

2205436 - Electronics I

by Jade Williams

Submission date: 10-Dec-2024 02:17PM (UTC+0000)

Submission ID: 245401398

File name: 49685_Jade_Williams_2205436_-_Electronics_I_634321_1739303418.docx (889.8K)

Word count: 2470

Character count: 13134

SamanthaJade Williams – 2205436
Electronics I – ACDB5002
Designing an Automatic Traffic Light Controller
for a T- Junction Road

Introduction:	3
Background Research:.....	3
Circuit Design:.....	6
Code:.....	7
Conclusion:	11
References:.....	13

Introduction:

The task provided in the brief was to design and build a traffic light control system specifically tailored for a T-junction. This project aimed to enhance traffic management and ensure the safety of both vehicles and pedestrians at this intersection. The requirements outlined in the brief focused on several critical elements necessary for effective operation.

One of the key requirements was to incorporate a set of 7-segment displays for each section of the traffic light system. These displayed a counter or timer, clearly showing the remaining time for each traffic light cycle. The timer was set for 40 seconds, which allows drivers and pedestrians to anticipate changes in traffic flow. Additionally, it was essential that the traffic lights change every 10 seconds, providing a predictable rhythm to the flow of traffic, thereby minimizing confusion and reducing the likelihood of accidents. The brief stipulated that the final 10 seconds of the cycle must be designated as a pedestrian crossing phase. During this phase, the traffic signals for vehicles must either display a red light or be completely turned off, ensuring that pedestrians can cross safely.

For the control of this circuit, the brief specified the use of two options: a PIC16F microcontroller or a PIC18F microcontroller. The chosen microcontroller was the PIC16F877A.

Background Research:

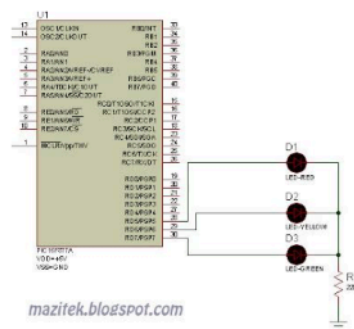


Figure 1: Simple example circuit [1]

Shown in figure 1 is an example circuit that illustrates the connections for a set of LEDs arranged to function as a traffic light. This example [1] features a simplified circuit design that slightly resembles the specifications outlined in the project brief, using the PIC16F877A microcontroller for control and functionality. The code associated with this example is designed to manage the operation of the three LEDs, saying that each light is to stay illuminated for varying amounts of time, simulating a more realistic example of

traffic signal timing. The initial design of this particular circuit consisted of the PIC16F877A, a crystal oscillator, two 22pF capacitors, a 9V battery, a 7805 voltage regulator, a 220uF capacitor, a 1kΩ resistor, three 220Ω resistors, a red LED, a yellow LED, and a green LED, although the final design, shown in figure 1, only included the PIC16F877A, the red LED, the yellow LED, the green LED, and a 220Ω resistor.

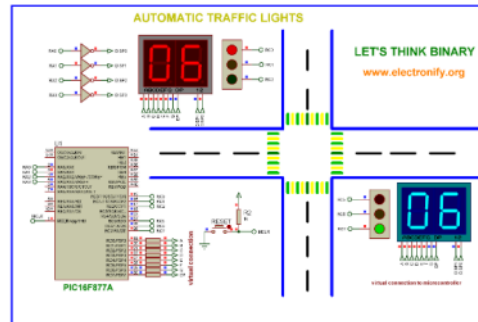


Figure 2: Advanced example circuit [2]

Shown in Figure 2 is a more advanced example of the same circuit [2], which incorporates two traffic lights and corresponding two-digit 7-segment displays. In this configuration, the two-digit 7-segment displays are used instead of a pair of one-digit 7-segment displays, which introduces a bit more complexity into the circuit. The primary difference lies in the number of input pins: while a one-digit 7-segment display typically requires only seven input pins, a two-digit 7-segment display requires ten. This increased number of pins results in a more intricate wiring setup, as the connections between the display and the microcontroller will differ. As a result, the code to control the two-digit 7-segment display becomes more complicated, as the system must manage added pins and address both digits simultaneously. Despite these challenges, the use of a two-digit display allows for a more compact and efficient design, reducing the need for extra displays in the system, as well as needing fewer connections to the microcontroller. The final design for this example includes the PIC16F877A, two traffic light modules, two 7-segment displays to show two digits, nine resistors, four inverters, and a button.

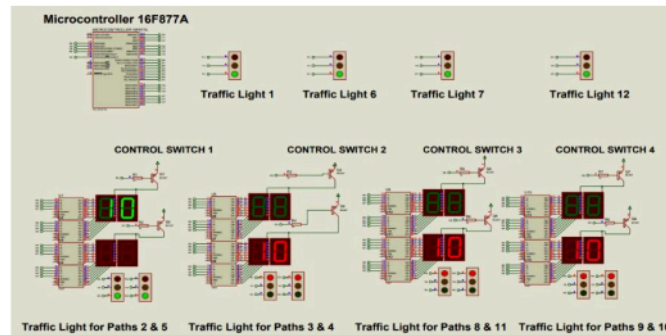


Figure 3: Complex example design

A more complex design for a traffic light circuit is shown above in figure 3. The design is for a six-way, four lane intersection [3] utilising a traffic light per lane and a set of 7-segment displays for specific lanes.  example features a PIC16F877A microcontroller to program in C++. This example has been used as it shows a variation of the circuit which could have been used which includes BCD to 7-segment display decoders. The design shows that there are two sets of 7-segment displays per junction. Shown below in figure 4 is a diagram showing the operation of the lights. This design has been used as an example as it shows a system that is safe for drivers in all lanes, although there is not any indication that there is a pedestrian crossing included in the design.

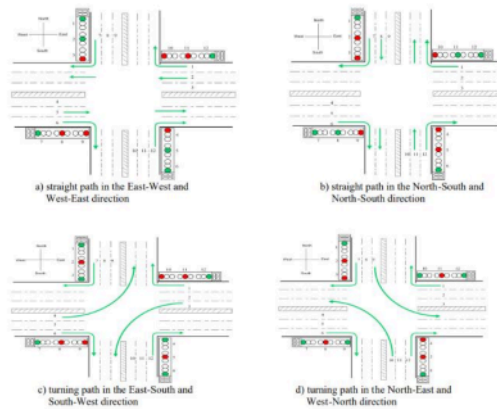


Figure 4: Road layout for complex example

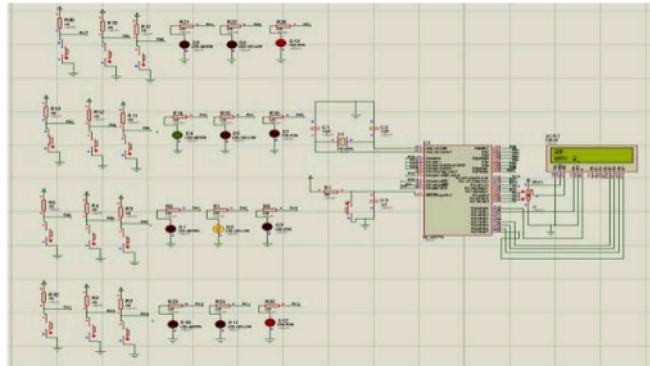


Figure 5: Circuit for density-based traffic lights

The circuit shown in figure 5 is of a density-based traffic light system [4]. It consists of a PIC16F877A, thirteen buttons, twenty-five resistors, a crystal oscillator, an LCD, four red LEDs, four yellow LEDs, four green LEDs, and a variable resistor. In this circuit, the buttons are used to set the density of the traffic from two states, high density and low density. When the button is pressed, it stands for high density traffic, which decides the timing of the lights. The LCD shows the amount of time between the changes in state of the traffic signals which are shown through the LEDs.

Circuit Design:

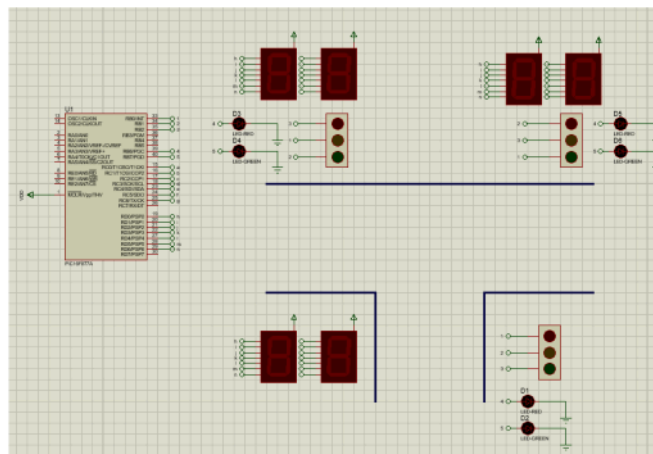


Figure 6: Circuit design on Proteus

Figure 6 illustrates the complete circuit when it is in the off state. The final design consists of several important components that work together to control traffic at the T-junction effectively. It features a set of two 7-segment displays, which are crucial for displaying timers and counters to inform both drivers and pedestrians of the current

traffic light phase. Additionally, the system includes a traffic light module that manages the red, yellow, and green lights for vehicles, ensuring orderly traffic flow. Each section of the T-junction is enhanced with two LEDs, specifically designed to show the state of the pedestrian crossing. These LEDs provide clear visual signals to inform pedestrians when it is safe to cross and when they should wait, thereby enhancing safety at the intersection.

All these components are meticulously connected to a PIC16F877A microcontroller, which acts as the brain of the system, orchestrating the timing and functioning of the lights and displays to ensure smooth and safe operation at the intersection. This integrated design shows careful planning and consideration for both traffic management and pedestrian safety.

The circuit has been designed with input nodes instead of wires to simplify the layout and enhance clarity. This approach is especially useful in digital circuit simulations, where too many wires can create confusion and hinder understanding of connections and circuit flow. Using input nodes helps keep a clean and organized appearance, making it easier to find components and their connections. This design choice ensures that the circuit stays intuitive and accessible, facilitating better analysis and troubleshooting.

Code:

```
1 #include <16f877a.h>    //Header file for the pic
2 #fuses XT, NOWDT, NOPROTECT, NOLVP
3 #use delay(clock=2000000) //sets the clock speed to 2MHz
4
5 //define LED pins
6 #define LED_ONE PIN_B0
7 #define LED_TWO PIN_B1
8 #define LED_THREE PIN_B2
9 #define P_RED PIN_B6
10 #define P_GREEN PIN_B7
```

Figure 7: Lines 1-10 of code

The first 10 lines of code, shown in figure 7, include a header file for the PIC16F877A microcontroller and set the clock speed to 2 MHz, ensuring correct timing for the traffic light control system. Additionally, these lines define each LED, including those on the traffic light and the two LEDs for the pedestrian crossing, enabling efficient control throughout the program.

The code used in line one, “#include <16f877a.h>” is used to include the header file for the specified PIC, which as stated in the introduction, is a PIC16F877A.

The term used in line two “XT” means that an oscillator value of up to 4MHz can be used in the circuit, “NOWDT” means no watch dog timer, “NOPROTECT” means that

the code is not protected from reading, and finally, “NOLVP” means no low voltage programming. NOLVP is used in PIC16 and PIC18 circuits.

```
12 //7-segment display pattern to show 0-9
13 Unsigned int display[10] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90}
```

Figure 8: 7-segment display array

Figure 8 displays an array of hexadecimal values for the 7-segment display, enabling the microcontroller to accurately illuminate the correct segments for each digit. Each hexadecimal value corresponds to a number shown on the 7-segment display with 0xC0 standing for 0, 0xF9 standing for 1, 0xA4 standing for 2, etc.

The hexadecimal values for each number on the display are shown below in figure 9 in a table of inputs. This will only be the case for common anode 7-segment displays and not in the case of common cathode displays being used. In the case of a common cathode 7-segment display being used, the values will be different, for example: 0x7E stands for the number 0 on the 7-segment display, 0x30 stands for 1, 0x6D stands for 2, etc.

Numbers	Common Anode		Numbers	Common Anode	
	(DP)GFEDCBA	Hex Code		(DP)GFEDCBA	Hex Code
0	11000000	0xc0	A	10001000	0x88
1	11111001	0xf9	B	10000011	0x83
2	10100100	0xa4	C	11000110	0xc6
3	10110000	0xb0	D	10100001	0xa1
4	10011001	0x99	E	10000110	0x86
5	10010010	0x92	F	10001110	0x8e
6	10000010	0x82	.	01111111	0x7f
7	11111000	0xf8			
8	10000000	0x80			
9	10010000	0x90			

Figure 9: Hexadecimal values for 7-segment display [5]

```
19 set_tris_b(0x00); //sets portB as an output
20 output_b(0x00); //Sets the output to 0
21 set_tris_c(0x00); //sets portC as an output
22 output_c(0x00); //Sets the output to 0
23 set_tris_d(0x00); //sets portD as an output
24 output_d(0x00); //Sets the output to 0
```

Figure 10: Output declarations

Shown above in Figure 10 are the declarations for outputs B, C, and D, which are connected to the two 7-segment displays and the LEDs. Shown on lines 20, 22, and 24 are the declarations setting the outputs to 0. The code provides clarity through comments, which explain what each line of code does.

```

28 //red LED on
29 for (i=0; i<10; i++) { //counts from 0-9
30   output_d(0xC0); // 0 on the static 7-segment display
31   output_c(display[i]); //update the second 7-segment display
32   output_high(LED_ONE); //LED on
33   output_low(LED_TWO); //LED off
34   output_low(LED_THREE); //LED off
35   output_high(P_RED); //Red LED for pedestrian crossing
36   delay_ms(1000); //1 second delay
37 }

```

Figure 11: C code for the first stage of lights

The code depicted in Figure 11 governs the operation of the traffic light system across multiple stages, ensuring efficient traffic flow and pedestrian safety at the T-junction. The code shown above in figure 11 states that the first 7-segment display is to show the number 0, while counting from 0-9 on the second 7-segment display. The first stage of the traffic lights is programmed to display a red light for one junction, a yellow light for the second junction, and a green light for the third junction. This is shown in the code with the use of “output_high” for LED one, and “output_low” for both LED two and LED three. During this first phase, the pedestrian crossing light is also set to red, signalling to pedestrians that it is not safe to cross. This stage runs for 10 seconds, which is shown on the 7-segment displays. The output for this section is displayed below in figure 12.

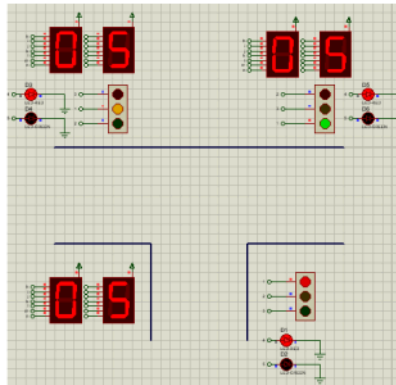


Figure 12: Section one output

In the second section of the code, the light configurations are adjusted slightly. The code for this section states that the first 7-segment display is to show the number 1, while counting from 0-9 on the second 7-segment display. The first junction displays a yellow light, the second junction shows a green light, and the third junction has a red light. This is done by changing the code slightly for the LEDs, with “output_high” being changed to “output_low” for LED one, and “output_low” to “output_high” for LED two. Once again, the pedestrian crossing light stays red to ensure pedestrian safety. This section also lasts for 10 seconds before continuing to the third segment of the program. The output of this section is shown in figure 13.

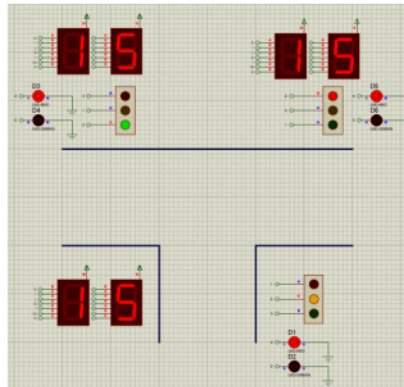


Figure 13: Section two output

The third section of code handles displaying green for the first junction, red for the second junction, and yellow for the third. This section of code is changed again for the LEDs, with “output_high” to “output_low” for LED two, and “output_low” to “output_high” for LED three. The code for this section also says that the first 7-segment display is to show the number 2, while counting from 0-9 on the second 7-segment display. The pedestrian crossing light will still be red during this phase, which lasts an added 10 seconds. This is shown in figure 14.

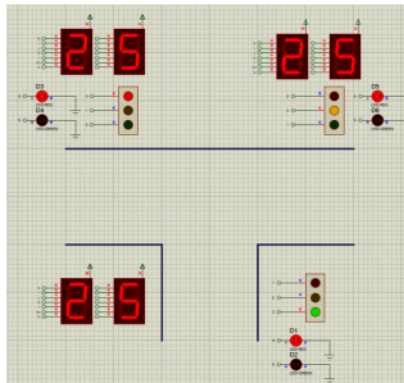


Figure 14: Section three output

Finally, the last section of code is dedicated solely to the pedestrian crossing, running for 10 seconds. During this time, all traffic lights are turned off, and the pedestrian crossing lights illuminate green, allowing pedestrians to cross safely. This is shown in the code with all LED code for the traffic lights written as “output_low”. The code for this section states that the first 7-segment display is to show the number 3, while counting from 0-9 on the second 7-segment display. There is a further section of code simply to display the number 40 on the two 7-segment displays. This structured approach ensures an orderly and safe traffic flow at the junction as shown in figure 15.

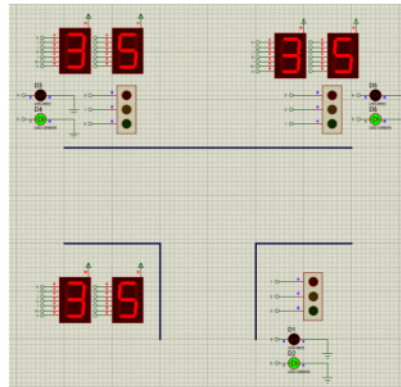


Figure 15: Section four output

Figure 16 displays a flow chart illustrating the state of the traffic signals for each junction, as well as the pedestrian crossing. This flow chart outlines the sequential changes in the signal states at the various junctions. It also shows the activation of the pedestrian crossing signal, ensuring safe passage for pedestrians. The flow chart serves as a clear visual representation of how the traffic lights and pedestrian signals interact, helping to keep orderly traffic flow while prioritizing pedestrian safety. By following this chart, it ensures synchronized operation across all signals, minimizing confusion and delays.



Figure 16: Flow chart of traffic signals

Conclusion:

The traffic light control circuit was designed with simplicity and flexibility in mind, making it easy to change as needs evolve. The straightforward design ensures that future adjustments can be done with minimal effort.

One potential advancement would be to adjust the timing of the traffic light. Currently the circuit uses fixed timers of ten seconds per section of the cycle. This could be adjusted to suit varying traffic patterns at various times of day. For instance, during off-peak times of the day the light cycles would be shorter than during busier times of the day.

The circuit could also evolve to become a density-controlled circuit which would rely on sensors to adjust traffic light timing based on real time conditions rather than be on a fixed timer. Infrared (IR) sensors can detect the presence of vehicles at a junction and adjust the lights accordingly. If there are many vehicles at a junction, the light could stay green for longer, and if there are not many vehicles at a junction, the light would not stay on green for as long. Alternatively, ultrasonic sensors could measure the distance between vehicles in order to assess the levels of congestion at each junction, which would help determine the optimal timing for the traffic lights.

Additionally, introducing manual controls for pedestrian use would allow pedestrians to cross the roads safely by extending the timer of the red light. Such improvements would improve traffic flow and safety for pedestrians and drivers, while also staying future-proof.

References:

- [1] Z. Mohammed, "BASIC TIMING TRAFIC CONTROL USING PIC16F877A," MaziTek Electronics, Oct. 29, 2016. <https://blog.mazitekgh.com/2016/10/basic-timing-traffic-control-using.html> (accessed Nov. 19, 2024).
- [2] S. Dahouathi, "Automatic Traffic Light Using PIC Microcontroller Code, Circuit Diagram and Explanation," *pdfcoffee.com*, Apr. 5, 2019. <https://pdfcoffee.com/automatic-traffic-light-using-pic-microcontroller-code-circuit-diagram-and-explanation-electronify-pdf-free.html> (accessed Nov. 19, 2024).
- [3] A. Awami and Z. Yusupov, "Microcontroller Based Traffic Light Control System," Sep. 29, 2017. <https://isites.info/PastConferences/ISITES2017/ISITES2017/papers/A6-ISITES2017ID187.pdf> (accessed Nov. 20, 2024).
- [4] P. A. Saiba, M. U. Afeefa, T. S. Aruna, J. Clincy, and V. M. Radhika, "Density Based Traffic Signal System Using PIC Microcontroller," *International Journal of Computer Trends and Technology*, vol. 47, no. 1, pp. 74–78, May 2017, doi: <https://doi.org/10.14445/22312803/ijctt-v47p109>.
- [5] "4-Digit 7-Segment Display," docs.sunfounder.com. https://docs.sunfounder.com/projects/uno-mega-kit/en/latest/components/component_4_digit.html (accessed Nov. 19, 2024).

FINAL GRADE

GENERAL COMMENTS

77 /100

PAGE 1

PAGE 2

PAGE 3

 **Comment 1** |  Engagement with Literature(10)

The task to design a traffic light system is identified.

 **Comment 2**

good way of representing the problem statement!

PAGE 4

 **Comment 3**

Excellent!

PAGE 5

 **Comment 4** |  Engagement with Literature(10)

Extensive range of relevant and credible literature were made. Including more journal papers will enhance the depth providing a stronger foundation for the analysis.

PAGE 6



Comment 5 | Knowledge & Understanding (20)

Various traffic light examples are included which is excellent !

PAGE 7

PAGE 8



Comment 6 | Cognitive & Intellectual (20)

It is commendable that the lines of the program are explained.

PAGE 9

PAGE 10

PAGE 11



Comment 7 | Practical Application Sk(30)

Clear Circuit Layout

PAGE 12

PAGE 13

GRADING FORM: EC1

JADE WILLIAMS

77.26

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. Information related to the traffic Light Controller is well identified and its significance

were captured. 4.8

b. Engagement with an extensive range of relevant and credible literature. 2.1

c. Consistently accurate application of referencing. 0.7

7.6

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. Good! Knowledge on various products/parts of traffic light Controller is reasonably detailed and accurate. Market Requirements and Specifications are limited. 5

b. Excellent understanding on the different Traffic Light controllers and adequate justification is made. 10

15

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. Evaluation on selection of Microcontroller and its supporting Modules, ICs were made. 4.9

b. Clearly distinguishes between fact, opinion, & acknowledges value judgments are seen. 5.25

c. Crystal Clear Pseudocode is written which runs and produces the expected outputs and results. 5.4

15.66

PRACTICAL APPLICATION SK(30)

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

25



- a. A reasonable circuit is designed for the Traffic Light Controller was designed by considering the traffic light modules. 8
- b. Exceptional! Outstanding use of compilation and simulation tool are made. Results are clearly presented with proper justification. 17

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)

14



- a. Work is coherent, very fluent and is presented proficiently. 5
- b. Excellent! Definitions, Acronyms and Abbreviations used appropriately and adequately. 4
- c. High level of professionalism is maintained throughout the report. Proper Presentation and the Quality of the report is excellent. 5