SOPHISTICATED OPERATION OF TRAFFIC LIGHTS BY MICROCONTROLLER LEARNED FOR TRAFFIC CONGESTION

S. Sankar[1], M. N. Saravana Kumar[2]

[1]Department of Electrical and Electronics Engineering, AMET University, Chennai, TN, India

[2]Department of Electronics and Instrumentation Engineering, Erode Sengunthar Engineering College, Erode, TN, India

ssankarphd@gmail.com, dr.mnskphd@gmail.com

Abstract – *Application possibilities of technically realized strategic planners on intermittent lighting of LED diodes (red, yellow, green) are considered and represented in the paper with programme-defined time intervals. This complex is being extended to the control of traffic lights for traffic management. The implementation of the complex is based on the INTEL 8051 microcontroller. The software is written in an assembly language and version 6.0 Professional is used by ISIS Proteus. One of the purposes of this paper is to illustrate the advantages of the complex over the electromechanical complex based on microcontrollers (microprocessors) and this is provided by a clear illustration that is practically realized.*

Keywords: *intermittent lighting LED, time intervals, traffic management, microcontroller*

1. INTRODUCTION

The regulation of the traffic intersection is realized by means of four traffic lights that work crosswise in pairs and in that way provide safety for both drivers and pedestrians. A special case was performed for pedestrians with a lack of sense of sight. The problem was solved by sound signaling using two different sounds of different frequencies [1]-[4]. Traffic light control is performed using an 80C51 microcontroller. A processor is a digital circuit that performs data processing and control-management functions and is controlled by a program that executes that processor. The structure of the processor consists of various digital circuits and circuits: logic circuits, flip-flops, counters, registers, encoders, decoders, shifters, addition circuits, connected into a functional unit[5]-[7]. On the other hand, the microcontroller is designed to be all in one.

Hardware-software control of the normal mode of operation of the device was performed. In this way, it is controlled whether the required condition obtained by the traffic light program is obtained at the outputs (bulbs). The traffic light program is realized on the basis of the 80C51 microcontroller which is programmed by the software package "ISIS Proteus Professional". VSM Proteus contains models for many different microcontrollers based on the MCS-51 family as well as derivatives of this family. In addition to this family, there are models for PIC microcontrollers, Microchip microcontrollers, Motorola 68HC00 family microcontrollers, and the AVR microcontroller family[8]-[10].

2. 80C51 MICRO CONTROLLER MODEL

- In the INTEL family of single-chip versions, the 80C51 microcontroller is synthesized specifically to serve as a processor in digital control systems as well as real-time control. It is realized on the basis of CMOS technology, which leads to lower energy consumption and increased speed. The basic characteristics of the 80C51 are given:
- - Eight-bit CPU for digital control applications,
- - 128 bytes of data memory,
- - 4K bytes of program memory,
- - fully duplex UART,
- - two 16-bit timers-counters,
- - interruption structure with two levels and five sources,
- - built-in clock generator,
- Boolean processor,
- RAM that can be addressed by bits,
- 64K bytes of program memory space,
- 64K bytes of data space, and
- CMOS versions 80C51 / 80C31 / 87C51.

The 80C51 microcontroller is a microcontroller of the MCS-51 family with 40 pins arranged in two-row Dip or four-row Quip packages. A realistic representation of this model is given in the following figure.

Fig.1. 40th pin DIP pack

The layout of the 80C51 microcontroller model in VSM Proteus is shown in the following figure.

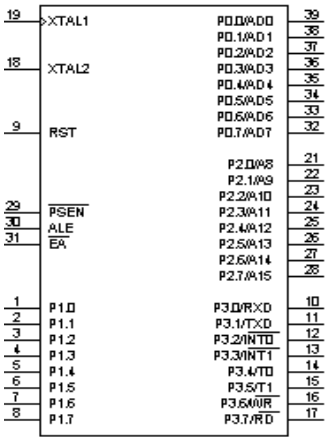

Fig.2. Appearance of the μC 80C51 in the DIL 40 package

Svaki of these pins and ports has its meaning which is described as follows:

- Port0 (P0.0-P0.7) -bidirectional port, gives a lower byte of address during an access to external memory,
- Port1 (P1.0-P1.7) -bidirectional port,
- Port2 (P2.0-P2.7) -bidirectional port, gives a higher byte of address during an access to external memory,
- Port3 (P3.0-P3.7) if it is used as a port then it is a bidirectional port or special functions used on the pins of this port are used. Special functions are,
- P3.0 / RXD serial input,
- P3.1 / TXD serial output,
- P3.2 / INT0-input for an external interrupt of higher priority,
- P3.3 / INT1- input for an external interrupt of lower priority,
- P3.4 / T0-external timer input0,
- P3.5 / T1-external timer input1,
- P3.6 / WR signal for writing to external data memory,
- P3.7 / RD signal for reading from external data memory,
- RST-Reset input: a high logic level on this pin resets the microcontroller for at least two machine cycles. ALE (Adress Latch Enable) - enable the reception register for the address,
- PSEN (Program Select Enable) - output for enabling external program memory,
- EA (External Access) enabling access from external program memory,
- XTAL1i XTAL2 inputs for an external oscillator with a quartz crystal.

3. TRAFFIC INTERSECTION REGULATED BY TRAFFIC LIGHTS

The goal of the intersection, which is regulated by traffic lights, is to ensure traffic safety both in city places and other smaller places, depending on the need. It contains four traffic lights that regulate traffic at the intersection, and traffic light control is performed using an 80C51 microcontroller.

A traffic light is a circuit that represents the alternating ignition of LEDs (red, yellow, and green) with program-defined time intervals and as such can find applications for traffic management. The traffic light can be in one of four states, which is shown in the picture.

- DONE (activated green LEDs),

- PENDING (activated yellow LEDs),

- WAITING (activated red LEDs),

- WAITING-PENDING (activated red and yellow LEDs).

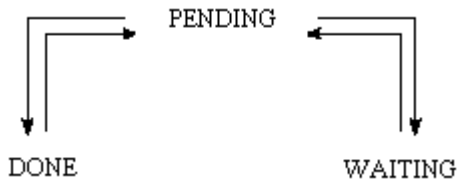

Fig.3. Traffic light transition protocol

When choosing an LED, care is taken of the diameter, which is 3 or 5 mm, the operating current, which is usually about 10 mA, and the color. In order for the LED to light up to the maximum, it is necessary to connect it correctly, ie the positive pole of the power supply is fed to the anode and the ground to the cathode. Usually, the diode housing is notched on the cathode side. In order for the diode to connect properly, it is necessary to add one resistor that will limit the current. The voltage on the diodes is usually 1.2 to 1.6 V.

4. HARDWARE-SOFTWARE REALIZATION

With the help of the software package "ISIS Proteus Professional," the hardware-software realization is performed. It is first necessary to perform a hardware approximation by selecting the appropriate components and connecting them. Next, it is necessary to program the microcontroller, that is, write the program code in the assembler and compile it. Based on this program code, the microcontroller controls the traffic lights. Appearance of the program work environment.

Proteus is quite similar to most commercial applications written for the Microsoft Windows operating system. The work environment contains toolbars, menus, component selectors, etc. The appearance of the work environment is shown in the following figure.

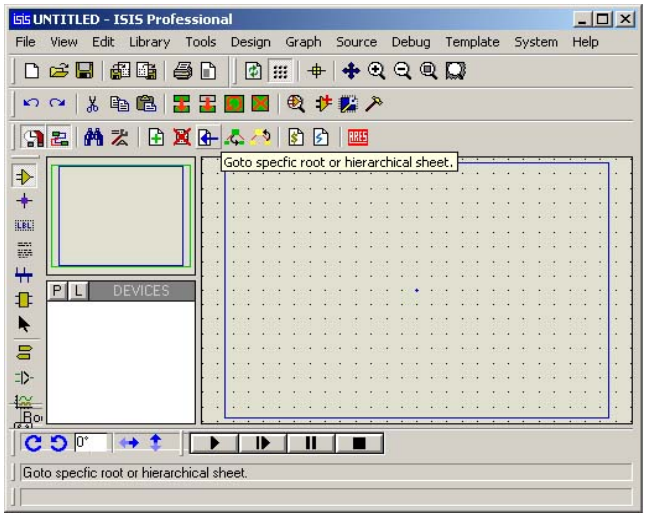

Fig.4. Proteus work environment

When this software package is started and the working environment window is obtained (Fig. 4), the possibility to work with microprocessors is realized. In order to add the source code to the project, it is necessary to select the “Add / Remove Source Files” command from the “Source” menu. . Then, select the code generator for the source file (Code Generation Tool), for the 8051 microcontroller it is ASEM51. In Fig.5. the appearance of the “Add / Remove Source Files” window is displayed.

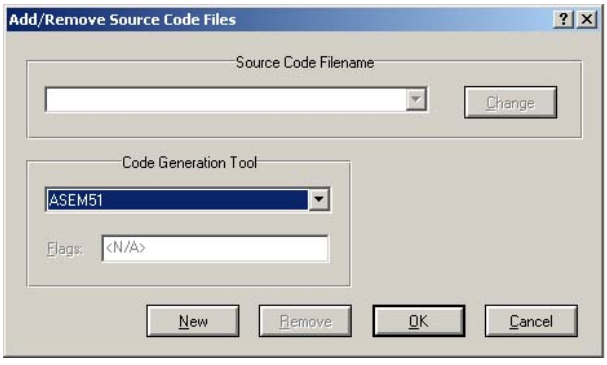

Fig.5. Add/Remove Source Files

The next step is to create a file that will contain the source code of the program. This is achieved by selecting the option "New" after which a window opens where you need to specify the directory where the project will be located with the microcontroller being created. After that, in the field "File Name" we enter the name of the file with the source code. Following this procedure and after a couple of simple selections, the file creation is completed.
Now in the "Source Editor" window, it is necessary to write the code based on when the microcontroller will work. For this example, the code is shown in Fig.6. When the necessary code for the appropriate function of the microcontroller (microprocessor) is written, it is necessary to translate the code using the "Build All" option on the desktop of the simulator and then perform an interactive simulation of the microcontroller. An interactive microcontroller simulation will be performed if the code compilation was successful.

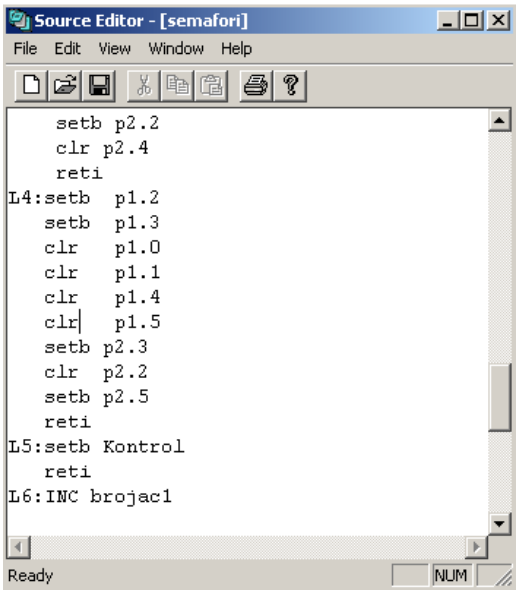

Fig.6. Microcontroller program sequence

The first step is to connect the microcontroller with the translated source code, ie. with a file that has a HEX existence by loading the program into a microcontroller. After that, the “Edit Component” window will appear on the desktop, which looks like the following image.

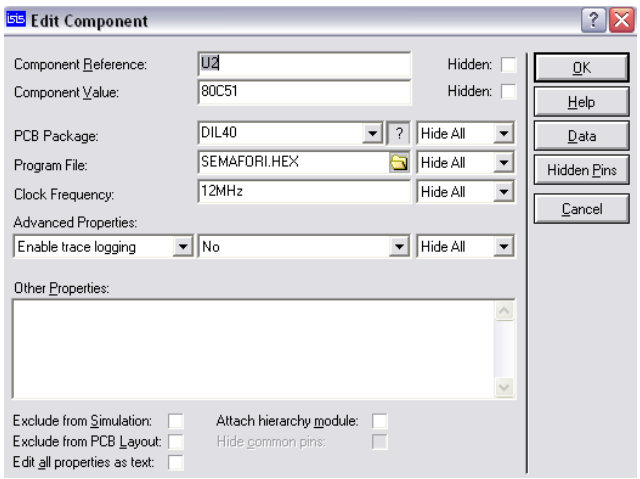

Fig.7. Choice of microcontroller model

For a microcontroller network with different controlled convergences, a simulation session is represented in Figure 1, illustrating the transport infrastructure, the automobiles, the signals in their present incarnation, and several quality measures regarding the various electrical components. After defining the dimension of the traffic grid network, a simulation session is started.

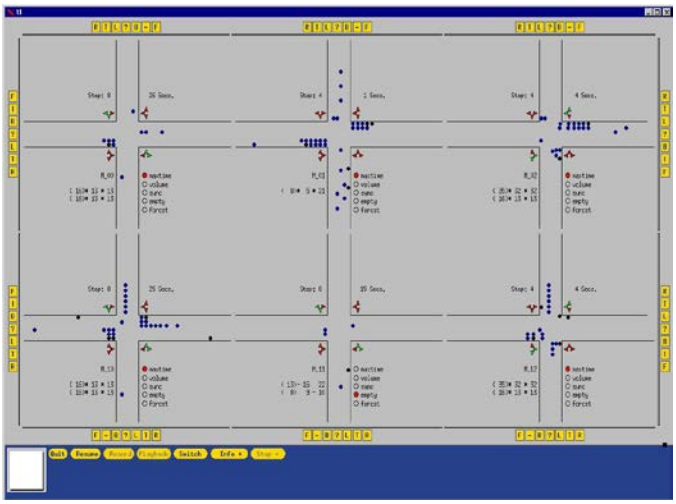

Fig 8. Sequence of microcontroller programme

In a congested area, this operating sequence is a common area here, this is the only signaled convergence in that area for the time. Four single-lane approaches constitute the convergence. Its organization and function are very complex. The several causes conflicts already within the same green process. In Figure 2, a representation of the convergence is given.

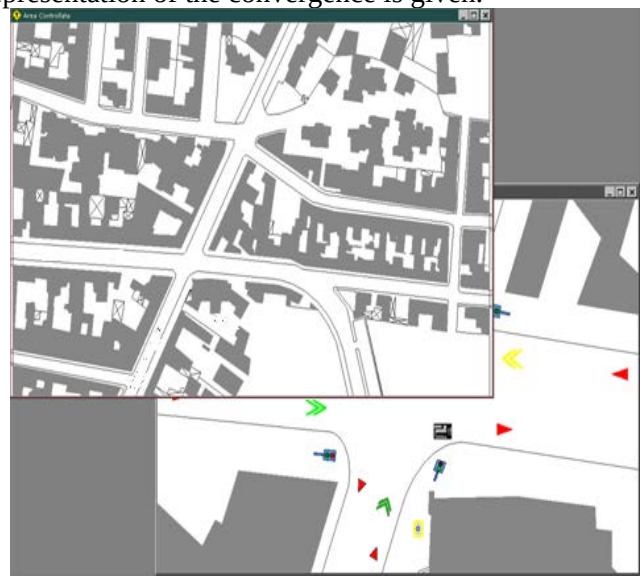

Fig 9. The confluence

In this control module, the overall framework, as shown in Figure 3, consists of various modules to access data using regular sequence protocols of the microcontroller software to demand traffic information,

execute the logic programme, and generate control decisions. If any errors are observed, such as traffic data inconsistency or counter inactivity, the other modules are informed. The module is responsible for communicating through an optical fiber connection with the signal.

There are 16 appearance sensors for cameras and 12 line sensors installed at intersection with real-time counts and occupancy times. The signal cycle can be made up of either two or four.

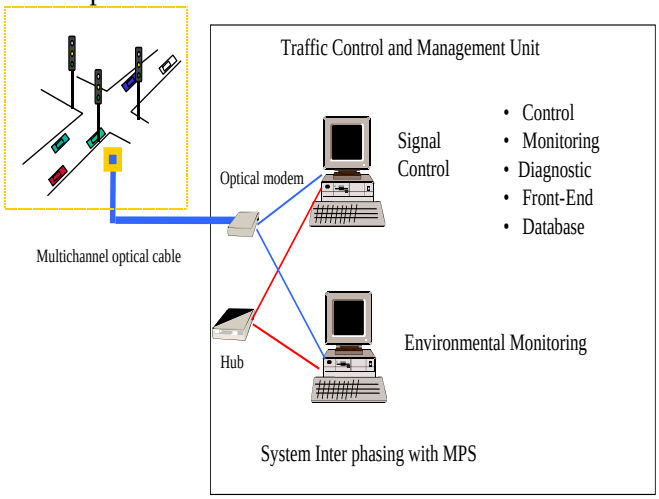

Fig 10. Architecture of a device

For this application, the logic software explicitly built is designed as follows:

- Logic control: It compare traffic levels with options for altering the current stage or adjusting the configuration of the current cycle;
  - Maxime reasoning comments that decide to modify stage after the greatest instance allocated for the current phase has expired;
  - Heavy traffic reasoning comments that plan to alter process when there are several waiting vehicles;
  - Null reasoning sentences that decide to modify stage when a small number of vehicles use the current green phase are available;
- Logical statements of camera failure: there are several faults and contradictory actions of the cameras. These problems can be detected by a diagnostic system and notify the manage unit if the camera is not functioning properly. In this case, for non-operating cameras, some statements provide an estimation of traffic levels, based on the traffic volumes of other methods, phase time and historical data.

The hardware realization of the traffic intersection is done by selecting the appropriate components and connecting them. An overview of the traffic intersection hardware represented by the software package "ISIS Proteus Professional" is given in Fig.7. The problem essentially comes down to controlling the operation of two traffic lights because cross traffic lights work in pairs. This means that they use the same connection lines to the microcontroller port. The control of switching on and off of certain light positions takes place via the pins of port P1. In this case, the duration of the red light at the traffic light is five seconds, which means that the duration of the green light is also five seconds.

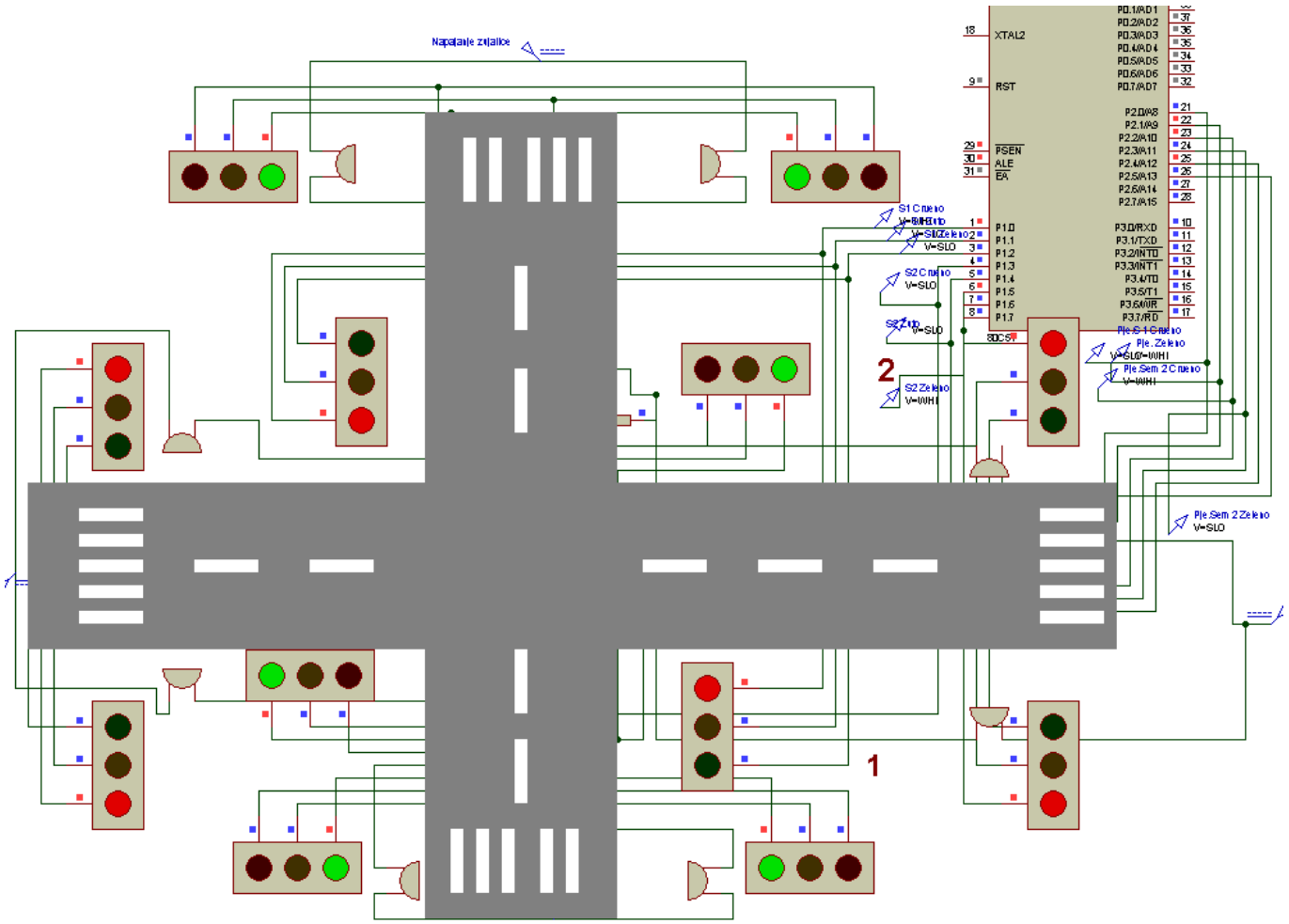

Fig.11. Hardware approximation of traffic intersection

The duration of the yellow light is two seconds. When the duration of the red light at the traffic light ends, the yellow light comes on, which still lights up for two seconds, which means that the transition from red to green light of one traffic light must go through a sequence of active red and yellow light for two seconds. The transition from green to red light at one traffic light must go through a sequence of active yellow light only for two seconds. Based on the previously described mode of operation in Fig. 9. a realistic overview of the functioning of traffic lights for drivers is given.

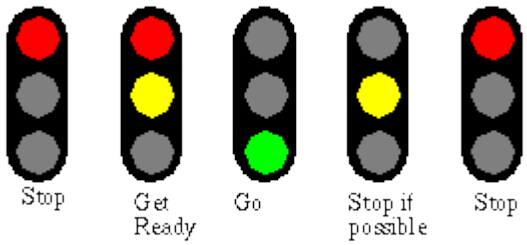

Fig.12. Realistic approximation of traffic light mode

In addition to the solution for drivers, a solution for pedestrians was also implemented. In the case of pedestrians who do not have a sense of sight, an audible signal was performed. Two audible alarms are operating at two different frequencies (100Hz and 500Hz). As there are two pairs of pedestrian crossings each pair contains four traffic lights that possess one sound while the other four traffic lights possess another sound. When a green light is activated on one pair of pedestrian crossings, then an audible signal is activated, which enables a pedestrian without a sense of sight to safely cross the pedestrian crossing. The following figure shows the control of a microcontroller with two traffic lights.

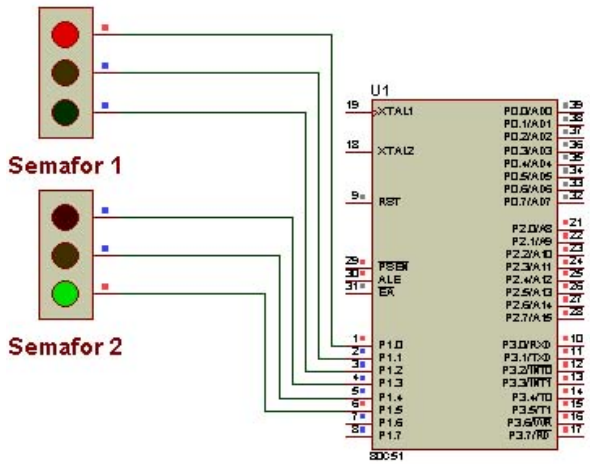

Fig.13. Microcontroller control with two traffic lights

The red light of traffic light 1 is connected to pin P1.0, the yellow light of traffic light 1 is connected to pin P1.1 and the green light of traffic light 1 is connected to pin P1.2. The red light of traffic light 2 is connected to pin P1.3, the yellow light

of traffic light 2 is connected to pin P1.4, and the green light of traffic light 2 is connected to pin P1.5. Figure 11 shows a diagram of light signals for all pairs of traffic lights. From Fig.11. status diagrams are displayed showing the traffic light operation function described in the previous section.

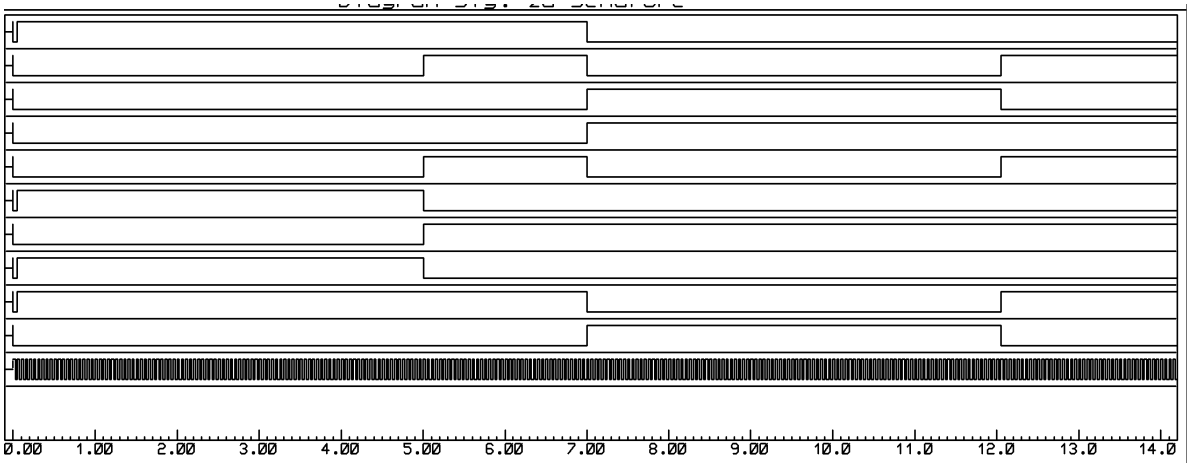

Fig.14 Time diagram of light signals for all traffic light pairs

4. Experimental Results

The experiments with two alternative systems on controlled convergence were carried out with the objective of the sequence of the Microcontroller programme described in the following Argument: A fixed schedule system, hereinafter referred to as 'MANUAL' wherewith the Proteus Professional system, The duration of the cycle and the duration of the stages have been established. A technical integrated system consists, hereinafter referred to as "DYNAMIC." built into the SelfSime distributed convergence devices.

For two types of detectors, the cameras mounted provide traffic data: Detectors and queue detectors count. In real-time, the vehicles operating in the sensed area as well as the moment overall utilization are given by all the detector forms. Similar information may not always be accurate and it is important that the stated rate of error is taken into account when the vehicles are counted.

Second, over a specified time period, we determine the condition of traffic at the convergence accumulated tenure instance of the line sensors. This significance is clearly regulated by the smoothness of the movement throughout the observed area; the variations in the average annual growth rates throughout the lines below the same traffic volumes need to be related to both the operation of the management techniques.

A relationship is then obtained between some of the quantity of the line place where people as well as the amount of people as the final performance measure.

If the two methods can be contrasted with the two indicators in the time interval (t,t '),:

$$IND(M_a)=\frac{\sum_{i=1}^{n} TOC(a_i)}{\sum_{i=1}^{n} VOL(a_i)} \qquad IND(M_b)=\frac{\sum_{j=1}^{n} TOC(b_j)}{\sum_{j=1}^{n} VOL(b_j)}$$

A preliminary combination of the various techniques is then provided by that of the proportion of

$$\frac{IND(M_a)-IND(M_b)}{IND(M_b)}\times 100$$

If method b indicates savings in waiting time for vehicles in contrast to method a, this has a positive value and is negative in the opposite case.

From all the experimental sessions running where data of good quality was available, we picked only a limited number of time intervals, and then we built the occupancy indicators as described above. We have also summarised the efficiency assessments of all of the other cohesive intervention phase carried out just use the same system model by each time period. Therefore, 16 various load length response time (30 to 60 minutes) overlapping growing season are accessible where, as shown in Table 1, the three methods (MANUAL, LOGIC, and DYNAMIC) can be compared. Underneath.

Table 1: Experiment Study

Time Interval		Performance Indicator		
		Manual	Logic	Dynamic
1	10-11	4,516	4,138	4,396
2	12,15-13	3,649	3,131	3,401
3	14,10-14,45	3,283	2,121	2,323
4	15,10-16	3,283	2,371	2,459
5	16,30-17,03	ND	3,181	3,460
6	16,01-20	ND	3,814	3,912
7	16,01-17	ND	2,871	3,027
8	17,01-18	ND	3,876	3,832
9	18,01-19	ND	4,116	4,137

10	19,01-20	ND	4,094	4,296
11	10,30-16	3,654	3,340	3,505
12	11-12	4,992	4,687	4,970
13	12,01-13	3,354	3,224	3,710
14	13,01-14	3,302	2,885	2,876
15	14,01-15	3,421	2,470	2,601
16	15,01-16	2,898	2,328	2,457

The table 1 indicates that in almost all sessions, the LOGIC approach results in savings in queuing time relative to the other two strategies. In the two following tables, the savings obtained are highlighted, Compared to the fixed plan approach and the dynamic adaptive management, we reflect the percentage changes achieved using our approach.

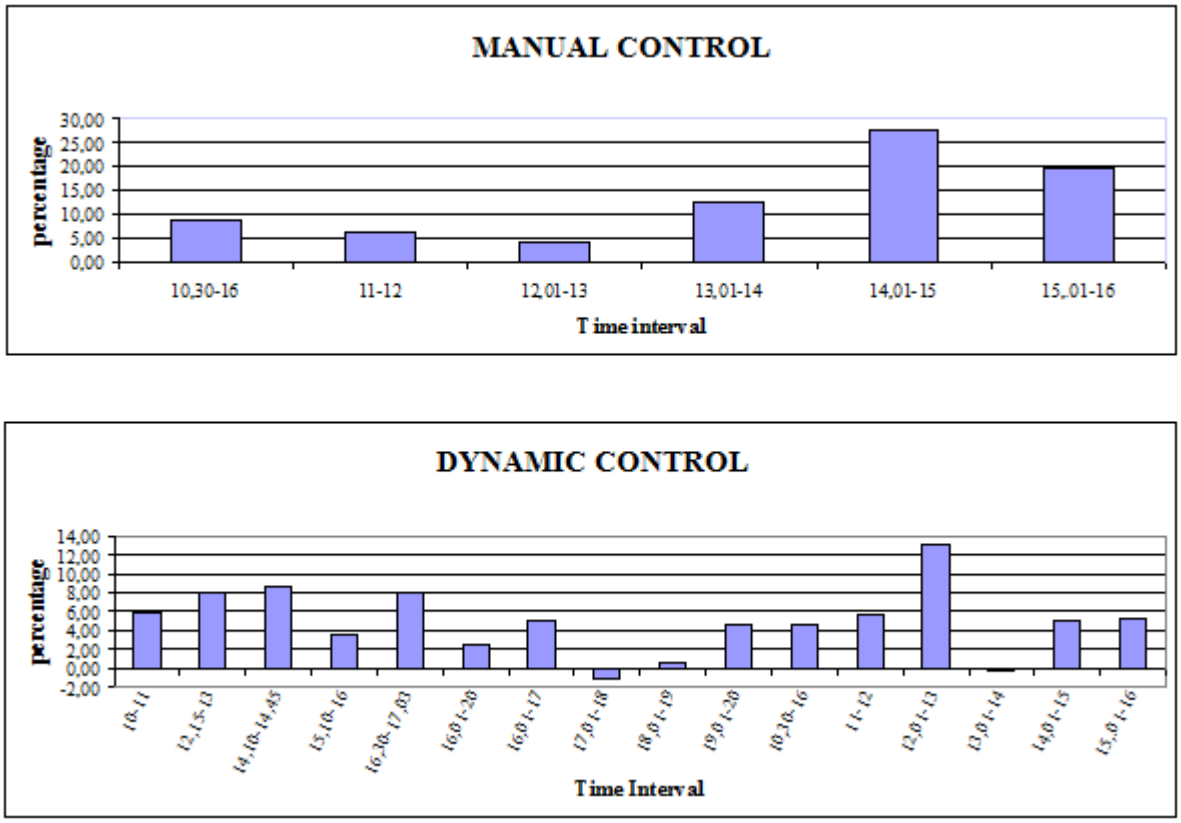

Fig.15. Comparison on manual control and dynamic control

5. CONCLUSION

This paper presents one of the ways to control and manage traffic using a microcontroller. Also, one of the goals of this paper is to show through a simple practical example the advantages of circuits made on the basis of microcontrollers (microprocessors) over electromechanical assemblies. Based on the presented, we can conclude that the circuits realized on the basis of microcontrollers (microprocessors) are much more favorable, in terms of dimensions much smaller, easier to maintain, and in terms of electricity consumption more profitable compared to electromechanical assemblies, which is even more pronounced in more complex assemblies. All these facts indicate the growing expansion of such devices in various spheres of application from the control of home appliances, traffic, and the automotive industry to various industrial devices.

References

[1]. Qiang Chen; Yinong Chen; Jinhui Zhu; Gennaro De Luca; " Traffic light and moving object detection for a guide-dog robot", The Journal of Engineering, Volume: 2020, Issue: 13, 2020. (10.1049/joe.2019.1137)

[2]. Hu Zhang; Shidong Liang; Yin Han; Minghui Ma, " Pre-Control Strategies for Downstream Bus Service Reliability With Traffic Signal", IEEE Access, Volume: 8, 2020. (10.1109/ACCESS.2020.3015982)

[3]. Stanley W. Smith; Yeojun Kim; Jacopo Guanetti; Ruolin Li; Roya Firoozi, " Improving Urban Traffic Throughput With Vehicle Platooning: Theory and Experiments", IEEE Access, Volume: 8, 2020. (10.1109/ACCESS.2020.3012618)

[4]. Ming Ye; Xiangyu Gongye; Yonggang Liu; Xiao Wang, " Research on Dynamic Coordination Active Mode Switching Control Strategy for Hybrid Electric Vehicle Based on Traffic Information", IEEE Access, Volume: 7, 2019. (10.1109/ACCESS.2019.2932585)

[5]. Manuel Contreras; Eric Games, " Real-Time Counting of Vehicles Stopped at a Traffic Light Using Vehicular Network Technology", IEEE

Access, Volume: 8, 2020. (10.1109/ACCESS.2020.3011195)

[6]. Tong Wu; Pan Zhou; Kai Liu; Yali Yuan; Xiumin Wang; Huawei Huang, " Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks", IEEE Transactions on Vehicular Technology, Volume: 69, Issue: 8, 2020. (10.1109/TVT.2020.2997896)

[7]. Xiaohong Zhang; Di Wang, " Adaptive Traffic Signal Control Mechanism for Intelligent Transportation Based on a Consortium Blockchain", IEEE Access, Volume: 7, 2019. (10.1109/ACCESS.2019.2929259)

[8]. Iain Guilliard; Felipe Trevizan; Scott Sanner, " Mitigating the impact of light rail on urban traffic networks using mixed-integer linear programming", IET Intelligent Transport Systems, Volume: 14, Issue: 6, 2020. (10.1049/iet-its.2019.0277)

[9]. Jinbao Mou, " Intersection Traffic Control Based on Multi-Objective Optimization", IEEE Access, Volume: 8, 2020. (10.1109/ACCESS.2020.2983422)

[10]. Pablo Javier Vidal; Ana Carolina Olivera, " Management of urban traffic flow based on traffic lights scheduling optimization", IEEE Latin America Transactions, Volume: 17, Issue: 01, 2019. (10.1109/TLA.2019.8826701)