COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE

CEL 324

Bahria University
Discovering Knowledge

Mohammad Abdullah (049)
Saad Ahmed (077)
Asad Ullah (013)
Zeeshan Aurengzeb (100)
Aroob Zuberi (029)
Saiqa Rahmat (081)

BS(Cs)-3B

Term Project

Two-Way Traffic Light Controller

TABLE OF CONTENTS

Two-Way Traffic Light Controller

1. Abstract:

One of the major problems faced in any metro city is traffic congestion. Getting straned in between heavy traffic is a headache for each and every person driving the vehicle and even to the traffic polic in controlling the traffic.

The function of Traffic Lights is to provide sophisticated control and coordination to ensure that traffic moves as smoothly and safely as possible.

2. Introduction:

Traffic congestion is a severe problem in many major cities across the world.

The main purpose of this project is to design the traffic light controller.

The first traffic light was developed on 9th December in 1868 by using Lit Gas and promoted by J.P.Knight.

The first Electric Traffic Light was developed in 1912 by Lester Wire.

2.1. Motivation:

- ✓ The project finds high practical and widespread use.
- ✓ It is very primitive application of the micro-controller.
- ✓ Easy and convinient to be built for a beginner as the coding comprises of basic instructions.

3. Literature Review:

- **Rongrong Tian, Xu Zhang** suggested to use the **TRANSYT** traffic modeling software to find the optimal fixed-time signal plan and

VISSIM micro-simulation software to affirm and evaluate the **TRANSYT model** and to help assess the optimal signal plan; build an adaptive

frame signal plan and refined and evaluated the plan using **VISSIM** with **VS-PLUS emulator**. Through micro-simulation, it was shown that

delay in the adaptive signal control was shortened noticeably than that in the fixed time control.

- **Jianhua Guo et al** introduced a new method for area-wide traffic signal timing optimization under user **equilibrium traffic**. The optimization

model was formulated as a multi-dimensional search problem aimed to achieve minimized product of the total travel time associated with

urban street network and the variance of travel time for unit distance of travel. A genetic algorithm was developed to derive the model

solution. A simulation control protocol embedded in **PARAMICS software tool** capable of conducting area-wide micro simulation is adopted

to design the logic frame and function module of the area-wide traffic signal control system. His results shown that mobility improvements are

achieved after applying the proposed model along with the genetic algorithm for area-wide signal timing optimization, assessed by extended

capacity ratio, and reductions in through and turning movement delays, as well as average and variance of travel time for unit distance of

travel.

4. Objectives:

The objective behind the proposal is to limit the stoppage time and also regulate the traffic flow by means of the introduction of the sensors at all major traffic signals.

The proposal aims at reducing the traffic jams in order to reduce traffic congestion, optimize traffic flow and help pro actively manage traffic conditions.

5. Problem Statement:

The purpose of this project is to develop a series of systems model for traffic passing through a 2-way intersection, controlled by traffic light. We will assume that arrangement of traffic lights and road lanes is fixed and that the lights switch from red to green to amber in a regular repetitive pattern. Moreover, we assume that driver behavior is constrained by the road rules (we keep this part really simple) and the desire to avoid vehicle collisions.

6. Methodology:

We have implemented this project using Atmega16A AVR Microc-controller and is programmed in 8051 Assembly Language.

6.1. Components:

- ✓ Atmega16A Microcontroller
- ✓ PCB (Printed Circuit Board)
- ✓ Crystal Oscillator
- ✓ 22pf Capacitors
- ✓ 10k and 1k resistors
- ✓ 7805 voltage regulator
- ✓ 1n4007 Diode
- ✓ 22 ohm resistor
- ✓ LEDs (Red,Green,Yello)
- ✓ 2 pin connector
- ✓ 9volt Power Supply Battery

Atmega16A Microcontroller:

Atmega16A is an 8-bit high performance microcontroller of Atmel's Mega AVR family with low power consumption.

Atmega 16A is 40 pin microcontroller.There are 32 I/O lines which are divided into four 8-bit ports designated as PortA, PortB, PortC and PortD.

LEDs:

LED means Light emiting Diode. It consists of two terminals anode and cathode.

We are using Red, Yellow and Green LED's based on their indication.

Resistors:

A resistor is a passive two terminal electrical component.

In electronic circuits, resistors are used to reduce current flow.

6.2. Working:

This project uses a LED light as an indicator.

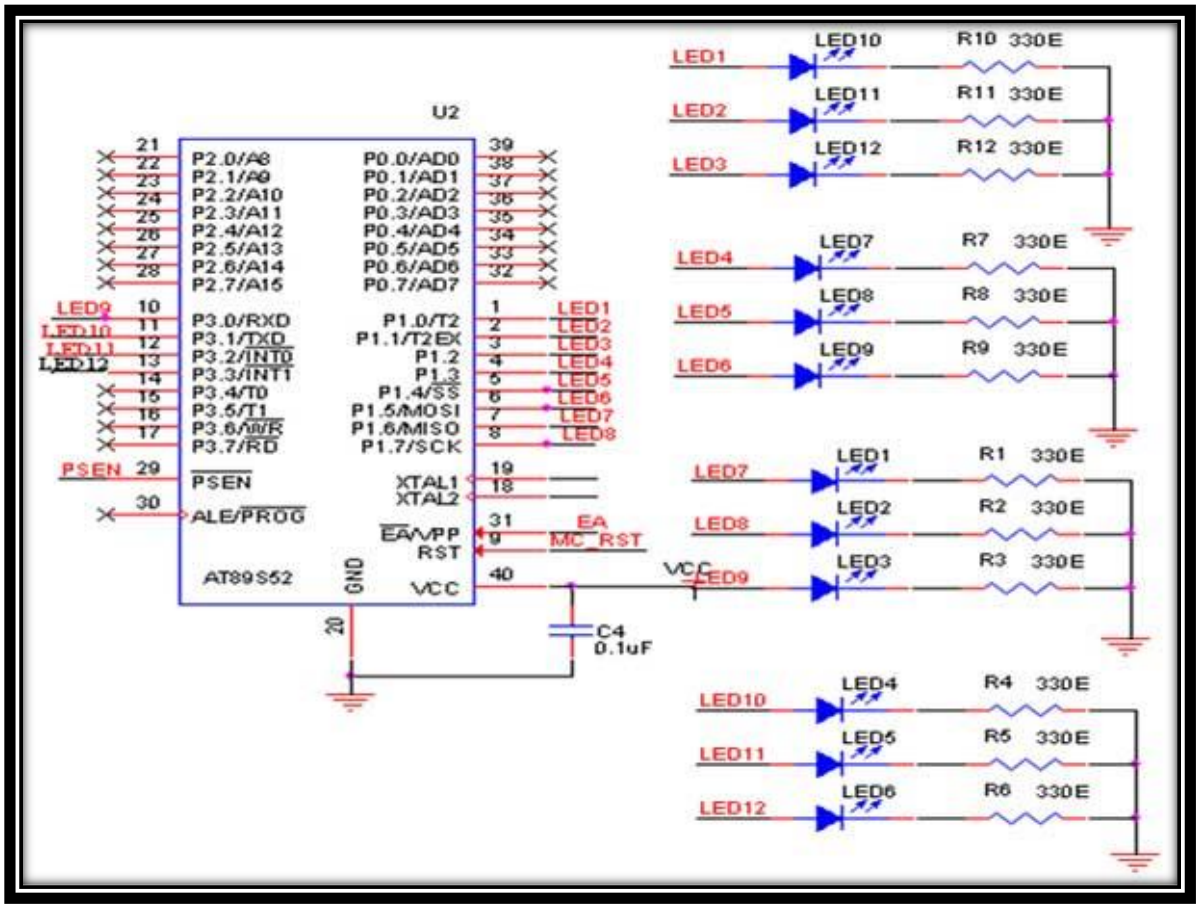
A microcontroller for auto change signal after a specific time interval.

The pins of the various input output ports of microcontroller are connected directly to the given LEDs.

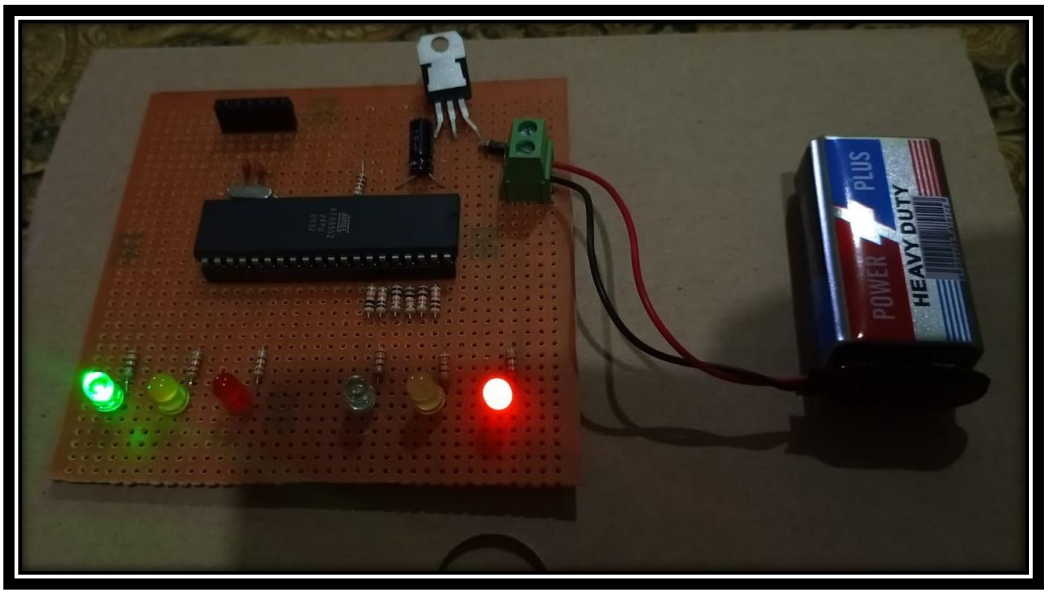
The LEDs are automatically on and off by making the corresponding port pin of the microcontroller high.

The 8051 is programmed in a mannaer that the respective LEDs glow by setting the required bit using assembly language and a certain amount of delay is provided depending on the user.

Circuit Diagram:



Project Visualization:



6.3. Software Used:

Studio 7 is the integrated development platform (IDP) for developing and debugging all AVR and SAM microcontroller applications. The Atmel Studio 7 IDP gives you a seamless and easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code.

6.4. Code Segments:

main.asm

```
0000 C285               CLR     LEDg2_pin
0002 7FE8               MOV     R7,#0E8H
0004 7E03               MOV     R6,#03H

0006 E4                 CLR     A
0007 FD                 MOV     R5,A
0008 FC                 MOV     R4,A
0009          ?C0004:
0009 C3                 CLR     C
```

```
000A ED                 MOV     A,R5
000B 9F                 SUBB    A,R7
000C EE                 MOV     A,R6
000D 6480               XRL     A,#080H
000F F8                 MOV     R0,A
0010 EC                 MOV     A,R4
0011 6480               XRL     A,#080H
0013 98                 SUBB    A,R0
0014 5015               JNC     ?C0010
                                        ; SOURCE
LINE # 69
                                        ; SOURCE
LINE # 70
;---- Variable 'j' assigned to Register 'R2/R3' ----
0016 E4                 CLR     A
0017 FB                 MOV     R3,A
0018 FA                 MOV     R2,A
0019            ?C0007:
                                        ; SOURCE
LINE # 71
                                        ; SOURCE
LINE # 72
0019 0B                 INC     R3
001A BB0001             CJNE    R3,#00H,?C0011
001D 0A                 INC     R2
001E            ?C0011:
001E EB                 MOV     A,R3
```

```
001F 6464                XRL     A,#064H
0021 4A                  ORL     A,R2
0022 70F5                JNZ     ?C0007



                                                    ; SOURCE
LINE # 73
0024           ?C0006:
0024 0D                  INC     R5
0025 BD0001              CJNE    R5,#00H,?C0012
0028 0C                  INC     R4
0029           ?C0012:
0029 80DE                SJMP    ?C0004
                                                    ; SOURCE
LINE # 74
002B           ?C0010:
002B 22                  RET     ;
```

main.c

```
#include<reg51.h> /* special function register declarations */
/* for the intended 8051 derivative */

sbit LEDr1_pin = P0 ^ 0; //Defining LED PIN
sbit LEDy1_pin = P0 ^ 1; //Defining LED PIN
sbit LEDg1_pin = P0 ^ 2; //Defining LED PIN
sbit LEDr2_pin = P0 ^ 3; //Defining LED PIN
sbit LEDy2_pin = P0 ^ 4; //Defining LED PIN
sbit LEDg2_pin = P0 ^ 5; //Defining LED PIN

void Delay(int); //Function prototype declaration

void main(void)
{
```

```
	LEDr1_pin = 0; //LED off initially
	LEDy1_pin = 0; //LED off initially
	LEDg1_pin = 0; //LED off initially
	LEDr2_pin = 0; //LED off initially
	LEDy2_pin = 0; //LED off initially
	LEDg2_pin = 0; //LED off initially

	while (1) //infinite loop
	{

		LEDr1_pin = 1; //LED off initially
		LEDy1_pin = 0; //LED off initially
		LEDg1_pin = 0; //LED off initially
		LEDr2_pin = 0; //LED off initially
		LEDy2_pin = 0; //LED off initially
		LEDg2_pin = 1; //LED off initially

		Delay(4000);

		LEDr1_pin = 1; //LED off initially
		LEDy1_pin = 0; //LED off initially
		LEDg1_pin = 0; //LED off initially
		LEDr2_pin = 0; //LED off initially
		LEDy2_pin = 1; //LED off initially
		LEDg2_pin = 0; //LED off initially

		Delay(1000);

		LEDr1_pin = 0; //LED off initially
		LEDy1_pin = 0; //LED off initially
		LEDg1_pin = 1; //LED off initially
		LEDr2_pin = 1; //LED off initially
		LEDy2_pin = 0; //LED off initially
		LEDg2_pin = 0; //LED off initially

		Delay(4000);

		LEDr1_pin = 0; //LED off initially
		LEDy1_pin = 1; //LED off initially
		LEDg1_pin = 0; //LED off initially
		LEDr2_pin = 1; //LED off initially
		LEDy2_pin = 0; //LED off initially
		LEDg2_pin = 0; //LED off initially

		Delay(1000);
	}
}

void Delay(int k)
{
	int j;
	int i;
	for (i = 0; i < k; i++)
	{
		for (j = 0; j < 100; j++)
```

```
            {
            }
      }
}
```

main.lst

```
0000 C285               CLR     LEDg2_pin
0002 7FE8               MOV     R7,#0E8H
0004 7E03               MOV     R6,#03H

0006 E4                 CLR     A
0007 FD                 MOV     R5,A
0008 FC                 MOV     R4,A
0009          ?C0004:
0009 C3                 CLR     C
000A ED                 MOV     A,R5
000B 9F                 SUBB    A,R7
000C EE                 MOV     A,R6
000D 6480               XRL     A,#080H
000F F8                 MOV     R0,A
0010 EC                 MOV     A,R4
0011 6480               XRL     A,#080H
0013 98                 SUBB    A,R0
0014 5015               JNC     ?C0010

0016 E4                 CLR     A
0017 FB                 MOV     R3,A
```

```
0018 FA                MOV     R2,A
0019          ?C0007:

0019 0B                INC     R3
001A BB0001            CJNE    R3,#00H,?C0011
001D 0A                INC     R2
001E          ?C0011:
001E EB                MOV     A,R3
001F 6464              XRL     A,#064H
0021 4A                ORL     A,R2
0022 70F5              JNZ     ?C0007




0024          ?C0006:
0024 0D                INC     R5
0025 BD0001            CJNE    R5,#00H,?C0012
0028 0C                INC     R4
0029          ?C0012:
0029 80DE              SJMP    ?C0004
```

7. Conclusion:

Finally we implemented the Traffic Light Control System using Atmega16A microcontroller on breadboard.

This project can be enhanced in such a way as to control automatically the signals depending on the traffic density on the roads using sensors like 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.

Thank You...