

Digital Systems Project

TRAFFIC Lights

Introduction to Computer Engineering | Semester 2 2017

Lecturer: Damith Herath Thanh Phan u3175008

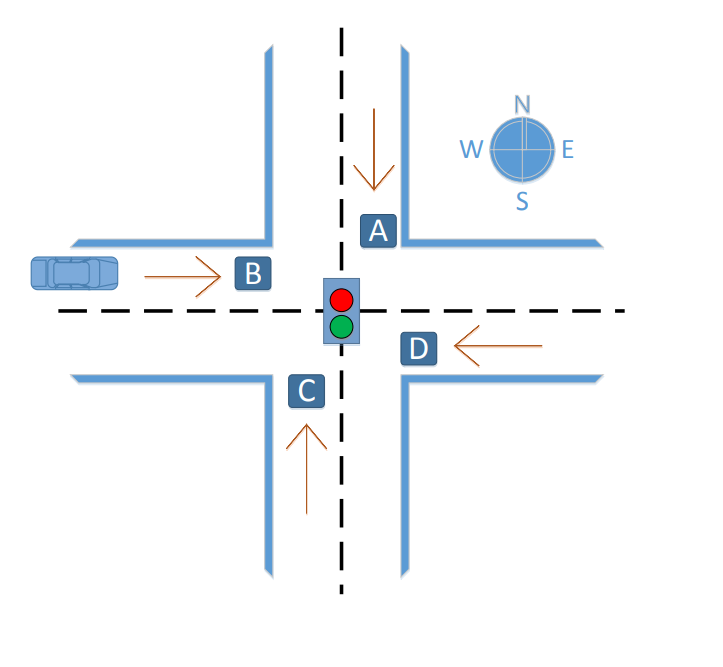
****

## Introduction

The purpose of this projects is built a small model of the traffic light by using knowledge about the basic features of a microcontroller, associated peripherals, and programming instructions to solve a simple problem. Using a project-based approach, the implement of an Arduino single-board Microcontroller based system as the final project.

**Traffic Light Rules**

Figure below shows the intersection of a highway with a local street. Vehicle detection sensors are placed along lanes B and D (highway) and lanes A and C (Street). These sensor outputs are LOW (0) when no car is present and HIGH (1) when a car is present. The intersection traffic light is to be controlled according to the following logic:

• The east-west (E-W) traffic light will be green whenever both lanes D and B are occupied.

• The E-W light will be green whenever either D or B is occupied but lanes A and C are not both occupied.

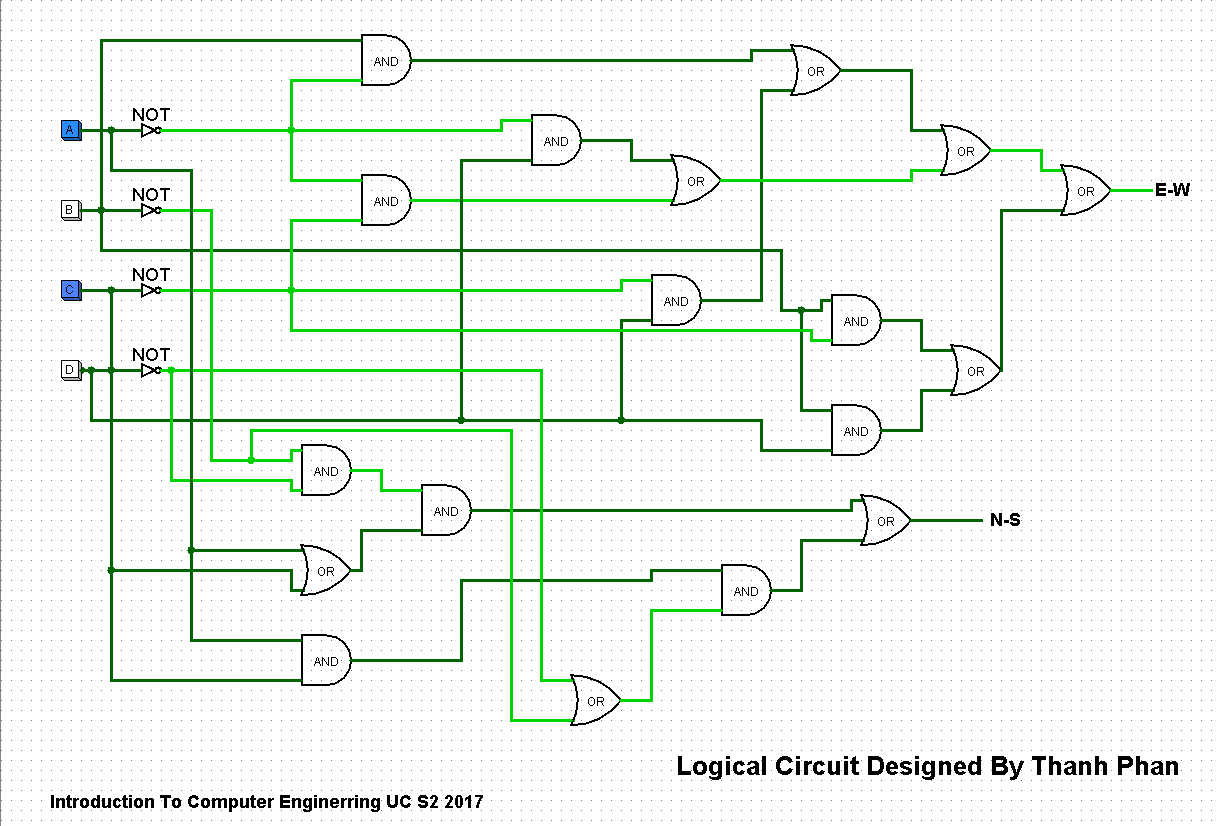
• The north-south (N-S) light will be green whenever both lanes A and C are occupied but D and B are not both occupied.

• The N-S light will also be green when either A or C is occupied while D and B are both vacant.

• The E-W light will be green when no vehicles are present.

**Part A: The Logic System (50 marks)**

By using sensor outputs A, B, C, and D as inputs, to control the traffic light the logic circuit was designed, with two outputs N-S and E-W that go HIGH when the corresponding light is to be green.



There are 8 different cases from the 4 inputs A, B, C and D to match all of the rules of the traffic lights. But in the table below there are 16 different cases, by including some of the special cases that haven’t had in the rules above.

**Truth Table for Traffic Light. (8 marks)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input | | | | Output | |
| A | B | C | D | X(Street) | Y(Highway) |
| 0 | 0 | 0 | 0 | **1** | 0 |
| 0 | 0 | 0 | 1 | **1** | 0 |
| 0 | 0 | 1 | 0 | 0 | **1** |
| 0 | 0 | 1 | 1 | **1** | 0 |
| 0 | 1 | 0 | 0 | **1** | 0 |
| 0 | 1 | 0 | 1 | **1** | 0 |
| 0 | 1 | 1 | 0 | **1** | 0 |
| 0 | 1 | 1 | 1 | **1** | 0 |
| 1 | 0 | 0 | 0 | 0 | **1** |
| 1 | 0 | 0 | 1 | **1** | 0 |
| 1 | 0 | 1 | 0 | 0 | **1** |
| 1 | 0 | 1 | 1 | 0 | **1** |
| 1 | 1 | 0 | 0 | **1** | 0 |
| 1 | 1 | 0 | 1 | **1** | 0 |
| 1 | 1 | 1 | 0 | 0 | **1** |
| 1 | 1 | 1 | 1 | **1** | 0 |

**Meaning:** BD = West-East = Highway = X

AC = North-South = Street = Y

Input 0 = Green Input 1 = Red Purple = to whenever the output is 1

**The Sum-of-Product form for the circuit without any simplification. By reusing the Truth Table above (6 marks)**

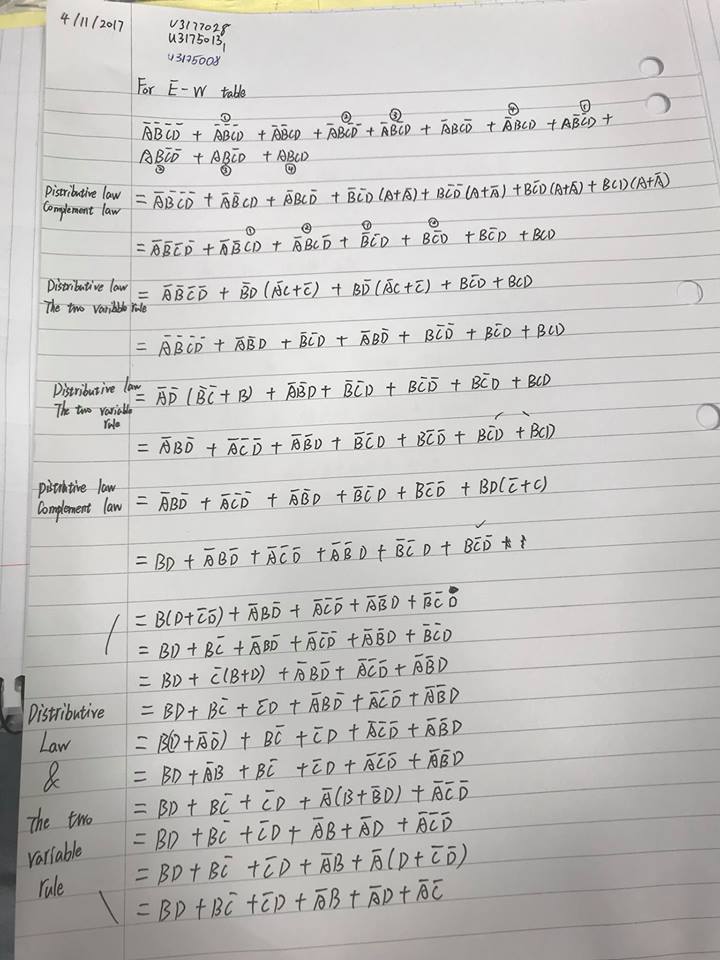
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Input | | | | Output | | Sum of Product |
| A | B | C | D | X(Street) | Y(Highway) |
| 0 | 0 | 0 | 0 | **1** | 0 | ~A\*~B\*~C\*~D |
| 0 | 0 | 0 | 1 | **1** | 0 | ~A\*~B\*~C\*D |
| 0x | 0 | 1 | 0 | 0 | **1** | **~A\*~B\*C\*~D** |
| 0x | 0 | 1 | 1 | **1** | 0 | ~A**\***~B**\***C**\***D |
| 0 | 1 | 0 | 0 | **1** | 0 | ~A**\***B**\***~C**\***~D |
| 0 | 1 | 0 | 1 | **1** | 0 | ~A**\***B**\***~C**\***D |
| 0x | 1 | 1 | 0 | **1** | 0 | ~A**\***B**\***C**\***~D |
| 0x | 1 | 1 | 1 | **1** | 0 | ~A**\***B**\***C**\***D |
| 1 | 0 | 0 | 0 | 0 | **1** | **A\*~B\*~C\*~D** |
| 1 | 0 | 0 | 1 | **1** | 0 | **A\*~B\*~C\*D** |
| 1x | 0 | 1 | 0 | 0 | **1** | **A\*~B\*C\*~D** |
| 1x | 0 | 1 | 1 | 0 | **1** | **A\*~B\*C\*D** |
| 1 | 1 | 0 | 0 | **1** | 0 | A**\***B**\***~C**\***~D |
| 1 | 1 | 0 | 1 | **1** | 0 | A**\***B**\***~C**\***D |
| 1x | 1 | 1 | 0 | 0 | **1** | **A\*B\*C\*~D** |
| 1x | 1 | 1 | 1 | **1** | 0 | A**\***B**\***C**\***D |

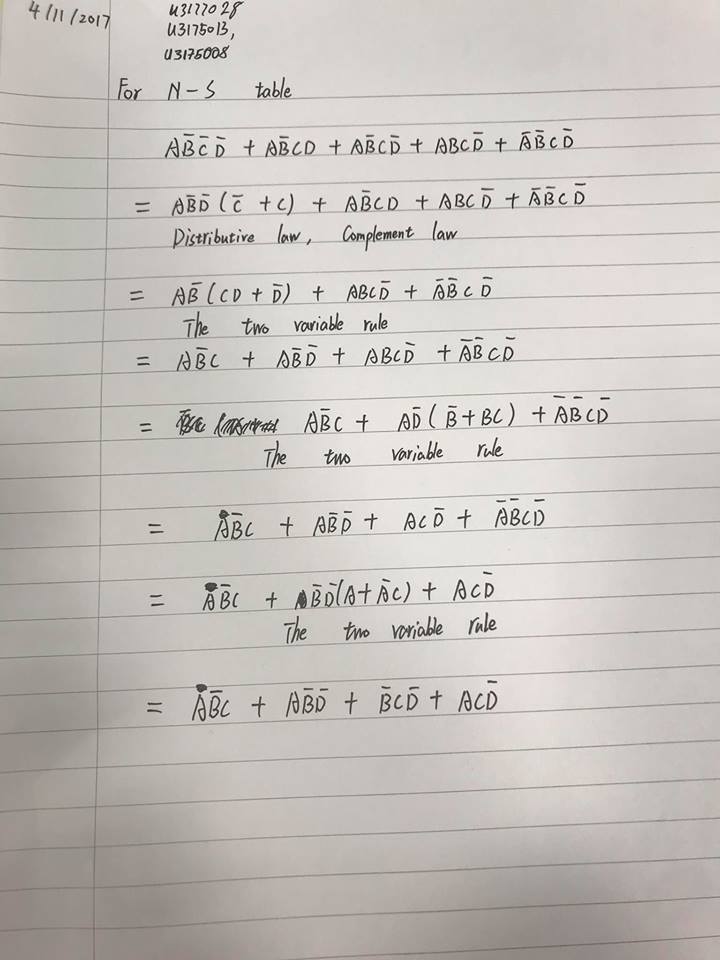
**Meaning:**

~A = o and A = 1; ~B = 0 and B = 1; ~C= 0 and C =1; ~D= 0 and D =1;

\*= And.

Yellow= West-East = Highway = X; Brown = North-South = Street = Y

**After the Sum-of-Product form for the circuit without any simplification then we** **Simplified logic expression obtained in Second table above by using logical equivalent**

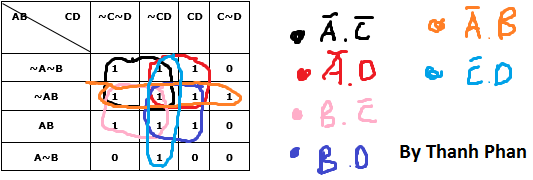


**Simplified** by Thanh, Ronnie, Lovevish

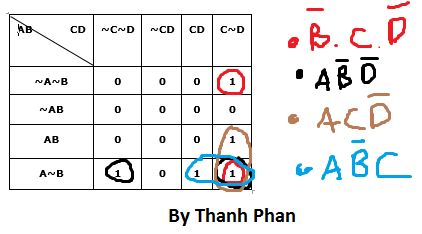
**Written** by Ronnie

**K-map was used as an alternative tool to simplify the logic expression achieved in First table above.**

K-map table for **STREET**



K-map table for **HIGH-WAY**

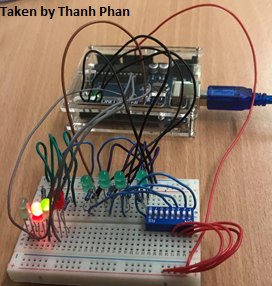


**How the circuit would operate if the sensor located at section C is stuck HIGH by its own or with others section.**

* At section C if the sensor is stuck high then traffic light in high-way will turn red and the traffic light will turn green for the street.
* At section C if the sensor is stuck high and D also, then traffic light in high-way will turn green and the traffic light will turn red for the street.
* At section C if the sensor is stuck high and B also, then traffic light in high-way will turn green and the traffic light will turn red for the street.
* At section C if the sensor is stuck high and B, D also, then traffic light in high-way will turn green and the traffic light will turn red for the street.
* At section C if the sensor is stuck high and A also, then traffic light in high-way will turn red and the traffic light will turn green for the street.
* At section C if the sensor is stuck high and A, D also, then traffic light in high-way will turn red and the traffic light will turn green for the street.
* At section C if the sensor is stuck high and A, B also, then traffic light in high-way will turn red and the traffic light will turn green for the street.
* At section C if the sensor is stuck high and A, B, D also, then traffic light in high-way will turn green and the traffic light will turn red for the street.

**Part B: Arduino Implementation**

After created the truth table to match all of the rules from the traffic light then we will a small model of it and using Cables, Swtich, Led lights, breadboad, Arduino Uno platform, and specially use C programing language to make it work. The image below is the complete look of the circuit built by the requited materials.



**Below is the program to make this circuit work**

//Author: Thanh Phan u3175008

//Version: 11/11/17

int redLight1 = 3; // East-West Red Light

int greenLight1 = 1; // East-West Green Light

int redLight2 = 4; // North-South Red Light

int greenLight2 = 2; // North-South Green Light

int switch1 = 8; // North-South A Road

int switch2 = 9; // East-West B Road

int switch3 = 10; // North-South C Road

int switch4 = 11; // East-West D Road

int Value = 0; // To store the input of switch 1 input A

int Value2 = 0; // To store the input of switch 2 input B

int Value3 = 0; // To store the input of switch 3 input C

int Value4 = 0; // To store the input of switch 4 input D

void setup() //setup the mode for the lights and the Switches

{

 pinMode(redLight1, OUTPUT);

 pinMode(greenLight1, OUTPUT);

 pinMode(redLight2, OUTPUT);

 pinMode(greenLight2, OUTPUT);

 pinMode(switch1, INPUT);

 pinMode(switch2, INPUT);

 pinMode(switch3, INPUT);

 pinMode(switch4, INPUT);

}

void loop()

{

 Value = digitalRead(switch1); // First Switch for input A

 Value2 = digitalRead(switch2); // Second Switch for input B

 Value3 = digitalRead(switch3); // Third Switch for input C

 Value4 = digitalRead(switch4); // Fourth Switch for input D

   if (Value == HIGH && Value2 == HIGH && Value3 == HIGH && Value4 == HIGH) // Case 1

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == LOW && Value2 == LOW && Value3 == LOW && Value4 == HIGH) // Case 2

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW);    // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == LOW && Value2 == LOW && Value3 == HIGH && Value4 == LOW) // Case 3

 {

   digitalWrite(redLight1, HIGH); // RED-LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(greenLight1, LOW); // GREEN LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(redLight2, LOW); // RED LIGHT = STREET = NS = LOW = OFF

   digitalWrite(greenLight2, HIGH); // GREEN LIGHT = STREET = NS = HIGH = ON

 }

 else if (Value == LOW && Value2 == HIGH && Value3 == LOW && Value4 == LOW) // Case 4

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW);   // GREEN LIGHT = STREET = NS= LOW = OFF

 }

 else if (Value == LOW && Value2 == HIGH && Value3 == LOW && Value4 == HIGH) // Case 5

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == LOW && Value2 == HIGH && Value3 == HIGH && Value4 == HIGH) // Case 6

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == HIGH && Value2 == LOW && Value3 == LOW && Value4 == LOW) // Case 7

 {

   digitalWrite(redLight1, HIGH); // RED-LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(greenLight1, LOW); // GREEN LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(redLight2, LOW); // RED LIGHT = STREET = NS = LOW = OFF

   digitalWrite(greenLight2, HIGH); // GREEN LIGHT = STREET = NS = HIGH = ON

 }

 else if (Value == HIGH && Value2 == LOW && Value3 == HIGH && Value4 == LOW) // Case 8

 {

   digitalWrite(redLight1, HIGH); // RED-LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(greenLight1, LOW); // GREEN LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(redLight2, LOW); // RED LIGHT = STREET = NS = LOW = OFF

   digitalWrite(greenLight2, HIGH); // GREEN LIGHT = STREET = NS = HIGH = ON

 }

 else if (Value == HIGH && Value2 == HIGH && Value3 == LOW && Value4 == HIGH) // Case 9

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == LOW && Value2 == LOW && Value3 == HIGH && Value4 == HIGH) // Case 10

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == LOW && Value2 == HIGH && Value3 == HIGH && Value4 == LOW) // Case 11

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == HIGH && Value2 == LOW && Value3 == LOW && Value4 == HIGH) // Case 12

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == HIGH && Value2 == LOW && Value3 == HIGH && Value4 == HIGH) // Case 13

 {

   digitalWrite(redLight1, HIGH); // RED-LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(greenLight1, LOW); // GREEN LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(redLight2, LOW); // RED LIGHT = STREET = NS = LOW = OFF

   digitalWrite(greenLight2, HIGH); // GREEN LIGHT = STREET = NS = HIGH = ON

 }

 else if (Value == HIGH && Value2 == HIGH && Value3 == LOW && Value4 == LOW) // Case 14

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

 else if (Value == HIGH && Value2 == HIGH && Value3 == HIGH && Value4 == LOW) // Case 15

 {

   digitalWrite(redLight1, HIGH); // RED-LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(greenLight1, LOW); // GREEN LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(redLight2, LOW); // RED LIGHT = STREET = NS = LOW = OFF

   digitalWrite(greenLight2, HIGH); // GREEN LIGHT = STREET = NS

 }

 else // Case 16

 {

   digitalWrite(redLight1, LOW); // RED-LIGHT = HIGH WAY = EW = LOW = OFF

   digitalWrite(greenLight1, HIGH); // GREEN LIGHT = HIGH WAY = EW = HIGH = ON

   digitalWrite(redLight2, HIGH); // RED LIGHT = STREET = NS = HIGH = ON

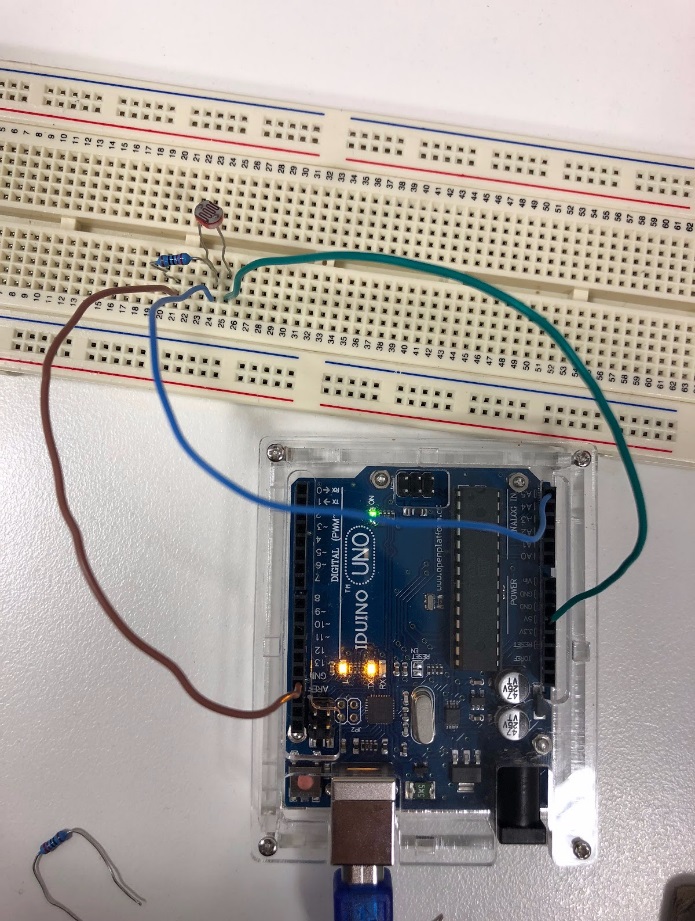
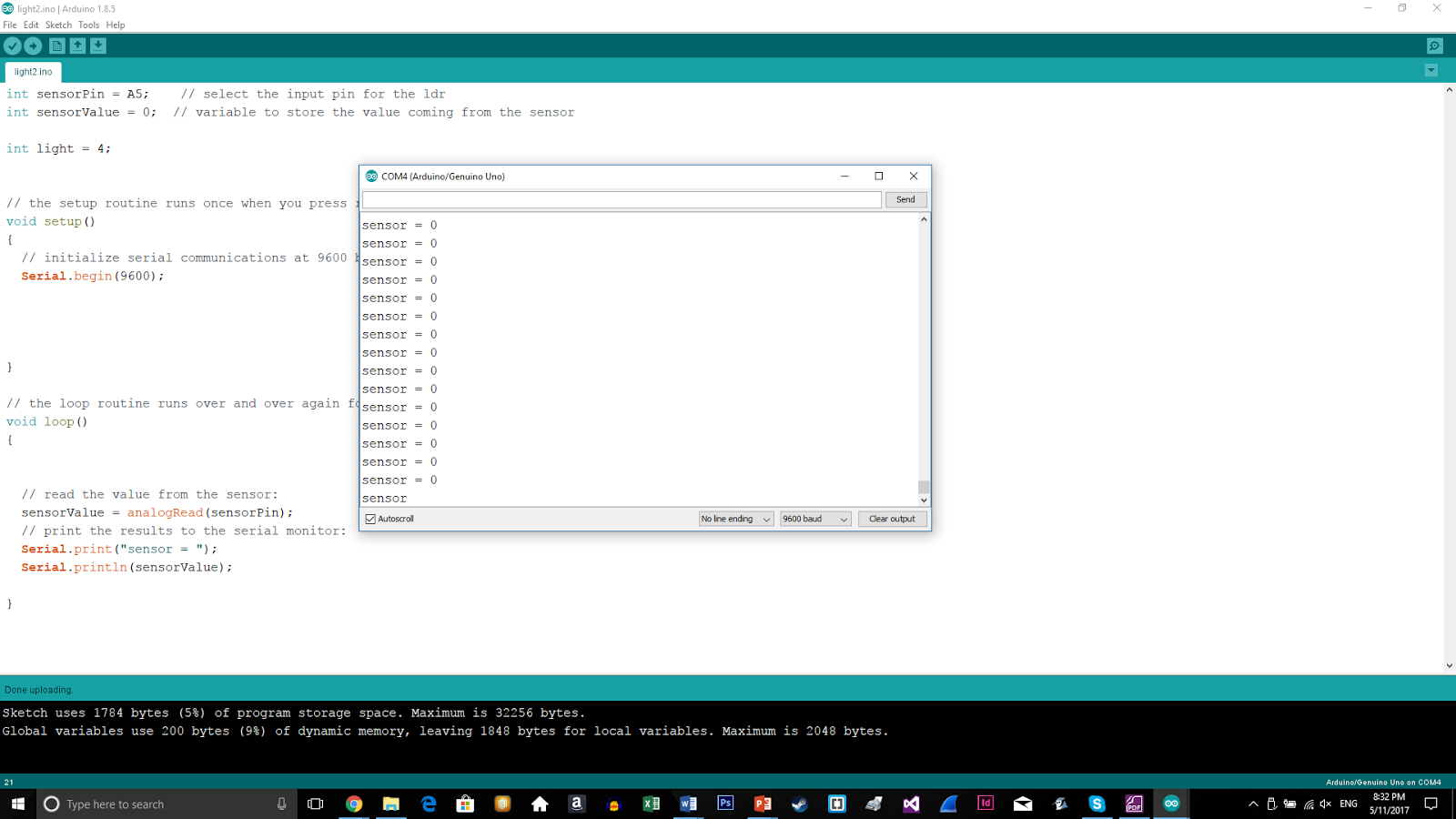
   digitalWrite(greenLight2, LOW); // GREEN LIGHT = STREET = NS = LOW = OFF

 }

}

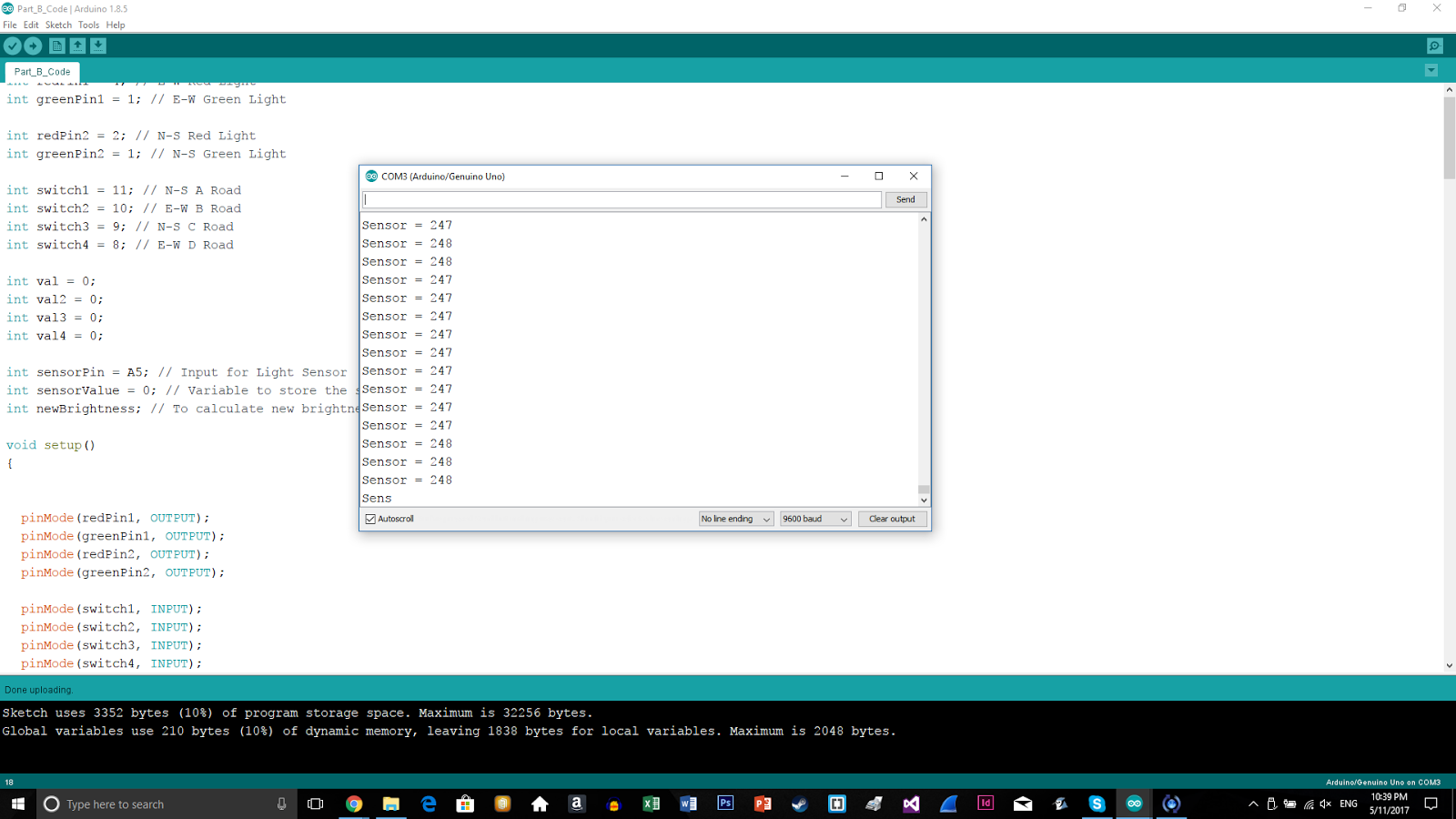
**Part C: Extended Arduino Implementation**

This is a picture of the setup of the light sensor and will be used to record the input values when it is in complete darkness. Below is the taken a screenshot of the output values from the light sensor when it was in completed darkness.



The values were obtained by covering the sensor completely

**Photos Taken** by Lovevish in Part C



Values without covering the sensor

**//Sensor code:**

//Author: Thanh Phan u3175008

//Version: 11/11/17

int sensorPin = A1;    // select the input pin for the ldr

int sensorValue = 0;  // variable to store the value coming from the sensor

int light = 4;

// the setup routine runs once when you press reset:

void setup()

{

 // initialize serial communications at 9600 bps:

 Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop()

{

 // read the value from the sensor:

 sensorValue = analogRead(sensorPin);

 // print the results to the serial monitor:

 Serial.print("sensor = ");

 Serial.println(sensorValue);

}

**CONCLUSION**

At the end the end of this project I have learned, got more knowledge and confident to build a small simple logical circuit. Specially understood how the traffic-lights work in the real life and apply these things/technique to the future traffic lights or other digital devices.