

Ethan Longley - Electronics 1 Assignment

by Ethan Longley

Submission date: 15-Apr-2024 10:59PM (UTC+0100)

Submission ID: 229499036

File name: 50234_Ethan_Longley_Ethan_Longley_-_Electronics_1_Assignment_568563_1309720472.docx
(3.74M)

Word count: 3070

Character count: 15930



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Designing an Automated Traffic Light Controller

ELECTRONICS 1 (ACDA5002)

ETHAN LONGLEY
2318473

Contents

Abstract	2
Introduction	2
Research	3
Microcontrollers	3
CPU	3
Memory	3
I/O Peripherals	4
Bus	4
Architecture	4
Traffic Light Systems	5
Flowchart	7
Circuit Diagram	9
Program	10
Simulation	13
Results	14
Conclusion	14
References	15
Bibliography	15
Appendix	16

Abstract

Throughout this report a program was created to run a four-way traffic light junction with pedestrian crossings and a timer. The programme was made to use two separate codes with two separate PIC16F877A microcontrollers to run on the same system. The simulation ran well and proved that it was safe for vehicles and pedestrians whilst staying to the 50 second limit. Before completing the program, a few issues were encountered due to attempting to use only one microcontroller and a double seven segment display. A flowchart was created, showing the desired sequence for the traffic lights, this proved to be useful when working through the program as it helped remain on track. This type of project will come in handy with future projects in the workplace to help with improving efficiency and automation.

Introduction

The aim of this report is to design and create a traffic light system for a four-way junction, this will include pedestrian crossings and timers. To achieve this a code must be written for a microcontroller using MikroC PRO for PIC which will then be simulated using Proteus. Throughout the report, research will be carried out on Microcontrollers, looking into depth on how the microcontroller works and how it is used as well as brief research on traffic light systems to ensure that a correct system is created with safety being considered. Researching the traffic light systems will assist with creating a flowchart which will give structure to what the program is required to achieve. With the flowchart in place, the program will then be written, ensuring that it matches the flowchart, using MikroC PRO for PIC, it will debug the code and ensure it is clean and able to be used. The program will then be simulated and will show how the system works, using the simulation, the system will then be analysed and a conclusion will be constructed.

Research

Throughout the research of this report, microcontrollers are the main area which will be researched. For microcontrollers there are many variations, due to this, it is important to locate the best one for the required task.

Microcontrollers

A microcontroller is a small integrated circuit which can control specific operations within embedded systems, an example of this can be seen in figure 1 below. Microcontrollers are used often in everyday life, most commonly in your car or your home appliances and even used commonly in industry, for things such as robotic arms. A microcontroller is effectively a small PC which can control smaller features of larger appliances, this is by using internal components which all work in correlation to each other to achieve their desired outcome.

When a microcontroller is embedded within a system, it uses the data within it and data that it receives to decipher it to ensure that the appropriate data can then be applied and communicated to the system itself through the Input and Output Peripherals [1].



Figure 1 – PIC16F17126 Microcontroller [2].

Knowing now the purpose of a Microchip, the internal circuit of a microchip can be researched to determine how each internal component plays a role when it is communicating to the system. To begin, the CPU at the core of the microcontroller will be researched.

CPU

The Central Processing Unit has the main responsibility for executing instructions and performing calculations. It can obtain instructions from memory to decode them before sequentially executing them. The CPU manages the movement of data within the microcontroller and therefore can control the overall operations of the system. By interacting closely with components such as the internal memory and the input and output peripherals, it can access data and then move the data through the required output as well as receive the data from the inputs and then storing it within the memory [3].

Memory

Memory is another important component of the microcontroller; its purpose is to store data within it for the CPU to use. There are different types of memory, the main being RAM and ROM.

RAM

Random Access Memory, this is a volatile memory which can only be accessed when the system is working. If the system was to lose power, it would also lose everything stored within its RAM memory. Its main use is storing temporary data such as data input from the system.

ROM

Read Only Memory can't be edited during operations, the only time this can be edited is during the programming phase. This is a non-volatile memory which can be retained by the microcontroller even when power is lost. The ROM is where the program will be stored within the Microcontroller. [4]

I/O Peripherals

The purpose of the input and output peripherals are to transfer data between the system and the CPU. Each port can either be programmed to be an input or an output and therefore provides the user with flexibility.

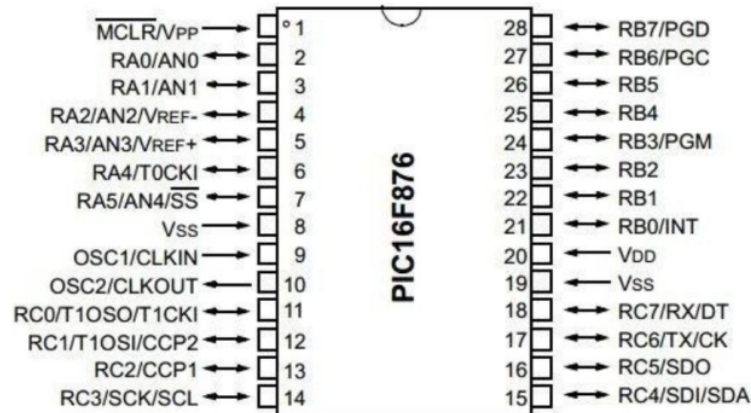


Figure 2 – PIC16F876 Microcontroller I/O Port Diagram [5].

Shown in Figure 2, there are multiple peripherals showing whether they are purely outputs, inputs, or bidirectional peripherals. These within their groups make up ports, a port could include different amounts of peripherals, for example, port A shown in figure 2 only has 6 peripherals when port B has 8. The double-sided arrows show bidirectional peripherals, this meaning that they can both output and input data, whereas the single sided arrows are only able to output or input data depending on their purpose. [6] [7]

Bus

The bus has the responsibility of connecting the different components within the microcontroller creating a link between them. Common types of buses include.

- Data Bus – Transfers data back and forth between the CPU and other components within the microcontroller. Transfers data such as Instructions.
- Address Bus – Specifies the address where the needed bit of data can be located within the memory and the I/O peripherals. This is unidirectional.
- Control Bus – Carries control signals from the CPU to the other components, this can also carry clock pulses. This is also unidirectional. [8]

Therefore each of these buses has different properties and can transfer different bits of data between the CPU and the Memory and even to locate the correct bits of data.

Architecture

Architecture refers to how the design and hardware components including the CPU and memory are organised within the Microcontroller, the architecture determines the microcontrollers capabilities and how it processes and performs instructions.

Harvard Architecture

Harvard architecture physically separates the memory storage from the program storage and the data. This means that there are separate bus lines which then means they can access both memory banks at the same time, overall increasing the speed of the microcontroller and completing instructions within one cycle. [3] [9]

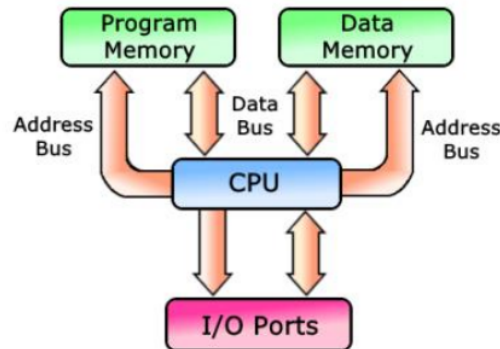


Figure 3 – Harvard Architecture Diagram [10].

Von Neumann Architecture

Von Neumann architecture is different to Harvard, due to it only having one memory for both data and program. Due to this there is only one bus required, therefore it requires two machine cycles to complete a task, this means it has a slower performance than the Harvard architecture. [11]

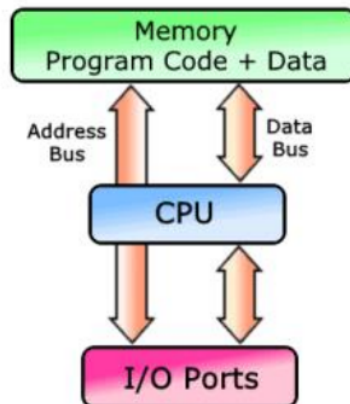


Figure 4 – Von Neumann Architecture Diagram [12].

Traffic Light Systems

Before planning the sequence, some research should be done on traffic light sequences themselves. In Wales and the UK, a traffic light sequence typically goes as follows below.

- Red
- Red & Amber
- Green
- Amber
- Red

Knowing this, it gives a better idea of the sequence of each of the traffic lights when writing the program for the four-way junction. [13] [14]

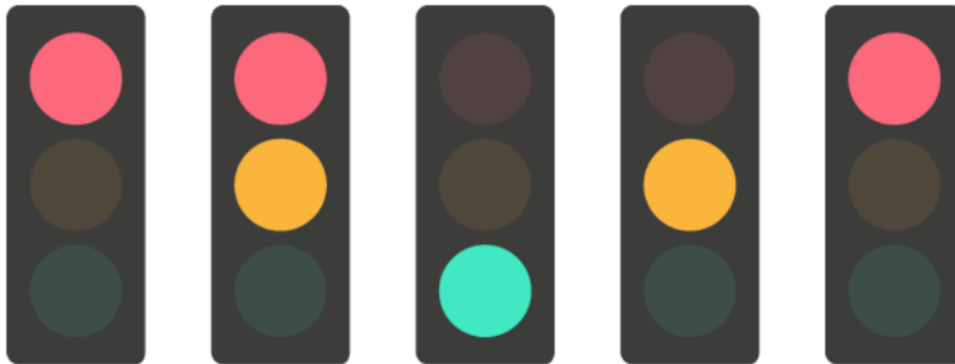


Figure 5 – Traffic Light Sequence [15].

Due to the brief of the assignment, the time must remain as 0-49 seconds, therefore the timings for each part of the sequence wont need to be considered.

Flowchart

Knowing the sequence of the traffic lights, it is now time to create a flowchart which shows the sequence of the four-way junction including the timer and pedestrian crossing. The time also must be within 0-49 seconds therefore it will give 0-9 seconds for the North traffic light, 10-19 seconds for the East traffic light, 20-29 seconds for the South traffic light, 30-39 seconds for the West traffic light and then finally 40-49 seconds for the Pedestrian Crossing. Flowchart 1 below shows the sequence for the four-way traffic light junction.



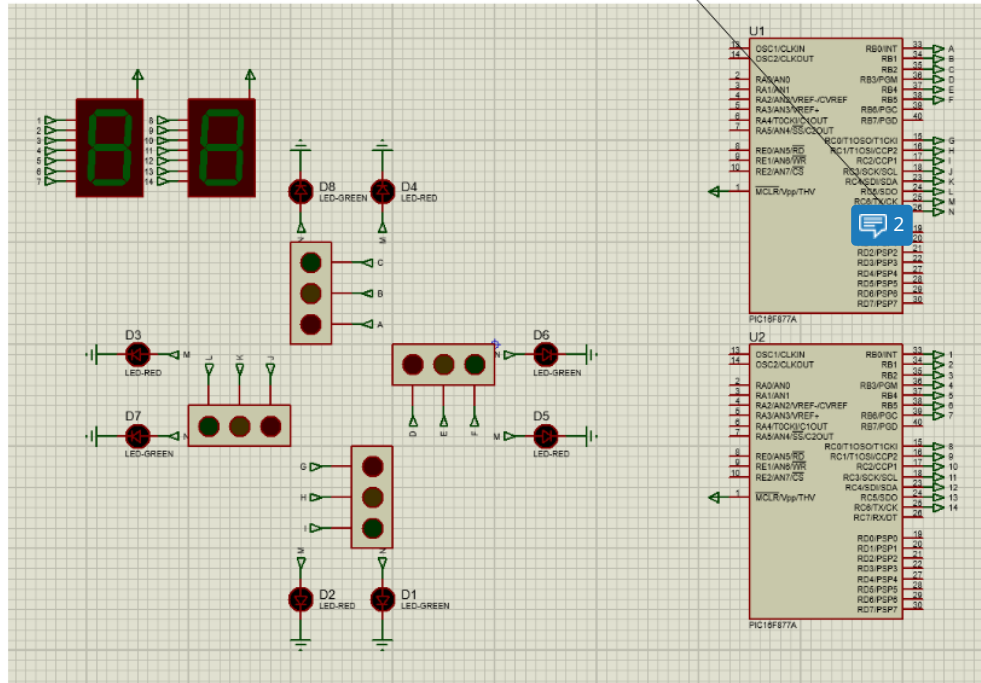
Flowchart 1 – Sequence of four-way traffic light junction

The purpose of the flowchart above is to show the entire sequence of the traffic light system, with this it will help visualise how the system will work and in what sequence the lights will change. The flowchart above uses downwards pointing triangles in green to represent the start of the sequence and then moves into different coloured squares to represent the colours of the light changing and the blue squares to show the corresponding time. At the bottom of the sequence is the upward pointing triangles to show the end of the sequence.

As shown in Flowchart 1, the sequence for the traffic light junction and the timer will start, it then shows the traffic lights and the pedestrian crossing branching off for each light. This then shows what state each light will be in at each time. On the right-hand side in the blue squares, the time is shown for example up until 1 second, all lights will be red, then up until 2.5 seconds the North light will turn Red and Amber and this will continue for 50 seconds until both sequences end. Although on the flowchart, it shows the timer ending at 50 seconds, on the simulation it will end at 49 seconds but will start at 0 instead of 1.

Circuit Diagram

With the flowchart created and a sequence put into place, it is now feasible to create the circuit diagram of the four-way traffic light junction. The purpose of this diagram is to simulate the program and check that it works with the created program installed within it.



Circuit Diagram 1 – Four Way Traffic Light Junction

Shown in Circuit Diagram 1 is the circuit which will be simulated using Proteus. The circuit includes 2 PIC16F877A Microcontrollers both ports B and C are being used on each of them with outputs coming from them to connect them to the external components. PIC16F877A Microcontrollers were chosen due to its ample 33 I/O pins, with this amount of Input and Output pins it would also allow for future improvements if needed as it would allow extra components to be included into the circuit. In the centre of the diagram is the four-way traffic light junction with North, East, South and West traffic lights including their related pedestrian crossings. The traffic lights have been assigned 3 output pins each on the U1 microcontroller to ensure that they can each change when needed. Due to each of the pedestrian crossings turning Green at the same time as all the traffic lights turn red, they have been given the same output pins in Port C so that they will each work at the same time and reduce the number of peripherals used on the Microcontroller. On the second microcontroller, U2, the two seven segment displays shown in the top left are connected to Ports B and C. The reasoning behind choosing to have two microcontrollers is to simplify the programming, rather than having one larger program which could become complex with all the aspects from the traffic lights and the seven segment display it could become confusing to understand. Since the PIC16F877A is cost effective, including a second one wouldn't be an issue.

3

Program

To begin, the program for the first microcontroller which is connected to the traffic lights. The beginning of the program included each of peripherals being stated, this was to ensure that the microcontroller knew which pin related to which part of the program.

```
• // North Traffic Lights
• sbit LED_NORTH_RED at RB0_bit;
• sbit LED_NORTH_YELLOW at RB1_bit;
• sbit LED_NORTH_GREEN at RB2_bit;
• // East Traffic Lights
• sbit LED_EAST_RED at RB3_bit;
• sbit LED_EAST_YELLOW at RB4_bit;
20 sbit LED_EAST_GREEN at RB5_bit;
• // South Traffic Lights
• sbit LED_SOUTH_RED at RC0_bit;
• sbit LED_SOUTH_YELLOW at RC1_bit;
• sbit LED_SOUTH_GREEN at RC2_bit;
• // West Traffic Lights
• sbit LED_WEST_RED at RC3_bit;
• sbit LED_WEST_YELLOW at RC4_bit;
• sbit LED_WEST_GREEN at RC5_bit;
• // Pedestrian Crossing
30 sbit LED_PED_RED at RC6_bit;
• sbit LED_PED_GREEN at RC7_bit;
```

Program 1 – Stating variables.

As is shown in program 1, “sbit” is used to declare the variable which is the LED and the states which bit it corresponds to.

```
• void main()
• {
• // North and East Traffic Lights
• TRISB = 0X00;
• PORTB = 0X00;
• // South, West Traffic Lights and Pedestrian Crossings
• TRISC = 0X00;
40 PORTC = 0X00;
• }
```

Program 2 – Stating variables.

“void main ()” executes the beginning of the program whereas the “TRISB” and “TRISC” is turning each of the port B and C pins into outputs, whereas the “PORTB” and “PORTC” statement turns each pin to logic low.

The next part of the program is the first phase of the light sequence. As seen in Program 3 below, the “do” function is the beginning of the do-while loop, the purpose of this is for the program to continuously run until a condition change to false.

```

do
{
    LED_NORTH_RED = 1;           //North Red ON
    LED_NORTH_YELLOW = 0;
    LED_NORTH_GREEN = 0;

    LED_EAST_RED = 1;           //East Red ON
    LED_EAST_YELLOW = 0;
    LED_EAST_GREEN = 0;

    LED_SOUTH_RED = 1;          //South Red ON
    LED_SOUTH_YELLOW = 0;
    LED_SOUTH_GREEN = 0;

    LED_WEST_RED = 1;           //West Red ON
    LED_WEST_YELLOW = 0;
    LED_WEST_GREEN = 0;

    LED_PED_RED = 1;            //Pedestrian Red ON
    LED_PED_GREEN = 0;

    Delay_ms(1000);
}

```

Programme 3 – Phase 1 of the Do-While loop.

Where the program says “LED_NORTH_RED = 1;” its stating that the corresponding pin turns to logic high and therefore will make the LED turn on. “Delay_ms(1000);” is stating the delay in time before moving on to the next phase of the program.

This program will continue the same in each phase with changes to the logic high command depending on which LED needs to be turned on. The delay will also change depending on how long the LED needs to be turned on for.

Moving on to the second microcontroller, U2, this will have the program for the timer. The first part of the program is “int i;”, this part of the program is a variable which is used as a loop counter for when the timer is counting. Program 4 also shows “unsigned int display [10] = {0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90};” the purpose of this is that the hexadecimal values represent the patterns needed on the seven-segment display to display the correct number on the seven-segment display.

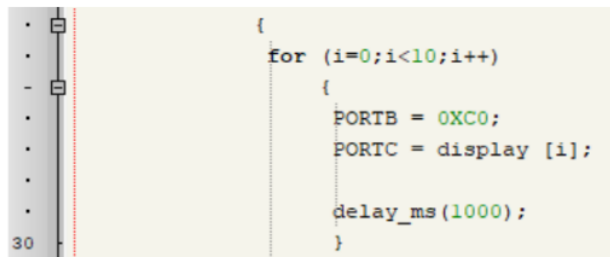
```

10
· int i;
· unsigned int display [10] = {0XC0, 0XF9, 0XA4, 0XB0, 0X99, 0X92, 0X82, 0XF8, 0X80, 0X90};

```

Programme 4 – Initialization of Loop Counter and Seven-Segment Display Patterns

After what is shown in program 4, the same as program 2 is included into the program. After that is the first phase of the timer program which counts from 0-9seconds. In program 5 the purpose of that part of the program is to count from 0 and then to count up to 9. In the second phase, the counter will go from 10– 19. With the loop function at the end, once the counter reaches 49 it will restart with the rest of the program in microcontroller U1.



```
{
    for (i=0; i<10; i++)
    {
        PORTB = 0XC0;
        PORTC = display[i];
        delay_ms(1000);
    }
}
```

Programme 5 – Sequential Display of Seven-Segment Digits

These two codes will be used to create the sequence shown in Flowchart 1 above.

Simulation

With the programs written, they can now be uploaded into the microcontroller as shown below in figure 6.

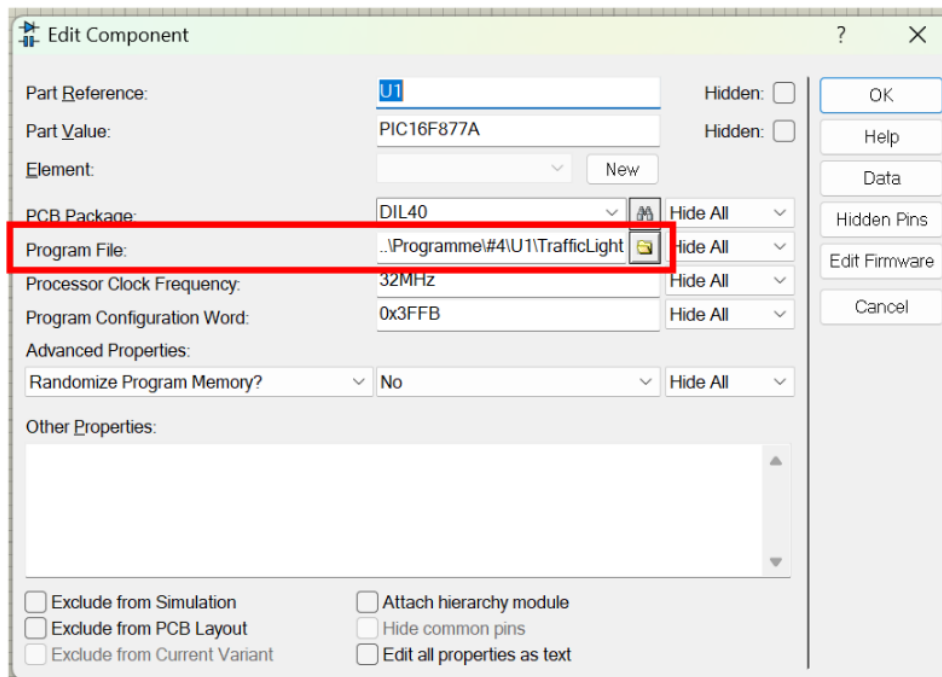


Figure 6 – Uploading Programme File to Microcontroller

The program when simulated runs well and keeps time accurately moving between phase in the correct sequence as was shown in the flowchart. Figure 7 below shows the simulation working. To see the full simulation, please see the video included in the Assignment File.

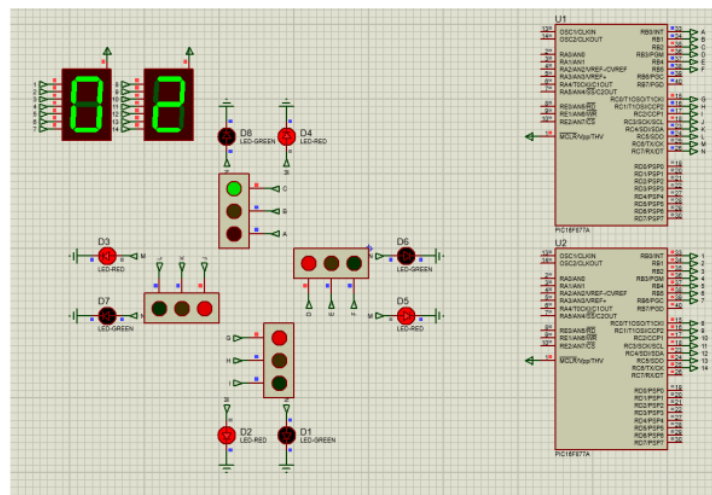


Figure 7 – Circuit Simulation

Results

Upon completing the program upload and carrying out the simulation, the results were as expected. The sequence of the traffic lights and pedestrian signals flawlessly adhered to the sequence, ensuring a secure and efficient traffic flow. Notably, the sequence was very safe for both vehicles and pedestrians due to there being no two green lights on at the same time.

Furthermore, the timing exhibited by the seven-segment display was very good, mirroring the traffic lights' operation with good accuracy. This synchronization played a pivotal role in ensuring smooth transitions at designated intervals.

A noteworthy aspect was the seamless coordination observed between the two microcontrollers, ensuring that they both remained in sync and kept their timings throughout.

In summary, the executed program proved that the program worked effectively and met each objective given within the brief.

Conclusion

Throughout the report, it has been evident that microcontrollers can be used within a wide variety of applications and have proved that they are an important component in today's technology. Throughout the report, microcontrollers were researched to ensure a good understanding of how they work was obtained prior to creating the simulation.

During the creation of the circuit, the first attempt was done using a double seven segment display rather than two singles. This was desired as it would have eliminated the need to use two microcontrollers. Due to the code not working using this method, it was decided that two microcontrollers would be used. This was a suitable idea as it was mentioned during the research section that two separate microcontrollers can be used for the same system.

After altering the first idea and using two single seven segment displays the program worked well in the simulation and as mentioned in the results, it met each of the requirements from the brief.

If the project was to be done again, it would be beneficial for cost saving to use a double seven segment display rather than the two singles as this would save a small amount on microcontrollers. As well as this, to enhance the program further, placing a function in the code so that when a button by the pedestrian crossings is pressed at any point, the cycle on the traffic light it is currently on would finish and then the pedestrian crossings will turn green for 10 seconds before continuing following the phases for the traffic lights. This way, it would become more realistic to real world traffic systems.

To conclude, the use of microcontrollers is used in a broad variety of applications and is very important to electronics in industry and everyday technology. Knowing about the uses and workings of microcontrollers is very important for electrical engineers as using them can prove to be very beneficial for a multitude of applications within the workplace, whether its automation or improving efficiency.



References

Bibliography

- [1] B. Lutkevich, "What is a Microcontroller and How Does it Work?," *IoT Agenda*, Nov. 2019. <https://www.techtarget.com/iotagenda/definition/microcontroller>
- [2] RS, *Microchip PIC16F17126-I/ST*. 2022.
- [3] S. Yousif, "Microcontroller Design," 2022. Available: https://www.uomus.edu.iq/img/lectures21/MUCLecture_2022_41025606.pdf
- [4] Myke Predko, *Programming and Customizing the PIC Microcontroller*. McGraw Hill Professional, 2007, pp. 32–33.
- [5] S. Masud Rana, *PIC16F876 Microcontroller*. 2013.
- [6] Myke Predko, *Programming and Customizing the PIC Microcontroller*. McGraw Hill Professional, 2007, pp. 29–31.
- [7] TOSHIBA, "I/O (Input, Output) | Toshiba Electronic Devices & Storage Corporation | Europe(EMEA)," *toshiba.semicon-storage.com*, 2023. <https://toshiba.semicon-storage.com/eu/semiconductor/knowledge/e-learning/micro-intro/chapter2/i-o-input-output.html#:~:text=I%2FO%20is%20a%20function>
- [8] BBC, "Architecture - Eduqas - Revision 2 - GCSE Computer Science - BBC Bitesize," *BBC Bitesize*, 2019. <https://www.bbc.co.uk/bitesize/guides/zhppfcw/revision/2>
- [9] A. Suttle, "Harvard Architecture," 2017. Available: https://tdck.weebly.com/uploads/7/7/0/5/77052163/03_-_harvard_architecture_comparison.pdf
- [10] S. Yousif, *Harvard Architecture*. 2022.
- [11] BBC, "Von Neumann architecture - Architecture - Eduqas - GCSE Computer Science Revision - Eduqas," *BBC Bitesize*, 2019. <https://www.bbc.co.uk/bitesize/guides/zhppfcw/revision/3>
- [12] S. Yousif, *Von Neumann Architecture Diagram*. 2022.
- [13] BSM, "Guide to traffic lights for beginners," *www.bsm.co.uk*, 2016. <https://www.bsm.co.uk/traffic-lights>
- [14] Theory test easy, "Traffic lights sequence theory test Uk," *www.youtube.com*, 2022. <https://www.youtube.com/watch?v=l6Eh8ilzkwc> (accessed Apr. 15, 2024).
- [15] S. Jones, *Traffic light sequence*. 2023.

Appendix

1. Traffic Light Simulation – Proteus Project
2. Traffic Lights (V4.2.U1) – PDF
3. Seven Segment Display (V4.1.U2) – PDF
4. Simulation Video – MP4 File

Ethan Longley - Electronics 1 Assignment

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

67/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5



Comment 1 | Knowledge & Understanding (20)

It is good that most of the basic parts are covered, but it still needs more in-depth analysis.

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10



Comment 2 | Practical Application Sk(30)

Great work on the designing a clear and precise circuit.



Comment 3 | Cognitive & Intellectual (20)

Needs further elaboration on the microcontroller selection criteria.



Comment 4 | **Transferable Skills (20)**

Good flow of structure is maintained through out the report

GRADING FORM: EC1

ETHAN LONGLEY

66.94

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)



Limited identification of relevant concepts, with several omissions.

Limited explanation of significance, with several key points missing. 2.85

Good background research, using a variety of quality sources. Demonstrates a solid understanding of the topic with minor gaps.

6.28

2.19

Reasonable variety of sources. 1.24

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 limited knowledge of traffic light controller products and their market requirements. Shows limited understanding, with several gaps.

5

Reflects limited understanding of the underlying principles in analysing different traffic light controllers. 5.9

10.9

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 an adequate review of the microcontroller selection, with basic analysis of supporting modules and ICs. Cost evaluation is present but may lack detail. 4.83

Proposes creative and effective solutions that enhance the project, with minor areas for improvement. 4.9

Shows a good understanding of programming concepts and logical flow, with minor areas for

14.26

improvement. Pseudocode is mostly easy to follow and communicates the system's operation effectively. 4.5

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. 7.5
Demonstrates thorough testing and debugging processes, identifying and resolving most issues effectively. 14

21.5

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)*



The report is well-organized, with a logical flow of ideas that is easy to follow. Most chapters and sub-chapters are appropriately structured and connected. 5
They generally aid understanding but with some gaps. 4
The report is professional and well-presented, with minor areas for improvement. 5

14