Simulate Modern Traffic Control System Using 8051 Microcontroller and ALP

Md. Mahmudur Rahman Limon

ID: 2018-1-60-253

Dept. of Computer Science and Engineering,

East West University, Bangladesh

***Abstract*— This paper concludes the details of how we can simulate a modern traffic control system and density control system using 8051 microcontroller in Proteus software with Assembly Language being the programming language. In this project I tried to simulate the modern traffic control system in proteus with four-way traffic where one lane starts, and 3 lanes stop. I also added a time counter in every traffic lane. To do that I used Assembly Language Programming and the tool I used was Keil uVision 5.**

***Keywords*— Simulation, Assembly, Traffic control, Lights, Component, Counter, Logic.**

I. INTRODUCTION

Traffic lights consist normally of three signals, transmitting meaningful information to drivers and riders through colors and symbols including arrows and bicycles. The regular traffic light colors are red, yellow and green, arranged vertically or horizontally in that order. Although this is internationally standardized, variations exist on national and local scales as to traffic light sequences and laws.

The first system of traffic signals was installed as a way to replace police officer control of vehicular traffic outside the Houses of Parliament in London on 9 December 1868. In the first two decades of the 20th century, semaphore traffic signals like the one in London were in use all over the United States with each state having its own design of the device. In many cases, it was controlled by a traffic officer who would blow a whistle before changing the commands on this signal to help alert travelers of the change. In 1912, the first electric traffic light was developed by Lester Wire, a policeman in Salt Lake City, Utah. It was installed by the American Traffic Signal Company on the corner of East 105th Street and Euclid Avenue in Cleveland, Ohio. The first four-way, three-color traffic light was created by police officer William Potts in Detroit, Michigan in 1920. He was concerned about how police officers at four different lights signals could not change their lights all at the same time. The answer was a third light that was colored yellow, which was the same color used on the railroad. In 1922 traffic towers were beginning to be controlled by automatic timers. The main advantage for the use of the timer was that it saved cities money by replacing traffic officers. The city of New York was able to reassign all but 500 of its 6,000 officers working on the traffic squad; this saved the city $12,500,000.

The control of traffic lights made a big turn with the rise of computers in America in the 1950s. One of the best historical examples of computerized control of lights was in Denver in 1952. In 1967, the city of Toronto was the first to use more advanced computers that were better at vehicle detection. The computers-maintained control over 159 signals in the cities through telephone lines.

Traffic simulation or the simulation of

transportation systems is the mathematical modeling of transportation systems (e.g., freeway junctions, arterial routes, roundabouts, downtown grid systems, etc.) through the application of computer software to better help plan, design, and operate transportation systems. Simulation of transportation systems started over forty years ago, and is an important area of discipline in traffic engineering and transportation planning today. Simulation in transportation is important because it can study models too complicated for analytical or numerical treatment, can be used for experimental studies, can study detailed relations that might be lost in analytical or numerical treatment and can produce attractive visual demonstrations of present and future scenarios. Simulation software is getting better in a variety of different ways. With new advancements in mathematics, engineering and computing, simulation software programs are increasingly becoming faster, more powerful, more detail oriented and more realistic.

II. RELATED WORKS

A. Intelligent Traffic Control System Based on DSP and Nios II.

The authors designed an intelligent traffic control system based on DSP and Nios II. Using Dual-CPU, intelligent traffic control system combined with logic control in FPGA implemented kinds of functions, which included cross phase adjustment, exchanging and establishing related information, live human-computer interaction and remote control etc. This system works mostly at the mode of timing and multiple phases according to the user demands dynamically, thus it breaks through the bottleneck of traditional traffic signal controller, and can accomplish the control in complicated and diversified traffic.

B. Distributed traffic control system based on model predictive control

This paper investigates a distributed control system scheme for urban road traffic management. The author's control algorithm is based on model predictive control (MPC) involving Jacobi iteration algorithm to solve constrained and nonlinear programming problem.

C. Modern traffic control system

The authors introduced Fuzzy Logic in the solution of traffic control. With the help of Fuzzy Logic, they set some rules that depended upon 2 factors i.e. number of vehicles and rate of vehicles approaching towards the traffic signal poles. According to the rules based on above two factors they set a timer. They performed the simulation in the Mamdani Fuzzy Inference System.

D. Design and Simulation of an Intelligent Traffic Control System

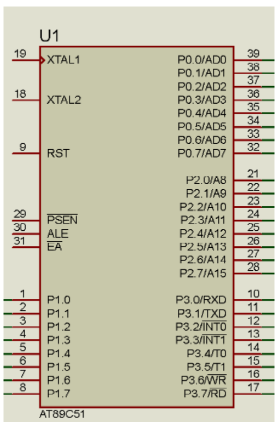
The authors described their research experiences of building an intelligent system to monitor and control road traffic in a Nigerian city. A hybrid methodology obtained by the crossing of the Structured Systems Analysis and Design Methodology (SSADM) and the Fuzzy-Logic based Design Methodology was deployed to develop and implement the system. They identified the problems with the current traffic control system at the '+' junctions thus necessitated the design and implementation of a new system to solve the problems.

E. Simulation of modern Traffic Lights Control Systems using SUMO

This research's purpose was the improvement of the flow throughput of junctions. The authors showed their work using the open source traffic Simulation "SUMO" (Simulation of Urban Mobility). This publication described the algorithm itself and how it was embedded within the simulation. Also, they provided the simulation results.

III. PROPOSED WORK

I tried to create a project where the modern traffic control simulation will be done using Proteus software. In order to do that I used simulation components 7SEG-MPX2-CA-BLUE, AT89C51, RESPACK-8 and TRAFFIC LIGHTS in Proteus.

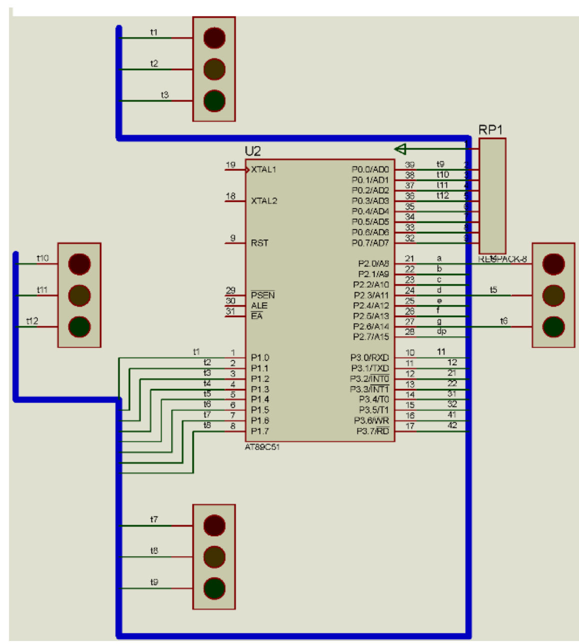


AT89C51 is an old 8-bit microcontroller from the Atmel family. It works with the popular 8051 architecture It is a 40 pin IC package with 4Kb flash memory. It has four ports and all together provide 32 Programmable GPIO pins. It does not have in-built ADC module and supports only USART communication.

RESPACK is a device just similar to resistance box used for the variation of the resistances as per use of the circuit. But there is subtle difference in the RESPACK that is the resistance present in it are of same value and here the RESPACK used RESPACK-8 which consists of 8 resistances of equal value i.e. 1 K ohm.

In Proteus there is a component called TRAFFIC LIGHTS that can behave exactly as a traffic light. I used that component in the project to simulate the behavior of a traffic light.

I completed the Proteus simulation project by starting with AT89C51. Then I added the RESPACK-8 component and connected the wires with AT89C51. After that I added 4 traffic lights representing a 4-way traffic lane. The simulation project looked like this:



Next, I added the time counter component 7SEG-MPX2-CA-BLUE. This component will show how many seconds after the light of a lane will turn to green, red and yellow. In order to determine the time to control the traffic I used Assembly Language. I developed the code with minding the logic that the lights will turn from red to yellow, yellow to green and green to red again. Then I developed the timer in a way that, only 1 light can be green at a time. And 2 lights will stay red at the same time. One light will be on yellow waiting for the green light to turn red so that it can turn green. The timer will follow the logic that, time for green is 10 seconds. Meaning any light can not stay being green more than 10 seconds. After every 10 seconds if a light is green, it will turn to yellow. The logic for yellow is same, time counter for yellow is 10 seconds. And will behave same as green. The red lights will behave same

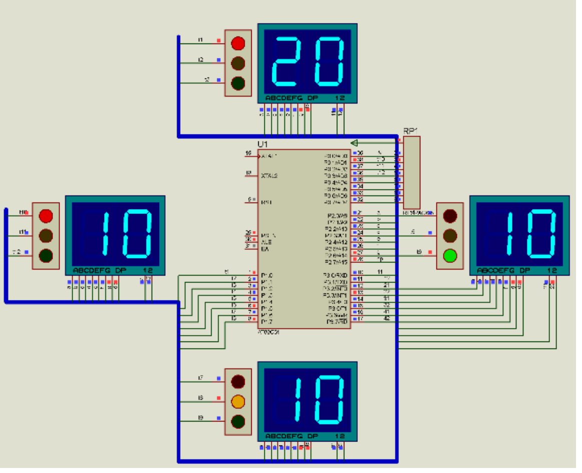
too. Every red light will stay red for 20 seconds. After that the light will turn to yellow. The assembly code:

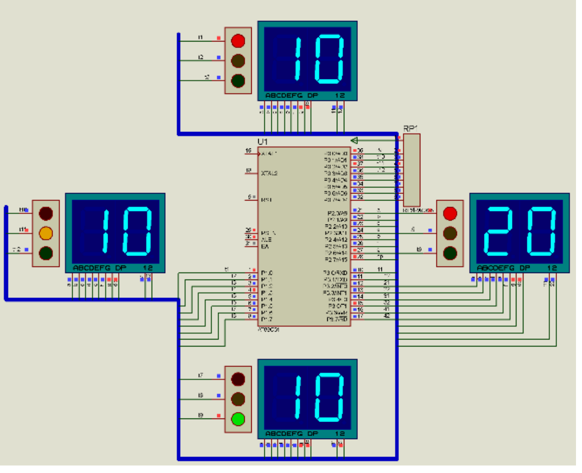
```
ORG 00H
        LJMP MAIN
        ORG 300H
TBL:    DB
0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,8
0H,90H
 ORG 30H
MAIN:   MOV P2,#00H
        MOV P3,#00H
        ACALL FRONT
        MOV DPTR,#TBL
        CLR A
        MOV 40H,#10
        MOV 43H,#10
        MOV 46H,#20
        MOV 49H,#20
        MOV R0,#35
        MOV R6,#30
        MOV R7,#40
X1:     MOV A,40H
        MOV B,#10
        DIV AB
        MOV 41H,A
        MOV 42H,B
A1:     SETB P3.0
        CLR P3.1
        MOV A,41H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        SETB P3.1
        CLR P3.0
        MOV A,42H
        MOVC A,@A+DPTR
        MOV P2,A
       ACALL DELAY
        MOV P3,#00H
        SJMP X3
X2:     SJMP X1
X3:     MOV A,43H
        MOV B,#10
        DIV AB
        MOV 44H,A
        MOV 45H,B
        SETB P3.2
        CLR P3.3
        MOV A,44H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        SETB P3.3
        CLR P3.2
        MOV A,45H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        MOV A,46H
        MOV B,#10
        DIV AB
        MOV 47H,A
        MOV 48H,B
        SETB P3.4
        CLR P3.5
        MOV A,47H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        SETB P3.5
        CLR P3.4
        MOV A,48H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        MOV A,49H
        MOV B,#10
        DIV AB
        MOV 50H,A
        MOV 51H,B
        SETB P3.6
        CLR P3.7
```

```
        MOV A,50H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        SETB P3.7
        CLR P3.6
        MOV A,51H
        MOVC A,@A+DPTR
        MOV P2,A
        ACALL DELAY
        MOV P3,#00H
        DJNZ R0,X2
        MOV R0,#35
        DJNZ 40H,Q1
        MOV 40H,#20
Q1:     DJNZ 43H,Q2
        MOV 43H,#10
        ACALL RIGHT
Q2:     DJNZ 46H,Q3
        MOV 43H,#20
        MOV 46H,#10
Q3:     DJNZ 49H,Q4
        MOV 49H,#10
        ACALL BACK
Q4:     DJNZ R6,X4
        ACALL LEFT
        MOV 40H,#10
        MOV 43H,#10
        MOV 46H,#30
X4:     DJNZ R7,L1
LJMP MAIN
L1:     LJMP X1
DELAY:  MOV R4,#5
H2:     MOV R5,#0FFH
H1:     DJNZ R5,H1
        DJNZ R4,H2
        RET
FRONT:  MOV P1,#54H
        MOV P0,#02H
        RET
RIGHT:  MOV P1,#0A1H
        MOV P0,#02H
       RET
BACK:  MOV P1,#09H
       MOV P0,#05H
       RET
LEFT:  MOV P1,#4AH
       MOV P0,#08H
       RET
```

IV. RESULT ANALYSIS

The simulation is the result the analysis will be of the simulation. The result will be accurate if every light turns to red, yellow and green in that order and follows the time counter properly. The simulation result:





Here the time counter is changing light colors every 10 seconds for green and yellow. And it is changing light color for red in every 20 seconds. One light remains at green and one light stays as yellow at the same time. While two lights stay red at the same time their time counter is not same. One light has already started its timer and counted

10 seconds already. The other red light has just turned into red, so it has to count 20 seconds before changing into yellow.

V. CONCLUSION

The project successfully simulates a 4-way modern traffic control where no traffic congestion is allowed. The simulation behaves a system where only one lane will be open for traffic at a time. But in this project the time counter sometimes displays unwarranted data even though it behaves properly. This could be a bug problem in Proteus simulation environment. It does not affect the logic and outcome of the result.

VI. REFERENCES

1. H. Zhao, X. Zheng and W. Liu, "Intelligent Traffic Control System Based on DSP and Nios II," *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, 2009, pp. 90-94, doi: 10.1109/CAR.2009.59.
2. Tettamanti, T., Varga, I. "Distributed traffic control system based on model predictive control", Periodica Polytechnica Civil Engineering, 54(1), pp. 3–9, 2010. https://doi.org/10.3311/pp.ci.2010-1.01
3. Chandel, Shiva, Shubhransh Yadav, and Sandeep Yadav. "Modern traffic control system." *Malaya J. Matematik S* 1 (2018): 22-25.
4. Chinyere, Osigwe Uchenna, Oladipo Onaolapo Francisca, and Onibere Emmanuel Amano. "Design and simulation of an intelligent traffic control system." *International journal of advances in engineering & technology* 1.5 (2011): 47.
5. Krajzewicz, Daniel, et al. "Simulation of modern traffic lights control systems using the open source traffic simulation SUMO." *Proceedings of the 3rd industrial simulation conference 2005*. EUROSIS-ETI, 2005.
6. Wikipedia contributors. (2022, September 1). Traffic light. In *Wikipedia, The Free Encyclopedia*. Retrieved 05:21, September 16, 2022, from https://en.wikipedia.org/w/index.php?title=Traffic_light&oldid=1107957039
7. Wikipedia contributors. (2022, August 20). Traffic simulation. In *Wikipedia, The Free Encyclopedia*. Retrieved 05:23, September 16, 2022, from https://en.wikipedia.org/w/index.php?title=Traffic_simulation&oldid=1105452137