

Mehrshad J Traffic Light Report

by Mehrshad Jafari

Submission date: 05-Dec-2025 12:59PM (UTC+0000)

Submission ID: 269737560

File name: 58220_Mehrshad_Jafari_Mehrshad_J_Traffic_Light_Report_634321_1270354624.docx (12.8M)

Word count: 3242

Character count: 17421

Electronics I

ACDA5002

Designing an Automatic Traffic Light Controller For A T-Junction
Road

Mehrshad Jafari - 2410644

Table of contents

1. Introduction
2. Research
 - 2.1 What are traffic control devices
 - 2.2 types of traffic control systems
 - 2.3 where are traffic systems used
 - 2.4 choosing a microcontroller
3. Circuit Design
4. Flowchart
5. Proteus Simulation
6. Breadboard Simulation
7. Added Features
8. Circuit Code
9. Conclusion
10. References

1. Introduction

This project focuses on the development of a fully functional T-junction traffic light control system. The aim is to design a system that manages traffic through a time-based sequence while incorporating a safe and functioning pedestrian crossing. This will be achieved by integrating a 7-segment countdown display, LEDs, timing logic, and a microcontroller to provide a reliable and efficient solution for a real-world traffic system. The system will be developed to operate with precise 10 second intervals over a 40 second time period in order to regulate traffic flow.

Description of problem

This system must provide a predictable signaling system with effective intervals to ensure safe procedure for vehicles on all three incoming routes, therefore the following requirements must be done in order to achieve this;

Traffic signal lights at each intersection. Each direction should be equipped with a set of traffic lights to ensure safe passage and also providing drivers and pedestrians with real time awareness therefore reducing uncertainty at the junction.

Timing traffic light intervals. The system completes a full cycle of 40 seconds, broken into four 10 second phases. The phases must be programmed to change automatically, enduring smooth and coordinated traffic flow.

Dedicated pedestrian crossing, as required by brief and as replicated from real world scenarios, during the final 10 seconds of the cycle a period of time will be dedicated to ensuring the safety of pedestrians.

2. Research

Traffic control systems play a vital role in improving road safety in modern transportation networks, reducing congestion, and ensuring orderly movement of vehicles and pedestrians. The need for reliable and well-coordinated traffic system is increasingly important as the number of road users increases every year. This report focuses on the design and simulation of a traffic light system tailored for a T-junction.

2.1 What are traffic control devices?

Traffic controllers are visual and physical devices used to moderate and guide traffic. [1] These devices could include lights, signs and road marking to help enforce road safety, with the main goal of:

- Reducing traffic congestion
- Ensuring both pedestrian and vehicle safety
- Ensuring efficiency in traffic

2.2 Types of traffic control systems

There are several types of traffic signal controllers that can be categorized, each addressing specific conditions and requirements. For this project, a fixed time traffic controller was used. This operates on a programmed cycle duration where sets of traffic lights have their own phases with a constant duration.[2] This system was chosen for its simplicity and reliability due to its ease of understanding and implementing in PIC to simulate. However, some disadvantages are that as it's a predetermined system it will not be responsive to real-time traffic therefore it is unpredictable in traffic fluctuations.[3] Another type of traffic control system is a sensor circuit or vehicle triggered systems. These can be seen on main roads with higher amounts of traffic, including detectors and cameras, so when a vehicle is detected, the system is triggered and then allocates a green light based on the demand of traffic.[4] The advantages of this system are its highly responsive and efficient traffic conditions, while the disadvantages could be the higher cost and difficulty to implement. Therefore, a Fixed time control system will be used for this circuit

2.3 Where they can be found

Traffic controls systems are used in a wide variety of settings and areas. They are used in train networks to ensure trains can run efficiently without risking collisions or derailments. In addition, helping with coordination and planning, allowing the train to run when the track is safe and ensuring no other train is allowed on the same track.[5] Traffic control systems can also be found in the air, these systems rely on a combination of technologies and human communication to manage air traffic. Air traffic systems are more expensive than other forms as they use systems such as radars to ensure precise control over traffic and help predict and prevent accidents.[6]

However, this project focuses on designing a traffic control system used for cars and pedestrians at a T-junction. T-junctions are more prone to accidents and congestion [7] as they include three roads that merge with different priorities. Without control systems drivers may face confusion resulting in delays or even unsafe crossing procedures. Therefore, implementing an automated and reliable control system will be essential to ensure driver and pedestrian safety while also allowing smooth traffic flow.

2.4 Choosing a microcontroller

The PIC16F877A was chosen for the traffic light system due to it being one of the most widely used PIC microcontrollers as its reliable and versatile. The PIC16F877A has sufficient input and output pins providing 33 usable pins, this is essential as the high availability allows for multiple features to be added with room for more features in the future. Furthermore, the PIC also has built-in timers for accuracy as this circuit is a time-based circuit with excellent support and compatibility with Proteus and CCS. Its low cost and ease of use make it accessible to the broader audience.[8]

3. Circuit design

Full circuit design implemented in a visual design to simulate a real-life T-Junction system

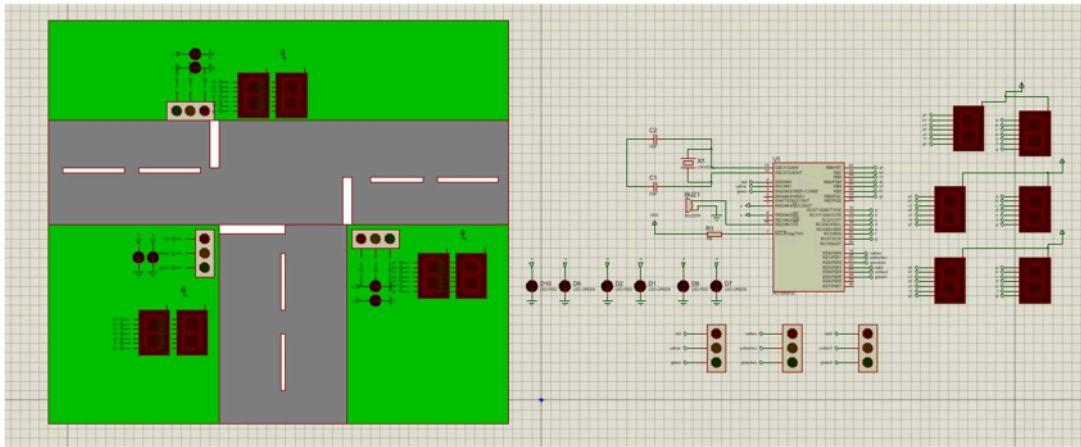


Figure 1: Shows the Full circuit design of system

Components used:

- 7 Segment display common anode - x6
- PIC16F877A
- Crystal oscillator
- 20pf Capacitor – x2
- Buzzer
- 10k resistor
- Red LED – x3

- Green LED – x3
- Traffic light LED system – x3

Circuit explanation

The traffic light control system relies on multiple stages of supporting components that allow the circuit to operate successfully.

7 Segment displays:

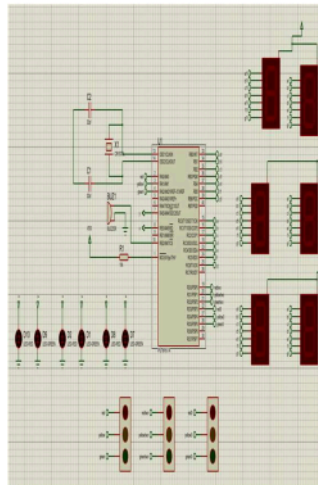


Figure 2: Shows 3 pairs of 7 segment displays

7 segment display is a single LED output device which displays decimal numbers from 0-9 [9] for this project the system requires a countdown timer from 40 seconds to 0. A 7-segment display provides a simple and clear output for both pedestrians and drivers. The 7-segment display was chosen as it's cost effective and easy to integrate in the PIC while also being coded with simple instructions.

Crystal oscillator:

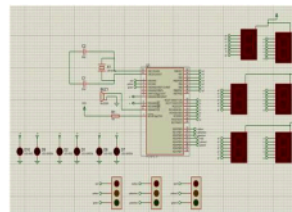


Figure 3: shows the block diagram of the crystal oscillator from circuit

This project requires a precise timing signal at a consistent rate. The use of a crystal oscillator provides a reliable and accurate timing reference [10]. This ensures that the 10 second phase of each road is accurate as even small delays can disrupt the system's timing. As for the capacitors C1 and C2, these are connected to ensure the stability of the crystal oscillator.

Buzzer:

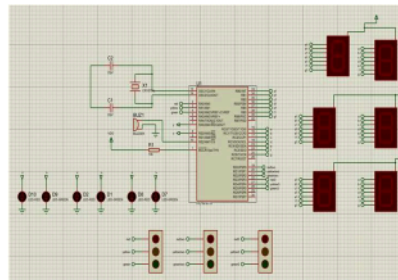


Figure 4: shows the components of a buzzer from the circuit

This additional feature provides an audible alert for visually impaired pedestrians allowing for safe travel for everyone. The feature allows for real-world realism in the traffic system simulation.

LEDs and traffic light:

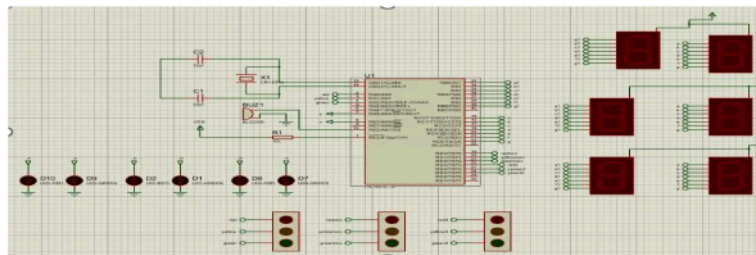


Figure 5: shows both Traffic light systems and Pedestrian signal LEDs

The use of green and red LEDs was specifically for pedestrians to ensure everyone's safety. The green indicates 'GO' and red indicates 'STOP'. These are important as they help confirm that the program is running correctly through the sequences whilst ensuring human safety.

The traffic light module provides clear visuals for drivers at the intersection connecting 3 roads safely. Each traffic light has a 10-second sequence in which they determine the priority of the roads. During the last 10 second phase all traffic lights a red and the pedestrian lights turn green. The traffic lights are connected by input and output pins to keep the circuit to be clean. The LEDs and traffic system have been integrated with the countdown sequence of the 7 segments to synchronize phases, presenting a real-life system.

4. Flowchart

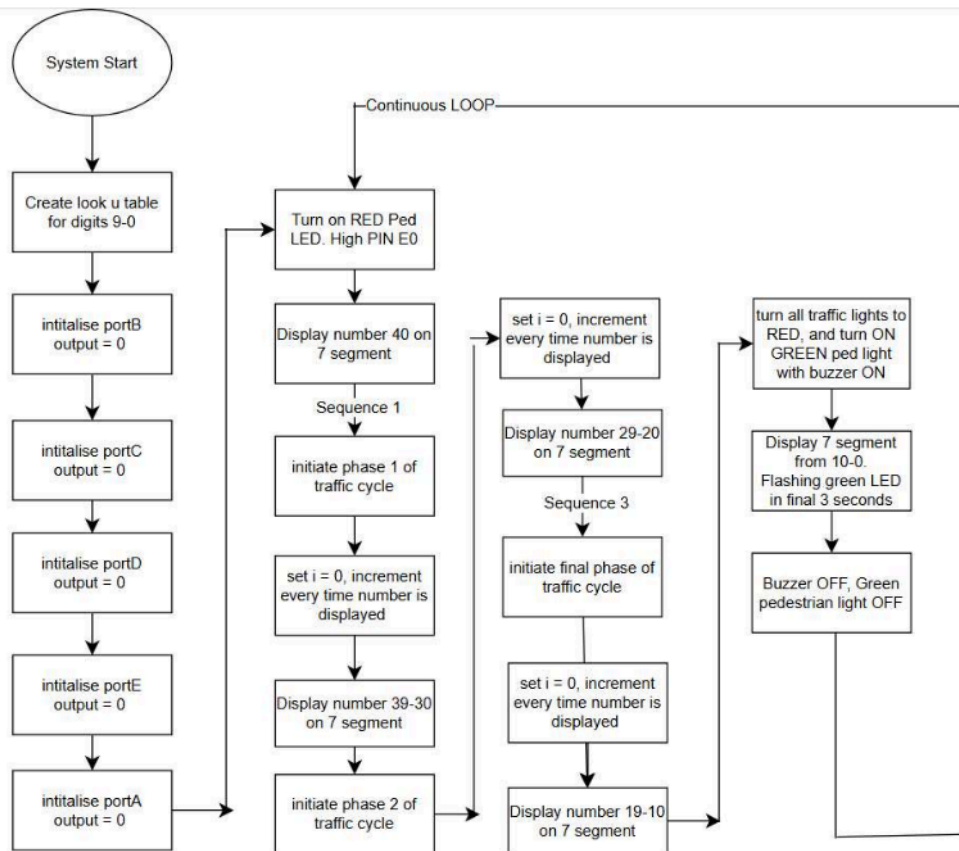


Figure 33: shows the circuit simulation explained via a flowchart

5. Simulation proteus

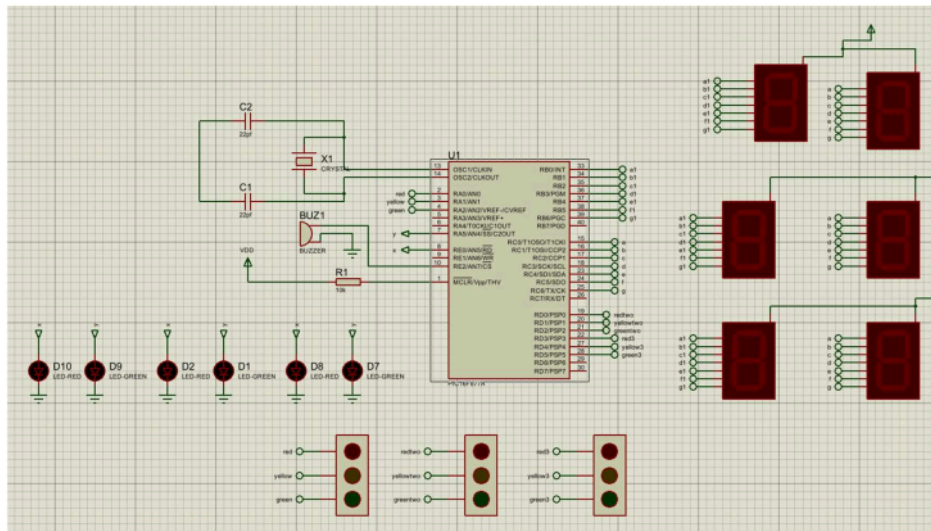


Figure 6: displays the full circuit design

This is the final project system which includes all the features that are required and additional safety applications which were added to improve the circuit system. A pair of 7 segment displays, one displaying the 10 second number and the other 1 second intervals, are connected to portB of the PIC and portC. As there were 3 different sequence for each traffic light system, this meant each traffic light had to have their own connection. Traffic light 1 is connected to portA, RA0, RA1, RA2. Traffic light 2 is connected to ports RD0, RD1, RD2. While traffic light 3 is also connected to portD, it is connected via ports RD3, RD4 and RD5. For the 3 pairs of pedestrian lights only 2 ports were needed and it was only used to indicate whether the LED was green or red.

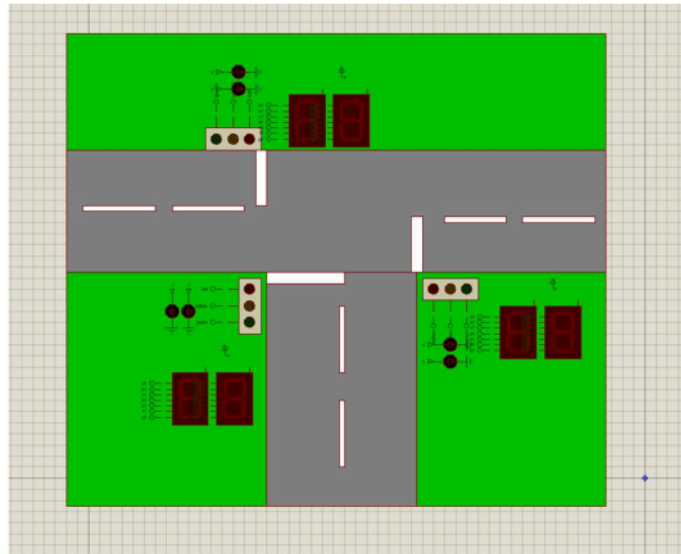


Figure 7: shows the circuit design implemented in a real-world design scenario

This is the real-life scenario as each junction has its own set of lights, pedestrian crossings, and countdown display. The circuit was implemented in a completed picture to show the necessity and format of the project. This indicates the need for this system as the roads can lead to several problems both for drivers and pedestrians.

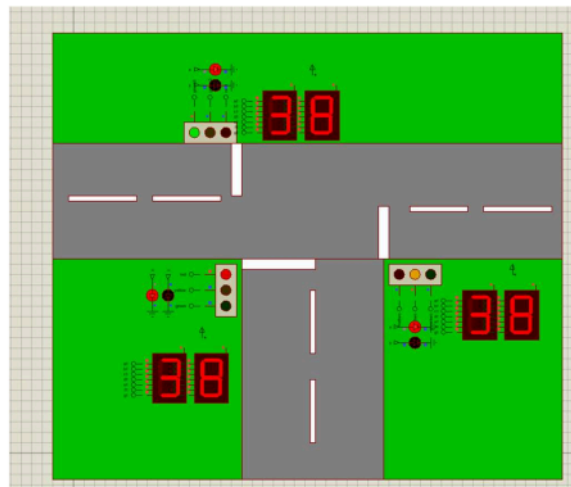


Figure 8: shows the working circuit on phase 1 of the cycle

This is the first cycle of the traffic light sequence, this initial phase immediately starts when the system has started or when the circuit loops back after completing the full 4 cycles.

During this stage the primary road is given priority to allow vehicle safe passage. This cycle happens from 40-30 seconds and is also displayed on the countdown display.

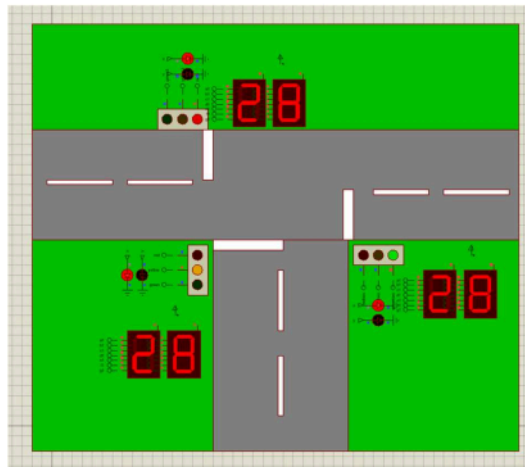


Figure 9: demonstrates the second cycle, countdown from 29-20

This is the second phase of the cycle, the countdown is from 29-20. traffic light 1 to yellow, traffic light 2 to green and finally traffic light 3 to red. Again, this all happens when the pedestrian light is red. This cycle gives priority to the right-side junction.

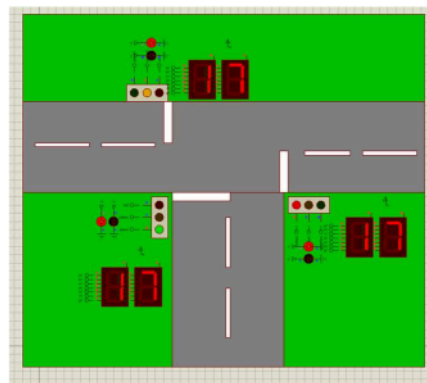


Figure 10: shows the 3rd cycle from 19-10

The final traffic light cycle from 19-10 ensures traffic light 1 is green, traffic light 2 is red, and traffic light 3 is yellow. This means now all traffic lights have had their respective cycles.

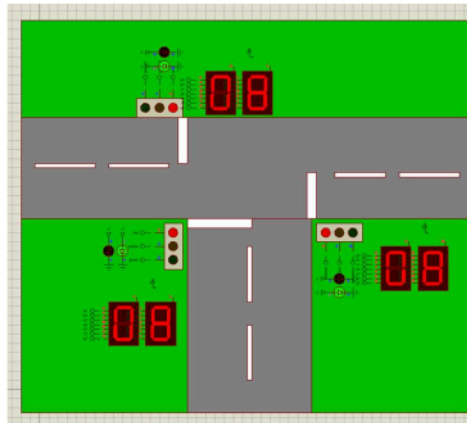
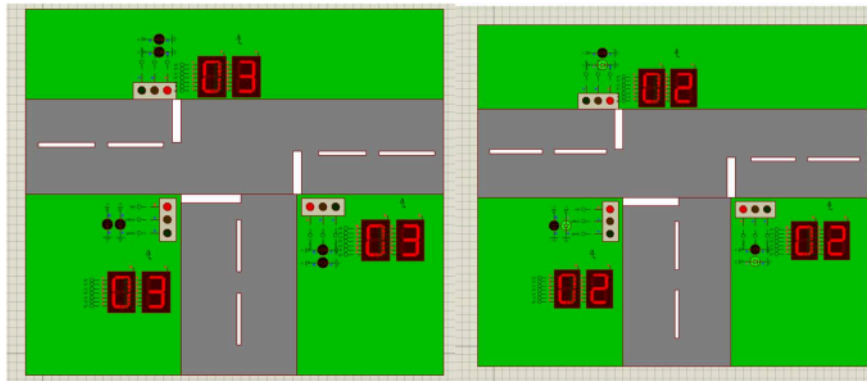


Figure 11: shows the pedestrian cycle

The final 10 seconds of this cycle are reserved for the pedestrians to ensure their safety as all traffic lights are red apart from the pedestrian lights which are green. In addition to this, a buzzer has been added to improve the features of this circuit as it adds a safety measure allowing visually impaired pedestrians to cross when they hear the sound.



Figures 12 & 13: show the flashing sequence

Another additional feature that was added to help improve the circuit was a flashing segment, which would occur in the last 3 seconds of the pedestrian cycle, flashing the green LED on and off, to allow the pedestrians to be aware that their time is coming to an end and they need to be cautious.

6. Simulation Breadboard

This is the completed real life breadboard connection of the circuit derived from Proteus.

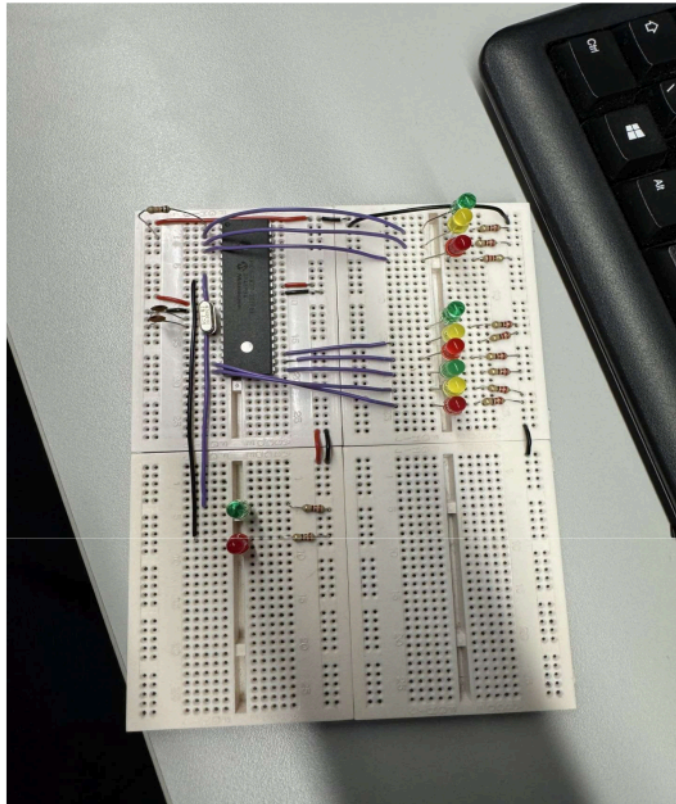


Figure 14: shows the Breadboard circuit

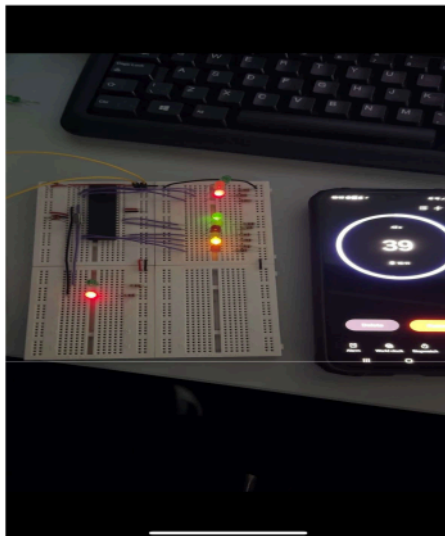


Figure 15: shows the working breadboard cycle from 40-30

This first 10 second phase of this cycle consists of traffic light 1 being red, traffic light 2 as yellow and green activated on traffic light 3, this all happens when the pedestrian light is red and the 7-segment countdown is from 40-30.



Figure 16: shows the second cycle on the breadboard

The second cycle which is from 29 to 20 switches traffic light 1 to yellow, traffic light 2 to green and finally traffic light 3 to red. Again, this all happens when the pedestrian light is red

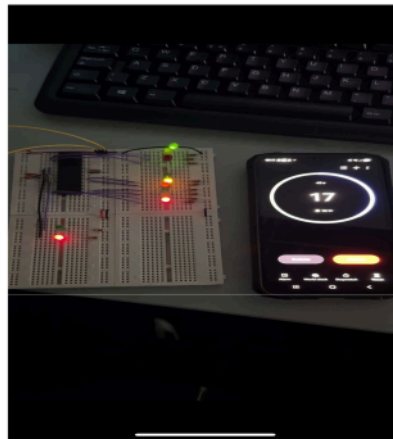


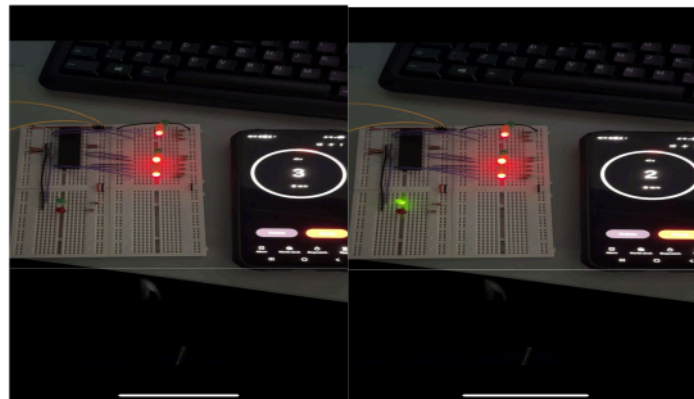
Figure 17: shows the final vehicle cycle

The final traffic light cycle from 19-10 ensures traffic light 1 is green, traffic light 2 is red, and traffic light 3 is yellow. This means now all traffic lights have had their respective cycles.



Figure 18: shows the pedestrian cycle

The final 10 seconds reserved for the pedestrians means all the traffic lights are red, and the pedestrian crossing is green along with the buzzer that sounds giving the visually impaired road users an alternative signal.



Figures 19 & 20: show the flashing sequence on breadboard

A feature that was added is a flashing sequence in the final 3 seconds of the pedestrian cycle to give users a visual warning that their time is ending and to remain cautious.

Visual presentation of traffic light cycle

Sequence 1	Traffic light 1	Traffic light 2	Traffic light 3
Time period 10s			

From 40-30s			
Pedestrian light			

Sequence 1	Traffic light 1	Traffic light 2	Traffic light 3
Time period 10s			
From 29-20s			
Pedestrian light			

Sequence 3	Traffic light 1	Traffic light 2	Traffic light 3
Time period 10s			
From 19-10s			
Pedestrian light			

Pedestrian seq	Traffic light 1	Traffic light 2	Traffic light 3
Time period 10s			
10-0s			
Pedestrian light			

7. Features added to improve the circuit

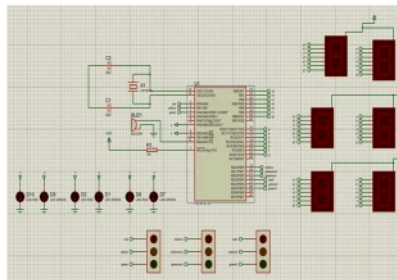
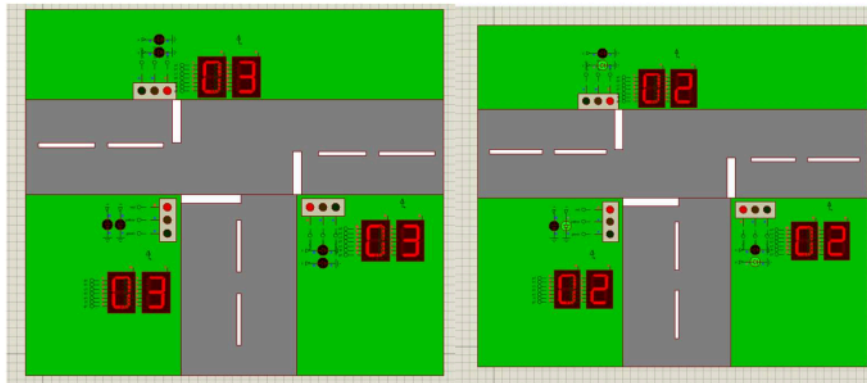


Figure 21: shows the buzzer block diagram



Figures 22 & 23: show the flash pedestrian sequence

The use of the buzzer in the pedestrian crossing improves both safety and usability of the traffic system, as buzzer are also used in real world traffic crossing systems, this is especially important as this is a busy junction and audible warnings are essential for vulnerable road users such as the elderly and visually impaired pedestrians. This reduces confusion as users have a clear cue to go.

The purpose of the flashing sequence in the final 3 seconds is to act as a warning which can be a critical safety feature. This indicates to users that crossing time is about to finish and those currently walking must cross quickly. This prevents late crossing which could place pedestrians in danger and creates a sense of urgency for those crossing.

With these combined safety features, it creates a multi pedestrian alert system as safety is one of the main goals. These improvements demonstrate a thoughtful and attentive design.

8. Circuit code

```
#include <main.h>

#define LED_RED PIN_A0
#define LED_YELLOW PIN_A1
#define LED_GREEN PIN_A2
#define BUTTON_RD7 PIN_D7
#define BLUELED_RB7 PIN_B7
#define LED_REDTHREE PIN_D3
#define LED_YELLOWTHREE PIN_D4
#define LED_GREENTHREE PIN_D5

#define LED_RED PIN_A0
#define LED_YELLOW PIN_A1
#define LED_GREEN PIN_A2
#define BUTTON_RD7 PIN_D7
#define BLUELED_RB7 PIN_B7
#define LED_REDTHREE PIN_D3
#define LED_YELLOWTHREE PIN_D4
#define LED_GREENTHREE PIN_D5

unsigned int display [10] = {0x90, 0x88, 0xF8, 0x82, 0x92, 0x99, 0xB8, 0xA4, 0xF9, 0xC0}; //Lookup table for the 7-segment display, Represents numbers 0-9
int i;
int button_pressed = 0; // sets the count to 0 to indicate whether button has been pressed
```

Figure 24: shows the code in Proteus

This is creating an array for the number 9 to 0 so it can be stored and then called upon to when it is needed. The “[10]” suggests that there are 10 numbers to be stored.

```
void main()
{
    output_low(PIN_B7); // Force RB7 low immediately at start
    delay_ms(10);      // Small delay

    set_tris_b(0x00); //set PORTB as output pin for the 10s of 7-segment display
    output_b(0x00);   //initialise PORTB to 0

    set_tris_c(0x00); //set portc as output pin for 1s 7 segments
    output_c(0x00);   // initialise portc to 0

    set_tris_d(0x00); //set portd as output for LEDs (traffic lights 2 and 3)
    output_d(0x00);   // initialise portd to 0

    set_tris_e(0x00); //set porte as outputs
    output_e(0x00);   // initialise porte to 0

    set_tris_a(0x00); //set porta as output for traffic light 1
    output_a(0x00);   // initialise porta to 0
}
```

Figure 25: shows the void main code

This section clears all the ports ensuring there is nothing to affect or disrupt the following code.

```
while (TRUE)
{
    if (input(PIN_D7) == 1) //checks if button connected to pin d7 is connected
    {
        button_pressed = 1; //if pressed sets count to 1 to indicate button pressed
    }
    if (button_pressed == 1)
    {
        output_high(PIN_B7); // turn on blue led if button is pressed
        output_high(PIN_E0); //Red pedestrian light is on
        output_low(PIN_A5); //Green pedestrian light is off
        output_b(0x00); //sets the 7 segment display number to 0 on the 10s
        output_c(0x00); // sets the 7 segment display number to 0 on the 1s
        delay_ms(1000); // small delay to show value 00 on 7 segment display
    }
}
```

Figure 26: shows the initiation of the while loop

This is within the ‘while true’ loop, meaning this is the start of the continuous loop. This part sets the pedestrian lights while also starting the 7-segment display from 40.

```

//sequence 1
output_high(LED_RED);           // RED led is on, on first traffic light

output_high(LED_YELLOWTWO);     // YELLOW led is on, on second traffic light

output_high(LED_GREENTHREE);    // GREEN led is on, on third traffic light

for (i = 0; i < 10; i++)
{
    output_b(0xB0);              // displays 3 on the 10s 7 segment display
    output_c(display[i]);        //Display numbers from 9-0 on 7 segment with 1 second delay in between
    delay_ms(1000);              //from 39-30
}

```

Figure 27: shows the code for sequence 1, cycle 1

Sequence 1 indicates phase 1 of the cycle, which sets the traffic lights and instructs a loop for a specific number of times, in this case 10. i is set to 0 before loop starts and is incremented by 1. When counting down on the 7 segments, each phase is 10 seconds. From 39-30

```

//sequence 2
output_low(LED_RED);            //Red LED off, first traffic light
output_high(LED_YELLOW);        //Yellow LED on, first traffic light
output_low(LED_GREEN);          //Green LED off, first traffic light
output_low(LED_REDTWO);         //Red LED off, second traffic light
output_low(LED_YELLOWTWO);      //yellow LED off, second traffic light
output_high(LED_GREENTWO);       //Green LED on, second traffic light
output_high(LED_REDTHREE);       //Red LED on, third traffic light
output_low(LED_YELLOWTHREE);     //Yellow LED off, third traffic light
output_low(LED_GREENTHREE);      //Green LED off, third traffic light

for (i = 0; i < 10; i++)
{
    output_b(0xA4);              // displays 2 on the 10s 7 segment display
    output_c(display[i]);        //Display numbers from 9-0 on 7 segment with 1 second delay in between
    delay_ms(1000);              // from 29-20
}

```

Figure 28: shows the code for cycle 2

Sequence 2 turns off the previously on LEDs and switches on the sequence for phase 2 and again set i to 0 and starts increments as long as i is less than 10. The code also displays the array on the 7-segment display.

```

//sequence 3
output_low(LED_RED);           //Red LED off, first traffic light
output_low(LED_YELLOW);        //Yellow LED off, first traffic light
output_high(LED_GREEN);        //Green LED on, first traffic light
output_high(LED_REDTHREE);     //Red LED on, second traffic light
output_low(LED_YELLOWTWO);     //yellow LED off, second traffic light
output_low(LED_GREENTWO);      //Green LED off, second traffic light
output_low(LED_REDTHREE);      //Red LED off, third traffic light
output_high(LED_YELLOWTHREE);  //Yellow LED on, third traffic light
output_low(LED_GREENTHREE);    //Green LED off, third traffic light

for (i = 0; i < 10; i++)
{
    output_b(0xF9);             // displays 1 on the 10s 7 segment display
    output_c(display[i]);       //Display numbers from 9-0 on 7 segment with 1 second delay in between
    delay_ms(1000);             // from 19-10
}

```

Figure 29: shows the cycle 3, final traffic light sequence

```

//Pedestrian light sequence

output_high(LED_RED);          //Red LED on, first traffic light
output_low(LED_YELLOW);        //Red LED off, first traffic light
output_low(LED_GREEN);         //Green LED off, first traffic light
output_high(LED_REDTHREE);     //Red LED on, second traffic light
output_low(LED_YELLOWTWO);     //yellow LED off, second traffic light
output_low(LED_GREENTWO);      //Green LED off, second traffic light
output_high(LED_REDTHREE);     //Red LED on, third traffic light
output_low(LED_YELLOWTHREE);   //Yellow LED off, third traffic light
output_low(LED_GREENTHREE);    //Green LED off, third traffic light
output_low(Pin_E0);            // Red pedestrian ligh off
output_high(Pin_A5);           // Green pedestrian light on
output_high(PIN_E2);           // Buzzer on when pedestrain light is green

```

Figure 30: shows the code that turns off all the traffic light sequences

for the pedestrian light sequence all the traffic lights are turned off, and the green pedestrian light is turned on

```

{
    output_low(PIN_B7);
    button_pressed = 0;

    output_b(0xC0);          // display 0 on 7 segment
    output_c(0x90);          // displays 9 on 7 segment
    delay_ms(1000);          // 1 second delay

    output_b(0xC0);          // display 0 on 7 segment
    output_c(0x80);          // displays 8 on 7 segment
    delay_ms(1000);

    output_b(0xC0);          // display 0 on 7 segment
    output_c(0xF0);          // displays 7 on 7 segment
    delay_ms(1000);

    output_b(0xC0);          // display 0 on 7 segment
    output_c(0x82);          // displays 6 on 7 segment
    delay_ms(1000);

    output_b(0xC0);          // display 0 on 7 segment
    output_c(0x92);          // displays 5 on 7 segment
    delay_ms(1000);

    output_b(0xC0);          // display 0 on 7 segment
    output_c(0x99);          // displays 4 on 7 segment
    delay_ms(1000);
}

output_b(0xC0);             // display 0 on 7 segment
output_c(0x80);             // displays 3 on 7 segment
output_low(Pin_A5);         //green led flashing off
delay_ms(400);
output_high(Pin_A5);        //green led flashing on
delay_ms(400);
output_b(0xC0);             // display 0 on 7 segment
output_c(0xA4);             // displays 2 on 7 segment
delay_ms(800);
output_low(Pin_A5);         //green led flashing off
delay_ms(200);
output_high(Pin_A5);        //green led flashing on
delay_ms(400);
output_c(0xF9);             // displays 1 on 7 segment
delay_ms(800);
output_low(Pin_A5);         //green led flashing off
delay_ms(400);
output_high(Pin_A5);        //green led flashing on
delay_ms(400);
output_c(0xC0);             // display 0 on 7 segment
delay_ms(800);
output_low(Pin_A5);         // Green pedestrian light off
output_low(pin_E2);         // Buzzer off
delay_ms(600);
}
}

```

Figure 31 & 32: show the manually countdown followed by a flashing sequence

This section manually counts down as the flashing sequence begins in the final 3 seconds

9. Conclusion

This project was successfully demonstrated by simulating a fully functional traffic light system via a PIC16F877A microcontroller and created in real life with a breadboard with the objective of implementing a fixed time-based traffic controller with countdown display, pedestrian crossing and a coordinated light sequence.

Throughout this project, the use of several circuits and systems was used to achieve a final working system. This included a PIC16F877A microcontroller which served as the main brain and central unit controlling the system. Along with this was a 7-segment display, LEDs, and a buzzer. The use of the crystal oscillator ensured stable timing, which is essential for this task. In addition, a common anode 7 segment display was included to visually represent the 40 second countdown for each traffic light, which required multiplexing and port configuration. The circuit also included red, yellow, and green LED traffic modules that are installed at each junction of the road along with separate pedestrian LEDs. These were programmed to operate at a fixed time sequence. Each direction has a 10 second cycle, and the final 10 seconds are dedicated to the pedestrians. During the pedestrian LED cycle, a buzzer was also implemented and activated as one of the improvements to the circuit to immediately alert pedestrians, another hazard prevention feature added was a flashing sequence at the end of the pedestrian cycle,

which activated at the last 3 seconds of the pedestrian cycle to give a visual alert to pedestrians that their cycle is coming to an end. The CCS Program was responsible for creating delays, switching traffic states, updating the 7-segment outputs, and displaying the pedestrian LED along with buzzer. The loop ensured continuous and predictable operation, which was the aim of this project. The code executed all required tasks, validated system states, and followed the operational cycle. Both simulations, one proteus and one real life breadboard, confirmed that the project requirements were met. The traffic light system changed sequence as intended at the correct intervals, countdown values updated accurately, and also the pedestrian crossing with the additional feature of a buzzer worked reliably.

Some future recommendations could be a pedestrian button that could alert users that the pedestrian button has been activated. Another improvement could be to use a sensor circuit system which can offer better traffic flow, preventing any problems or buildup of traffic. Overall, the completed circuit fulfills the assignment objectives and provides a fully functional traffic control system with additional safety features.

10. References

- [1] Ryan Fisher, "Traffic Control Devices for Facilities", Reliance Foundry, 19 March 2025, [online] Available: <https://www.reliance-foundry.com/blog/traffic-control-devices>, [Accessed, 03 December 2025]
- [2] "Fixed-time control systems", 1 Library, [online] Available: https://1library.net/article/fixed-time-control-systems-urban-traffic-control-systems.ynnl69ly#google_vignette, [Accessed, 01 December 2025]
- [3] "Fixed-time control systems", 1 Library, [online] Available: https://1library.net/article/fixed-time-control-systems-urban-traffic-control-systems.ynnl69ly#google_vignette, [Accessed, 01 December 2025]
- [4] "How modern traffic light signals work", automate systems, [online] Available: <https://www.automatesystems.co.uk/traffic-lights-signals-work>, [Accessed 01 December 2025]
- [5] "Signals Explained", Network rail, 5 September 2018, [online], Available: <https://www.networkrail.co.uk/stories/signals-explained/> [Accessed, 02 December 2025]
- [6] "How air traffic control systems work", Sheffield school of aeronautics, [online] Available: <https://www.sheffield.com/articles/how-air-traffic-control-systems-work>, [Accessed 02 December 2025]
- [7] "Hazards that you'll encounter at T-Junctions", Learn Automatic, [online] Available: https://www.learn-automatic.com/tutorials/t-junctions/hazards-at-t-junctions/#google_vignette, [Accessed 03 December 2025]
- [8] "PIC16F877A: A breakdown of the basics and how to select one for your project" EMS, 3 July 2024, [online] Available: <https://electronicmanufacturingservice.org/pic16f877a-a-breakdown-of-the-basics-and-how-to-select-one-for-your-project/>, [Accessed 01 December 2025]
- [9] "7 Segment Display" Electronics Tutorials [online] Available: <https://www.electronicstutorials.ws/blog/7-segment-display-tutorial.html>, [Accessed 02 December 2025]
- [10] "Crystal oscillator: Circuit, Frequency and working principle" Electrical 4 u, 24 June 2024 [online] Available: <https://www.electrical4u.com/crystal-oscillator/> [Accessed 03 December 2025]