Microcontroller-based Traffic Light Controller

Submitted 26/04/13 in partial fulfilment of the requirements for the Embedded Systems CT5003

Olivier Zoude

10034346 / org0020@my.londonmet.ac.uk

Faculty of Life Sciences & Computing
Communications Technology
London Metropolitan University

I hereby declare that this report is all my own work, except as indicated in the text:

Signature: ..............................................

Date: May 2013
CASE STUDY REPORT:

Microcontroller-based Traffic Light Controller

CONTENTS

3. ABSTRACT .................................................................Page 3

3. INTRODUCTION ..........................................................Page 3

3. SECTION 1: Preparatory Tasks .........................................Page 4

4. SECTION 2: Case Study ..................................................Page 14

a) HARDWARE DEVELOPMENT .........................................Page 15
   • TASK 1: pin configuration of main components ..Page15
   • TASK 2: Board layout and Schematic ......................Page18

b) SOFTWARE DEVELOPMENT ..........................................Page 22
   • FLOW CHART ..............................................................Page 22
   • C-PROGRAM ..............................................................Page 23
   • EXPLANATION ............................................................Page 27

c) TESTING .........................................................................Page 30

5. CONCLUSION ..............................................................Page 35

6. REFERENCES .................................................................Page 35
1. ABSTRACT

Traffic junctions are generally controlled by traffic lights controller. The function of traffic lights requires sophisticated control and accurate coordination to ensure that traffic moves as safely and smoothly as possible. Our case study is developed to meet the requirements of solid state traffic light controller by adopting microcontroller as the main controlling element, and led’s as the indication of light. A microcontroller is interfaced to led’s provide for centralized control of the traffic signals. Microcontroller is programmed in such a way to adjust their timing and phasing to meet changing traffic conditions.

2. INTRODUCTION

Traffic lights are signaling devices positioned at road intersections, pedestrian crossings and other traffic locations to control competing flows of traffic. They assign the right of way to road users by the use of lights in standard colors (red - amber/yellow - green), using a universal color code. Design of an effective traffic light system in a modern city is a task of traffic engineers, and is an important undertaking. It has therefore always being a real concern for traffic authorities in every country. Ever since Roman times, society has tried to control traffic. Even the fabled Roman road system created a conflict between pedestrian and equine travelers until the first traffic signal project was installed near Westminster Abbey in London, England in 1868. But J. P. Knight, project was having a safety issue and its implementation caused the death of a police officer. In our case study, Microcontroller-based Traffic Light Controller, composed respectively of two sections (Preparatory Tasks and Case Study) we will use the appropriate tools developed in the first section to implement a program in the second section, controlling the safety aspect of the traffic light board function, using C-program, compile it and debug it with compiler software (Ride) and then embed the program into the traffic light circuit board using tester software (Win loader) to effectively see the outcome of the C-program embedded into the microcontroller circuit board using a 7-segment display. Finally we will build the circuit board, base on the case study guideline, using appropriate software (Eagle) to emphasize in details on the components used to build the traffic light circuit board.

3. SECTION 1: Preparatory Tasks
The 7-Segment Display Module – 4-Digits

This module uses the M5450 Driver. The physical layout of this module is shown below:

The pin configuration of the RJ45 connector is shown above and in the posters in the laboratory.

One of the microcontroller ports is connected to the 7-segment module via an interface board as shown below:
Task 1:
Using the data for the interfacing cards and the 7-segment display module (Displayed on the posters in the lab). We determine which port is the 7-Segment has been connected to via the RJ45 cable.

We determine which pin of the M5450 is connected to pin 1 of the RJ45 in the 7-segment module. We write these in the table in the row for pin 1. We continue until all the 8-pins have been checked.

Note that some of the pins may have no connection in one or either side. In this case simply enter ‘NC’ for ‘No Connection’.

Task 2:
Task 2.1: From table 1, we enter the information in the definition part of the program

Table 1

<table>
<thead>
<tr>
<th>RJ45 Pins</th>
<th>RJ45 in the Interface Card</th>
<th>RJ45 in the 7-Segment Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>2</td>
<td>P1.3</td>
<td>P1.2</td>
</tr>
<tr>
<td>3</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>4</td>
<td>P1.1</td>
<td>P1.4</td>
</tr>
<tr>
<td>5</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>6</td>
<td>P1.0</td>
<td>P1.6</td>
</tr>
</tbody>
</table>
**Task 2.2:** The function `<void M5450_Delay (void)>` is a very short delay of about 10 microseconds. We complete the function by writing the appropriate code. This is just an approximate.

The skeleton of a typical C-program for microcontroller-based applications is shown below.

```c
/* *******************************************************
*               A Program for the 7-Segment Display Module *
* *******************************************************
* Developed By:  Olivier Zoude *
* *******************************************************
* Date: 20/12/12 *
* Module:  CT5003 - Microprocessor & Embedded Systems *
* *******************************************************

#include <reg51.h>
#include <string.h>

/* Declaration of the Data and Clock Bits for the M5450 */
sbit Data = P1^4;
sbit Clock = P1^6;

/* A very short Delay of about 10 microseconds */

void M5450_Delay(void)
{
    int i;
    for (i=0; i<10000; i++)
    {
        ;
    }
}

/* The Main Program */

void main (void)
{
    /* Declaration of all local variables in the main part of the program */
    M5450_Start();

    for(;;)
    {
        ;
    }
} /* End of the main program */
```

**Task 2.3:** Questions

**Task 2.3(a):** What is the purpose of `#include <reg51.h>`?
Answer: It is used to include a header-file for the registers (SFRs) in the source file.

Task 2.3(b): What is the purpose of < for(; ;) > in the main program?

Answer: The statements in the for loop repeat continuously for a specific number of times.

Task 3:
To display a number or a letter, the program requires the 7-segment code for each digit from 0 to 9. Using figure 4 and the table provided, determine the 7-segment codes for digits 0 to 9 and complete the table.

![7-segment display diagram]

Task 3.1:
There are some limited numbers of characters that can also be displayed (e.g. H). We write down the codes for all alphabetical letters that we think may be displayed on a 7-segment display.

<table>
<thead>
<tr>
<th>Dec</th>
<th>dp</th>
<th>g</th>
<th>f</th>
<th>e</th>
<th>d</th>
<th>c</th>
<th>b</th>
<th>a</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0x3F</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0x06</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0x5B</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0x4F</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0x66</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0x6D</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0x7D</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0x07</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0x7F</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0x6F</td>
</tr>
</tbody>
</table>
Task 3.2:
The above 7-segment is assumed to be of Common-Anode type. The other type is Common-Cathode.

What is the difference between the two types?

- **In common cathode type display**, all the cathodes of the segments are tied together and connected to ground. The supply will be given to the required segment from the decoder or driver.

- **In common anode type display**, the anodes of all the segments are tied together and connected to supply and the required segments will be connected to ground from the decoder or driver.

Is M5450 suitable for both types?

M5450 is not suitable for both type displays because when the power supply is ON all LEDs must be ON then the required segment will be connected to ground (OFF) from the decoder or driver.

Task 4: Display ‘1234’ on the module.
This program displays ‘1234’. The skeleton of a main program may be written as follows:

```c
/
/* ****************************************** */
/* A Program for the 7-Segment Display Module */
/* ****************************************** */
/* Developed By: Olivier Zoude */
```
#include <reg51.h>
#include <string.h>

/* Declaration of the Data and Clock Bits for the M5450 */
sbit Data = P1^4;
sbit Clock = P1^6;

/* A very short Delay of about 10 microseconds */
void M5450_Delay(void)
{
    int i;

    for (i=0; i<10000; i++)
    {
    }
}

void M5450_Pulse(void)
{
    /* This function sends a single clock pulse to the M5450 Driver */

    Clock=0;
    Clock=1;
    M5450_Delay(); /* A short Delay */
    Clock=0;
}

void M5450_Send(unsigned char D)
{
    /* This function sends the 8-bit (byte) D to the M5450 in serial form */
    /* with the least significant bit first */

    unsigned char k;
    for (k=0; k<8; k++)
    {
        B=D&0x01;
        if(B==0)
        Data=0;
        else
        Data = 1;
        M5450_Pulse();
        D=D>>1;
    }

    /* The following function sends a start pulse to the M5450 */
    void M5450_Start(void)
    {
        Data = 0;
        M5450_Pulse();
        Data = 1;
        M5450_Pulse();
    }

    // The Main Program */
    void main (void)
    {
        /* Declaration of all local variables in the main part of the program */
        /* First Send a Start Pulse to the M5450 */
        M5450_Start();
    }
for(;;)
{
  M5450_Send(0x66);
  M5450_Send(0x4F);
  M5450_Send(0x5B);
  M5450_Send(0x06);

  /* Send a further Three Clock Pulses to Complete the 35 Pulses */
  M5450_Pulse();
  M5450_Pulse();
  M5450_Pulse();
}

} /* End of the main program */

Task 4 - Figure 1

Task 5: A Simple Counter
This program is a simple counter. The display should start with 0000 and count at a rate of every second until it reaches to 9999 where it rolls over to 0000 and repeats the counting sequence again. The skeleton for this task is shown below:

/************************************************************************
*               A Program for the 7-Segment Display Module *
*************************************************************************
* Developed By:  Olivier Zoude                                       *
*                                                                       *
* Date:  20/12/12                                                     *
*                                                                       *
* Module:  CT5003 – Microprocessor & Embedded Systems                 *
*************************************************************************/

#include <reg51.h>
#include <string.h>
/ * Declaration of the Data and Clock Bits for the M5450 */
sbit Data = P1^4 ;
sbit Clock = P1^6 ;

void Delay(void);
void M5450_Start(void);
void M5450_Pulse(void);
void M5450_Delay(void);
void M5450_Send (unsigned char D);
void M5450_Display(unsigned int D);

void main (void)
{
    /* Declaration of all local variables in the main part of the program */

    /* First Send a Start Pulse to the M5450 */
M5450_Start();

    for(;;)
    {
        unsigned int k;
        for ( k = 0; k<9999; k++)
        {
            M5450_Display(k);
            Delay();  /* A 1 Second Delay */
        }
    }

    /* A very short Delay of about 10 microseconds */
    /* This doesn’t have to be accurate as explained in the lecture classes */
    void M5450_Delay(void)
    {
        int i;
        for (i=0; i<10000; i++)
        {
            ;
        }
    }
    void Delay(void)
    {
        int j,k;
        for (j=0; j<50; j++)
        {
            for(k=0; k<10000; k++)
            {
                ;
            }
        }

    /* This function sends a single clock pulse to the M5450 Driver */
    void M5450_Pulse(void)
    {
        Clock=0;
        Clock=1;
        M5450_Delay(); /* A short Delay */
        Clock=0;
    }
/* This function sends the 8-bit (byte) D to the M5450 in serial form */
/* with the least significant bit first */
void M5450_Send (unsigned char D)
{
    unsigned char k;
    for (k=0; k<8; k++)
    {
        B=D&0x01;
        if(B==0)
        
            Data=0;
        else

            Data = 1;
        M5450_Pulse();
        D=D>>1;
    }
}

/* The following function sends a start pulse to the M5450 */

void M5450_Start(void)
{
    Data = 0;
    M5450_Pulse();
    Data = 1;
    M5450_Pulse();
}

/* this function to display 7-segment display. D is a 4 digit number from 0 to 9999 */

void M5450_Display(unsigned int D)
{
    unsigned int D3,D2,D1,D0,K,J,T0,T1,T2,T3;
    unsigned int Seg[] = {0x3F, 0x06 , 0x5B , 0x4F, 0x66, 0x6D, 0x7D, 0x07,
                          0x7F,0x6F};
    K=(int)(D/10);
    D0=D-K*10;
    J=(int) (K/10);
    D1=K-J*10;
    D3=(int) (J/10);
    D2=J-D3*10;
    /* the 4 digits are separated and assigned to D0, D1, D2, and D3 */
    T0=Seg[D0];
    T1=Seg[D1];
    T2=Seg[D2];
    T3=Seg[D3];
    M5450_Start();
    M5450_Send(T0);
    M5450_Send(T1);
    M5450_Send(T2);
    M5450_Send(T3);
    M5450_Pulse();
    M5450_Pulse();
    M5450_Pulse();
Task5 - Figure1

/* End of the main program */

Task5 – Figure2
4. SECTION 2: Case Study

We are required to design a microcontroller-based Traffic Light Controller. The constraints (what we must use...) are as follows:

The unit:
1. We must use an 8051 microcontroller
2. We must use the available hardware modules: The 7-segment display and the traffic light modules.
3. All programs must be properly documented and have a corresponding flowchart (where appropriate).

Our Initial Task
Our first task is to decide how the traffic light should operate in order to control the traffic in all directions.

General points
As already mentioned, our organization in the lab is an important part of this case study. Before we start the work, we MUST ensure that we are clear on how to use:

- Ride Development System to Enter, Compile, Debug, Simulate, Single Step a program written in C.

Figure 7. A simplified diagram of the traffic light junction
• The down loader to download our HEX programs onto the Microcontroller Memory.
• We read the relevant lecture / tutorials from the lecture classes as well as the web site before attending the laboratory.

a) HARDWARE DEVELOPMENT

This phase of the project concerns: the design of the Microcontroller board. It involves designing the hardware on paper and design a PCB using the Eagle software.

**Task 1:** Pin configuration of main components of developed circuit board.

- Pin configuration of the microcontroller device
- The microcontroller Board other components

- To work with a 9v battery. This requires a small regulator

- An External SRAM (Static RAM) for external Data Storage pin configuration

- 9 pin D-con Serial Port using RS232 Protocol
· Expansion port on the edge of the board for user’s applications

· Atmel’s 2-Wire Serial EEPROM (AT24Cxx Series)

· RS232 Transceiver

· A NAND gate

· An inverter
Task 2: Schematic and board layout (screen print)

- SCHEMATIC

CT5003: Microprocessors & Embedded Systems Case Study report / Microcontroller-based Traffic Light Controller / By Olivier Zoude / ID: 10034346
CT5003: Microprocessors & Embedded Systems Case Study report / Microcontroller-based Traffic Light Controller / By Olivier Zoude / ID: 10034346

Page 20 of 35
- BOARD LAYOUT (of the designed PCB)
**Task 3:** See logbook for drawing of entire circuit diagram for the entire board.

**b) SOFTWARE DEVELOPMENT**

- **FLOW CHART**

```
Main Start

K=10

K=K-1

Yes

is K>0?

No

Move to next traffic light

Display Start

Red or Green light on each road sides with yellow light intersecting

No

Any walk request button pressed?

Yes

Red light on both road sides

K=15

K=K-1

Amber light with walking permission

is K>0?

No

Standard traffic light

Set next countdown

Return
```

CT5003: Microprocessors & Embedded Systems Case Study report / Microcontroller-based Traffic Light Controller / By Olivier Zoude / ID: 10034346

Page 22 of 35
• C-PROGRAM (Software)

/**************************************************************************
* A Program for the 7-Segment Display Module
**************************************************************************
* Developed By: Olivier Zoude
**************************************************************************
* Date: 20/12/12
* Module: CT5003 – Microprocessor & Embedded Systems
**************************************************************************/

#include <reg51.h>
#include <string.h>

sbit Data = P1^4;
sbit Clock = P1^6;
sbit Pedestrian_Buzzer = P3^2;

void M5450_Switch(void);
void M5450_Start(void);
void M5450_Pulse(void);
void M5450_Delay(void);
void Delay1 (void);
void Delay2 (void);
void M5450_Send(unsigned char D);
void M5450_Display(unsigned int D);

/* This main function regulates all traffic lights */

void main (void)
{

    // RS-NS = Road-Side North-South
    // RS-WE = Road-Side West-East

    unsigned int K;
    P3=0;

    for (;;) {

        P3=0x84; /* Pin P3^7 and P3^2 are ON which is green from RS-NS and
                red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x46;/* Pin P3^6, P3^2 and P3^1 are ON which is yellow from
RS-NS and red, yellow from RS-WE */

        Delay1();/* delay of 3 microsecond */

        M5450_Display(0);

        P3=0x21; /* Pin P3^5 and P3^0 are ON which is red from RS-NS and
green from RS-WE*/

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x04; /* Pin P3^4 is ON which is green from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        // RS-NS and red, yellow from RS-WE */

        Delay1();/* delay of 3 microsecond */

        M5450_Display(0);

        P3=0x62; /* Pin P3^1 is ON which is red from RS-NS and
green from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x24; /* Pin P3^3 is ON which is red from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x30; /* Pin P3^4 is ON which is red from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x25; /* Pin P3^5 is ON which is red from RS-NS and
green from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x02; /* Pin P3^2 is ON which is red from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x12; /* Pin P3^1 is ON which is red from RS-NS and
green from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x00; /* Pin P3^0 is ON which is red from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x20; /* Pin P3^3 is ON which is red from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

        P3=0x06; /* Pin P3^4 is ON which is red from RS-NS and
red from RS-WE */

        for (K=10; K>0; K--)
        
            M5450_Display(K);/* 7-segment displays count down from 10
to 0*/

        Delay2();/* delay of 1.5 microsecond */

        M5450_Display(0);

    }

}
{ M5450_Display(K); /* 7-segment displays count down from 10 to 0 */
  Delay2(); /* delay of 1.5 microsecond */
}

M5450_Display(0);
P3=0x62; /* Pin P3^6, P3^5 and P3^1 are ON which is yellow, red from RS-NS and yellow from RS-WE */
{ Delay1(); /* delay of 3 microsecond */
}

M5450_Display(0);
P3=0x84; /* Pin P3^7 and P3^2 are ON which is green from RS-NS and red from RS-WE */
for (K=10; K>0; K--)
{ M5450_Display(K); /* 7-segment displays count down from 10 to 0 */
  Delay2(); /* delay of 1.5 microsecond */
}

M5450_Display(0);
P3=0x64; /* Pin P3^6, P3^5 and P3^2 are ON which is red from RS-NS and red, yellow from RS-WE */
{ Delay1(); /* delay of 3 microsecond */
}

M5450_Delay(void)
{
  int i;
  for (i=0; i<10000; i++)
  {
  }
}

/* This function would clear all green and yellow LEDs */

void M5450_Switch(void)
{
  Pedestrian_Buzzer = 1; /* pedestrian buzzer is pressed, amber light is ON */
P3 = 0x3C; /* red light on both roads RS-NS and RS-WE are ON for 15 seconds when pedestrian press buzzer and its amber light is ON */
}

/* This function sends a very short delay of 10 microseconds (us) */

void M5450_Delay(void)
{
  int i;
  for (i=0; i<10000; i++)
  {
void Delay1 (void)
{
    int j,k;
    for (j=0; j<60; j++)
    {
        for(k=0; k<20000; k++)
        {
        ;
        }
    }
}

void Delay2 (void)
{
    int j,k;
    for (j=0; j<30; j++)
    {
        for(k=0; k<20000; k++)
        {
        ;
        }
    }
}

void M5450_Pulse(void)
{
    Clock=0;
    Clock=1;
    M5450_Delay();  /*delay of 10 microseconds */
    Clock=0;
}

void M5450_Send(unsigned char D)
{
    unsigned char B,i;
    for (i=0; i<8; i++)
    {
        B=D&0x01;
        if (B == 0) Data=0;
        else Data=1;
        M5450_Pulse();
        D=D>>1;
    }
}

void M5450_Start(void)
{  
    Data =0 ;  
    M5450_Pulse();  
    Data =1 ;  
    M5450_Pulse();  
}

/* this function to display 7-segment display. D is a 4 digit number from 0000 to 9999 */

void M5450_Display(unsigned int D)
{
    unsigned char D3,D2,D1,D0,K,J,T0,T1,T2,T3;
    unsigned char Seg[ ] = {0x3F, 0x06 , 0x5B , 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F,0x6F};

    /* first we separate the 4 digits and assign them to D0,D1,D2, and D3 */
    K=(int)(D/10);
    D0=D-K*10;
    J=(int) (K/10);
    D1=K-J*10;
    D3=(int) (J/10);
    D2=J-D3*10;

    /* Then we convert each digit to its seven-segment code using the lookup table method */
    T0=Seg[D0];
    T1=Seg[D1];
    T2=Seg[D2];
    T3=Seg[D3];
    M5450_Start();
    M5450_Send(T0);
    M5450_Send(T1);
    M5450_Send(T2);
    M5450_Send(T3);
    M5450_Pulse();
    M5450_Pulse();
    M5450_Pulse();
}

}/* End of program */
EXPLANATION (Using program description language: PDL)

1 - Main program

Start
Set port 3 (P3) to 0

Do forever

Set P3 to 0x84

Decrement count (converted data) from 10 to 0
Call Function M5450_Display to display count
Call Function Delay2 to delay for 1.5 microseconds

Set function M5450_Display to 0
Set P3 to 0x46

Call Function delay1 to delay P3 for 3 microseconds

Set function M5450_Display to 0
Set P3 to 0x21

Decrement count (converted data) from 10 to 0
Call Function M5450_Display to display count
Call Function Delay2 to delay P3 for 1.5 microseconds

Set function M5450_Display to 0
Set P3 to 0x62

Call Function Delay1 to delay P3 for 3 microseconds

Set function M5450_Display to 0
Set P3 to 0x84

Decrement count (converted data) from 10 to 0
Call Function M5450_Display to display count
Call Function Delay2 to delay P3 for 1.5 microseconds

Set function M5450_Display to 0
Set P3 to 0x44

Call Function Delay1 to delay P3 for 3 microseconds

Set function M5450_Display to 0
Call Function M5450_Switch
Decrement count (converted data) from 15 to 0
Call Function M5450_Display to display count
Call Function Delay2 to delay for P3 for 1.5 microseconds

Set function M5450_Display to 0
Set P3 to 0x64

Call Function Delay1 to delay P3 for 3 microseconds

EndDo

End

2 – Function M5450_Display

Input: Digit value
Output: None

Start

Convert data into 4 decimal digits

Set T0 to decimal digit D0
Set T1 to decimal digit D1
Set T2 to decimal digit D2
Set T3 to decimal digit D3
Call Function M5450_Start to set data

Call Function M5450_Send to send T0
Call Function M5450_Send to send T1
Call Function M5450_Send to send T2
Call Function M5450_Send to send T3

Call Function M5450_Pulse to set clock
Call Function M5450_Pulse to set clock
Call Function M5450_Pulse to set clock

End

3 – Function M5450_Start

Input: None
Output: None

Start
Set data to 0
Call Function M5450_Pulse to set clock
Set data to 1
Call Function M5450_Pulse to set clock
End

4 – Function M5450_Pulse

Input: None
Output: None

Start

Set clock to 0
Set clock to 1
Call Function M5450_Delay to delay clock for 10 microseconds
Set clock to 1
End

5 – Function M5450_Delay

Input: None
Output: None

Start

Delay of 10 microseconds
End

6 – Function M5450_Send

Input: Digit value
Output: None

Start

Get bit map of the digit to be sent
Increment the bit to 8
Get top bit of the bit pattern

If top bit is set to 0 then
Send 0 to data input
Else
Send 1 to data input
EndIf
Call function M5450_Pulse to set clock
Shift bit map right by 1 bit

End

7 – Function Delay2

Input: None
Output: None

Start

Delay of 1.5 microseconds

End

8 – Function Delay1

Input: None
Output: None

Start

Delay of 3 microseconds

End

9 – Function M5450_Switch

Input: None
Output: None

Start

Set Pedestrian_Buzzer port pin P3^2 to 1
Set Port 3 (P3) to 0x3C

End

c) TESTING

From the Lab, the C program above was successfully tested and the results are as below for evidence.
NB: LEDs green, yellow and red are for traffic light and LEDs amber are for pedestrian buzzer on both roads side (RS-NS and RS-WE).

Port 3

<table>
<thead>
<tr>
<th>P3(^7)</th>
<th>P3(^6)</th>
<th>P3(^5)</th>
<th>P3(^4)</th>
<th>P3(^3)</th>
<th>P3(^2)</th>
<th>P3(^1)</th>
<th>P3(^0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GREEN</td>
<td>YELLOW</td>
<td>RED</td>
<td>AMBER</td>
<td>AMBER</td>
<td>RED</td>
<td>YELLOW</td>
<td>GREEN</td>
</tr>
</tbody>
</table>

For P3 = 0x84: Pin P3\(^7\) and P3\(^2\) of port 3 are ON which is green from RS-NS and red from RS-WE.
7-Segment display counts down from 10 to 0.
For P3 = 0x46: Pin P3^6, P3^2 and P3^1 of port 3 are ON which is yellow from RS-NS and red, yellow from RS-WE.

For P3 = 0x21: Pin P3^5 and P3^0 of port 3 are ON which is red from RS-NS and green from RS-WE.
7-Segment display counts down from 10 to 0.
For $P3 = 0x62$: Pin $P3^6$, $P3^5$ and $P3^1$ of port 3 are ON which is yellow, red from RS-NS and yellow from RS-WE.

For $P3 = 0x44$: Pin $P3^6$ and $P3^2$ of port 3 are ON which is yellow from RS-NS and red from RS-WE.
Calling M5450_Switch (): Pin P3^5, P3^4, P3^3 and P3^2 of port 3 are ON which is red, amber from RS-NS and red, amber from RS-WE. 7-Segment display counts down from 15 to 0.

For P3 = 0x64: Pin P3^6, P3^5 and P3^2 of port 3 are ON which is red from RS-NS and red, yellow from RS-WE
5. CONCLUSION

In this Case Study we have studied the importance of embedded system in real life by implementing the optimization of Microcontroller-based Traffic Light Controller to reduce the possibility of traffic jams and improve the safety issues surrounding major traffics by applying accurate timing and delay over traffic light sequences. But this case study outcome can be enhanced in such way as to control automatically the signals depending on the traffic density on the roads using sensors like infrared (IR) detector/receiver module extended with automatic turn off when no vehicles are running on any side of the road which helps in power consumption saving.

6. REFERENCES

Common anode common cathode image

http://arduinoexplained.blogspot.co.uk/2012_04_01_archive.html

Intelligent traffic light and density control using IR sensor and microcontroller pdf

http://www.ijater.com/Files/IJATER_03_06.pdf.

Microcontroller Projects in C for the 8051


MINI-MAX/51-C2 PDF from module Tutorial document


MINI-MAX/908-C PDF


Development of a traffic light control system PDF