
  digitalWrite(9, HIGH);
  digitalWrite(10, LOW);
  delay(1000);
  digitalWrite(12, LOW);
  digitalWrite(13, HIGH);
}
```

```
void signal2() {
  digitalWrite(9, LOW);
  digitalWrite(13, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(8, HIGH);
  delay(4000);
  digitalWrite(9, HIGH);
  digitalWrite(8, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
  delay(1000);
  digitalWrite(9, LOW);
  digitalWrite(10, HIGH);
}
void signal3() {
  digitalWrite(6, LOW);
  digitalWrite(13, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
```

```
  delay(4000);
  digitalWrite(6, HIGH);
  digitalWrite(5, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  delay(1000);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
}
void signal4() {
  digitalWrite(3, LOW);
  digitalWrite(13, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(2, HIGH);
  delay(4000);
  digitalWrite(3, HIGH);
  digitalWrite(2, LOW);
  digitalWrite(12, HIGH);
  digitalWrite(13, LOW);
  delay(1000);
```

```
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
}
void loop() {
  signal1();
  signal2();
  signal3();
  signal4();
 // anydata();
 // delay(1000);
}
```

Observations

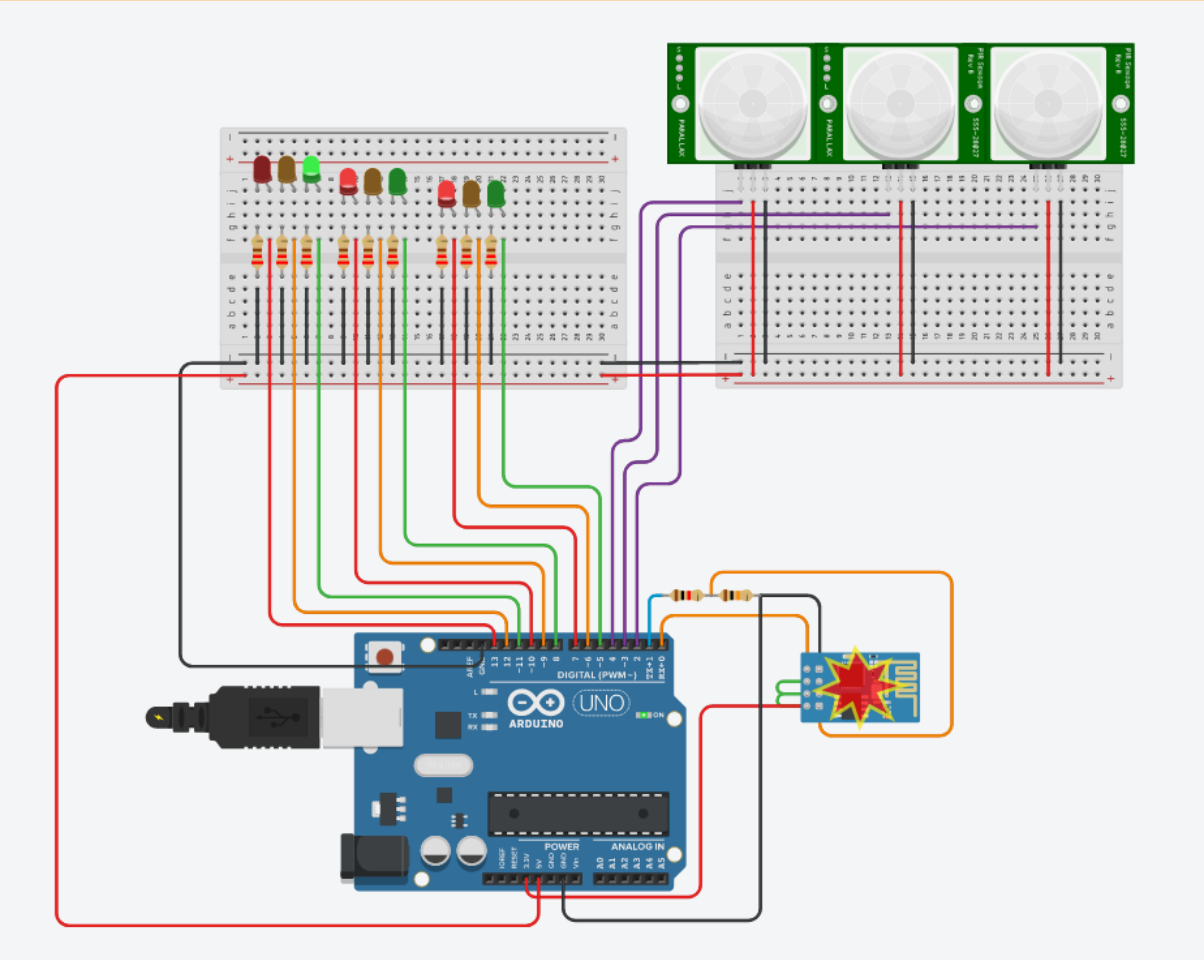
Program working as expected – minimal latency and user adjustable time delay which makes it easily manipulated and thus it is also effective for maintenance.

Conclusion

Therefore, by using Tinkercad, we simulated a circuit for a regular traffic signal system based on only delay calculation. Using PIR sensor in TinkerCad, we also can detect motion which can help to make the system smart and efficient – also implemented and attached to the same document.

SMART TRAFFIC SIGNAL SYSTEM

Circuit Diagram



Code

```
void setup() { // Initializing all pins
  for(int i=5;i<=13;i++){pinMode(i, OUTPUT);}
  for(int j=2;j<=4;j++) {pinMode(j, INPUT);}
```

```
  Serial.begin(9600);
}

void transition(int al){ // Single function for
all transitions
  int a=0;
  if(al==7) {a=al+9;}
  else {a=al;}
  digitalWrite((al-1), HIGH);
  digitalWrite((al-2), LOW);
  digitalWrite((a-4), HIGH);
  digitalWrite((a-3), LOW);
  delay(300);
  digitalWrite((al-1), LOW);
  digitalWrite(al, HIGH);
}

void on(int al) { // Single function for
turning on signals
  for(int x=5;x<=13;x++) {digitalWrite(x,LOW);}
  int a=0,b=0;
```

```
  if(al==7) {a=al+9;b=al+9;}
  else if(al==10) {a=al;b=al+9;}
  else {a=al;b=al;}
  digitalWrite(al-2,HIGH);
  digitalWrite((a-3),HIGH);
  digitalWrite((b-6),HIGH);
  delay(1200);
}
void call(int al) {on(al); transition(al);}

void loop() {
  if(digitalRead(4)==HIGH &&
digitalRead(3)==LOW && digitalRead(2)==LOW) {
    Serial.println("Congestion at Signal 1");
    while(digitalRead(4)==HIGH) {on(13);}
    transition(13);
    Serial.println("Signal 1 Cleared");
    call(10); call(7);
  }
  else if(digitalRead(4)==LOW &&
digitalRead(3)==HIGH && digitalRead(2)==LOW) {
```

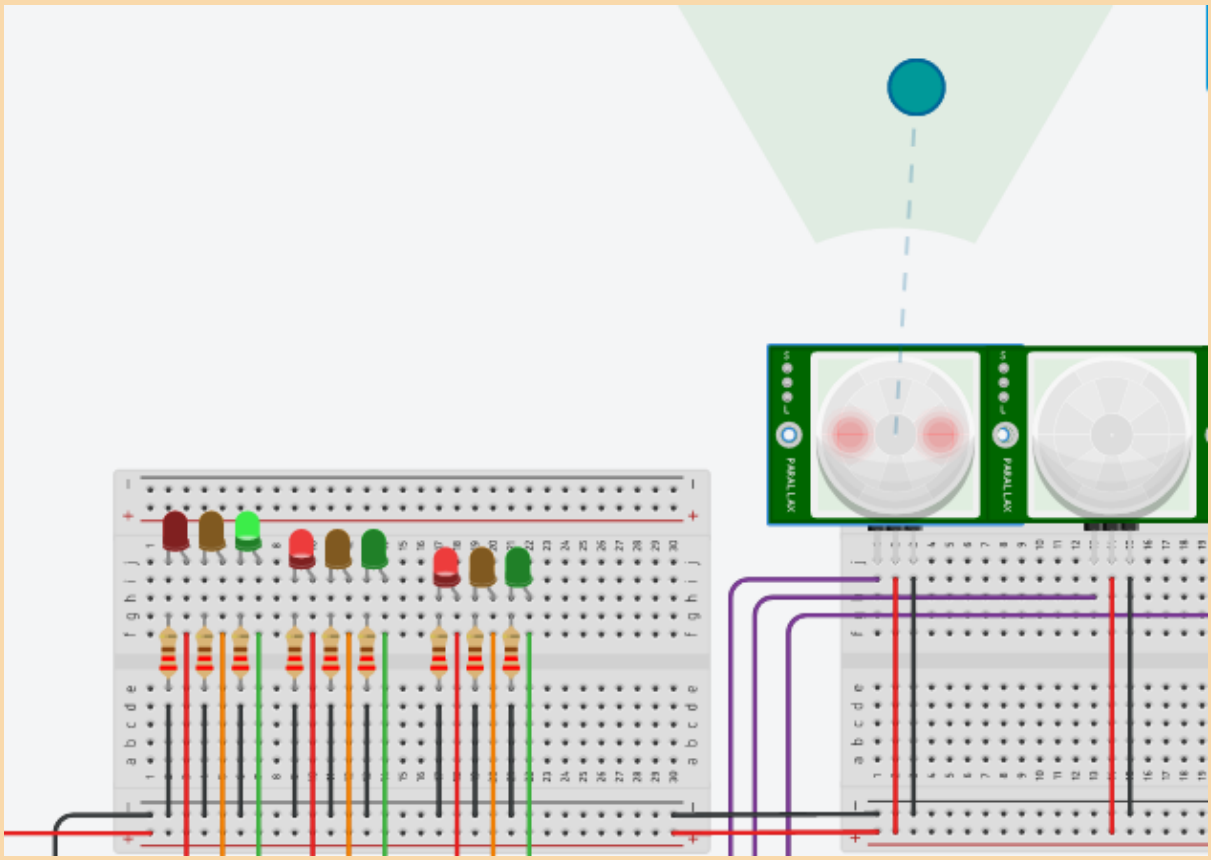
```
    call(13);
    Serial.println("Congestion at Signal 2");
    while(digitalRead(3)==HIGH){on(10);}
    transition(10);
    Serial.println("Signal 2 Cleared");
    call(7);
  }
  else if(digitalRead(4)==LOW &&
digitalRead(3)==LOW && digitalRead(2)==HIGH) {
    call(13); call(10);
    Serial.println("Congestion at Signal 3");
    while(digitalRead(2)==HIGH) {on(7);}
    transition(7);
    Serial.println("Signal 3 Cleared");
  }
  else {
    call(13);
    call(10);
    call(7);
  }
}
```

Output from TinkerCad

Serial Monitor

```
Congestion at Signal 1
Signal 1 Cleared
```

Working circuit



The red glow shows that the sensor is active and is detecting the traffic congestion at signal 1. This turns the signal green until the sensor becomes inactive – after which, the regular cycle repeats.