

Electronics 1

by Greg Dickson

Submission date: 16-Apr-2024 03:00PM (UTC+0100)

Submission ID: 229577553

File name: 42256_Greg_Dickson_Electronics_1_568563_1247326973.docx (629.21K)

Word count: 2100

Character count: 10067

Electronics I (ACDA5002) Assignment

GREG DICKSON (2204524)

Sections

Introduction

Background research

Circuit Design

Flowchart

Pseudocode

Software code

Screen captures of software

Testing information and outcomes

Introduction

The report contains information on how traffic light systems are implemented and how to implement a basic fixed time traffic light system.

A four-way traffic light system has been used to implement the fixed time traffic light system with the addition of a timer and pedestrian lights to allow pedestrians to cross the four way traffic light junction.

Background research has been used to figure out necessary information on how traffic control systems work and what type of traffic control systems are available to implement to help the flow of traffic whether this is a timer based traffic light system or a sensor based traffic light system.

Traffic management was also researched to see how important it is to have traffic management and what benefits it can provide to road users to help prevent accidents from occurring.

Some research has been done into the PIC16F877A to see if it would be suitable for the conditions needed to run the time-based traffic light control system and also what beneficial features it has for this simulation.

The circuit design is critical for the traffic light system to work fluidly, ensuring that the pins are correctly configured to allow the output of the pins to enable the correct lights are on at the set intervals. Having the traffic light is necessary to control the flow of traffic as are the pedestrian lights to ensure that the pedestrians know when to cross. The seven segment displays are also critical to the design of the circuit.

Showing the flowchart of how the system operate through the code allows a clearer vision on what the operations of the program do at each of the set intervals of 10 seconds and how exactly the program will be able to flow.

The pseudocode allows a simpler look into what the code is doing during its operation. This will give more detailed information on what is happening throughout the code at each of the set intervals and how that translates into the operation of the lights on each of the timer. There is also information on the background code such as the variables.

The testing of the simulation did provide results on if the simulation was functional or not and suitable for the task at hand. It allowed the results to be used to reflect on what went right or wrong within the simulation.

Background Research

Types of traffic light

Traffic lights are used all around the world to help control the flow of traffic. The use of traffic lights requires knowledges on how to time manage electronics and send signals at the relevant times to enable or disable a light or sound.

Traffic lights can come in different configurations on how they operate themselves from having fixed time or sensor activated traffic lights. Fixed time traffic light systems are much simpler to implement due to the fact they are programmed to run off a set pattern for a certain period of time, with no interruptions unless the timer has gotten to a certain pre-determined within the code written for it. [1]

The other type of traffic light configuration is a sensor activated traffic light. This type of traffic light will sense if a car is present at the lights by either using sensor underneath the road surface or a sensor to detect if a car is present on top of the traffic lights. The under the road surface sensors work off coils to detect if a car is present via induction and a change within a magnetic field. The sensors on top of the lights work through either laser or camera to detect if a car is there waiting for the lights to change. [2]

The sensor activated traffic light system will be more complex to implement than a pure fixed time system due to needing to program the traffic lights to have sensors to activate the traffic lights under certain conditions. The sensor based traffic light systems will still run a timer based program within them to ensure that the lights do change without any need for a vehicle to be present.

Traffic Management

Traffic management is critical for the flow of traffic to keep going. Having good traffic management helps to avoid congestion from building up on the roads, therefor traffic signals need to be configured for how busy the roads will be.

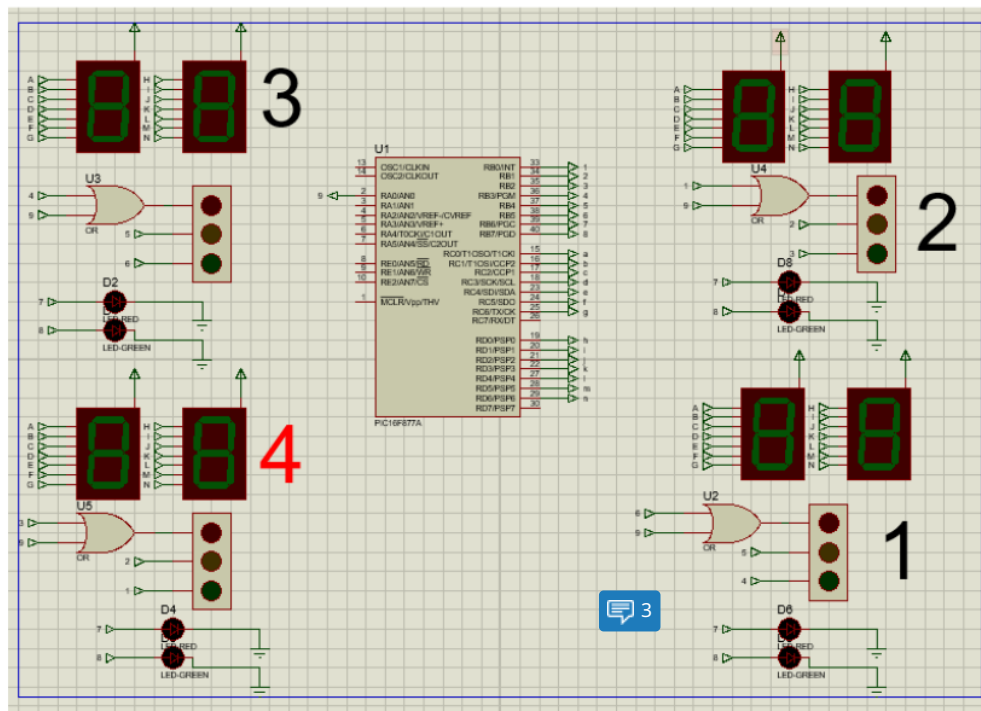
Ensuring that road safety is improved is also a part of traffic management, with the lights being correctly configured there will be less of a risk of accidents from occurring. [5]

Microcontroller

The microcontroller researched is the PIC16F1887A, this microcontroller allows a traffic light program to be written within it. Having 40 pins on the microcontroller allows the use of multiple outputs being send at the same time to control the traffic light system. It is able to handle a four way traffic system synchronously with no issue of confliction or errors. It contains 256 bytes of EEPROM memory to store the code within its data, which is ple²il for the use of a traffic light system.

The PIC16F1887A is able to communicate via SPI, I2C and USART, allow various ways for the microcontroller to communicate with external components. [3]

Circuit Design

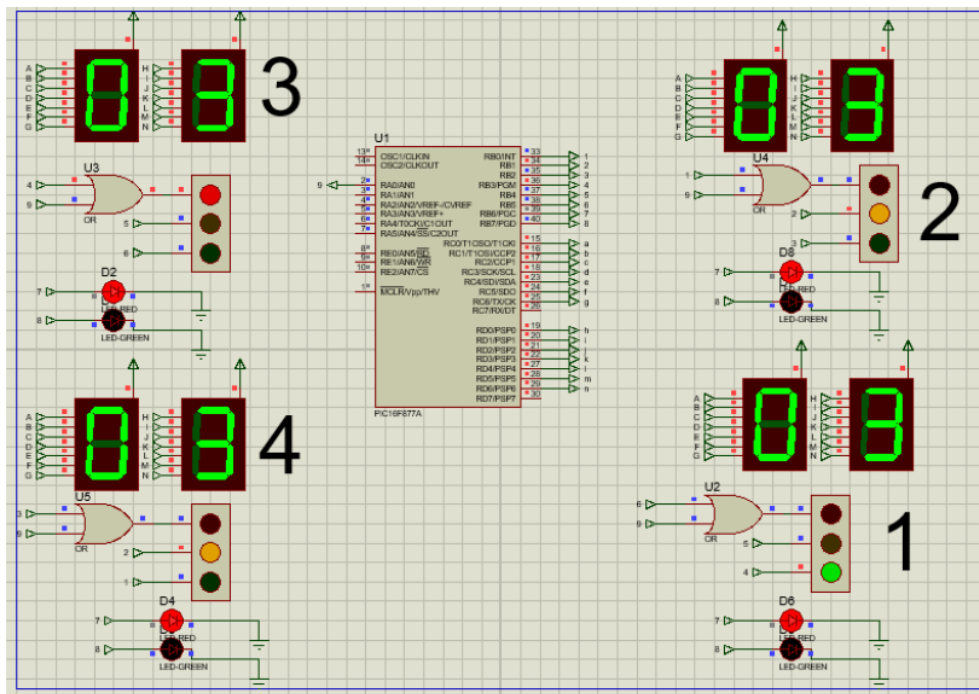


[6]

The circuit has been designed with thought into how a real-life traffic light would operate. Certain components have been used within the circuit to get a basic traffic light system working.

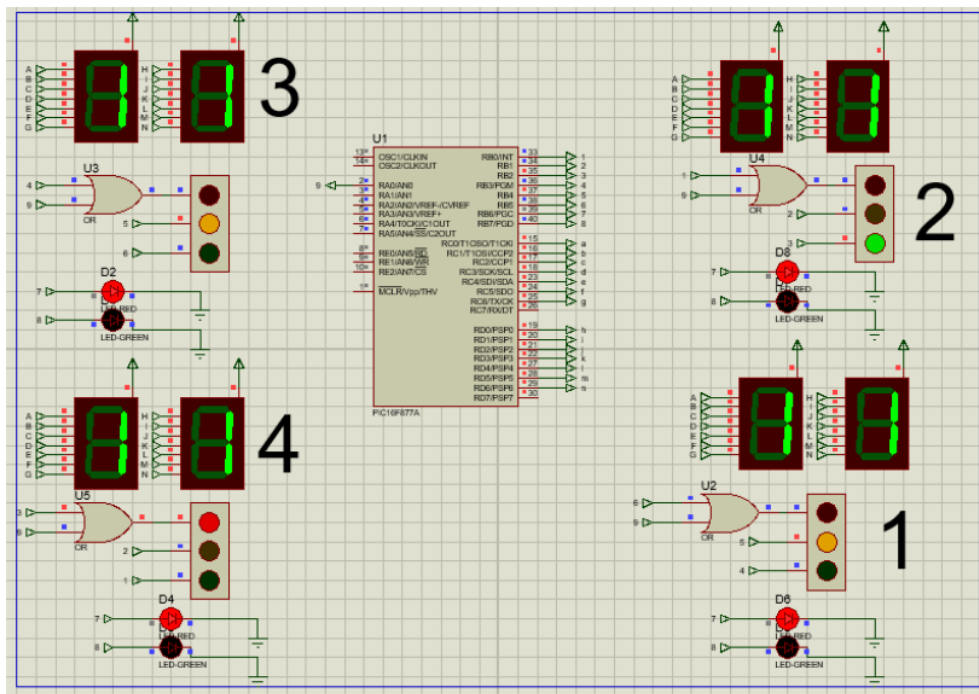
The microcontroller used is a PIC16F877A, this microcontroller will be able to support the code to output instructions to the traffic light system so that it is able to work fluidly. The microcontroller will be able to support the necessary 23 pins which are required to make this circuit work with the output to the lights. This microcontroller is relatively inexpensive and widely available. It also features a clock frequency of up to 20MHz, allowing it to handle instructions quickly. [4]

The initial image above is the simulated circuit off. The circuit has a pattern on how it operated above, starting at "1" being the start of the green light routine through the set of traffic lights.



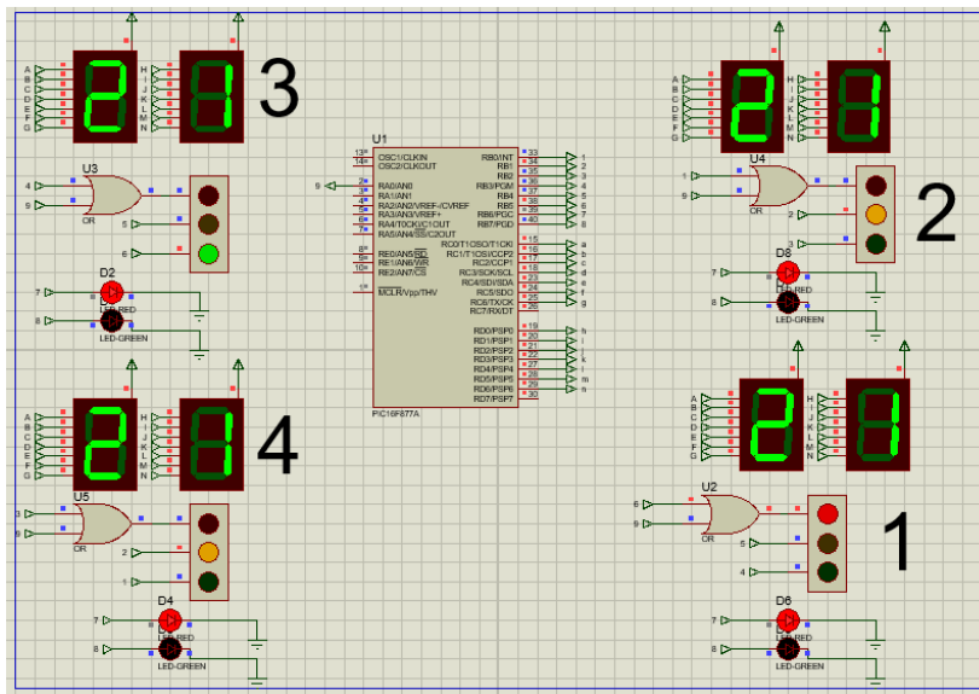
[7]

The screen shot above is the simulation at the 1 to 10 second phase, during this phase the first traffic light is green allowing the traffic through the “1” section and showing how long there is before it reaches the next 10 second increment.



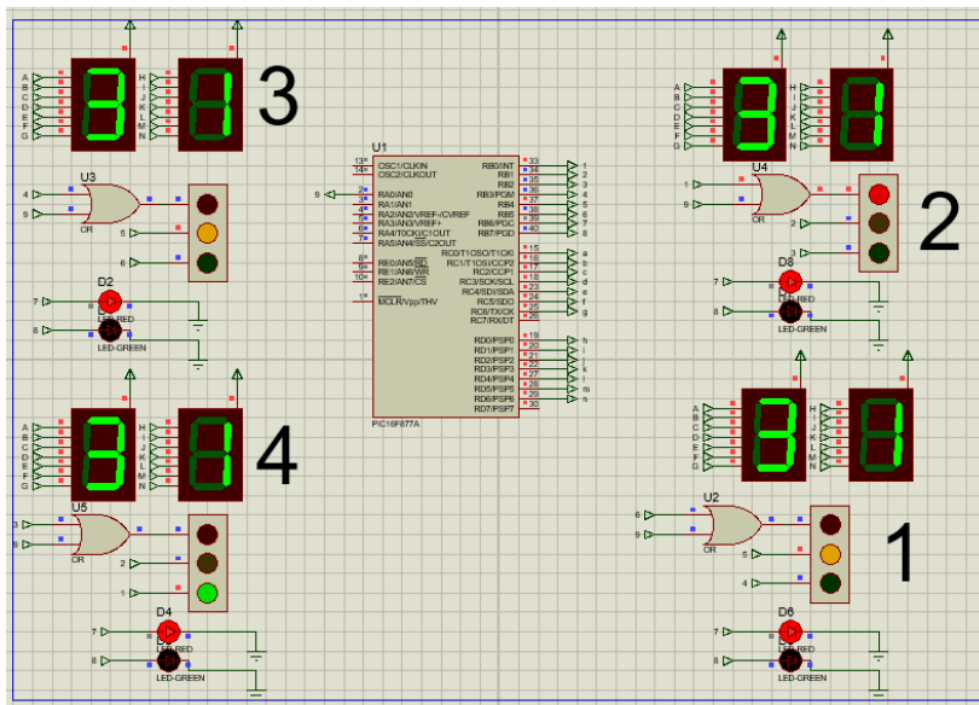
[8]

This is showing the circuit between the 10 and 20 second phase. The sets of lights have changed which lights are illuminated. The second set of lights are now green where the first set has changed itself to amber. The third set of lights are amber in preparation for the next light change.



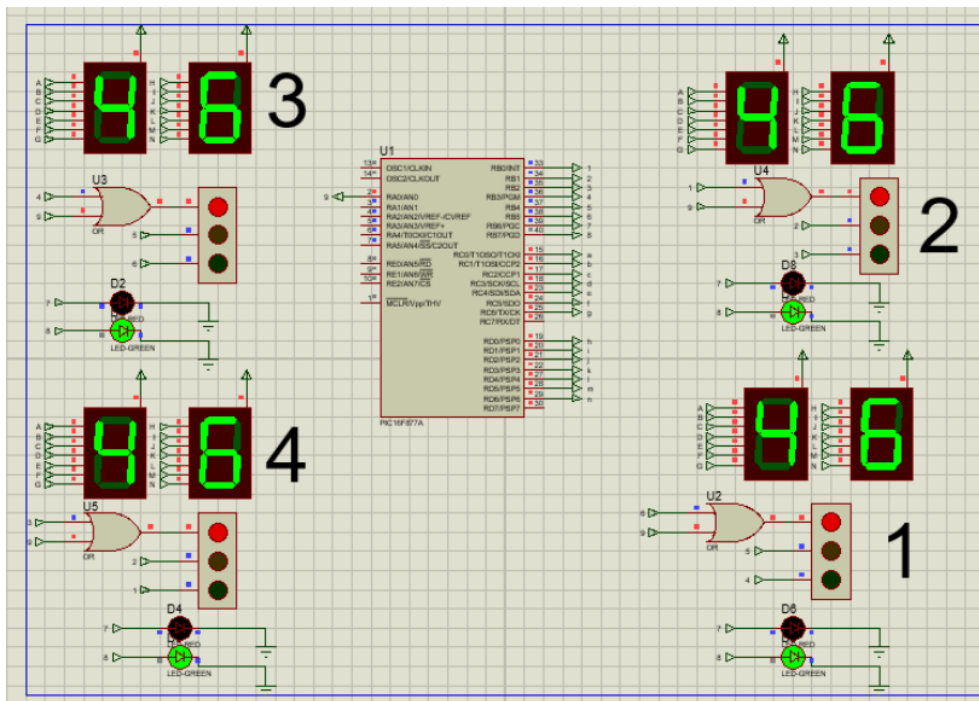
[9]

The 20 to 30 second phase shows the third set of traffic lights appearing green allowing traffic through with the second set turning amber to stop the traffic from going. The fourth set of lights are amber in preparation for the light turning green and the first set of lights are red, not allowing traffic through.



[10]

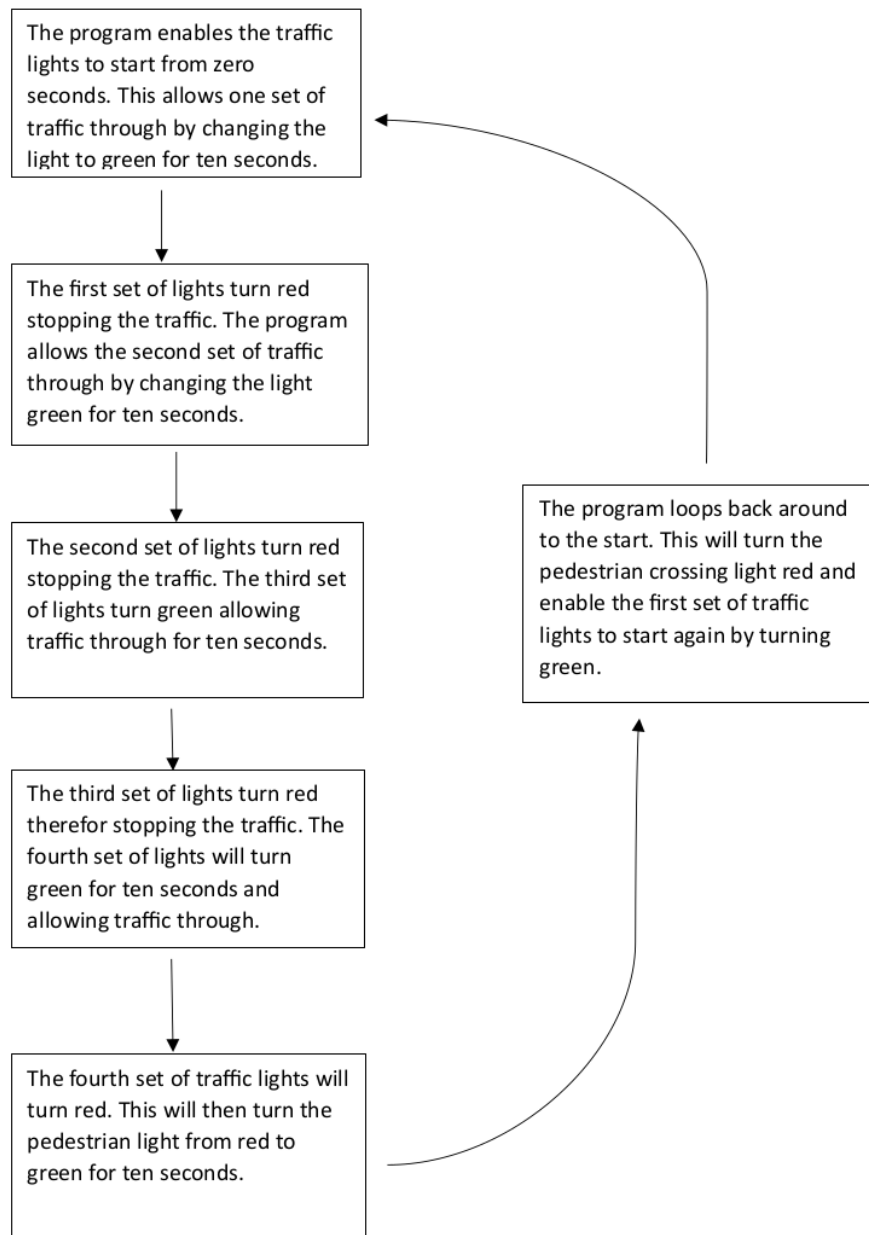
This is the phase of traffic lights at the 30 – 40 second mark. The fourth set of lights have turned green allowing vehicles through and the third set of light have turned orange stopping vehicles from going through. The second set of lights are red stopping vehicles from going and the first set are amber also stopping vehicles from going through.



[11]

The traffic lights have all been set to red to stop all the vehicles from moving through the junction. The pedestrian lights have been turned to green to allow pedestrians to walk across the road for ten seconds between 40 and 50 seconds. OR gates had to be introduced to stop all the lights and thus making them show red.

Flowchart



Pseudocode

1. State “increment” as an integer variable
2. State “sevensseg” as an unsigned int to allow the seven segment displays to show the numbers 0 to 9 via hexadecimal values.
3. Enable the pins on the microcontroller to be assigned to a certain name
4. Enable the ports as output ports by changing the tristate and port outputs.
5. Start the main loop of the program using the “do” command.
6. Start the timer to increment until it hits “10”.
7. Enable the first set of traffic lights to turn green for 10 seconds whilst the other lights are either amber or red.
8. Set the first set of traffic lights to red and enable the second set of traffic lights to green for 10 seconds whilst the other lights remain red or amber.
9. Set the second set of traffic lights to red and enable the third set of traffic lights to turn green for 10 seconds whilst the other lights remain red or amber.
10. Set the third set of traffic lights to red and change the fourth set of traffic lights to green for 10 seconds whilst the other lights remain amber or red.
11. Set the fourth set of traffic lights to red whilst the rest of the traffic lights also remain red.
12. Turn the pedestrian crossing to green for 10 seconds, allowing pedestrians to cross the road.
13. Loop back around to the start of the program, turning the pedestrian crossing lights to red and enabling the first set of traffic lights to turn green and keeping the other traffic lights red or amber.



Testing details

Testing the circuit provided a good number of results. The circuit has appeared work as intended; the circuit was assembled as the circuit design showed had worked allowing a fluid time-based traffic system to work across the fifty seconds it is programmed to using the PIC16F877A and the matrix based signal system.

The circuit did have a few issues if it were to be configured differently. The matrix system can cause serious issues if a single pin is configured differently leading it to being very fragile. The lights could be in a much very different order if care wasn't taken to see what exactly each pin output was timed at or if one was changed.

An introduction of an OR gate and an extra output pin was necessary to the circuit as the time of forty to fifty seconds whilst the pedestrian lights turn green, would allow the lights to just completely  off. Adding the OR gate to the circuit ensured that it could be programmed with an extra output to ensure that the traffic lights are red whilst the pedestrian lights are green.

Program screen shots

```
int increment;
unsigned int sevenseg [10] = {0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0XB0, 0X90};

sbit LED_Red at RB0_bit;
sbit LED_Yellow at RB1_bit;
sbit LED_Green at RB2_bit;
sbit LED_Red1 at RB3_bit;
sbit LED_Yellow1 at RB4_bit;
sbit LED_Green1 at RB5_bit;
sbit LED_PED_Red at RB6_bit;
sbit LED_PED_Green at RB7_bit;
sbit LED_Red_On at RA0_bit;

//Pin B0 controls a red light input
//Pin B1 controls a yellow light input
//Pin B2 controls a green light input
//Pin B3 controls another red light input
//Pin B4 controls another yellow light input
//Pin B5 controls another green light input
//Pin B6 controls the red pedestrian light
//Pin B7 controls the green pedestrian light
//Pin A0 controls the lights to be all red

void main()
{
    trisb = 0;
    portb = 0;
    trisc = 0;
    portc = 0;
    trisd = 0;
    portd = 0;
    trisa = 0;
    porta = 0;

    //Set tristate b as output
    //Set Port B to output low
    //Set tristate c as output
    //Set Port C to output low
    //Set tristate d as output
    //Set Port D to output low
    //Set tristate a as output
    //Set Port A to output low

    do
    {
        for (increment=0;increment<10;increment++)
        {
            portc = 0XC0;
            portd = sevenseg [increment];

            //Increment by 1 until 9 is reached
            //Display Port C sevensegment to 0
            //Increment the sevensegment display at port d
        }
    }
}
```

```
// Case 1
// Traffic Light 1
LED_Red = 0;
LED_Yellow = 1;
LED_Green = 0;

// Traffic Light 2
LED_Red1 = 1;
LED_Yellow1 = 0;
LED_Green1 = 0;
LED_PED_Red = 1;
LED_Red_On = 0;
Delay_ms (1000);

//Increment by 1 until 9 is reached
//Display Port C sevensegment to 1
//Increment the sevensegment display at port d

for (increment=0;increment<10;increment++)
{
    portc = 0xF9;
    portd = sevenseg [increment];

    // CASE 2
    // Traffic Light 1
    LED_Red = 0;
    LED_Yellow = 0;
    LED_Green = 1;

    // Traffic Light 2
    LED_Red1 = 0;
    LED_Yellow1 = 1;
    LED_Green1 = 0;
    LED_PED_Red = 1;
    Delay_ms(1000);

    //Set the first red led low
    //Set the first yellow led low
    //Set the first green led to high

    //Set the second red led to high
    //Set the second yellow led to low
    //Set the second green led to low
    //Set the pedestrian red light to high
    //Set the all red led's to low
    //Wait one second
}
```

<pre> PortC = 0xA4; PortD = sevenseg [increment]; // Case3 // Traffic Light 1 LED_Red = 0; LED_Yellow = 1; LED_Green = 0; // Traffic Light 2 LED_Red1 = 0; LED_Yellow1 = 0; LED_Green1 = 1; LED_PED_Red = 1; Delay_ms(1000); } // FOR 30-39 display for (increment=0;increment<10;increment++) { </pre>	<pre> //Change the first seven segment display to 2 //Increment the second seven display //Set the first red led low //Set the first yellow led to high //Set the first green led low //Set the second red led to low //Set the second yellow led to low //Set the second green led to high //Set the pedestrian red light to high //Wait one second //Increment by 1 until 9 is reached </pre>
--	--

<pre> PortC = 0xB0; PortD = sevenseg [increment]; // Case4 //Traffic Light 1 LED_Red = 1; LED_Yellow = 0; LED_Green = 0; // Traffic Light 2 LED_Red1 = 0; LED_Yellow1 = 1; LED_Green1 = 0; LED_PED_Red = 1; Delay_ms(1000); } for (increment=0;increment<10;increment++) { PortC = 0X99; PortD = sevenseg [increment]; //ALL THE TRAFFIC SIGNALS ARE RED // CASE 5 //Traffic Light 1 LED_Red = 0; LED_Yellow = 0; LED_Green = 0; // Traffic Light 2 LED_Red1 = 0; LED_Yellow1 = 0; LED_Green1 = 0; LED_PED_Green = 1; LED_Red_On = 1; Delay_ms(1000); } for (increment=0;increment<1;increment++) { PortC = 0X92; PortD = sevenseg [increment]; //ALL THE TRAFFIC SIGNALS ARE RED // CASE 5 </pre>	<pre> //Set the first seven segment display to 3 //Increment the second seven segment display //Set the first red led high //Set the first yellow led low //Set the first green led low //Set the second red led to low //Set the second yellow led to high //Set the second green led to low //Set the pedestrian red light to high //Wait one second //Increment by 1 until 9 is reached //Set the first seven segment display to 4 //Increment the second seven segment display //Set the first red led low //Set the first red led yellow //Set the first green led low //Set the second red led to low //Set the second yellow led to low //Set the second green led to low //Set the pedestrian green light to high //Turn all traffic lights red //Wait one second //Increment by 1 until 1 is reached //Set the first seven segment display to 5 //Increment the second seven segment display </pre>
---	--

Conclusion

Concluding the testing of the simulation and creation of the code, it can be found that the program works well with the circuit system with the code and hardware being set up a certain way.

The matrix system of setting up the circuit and integrating it with the code can be very fragile if changed ever so slightly, allowing the traffic lights to different signals than intended or not work at all. This could be resolved if additional pins were used to control the individual light signals rather than light signals sharing the same output pin.

The use of an OR gate to fix the issues of the lights all appearing off during the pedestrian crossing phase of the traffic signals is also a potential major flaw within the circuit. This could perhaps be rectified or improved on by using additional pins so more controls could be used on the signals at an individual basis.

Overall, the program is functional and would work in a real-world scenario without any issues. There are some improvements which could be made but are not necessary to the functionality of this traffic signal simulation.

References

- [1] [How Does a Traffic Light System Work? - Amperite Co.](#) (Judy Johnson, April 2022)
- [2] [How Does a Traffic Light System Work? - Amperite Co.](#) (Judy Johnson, April 2022)
- [3] [PIC16F877A | Microchip Technology](#) (Microchip)
- [4] [PIC16F877A - More Microcontroller Features | Microcontroller Tutorials \(teachmemicro.com\)](#) (Roland Pelayo, April 2018)
- [5] [The Top Five Reasons Why Traffic Management Is Important | by Kitty Schowalter | Medium](#) (Kitty Schowalter, February 2023)
- [6] Proteus simulation off Screen shot (Greg Dickson, April 2024)
- [7] Proteus simulation 1-10s Screen shot (Greg Dickson, April 2024)
- [8] Proteus simulation 10-20s Screen shot (Greg Dickson, April 2024)
- [9] Proteus simulation 20-30s Screen shot (Greg Dickson, April 2024)
- [10] Proteus simulation 30-40s Screen shot (Greg Dickson, April 2024)
- [11] Proteus simulation 40-50s Screen shot (Greg Dickson, April 2024)

Electronics 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

72/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5



Comment 1 | Knowledge & Understanding (20)

Good. Different Traffic Lights have been found. Still needs bit elaborated sentences regarding their types and their operations

PAGE 6



Comment 2 | Cognitive & Intellectual (20)

Good Work. More research on which criteria have been considered in selecting the particular microcontroller should be included in the report. Example, research on the power consumption, performance, cost and reliability.

PAGE 7



Comment 3 | Practical Application Sk(30)

Designing the circuit diagram is clear and neat. It could be more presentable if the road layout is added to the design.



Comment 4 | Transferable Skills (20)

The idea of numbering the sequence of work is much appreciated. This in case, give a clear idea for the persons who see the simulation for the first time.



Comment 5

True

PAGE 8

PAGE 9

PAGE 10



Comment 6 | Knowledge & Understanding (20)

Good explanation is made on each of the simulation criteria's which clearly shows that you have a good idea in the operation of the traffic light controller.

PAGE 11

PAGE 12

PAGE 13

PAGE 14



Comment 7 | Cognitive & Intellectual (20)

The lines of operations are clearly explained here. Keep the work

PAGE 15



Comment 8 | Practical Application Sk(30)

Good Idea to include the OR gates to the circuit.

PAGE 16

PAGE 17

PAGE 18

PAGE 19

GRADING FORM: EC1

GREG DICKSON

71.83

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)



Good identification of most relevant concepts, with minor omissions or inaccuracies. 3.5

6.7

Adequate background research, with a reasonable range of sources. Shows basic understanding with some gaps in coverage. 1.8

Good variety of sources. Most references are correctly formatted and cited, with minor errors.

1.4

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)



Demonstrates good knowledge of a variety of traffic light controller products. 7.3

13.8

Reflects an adequate understanding of the underlying principles in analysing different traffic light controllers. Provides basic analysis with some significant gaps. 6.5

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)*



Provides a limited review of the microcontroller selection. 4.2

Demonstrates exceptional innovative thinking in problem-solving and feature additions. 5.95

Provides clear, detailed, and well-structured pseudocode that accurately represents the system operation. Demonstrates a strong understanding of programming concepts and logical flow. 5.1

15.33

PRACTICAL APPLICATION SK(30)

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



The circuit design layout is clear and well-presented, showing a solid understanding of circuit design principles. Most necessary components are effectively incorporated. 7

Provides clear and accurate simulated output results that validate the functionality of the traffic light controller. 15

22

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.*



The report is well-organized, with a logical flow of ideas that is easy to follow. 5

Most necessary definitions, acronyms, and abbreviations are provided and correctly used.

They are generally clear and enhance understanding. 4

The report is professional and well-presented, with minor areas for improvement. 5

14