

Designing an Automatic Traffic Light System

by Ellis Ashton

Submission date: 12-Dec-2024 09:36AM (UTC+0000)

Submission ID: 245703122

File name: 58247_Ellis_Ashton_Designing_an_Automatic_Traffic_Light_System_634321_955777247.docx
(1.21M)

Word count: 2348

Character count: 12593

Introduction

The task has been set to generate an electrical circuit with a corresponding written program to control a T-junction set of traffic lights. This T-junction consists of 3 roads that converge along with pedestrian crossings, each traffic light should include a 7-segment display with a counter to allow for the pedestrians to cross the road safely and successfully. The lights and operations included must adhere to the following sequencing:

- Traffic lights to transition to display for 10 seconds each in a predictable and repeatable fashion.
- A 7-segment display at each side of the junction with counts from 40 seconds.
- 10 seconds and the end of the traffic light cycle to allow for pedestrian crossing.
- At the time of the pedestrians crossing, the traffic lights must all show red or not display anything.

There several ancillary programs required to complete this  and program. This includes Proteus, PWC and CCS compiler to first build the electronic circuit, then to produce the support program and then finally to compile to program so that it can be imported to Proteus and executed. 

Components research

To complete this project there are several components required to create a functional system, these are being simulated to validate the code that has been written.

PIC16F877A

A microcontroller is a control device that can be incorporated into an electronic circuit as it contains a miniature circuit board that is capable of storing a program in memory and contains a processor that can enact the instruction defined by the program. This microcontroller has many characteristics that make it suitable for the task, physically the microcontroller has a suitable number of pins to accommodate the number of outputs required for this system to operate.

Advantages of PIC16F877A

- 16 input/output pins
- Compatible with software used (Proteus 8 professional)
- Sufficient RAM and ROM to store the program 14Kb (operational program utilises 5% of available RAM and 5% of available ROM)
- In comparison with other microcontrollers, this is a low-cost component 

[1]

Disadvantages of PIC16F877A

- The maximum processing speed is 20Mhz, this is suitable for the project, but it is operating at its maximum processing speed.
- The assembly language used to program this model can be challenging to beginners.

[2]

Although this specific model of microcontroller has some disadvantages, the characteristics are sufficient to complete the project. This microcontroller is intuitive and provides troubleshooting capability that can provide assistance with the program. This a helpful and intuitive feature for a programming beginner and increases the accessibility to writing programs to operate smaller systems.

7-segment display

A 7-segment display is a grouping of LED's that can be energised in different combinations to display digits 0-9. Many 7-segment displays, including the display being used for the current project, use an 8th LED to display a decimal point after the digit. This type of display is widely used, often as a counter or a timer where many displays can be grouped together to display for than one digit i.e a digital clock display. The display has electrical pins that are connected to each of the LED's that can be energised in different combinations to allow the display to present the shape of digits. This type of display can be instructed using an 8bit binary or hexadecimal operation in the microcontroller that will energise the pins required to display the digits.

There are two common forms of 7-segment displays that differ in their wiring configuration, they are defined by either a common anode or common cathode. Those with common anodes have a permanently energised anode that supplies power to all the Light Emitting Diodes, the microcontroller provides a negative or ground the those energising to complete the circuit. Those that are made using a common cathode use a common negative/ground connection where the microcontroller provides power to the pins that are energising where the common cathode completes the circuit. Traffic light project uses a common cathode display where a common ground is included in the circuit for the LED's in the display. [3]

The 7-segment display is a suitable component to be used for this project as it allows for the display of the counter and can be grouped to produce a multiple digit counter. This is an intuitive system and operates using principle logic, making this the ideal component to use.

LED traffic light display

Traffic light displays are very commonly used across the world to control the flow of traffic. It consists of 3 lights of different colours – Red, Yellow and green. Each colour light is illuminated by LED's where the PVC cover of the LED's provides the colour, rather than using different colour LED's which reduces the production costs of the component.

The traffic light display uses a common cathode configuration where the microcontroller provides the power to energise the LED's in the desired sequence. The sequence is outlined within the program written for the microcontroller, the traffic light displays the three colours in a sequence that provides instruction for those who observe the display. The program of the microcontroller outlines the pins that are energised at different stages of the program.

The LED traffic light is the required component to simulate this system as it is an accurate representation of the project.

Software

Proteus

Proteus is a software design and simulation package that allows for the user to create circuits using simulated components. This software is used to create circuits using simulated components so that circuits can be designed accurately and improved using a low-risk trial and error method. Using this software there are many changes that can be made such as altering the layouts, wiring and characteristics of the components to ensure the optimisation of the circuit. The software provides both the physical and the characteristic composition of many different components including many microcontrollers and electronics components including the traffic lights, LED's and 7-segment displays providing all the requirements to complete this project.

This software package also has the capability to live simulate the electrical circuit to see the behaviour of the components. Once the circuits have been designed and built using proteus, the circuit can be simulated to understand how it operates. This can be used to troubleshoot a circuit or to optimise the design by understanding problems or easily substituting components. This capability extends to microcontrollers allowing the circuit to be built including the specific model of microcontroller, and a compiled program can be imported and simulated using Proteus. Utilising this feature allows for not only the circuit, but the program to be tested to understand the behaviour, troubleshoot and make improvements to the program. This is a time and cost efficient method of fine-tuning a program and a circuit, although this method requires an external software package to compile the program before it is imported into Proteus. [4]

Proteus proves to be a capable software for the traffic lights project, this program provides many options of microcontrollers and components. This also has the advantage of simulation which is essential to completing the project by simulating the entire system to debug and optimise the program and its components.

CCS PCW Compiler

A Code Composer Studio (CCS) is a program composing software tool that allows the user to input a program and to compile the code and to outline errors within the formatting of the program. This helps the user to find and correct the formatting errors before uploading them to the simulation. This combination of compiler and simulator help to reduce the time and cost of producing a program and complex circuit as alterations can be made very quickly to the program and then further testing and simulating can continue.

Although, this software does not assist with troubleshooting the program directly as it only provides assistance with the formatting issues. Therefore, knowledge of the program and operation of the coding language is required to troubleshoot more complex issues so that this program can function correctly.

CCS PCW Compiler is a suitable tool to use for this project as it is an intuitive software that integrates well with Proteus allowing for a more seamless experience with some assistance with troubleshooting.

Circuit design

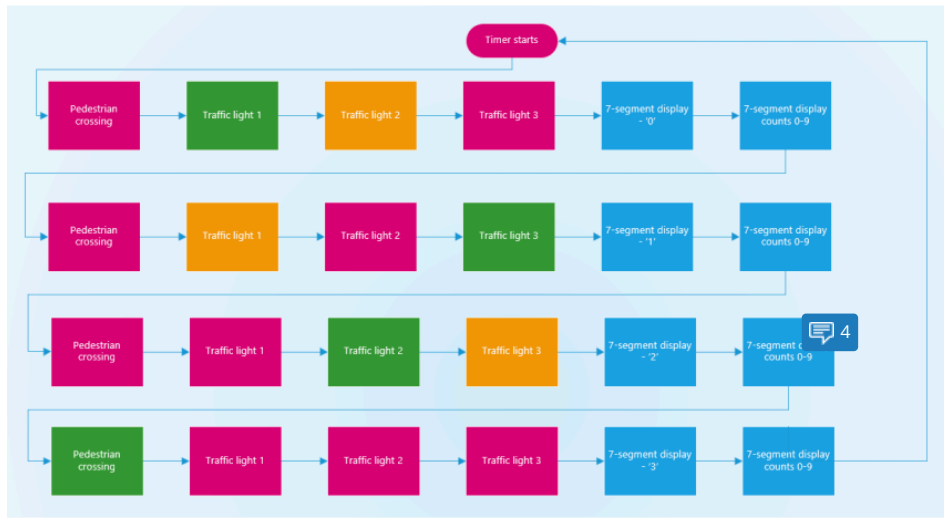
The circuit design for the T-junction traffic signals firstly requires the microcontroller, in this instance a PIC16F877A microcontroller was used. The microcontroller had four sets of pins (Pin sets A-D) each with 8 pins (Pins numbered 0-7), the three traffic signals are connected to Pin set C. The Red, amber and green LED anode pins of each traffic signal are connected to three pins on the microcontroller, these pins alternate with each of the traffic signals to ensure that a different colour LED is energised on each of the signals at any one time. One improvement to the basic operation of the program is to ensure that each of the traffic lights display red during the pedestrian crossing period, this required an additional pin to the red LED of each traffic light to allow all the red LED's to be energised when the green pedestrian LED is energised.

The pedestrian crossing lights are also connected to the pins in pin set C, these pins become energised when instructed by the program. The LED's are then connected to a ground to allow for completion of a circuit for the current to flow when energised.

The third type of component within the circuit are the 7 segment displays. The 7-segment displays require 8 pins to operate each of the LED's within, the pins for the LED's are connected to pin set D, the correct combination of energised pins is dictated by the program to display the numbers on the counter increasing every second. The first display counts from 0-9 representing the single integer, the second display counts from 0-4 increasing every 10 seconds to represent the second integer, this combination of two 7-segment displays placed next adjacent allow for the display of a counter the increases in seconds from 0-40. Each set of two displays are required to represent the same counts at each of the three traffic light positions so therefore can be replicated using the same pins in all three positions.

To ensure that the circuit is displayed clearly and to remove any ambiguity, the wiring has been reduced using remote pins to ensure that the components are displayed clearly, an image of the circuit design is displayed below:

Flow chart



Testing the program and circuit

Testing the program and circuit using the simulation tools allow for a system to be tested in operation, this simulated how the program will operate which allows for the system to be improved and to work out any bugs in the program/circuit.

When testing the system in simulation there were a combination of formatting errors and errors within the program. The formatting errors that were rectified included semi-colons that were required for the correct operation of the program and brackets that were incorrectly placed which can be categorised as typing errors. Also, the developing of the red lights energising during the pedestrian crossing green light displaying, required some changes to the program and circuit diagram in which the simulation allowed for an efficient process to incorporate this improvement into the system.

The compiling software package also gives insight into the formatting issues of the program which allow for a smoother testing process, some of the typing errors and formatting errors were outlined during the compiling process.

Conclusion

The T-junction traffic light system both an interesting and educational project to learn and understand the tools involved with generating a microcontroller system with a functional program. The components that were used are suitable for the task, if a more complex program were required with a faster processing capacity as the currently used microprocessor is operating at its maximum capability of 20MHz.

**Designing an Automatic Traffic
Light Controller for a T- Junction Road**

Ellis Ashton (2406269)

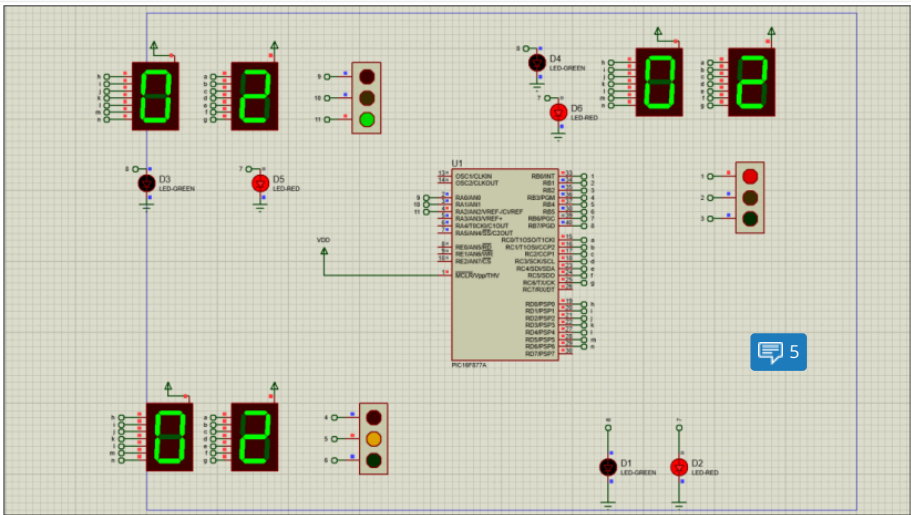
The Proteus software is intuitive to use and provided all the necessary components that were required for the circuit to be built efficiently with power supplies and ground references available within the circuit. The CCS PCW compiling software is an insightful tool allowing for a program to be written using a microcontroller directory that allows for the import of the logic library for the unit. This software separates each line of code allowing for formatting concerns to be outlined during the compiling process. As this program integrates with proteus seamlessly, the Hex file can then be directed from the proteus software and be encompassed into the simulation on Proteus.

With the combination of Proteus and CCS PCW compiler, the project was successfully completed to display three traffic light display which alternate in the LED displayed every 10 seconds resulting in a 30 second display of traffic lights, as this cycle completes the pedestrian red light is displayed to deter pedestrians from crossing whilst vehicles are moving. The final 10 seconds displays the green pedestrian light allowing for 10 seconds of pedestrian crossing whilst simultaneously displaying a red light on all traffic light junctions to ensure the pedestrians can cross safely.

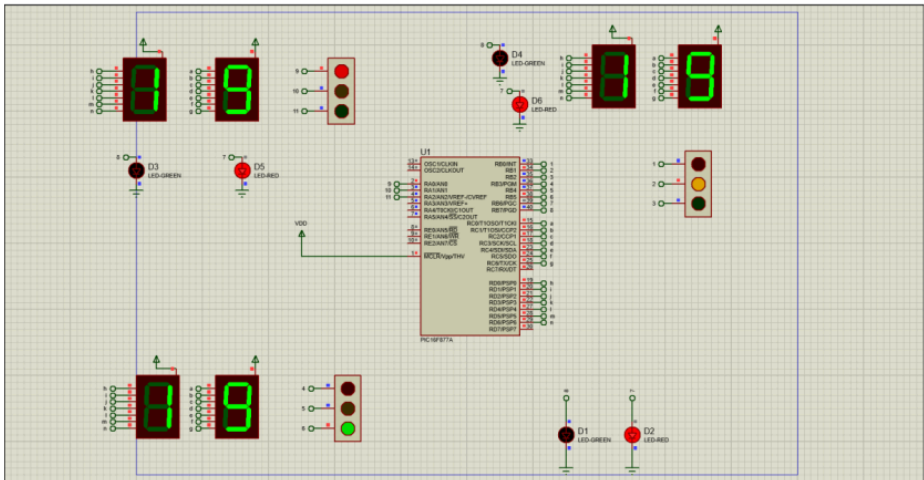
The program results in a safe and functional T-junction traffic light system that operates safely and efficiently for both the pedestrians and vehicles moving through the junction. Although, a further improvement that would make the system more efficient would be to install a push button at each of the pedestrian crossings to allow for the pedestrians to indicate their presence, if the button is not pressed in the initial 30 seconds of the traffic lights section of the program, then the pedestrian crossing 10 second section of the program would not be executed. This improvement would allow for the pedestrians to cross when present and for the vehicle traffic to keep moving if there were no pedestrians present attempting to cross.

Circuit in operation

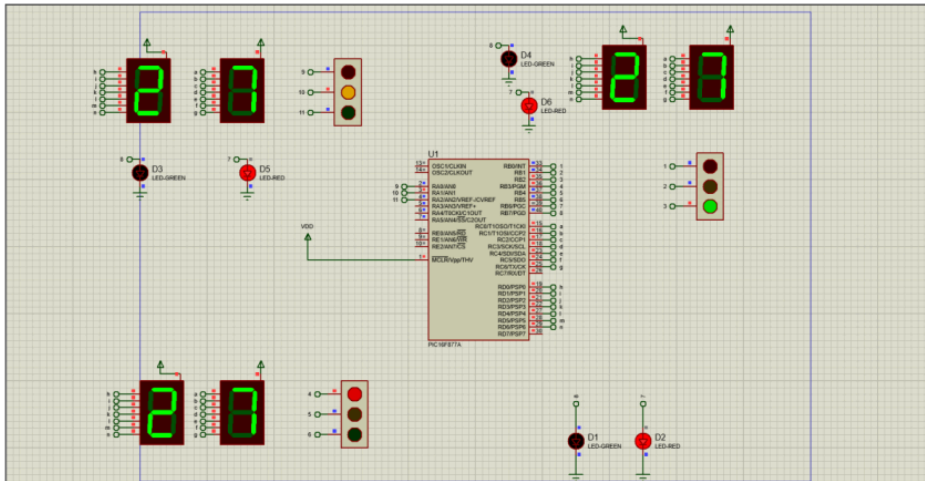
Condition 1 (0-10seconds)



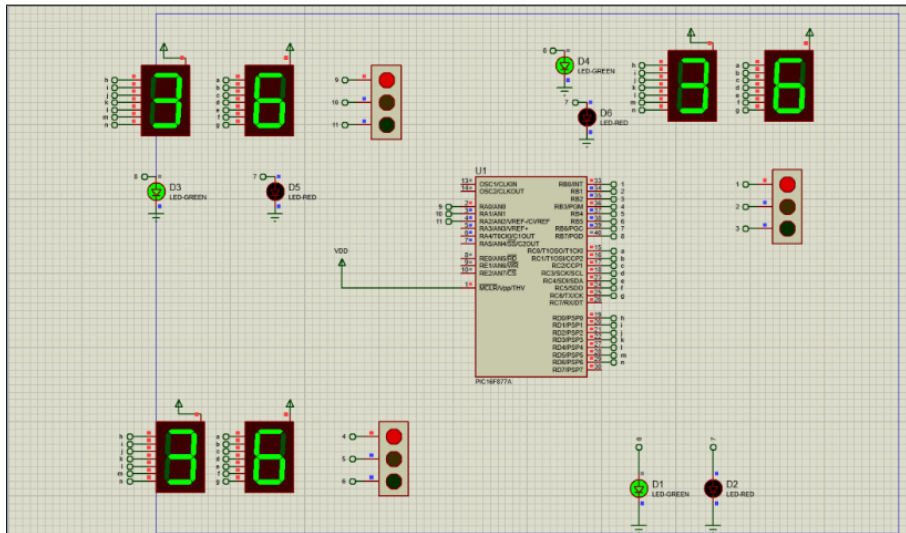
Condition 2 (11-20seconds)



Condition 3 (21-30seconds)



Condition 4 (31-40seconds)



Designing an Automatic Traffic Light Controller for a T- Junction Road

Ellis Ashton (2406269)

Program

```
1 #include <16F877A.h>
2 #fuses NOWDT, XT, NOPROTECT, NOLVP
3 #fuse delay(clock=2000000) // 20 MHz clock
4
5 // Define LED pins
6 #define LED_RED PIN_B0
7 #define LED_YELLOW PIN_B1
8 #define LED_GREEN PIN_B2
9 #define P_RED PIN_B6
10 #define P_GREEN PIN_B7
11 #define LED_RED2 PIN_B3
12 #define LED_YELLOW2 PIN_B4
13 #define LED_GREEN2 PIN_B5
14 #define LED_RED3 PIN_A0
15 #define LED_YELLOW3 PIN_A1
16 #define LED_GREEN3 PIN_A2
17
18
19
20 // 7-segment display digit patterns (assuming common cathode)
21 unsigned int display[10] = {0xC0, 0xF9, 0xA4, 0x80, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90};
22 int i;
23
24 void main()
25 {
26     set_tris_a(0x00); // Set PortA as input
27     output_a(0x00); // Initialise PORTA to 0
28     set_tris_b(0x00); // Set PORTB as output
29     output_b(0x00); // Initialise PORTB to 0
30     set_tris_c(0x00); // Set PORTC as output for 7-segment display
31     output_c(0x00); // Initialise port c to 0
32     set_tris_d(0x00); // Set PORTD as output for 7-segment display 2
33     output_d(0x00); // Initialise port d to 0
34     while(1)
35     {
36         // RED LED ON, Others off
37         for (i = 0; i < 10; i++)
38         {
39             output_d(0xC0); // '0'
40             output_c(display[i]); // 7-segment display with count 0-9
41             delay_ms(1000);
42             output_high(LED_RED); // Turn on red LED
43             output_low(LED_RED2); // Turn on red LED2
44             output_low(LED_RED3); // Turn on red LED3
45             output_low(LED_YELLOW); // Turn off yellow LED
46             output_high(LED_YELLOW2); // Turn off yellow LED2
47             output_low(LED_YELLOW3); // Turn off yellow LED3
48             output_low(LED_GREEN); // Turn off green LED
49             output_high(LED_GREEN2); // Turn off green LED2
50             output_low(LED_GREEN3); // Turn off green LED3
51             output_high(P_RED); // Turn pedestrian red light on
52             // output_low(P_GREEN); // Turn pedestrian green light off
53             delay_ms(1000); // 1 second delay
54         }
55
56         // Yellow LED ON, Others off
57         for (i = 0; i < 10; i++)
58         {
59             output_d(0xF9); // '1'
60             output_c(display[i]); // 7-segment display with count 0-9
61             output_low(LED_RED); // Turn on red LED
62             output_low(LED_RED2); // Turn on red LED2
63             output_high(LED_RED3); // Turn on red LED3
64             output_high(LED_YELLOW); // Turn off yellow LED
65             output_low(LED_YELLOW2); // Turn off yellow LED2
66             output_low(LED_YELLOW3); // Turn off yellow LED3
67             output_low(LED_GREEN); // Turn off green LED
68             output_high(LED_GREEN2); // Turn off green LED2
69             output_low(LED_GREEN3); // Turn off green LED3
70             output_high(P_RED); // Turn pedestrian red light on
71             // output_low(P_GREEN); // Turn pedestrian green light off
72             delay_ms(1000); // 1 second delay
73         }
74
75         // Green LED ON, Others off
76         for (i = 0; i < 10; i++)
77         {
78             output_d(0xA4); // '2'
79             output_c(display[i]); // Update 7-segment display with count
80             output_low(LED_RED); // Turn on red LED
81             output_high(LED_RED2); // Turn on red LED2
82             output_low(LED_RED3); // Turn on red LED3
83             output_low(LED_YELLOW); // Turn off yellow LED
84             output_low(LED_YELLOW2); // Turn off yellow LED2
85             output_high(LED_YELLOW3); // Turn off yellow LED3
86             output_high(LED_GREEN); // Turn off green LED
87             output_low(LED_GREEN2); // Turn off green LED2
88             output_low(LED_GREEN3); // Turn off green LED3
89             output_high(P_RED); // Turn pedestrian red light on
90             // output_low(P_GREEN); // Turn pedestrian green light off
91             delay_ms(1000); // 1 second delay
92         }
93
94         // Pedestrian Green LED ON, Others off
95         for (i = 0; i < 10; i++)
96         {
97             output_d(0x80); // '3'
98             output_c(display[i]); // Update 7-segment display with count
99             output_low(LED_RED); // Turn on red LED
100             output_high(LED_RED2); // Turn on red LED2
101             output_low(LED_RED3); // Turn on red LED3
102             output_low(LED_YELLOW); // Turn off yellow LED
103             output_low(LED_YELLOW2); // Turn off yellow LED2
104             output_high(LED_YELLOW3); // Turn off yellow LED3
105             output_high(LED_GREEN); // Turn off green LED
106             output_low(LED_GREEN2); // Turn off green LED2
107             output_low(LED_GREEN3); // Turn off green LED3
108             output_high(P_GREEN); // Turn pedestrian green light on
109             // output_low(P_RED); // Turn pedestrian red light off
110             delay_ms(1000); // 1 second delay
111         }
112     }
113 }
```

Designing an Automatic Traffic Light Controller for a T- Junction Road

Ellis Ashton (2406269)

```
{
    output_d(0x80); //'3'
    output_c(display[i]); // Update 7-segment display with count
    output_high(LED_RED); // Turn on red LED
    output_high(LED_RED2); // Turn on red LED2
    output_high(LED_RED3); // Turn on red LED3
    output_low(LED_YELLOW); // Turn off yellow LED
    output_low(LED_YELLOW2); // Turn off yellow LED2
    output_low(LED_YELLOW3); // Turn off yellow LED3
    output_low(LED_GREEN); // Turn off green LED
    output_low(LED_GREEN2); // Turn off green LED2
    output_low(LED_GREEN3); // Turn off green LED3
    // output_low(P_RED); // Turn pedestrian red light on
    output_high(P_GREEN); // Turn pedestrian green light off
    delay_ms(1000); // 1 second delay
}

for (i = 0; i < 1; i++)
{
    output_d(0x99); //'4'
    output_c(display[i]);
    delay_ms(1000)
    ;}}
}
```

References

[1] "PIC16F877A," *Microchip.com*, 2024. <https://www.microchip.com/en-us/product/pic16f877a#document-table> (accessed Nov. 28, 2024).

[2] T. Wilmshurst, *Designing embedded systems with PIC microcontrollers : principles and applications*. Amsterdam: Newnes, 2007.

[3] Electronics Tutorials, "7-segment Display and Driving a 7-segment Display," *Basic Electronics Tutorials*, Feb. 16, 2018. <https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>

[4] "PCB Design and Circuit Simulator Software - Pro  6 www.labcenter.com.
<https://www.labcenter.com/>

Designing an Automatic Traffic Light System

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

62/100

PAGE 1



Comment 1 | Engagement with Literature(10)

The task of designing the traffic light controller is identified.



Comment 2 | Knowledge & Understanding (20)

It is also essential to write about the various types of traffic light controllers.



Comment 3 | Cognitive & Intellectual (20)

It is good that some advantages are seen by It is also needed to compare the selected microcontroller with the other microcontroller in order to highlight its advantages.

PAGE 2

PAGE 3

PAGE 4

PAGE 5



Comment 4 | Cognitive & Intellectual (20)

Brilliant design!

PAGE 6

PAGE 7



Comment 5 | Practical Application Sk(30)

Circuit design are neat. Could be better if road layouts are also included.

PAGE 8

PAGE 9

**Comment 6** | Engagement with Literature(10)

More references could help to create a comprehensive work which makes the report much stronger.

GRADING FORM: EC1

ELLIS ASHTON

61.68

ENGAGEMENT WITH LITERATURE(10)

a. Identification on Concepts of Traffic Light Controller (4);its significance (2) b. Background Research (3) c. Diversity of sources and Referencing Skills(1)



- a. Good! Information related to the traffic Light Controller is very general and its significance were seen. 3.6
- b. Engagement with a limited range of mostly relevant and credible sources. 1.35
- c. Referencing may show minor inaccuracies or inconsistencies. 0.5

5.45

KNOWLEDGE & UNDERSTANDING (20)

a. Knowledge on various products of traffic light controller and Market Requirements and Specifications(10) b. Reflect clearly the underlying principles on analysing different traffic light controllers with adequate and appropriate Justifications(10)



- a.Limited knowledge on various products/parts of traffic light Controller and critical understanding of the well-established concepts and principles within the subject area. Market Requirements and Specifications are not Mentioned. 4

8

b. Analysing various traffics light controller is limited. 4

COGNITIVE & INTELLECTUAL (20)

a. Critical review on the selection of Microcontroller and its supporting Modules, ICs, costs(7) b. Innovative thinking in problem solving and feature additions(7) c. Pseudocode showing the system operation(6)



a. Limited Attempt on selection is made.
Supporting Modules are identified. 3.85
b.Examining the evidences of the design , its
accuracy, precision, relevance, completeness are
up to the satisfaction. 4.55
c.Pseudocode is relevant and produces the
expected outputs and results. 3.9

12.33

PRACTICAL APPLICATION SK(30)

a. Circuit Design Layout(10). b. Testing and Debugging and Simulated Output Results(20)



a. A reasonable circuit is designed for the Traffic
Light Controller was designed by considering the
traffic light modules. 7.5
b.Very Good! Compilation and simulation tools are
used sufficiently. Results are clearly presented for
the Traffic Light Controller. 15.2

22.7

TRANSFERABLE SKILLS (20)

*a. Logical order of ideas, Flow of ideas in the report from the beginning to end with appropriate chapters, sub chapters and continuation(7)"
b. Definitions, Acronyms and Abbreviations (6) c.
Professionalism, Spelling and Grammar, Complete Report, Proper Presentation, Quality of the report(7)*



a. Work is coherent, fluent, well-structured and
organised. 5

b. Most of the Definitions, Acronyms and Abbreviations are captured. 4

c. Professionalism is maintained in the complete report. Proper Presentation and the Quality of the report is appreciable. 5.2

13.2