

Electronics 1 Coursework

Stefan-Cristian Daitoiu

2317310

Introduction	3
Background Research	4
Traffic Light Control Systems	4
Pedestrian Safety Systems	5
Microcontroller Applications in Embedded Systems	5
PIC16F877A Microcontroller	6
Seven-Segment Display Operation	7
Circuit Design and System Implementation	8
Traffic Signal Circuit Design	9
Pedestrian Crossing Circuit Design	9
Seven-Segment Countdown Display Design	11
Simulation and Testing	14
Traffic Light Testing	14
Pedestrian System Testing	14
Seven-Segment Display Testing	15
Conclusion	15
Bibliography	16

Introduction

Traffic management is an essential part of modern infrastructure, ensuring the safe and efficient movement of vehicles and pedestrians. Intersections require effective control systems because multiple traffic flows interact within a limited area. Without suitable control, intersections can experience congestion, increased waiting times, and a higher risk of accidents. T-junctions require carefully coordinated signalling to control traffic movements safely and independently. [1]

The aim of this project was to design and simulate an automated traffic light control system for a T-junction using a PIC16F877A microcontroller. The system was developed using Proteus simulation software, with the control program written using the CCS C Compiler. The project demonstrates the application of embedded systems principles through the integration of a microcontroller with electronic components to create an automated control system.

The traffic controller consists of three vehicle signal units, each containing red, amber, and green LED indicators to represent real-world traffic lights. The system operates using a fixed timing sequence, allowing each road to receive priority before transitioning to the next stage. A pedestrian crossing phase was also included, where all vehicle signals change to red before the pedestrian signal turns green, preventing conflicting movements.

A two-digit common anode seven-segment display was implemented to provide a countdown indication for each traffic phase. Multiplexing was used to control the display digits, reducing the number of required microcontroller pins while increasing software timing requirements. The PIC16F877A was selected due to its suitable number of digital outputs, timer functionality, and compatibility with C-based embedded programming. Timer1 was used to control the timing sequence, demonstrating effective resource management within the microcontroller.

The main objectives of the project were:

- To design a working automated T-junction traffic light controller.
- To control three independent vehicle traffic signals.
- To implement a pedestrian crossing safety phase.
- To display countdown timing using a seven-segment display.
- To develop the circuit using Proteus simulation software.
- To program the PIC16F877A using CCS C.
- To analyse the limitations and performance of the final design.

The completed system demonstrates how embedded microcontrollers can be used to replace complex hardware logic circuits while providing increased flexibility and easier modification.

Background Research

Traffic Light Control Systems

Traffic light systems are electronic control systems used to regulate the movement of vehicles and pedestrians at road intersections. The operation of a standard traffic light system is based on three main signal states. A green signal allows vehicles to proceed when the route is clear, an amber signal warns drivers that the signal is about to change and prepares them to stop, while a red signal requires vehicles to stop and wait until movement is permitted again. [2]

The correct sequencing of these states is important because incorrect timing could result in dangerous situations. For example, if two roads were given green signals simultaneously at a T-junction, vehicles could enter the same area and potentially collide. [3]

Traditional traffic light systems were often implemented using dedicated electronic logic circuits, counters, and timing components. Although these systems can operate effectively, they require significant hardware complexity and are difficult to modify. A microcontroller-based solution provides a more flexible alternative because the traffic sequence can be modified through software without changing the physical circuit. [4]

For this project, a fixed timing approach was selected. Each road receives a 10-second green phase followed by an amber transition period. After all three vehicle phases are completed, a pedestrian crossing phase occurs for 10 seconds. This creates a complete 40-second cycle.

Pedestrian Safety Systems

Pedestrian safety is an important consideration in traffic management. In real-world traffic systems, pedestrian crossings are normally integrated into traffic signal controllers to ensure that pedestrians are provided with a safe crossing period.[5]

A pedestrian crossing system generally operates by stopping all vehicle movement before allowing pedestrians to cross. This prevents situations where pedestrians and vehicles are given priority simultaneously.

The implemented design follows this principle. During the pedestrian phase:

- Road 1 displays red.
- Road 2 displays red.
- Road 3 displays red.
- Pedestrian green signal activates.

This creates a dedicated crossing period where pedestrians have priority.

Although commercial systems often include pedestrian push buttons, infrared sensors, cameras, or intelligent traffic monitoring systems, a fixed-time pedestrian phase was suitable for this prototype because the aim was to demonstrate the fundamental operation of a traffic controller.[6]

Microcontroller Applications in Embedded Systems

Microcontrollers are widely used in modern electronic systems because they combine processing capability, memory, and input/output control within a single integrated circuit. They are commonly applied in areas such as automotive systems, industrial automation, consumer electronics, robotics, and communication systems. [6] Compared with purely hardware-based designs, microcontroller-based systems offer advantages including reduced component count, lower cost, increased flexibility, easier modification, and the ability to implement complex control logic through software. This removes the need for multiple hardware timers and logic circuits, as system behaviour can be programmed directly into the microcontroller. [7]

For this project, the PIC16F877A was selected due to its balance between functionality and simplicity. Its multiple digital input/output ports and integrated timer peripherals make it suitable for controlling the traffic light LEDs, pedestrian indicators, and countdown display used within the system.

PIC16F877A Microcontroller

The PIC16F877A is an 8-bit microcontroller produced by Microchip Technology. It is commonly used in educational and industrial embedded applications due to its reliability and available features. [8]

Important features used within this project include:

- 33 input/output pins.
- Multiple timer modules.
- Interrupt support..
- Compatibility with CCS C programming.

The microcontroller was configured so that different ports controlled different parts of the system.

The output allocation was:

Port	Function
PORTB	Pedestrian LEDs and seven-segment segments
PORTC	Seven-segment segments, digit control, Road 1 green
PORTD	Vehicle traffic lights

The use of Timer1 allowed the program to generate accurate timing intervals. The timer interrupt periodically updates the cycle counter, allowing the main program loop to focus on updating outputs and refreshing the display.

Using a single timer was a design constraint of the project. While additional timers could simplify display refreshing, using one timer demonstrates efficient management of limited microcontroller resources.

Seven-Segment Display Operation

Seven-segment displays are commonly used in embedded systems where numerical information needs to be displayed. They consist of seven LED segments labelled A to G, which can be individually controlled to create numerical characters. [9]

This project uses a common anode seven-segment display to provide a countdown timer showing the remaining time of the current traffic phase. A two-digit display was selected because the complete traffic cycle requires values to be displayed from 40 seconds down to 0 seconds. Since both digits share the same segment connections, multiplexing [10] was implemented to allow the microcontroller to control both digits using fewer output pins. The display operates by rapidly switching between the two digits, where the first digit is enabled and updated with the required segments before being disabled, followed by the same process for the second digit. This process repeats continuously at a high speed, creating the appearance of a continuously illuminated two-digit display.

The switching speed is fast enough that the human eye perceives both digits as continuously illuminated.

However, software multiplexing has limitations. The refresh rate depends on program execution speed and delays within the software. During Proteus simulation, slight flickering was observed. This occurs because the display is not continuously driven like a dedicated hardware display controller. Increasing the refresh rate or using a dedicated display driver would improve performance. [11]

Circuit Design and System Implementation

The circuit was designed using Proteus simulation software to represent a complete embedded traffic control system. The design consists of the PIC16F877A microcontroller as the central processing unit, discrete LED components representing traffic signals, a pedestrian crossing indicator, a two-digit seven-segment countdown display, and supporting components including a crystal oscillator and power connections.

The design process involved selecting suitable components, assigning microcontroller input/output pins, developing the control software, and testing the complete system operation through simulation.

The main design consideration was ensuring that the PIC16F877A had enough available output pins to control all required components. Since the project required three traffic lights, each containing three colours, a total of nine vehicle signal outputs were required. Additional outputs were required for the pedestrian system and seven-segment display. The available PIC16F877A ports were therefore carefully allocated to ensure correct operation.

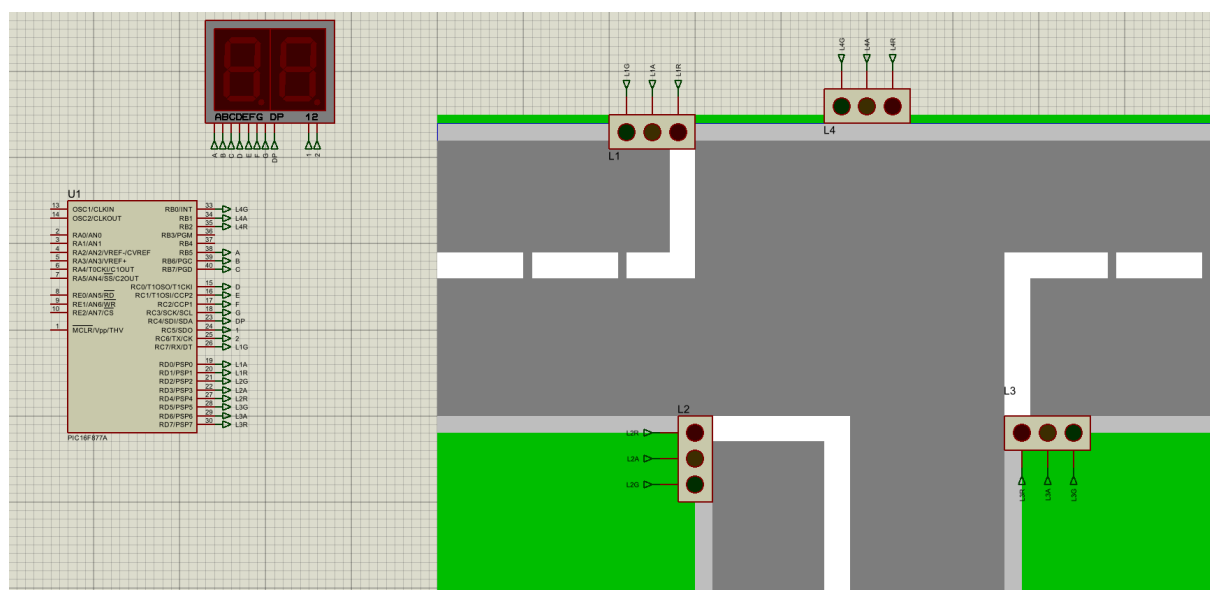


Figure 1: Complete circuit

Traffic Signal Circuit Design

Three traffic lights were implemented to represent the three roads connected to the T-junction. Each traffic light consisted of three LEDs:

- Green LED
- Amber LED
- Red LED

Each LED was connected to an individual output pin of the PIC16F877A through a current-limiting resistor. The resistor protects the LED by reducing current flow and prevents damage to both the LED and microcontroller output pins.

The traffic light connections were assigned as follows:

Traffic Light	Colour	PIC16F877A Pin
Road 1	Green	RC7
Road 1	Amber	RD0
Road 1	Red	RD1
Road 2	Green	RD2
Road 2	Amber	RD3
Road 2	Red	RD4
Road 3	Green	RD5
Road 3	Amber	RD6
Road 3	Red	RD7

Table 1: Individual lights pinout

The software controls these outputs by setting individual pins high or low depending on the required traffic phase.

To improve safety, an "all red" function was included within the program. Before activating a new road phase, all vehicle signals are temporarily returned to red. This prevents the possibility of conflicting green signals occurring during transitions.

Pedestrian Crossing Circuit Design

The pedestrian crossing system was implemented using an RGB LED to represent pedestrian signals. This approach simplified the circuit while still demonstrating the required functionality.

The pedestrian connections were:

Signal	PIC16F877A Pin
Pedestrian Green	RB0
Pedestrian Amber	RB1
Pedestrian Red	RB2

Table 2: Pedestrian light connections

During normal vehicle operation, the pedestrian red signal remains active. During the final 10 seconds of the traffic cycle, the program disables vehicle movement and activates the pedestrian green signal.

This design follows the same principle used in real traffic systems, where pedestrian movement is only permitted when vehicle traffic has been stopped.

Although the pedestrian amber output was included in the circuit, it was not used in the final sequence. Future improvements could implement a flashing pedestrian amber warning phase before returning control to vehicle traffic.

Seven-Segment Countdown Display Design

A two-digit common anode seven-segment display was used to display the remaining time for each traffic phase.

The display connections were:

Segment	PIC16F877A Pin
A	RB5
B	RB6
C	RB7
D	RC0
E	RC1
F	RC2
G	RC3
Decimal Point	RC4

Table 3: 7 segment display digit pinout

The digit control pins were:

Digit	PIC16F877A Pin
Left Digit	RC5
Right Digit	RC6

Table 4: Individual digit selection

The common anode pin of the display was connected to the positive supply voltage. Since the display is a common anode one, the PIC outputs control the segments by sinking current.

The countdown display operates by showing the remaining seconds of the active phase. For example:

- Road 1 phase begins with 10 seconds displayed.
- The display decreases until the phase ends.
- Road 2 begins and the countdown restarts.
- The same operation occurs for Road 3.
- The pedestrian phase displays the remaining crossing time.

The display uses software multiplexing. The PIC rapidly alternates between the left and right digits, creating the appearance of a continuously illuminated two-digit display.

A limitation of this method is that the display depends on software timing. During simulation, some flickering was observed due to the refresh rate and processing time of the PIC. This was analysed as a limitation of the chosen implementation rather than a complete system failure.

Software Design and System Operation

The control program was written in C using the CCS C Compiler. The software was designed using separate functions to improve readability and simplify debugging.

The main software functions include:

- `display_number()`
- `all_red()`
- `update_lights()`
- `update_countdown()`
- `timer1_isr()`

The program operates using an interrupt-driven timing system.

Timer1 was configured to periodically interrupt the microcontroller. Each interrupt increases the value of the “seconds” variable. When the counter reaches 40 seconds, it resets and begins the traffic sequence again.

The main program loop continuously performs three tasks:

1. Update traffic light states.
2. Update the countdown value.
3. Refresh the seven-segment display.

This structure allows the hardware outputs to continuously reflect the current system state.

Pseudocode Description

The operation of the system can be described using the following pseudocode:

WHILE system is running:

 Check current timer value

 IF seconds < 10:

 Set all lights red

 Activate Road 1 green

 IF final second:

 Activate Road 1 amber

 ELSE IF seconds < 20:

 Set all lights red

 Activate Road 2 green

 IF final second:

 Activate Road 2 amber

 ELSE IF seconds < 30:

 Set all lights red

 Activate Road 3 green

 IF final second:

 Activate Road 3 amber

 ELSE:

 Set all vehicle lights red

 Activate pedestrian green

 Calculate remaining countdown time

 Refresh seven-segment display

END

The flow of operation ensures that only one traffic direction is active at any point. The pedestrian crossing stage occurs only after all vehicle phases have completed.

Simulation and Testing

The circuit was tested using Proteus simulation software. Testing was performed by observing the behaviour of each individual subsystem before evaluating the complete system.

Traffic Light Testing

The first stage of testing involved verifying the operation of each traffic LED.

The following tests were performed:

- Confirming each output pin controlled the correct LED.
- Checking that each colour activated correctly.
- Confirming that conflicting green signals did not occur.
- Checking that amber the signals activated before changing to red.

The results showed that the traffic lights followed the programmed sequence correctly:

1. Road 1 green.
2. Road 1 amber.
3. Road 2 green.
4. Road 2 amber.
5. Road 3 green.
6. Road 3 amber.
7. Pedestrian crossing.

The "all red" safety stage successfully prevented multiple roads from being active simultaneously.

Pedestrian System Testing

The pedestrian crossing stage was tested by monitoring the final 10 seconds of the traffic light cycle. During this stage, the expected operation was for all vehicle lights to change to red, the pedestrian red light to turn off, and the pedestrian green light to activate. The simulation confirmed that the pedestrian crossing sequence operated correctly, with vehicle movement being prevented while pedestrians were given priority to cross. This demonstrates that the system meets the design requirement of prioritising pedestrian safety.

Seven-Segment Display Testing

The countdown display was tested by observing the displayed values throughout the traffic light sequence. The display correctly counted down from 10 seconds, decreased every second, and reset at the beginning of each new phase, confirming that the countdown logic operated correctly.

Some flickering was observed during simulation due to the multiplexing technique used to control the two-digit seven-segment display. Because both digits share the same segment connections, the microcontroller rapidly switches between them, and the refresh rate is limited by the execution speed of the software. This could be improved by using timer interrupts, increasing the refresh frequency, or using a dedicated display driver. Despite the flickering, the countdown remained fully functional throughout testing.

Conclusion

The project successfully achieved the design and simulation of a functional PIC16F877A-based T-junction traffic light controller. The completed system demonstrated automated control of three vehicle traffic signals, a pedestrian crossing phase, and a countdown display.

The use of a microcontroller provided a flexible and efficient solution compared with traditional hardware-based logic systems. The software approach allowed the traffic sequence to be easily modified while reducing the number of required hardware components.

The Proteus simulation successfully demonstrated the operation of the traffic sequence, including correct vehicle prioritisation and pedestrian safety control.

A limitation identified during testing was the flickering of the multiplexed two-digit display. Although this affected visual quality, the problem was understood and analysed as a consequence of software-based multiplexing. The system remained functional and met the main design objectives.

Overall, the project provided practical experience in embedded system design, microcontroller programming, circuit simulation, and engineering problem-solving.

Bibliography

- [1] C. K. WONG, "Optimization of a Signalized Cell Transmission Model for Network Signal Coordination," *Proceedings of the 3Rd International Conference on Material Engineering and Application (Icmea 2016)*, 2016, doi: 10.2991/icmea-16.2016.76.
- [2] "Traffic Light Controller - FBSwiki," *Caltech.edu*, 2026.
https://www.cds.caltech.edu/~murray/books/AM05/wiki/Traffic_light_controller.html
- [3] R. A. Retting, J. F. Chapline, and A. F. Williams, "Changes in Crash Risk Following Re-Timing of Traffic Signal Change Intervals," *Accident Analysis & Prevention*, vol. 34, no. 2, pp. 215–220, Mar. 2002, doi: 10.1016/s0001-4575(01)00016-1.
- [4] M. R. Gordon, "Review on Traffic Light Control Systems Based on Single-Chip Microcomputers and Smart Traffic Light Control Systems," *Applied and Computational Engineering*, vol. 8, no. 1, pp. 213–221, Aug. 2023, doi: 10.54254/2755-2721/8/20230131.
- [5] "Traffic Signs Manual," [Online]. Available:
<https://assets.publishing.service.gov.uk/media/5df0e29fed915d15f42c4820/dft-traffic-signs-manual-chapter-6.pdf>
- [6] J. Schneider and I. Smalley, "What Is a Microcontroller? | IBM," June 2024.
[Online]. Available: <https://www.ibm.com/think/topics/microcontroller>
- [7] GeeksforGeeks, "Difference Between Hardwired and Microprogrammed Control Unit | Set 2," Oct. 2018. [Online]. Available:
<https://www.geeksforgeeks.org/computer-organization-architecture/difference-between-hardwired-and-micro-programmed-control-unit-set-2/>
- [8] "Pic16f877a-I/P Technical Data." [Online]. Available:
<https://www.futurlec.com/Microchip/PIC16F877A.shtml>
- [9] "Product Documentation - NI," *Ni.com*, 2026.
<https://www.ni.com/docs/en-US/bundle/multisim/page/topics/seven-segment-display.html>

- [10] asathiskumar98-byte, "GitHub - asathiskumar98-byte/Multiplexing-2-Digit-7-Segment-Display-using-PIC16F877A: This Project Demonstrates Two-Digit Number Display (98) Using 7-Segment Multiplexing with the Pic16f877a Microcontroller. Multiplexing Allows Multiple 7-Segment Displays to Share the Same Segment Lines, Saving I/O Pins and Reducing Hardware Complexity.," 2025. [Online]. Available: <https://github.com/asathiskumar98-byte/Multiplexing-2-Digit-7-Segment-Display-using-PIC16F877A>
- [11] Two multiplexed 7-segment displays flickering, "Electrical Engineering Stack Exchange," *Electrical Engineering Stack Exchange*, 2017. <https://electronics.stackexchange.com/questions/287544/two-multiplexed-7-segment-displays-flickering>