International Journal For Technological Research In Engineering
Volume 4, Issue 9, May-2017 *ISSN (Online): 2347 - 4718*

TRAFFIC DENSITY CONGESTION SENSING AND CONTROLLING BY USING WIRELESS AD HOC NETWORK

Vanita[1], Umesh Kumar[2]
[1]HOD ECE, [2]M.TECH SCHOLORS, MRKIET REWARI

ABSTRACT: In the present scenario, vehicular traffic is increasing throughout the world, especially in large urban areas. As the number of road users constantly increase and resources provided by current infrastructures being limited, a smart traffic control will become a very important issue in the future. These needs have led to an ever increasing demand for an "intelligent" traffic control system. Therefore, optimization of traffic control to better accommodate this increasing demand is needed. In this project, we will demonstrate the optimization of traffic light controller in a city using wireless sensors. Traffic light optimization is a complex problem. With multiple junctions, the problem becomes even more complex as the state of one light node influences the flow of traffic towards many other nodes. We have proposed a traffic light controller and simulator that allow us to study different situations of traffic density. Using wireless sensors, we can easily sense the density of traffic because the general architecture of wireless sensors is an infrastructure -less communication network. The brain behind the implementation is an ARM 7(AT89S52) programmed using Embedded C, which performs processing of the real time data provided by the wireless sensors, eventually controlling the traffic flow via the LED traffic lights.

I. INTRODUCTION TO EMBEDDED SYSTEMS

History: In the present scenario, vehicular traffic is increasing throughout the world, especially in large urban areas. As the number of road users constantly increase and resources provided by current infrastructures being limited, a smart traffic control will become a very important issue in the future. These needs have led to an ever increasing demand for an "intelligent" traffic control system. Therefore, optimization of traffic control to better accommodate this increasing demand is needed. In this project, we will demonstrate the optimization of traffic light controller in a city using wireless sensors. Traffic light optimization is a complex problem. With multiple junctions, the problem becomes even more complex as the state of one light node influences the flow of traffic towards many other nodes. We have proposed a traffic light controller and simulator that allow us to study different situations of traffic density. Using wireless sensors, we can easily sense the density of traffic because the general architecture of wireless sensors is an infrastructure -less communication network. it is observed that the proposed intelligent Traffic Light controller is more efficient than the convenctional controller in respect of less waiting time, more distance travelled by average vehicles and efficient operation during emergency mode and GSM designed system as simple architecture, fast response time, user friendliness and scope for further expansion.

Overview

The main objective of this project to control the traffic lights based on the density of the vehicles .In this system IR sensors are used to measure the density of the vechicles which are fixed with in a fixed distance. All the sensors are interfaced with the microcontroller which in turn controls the traffic signals system according to density detected by the sensors. The traffic density is high on particular side more priority is given for that side. The sensors continuously keep sensing density on all sides and the green signal is given to the side on priority basis, where the sensors detect high density.The side with next priority level follows the first priority level. By using this system traffic can be cleared without irregularities and time delay seven though there is on the other side can be avoided

Embedded System: An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions, sometimes with real-time computing constraints. It is usually embedded as part of a complete device including hardware and mechanical parts. In contrast, a general-purpose computer, such as a personal computer, can do many different tasks depending on programming. Embedded systems have become very important today as they control many of the common devices we use. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale. Physically, embedded systems range from portable devices such as digital watches and MP3 players, to large stationary installations like traffic lights, factory controllers, or the systems controlling nuclear power plants. Complexity varies from low, with a single microcontroller chip, to very high with multiple units, peripherals and networks mounted inside a large chassis or enclosure. In general, "embedded system" is not an exactly defined term, as many systems have some element of programmability. For example, Handheld computers share some elements with embedded systems such as the operating systems and microprocessors which power them but are not truly embedded systems, because they allow different applications to be loaded and peripherals to be connected. An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular kind of application device. Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines, and toys (as well as the more obvious cellular phone and PDA) are among the myriad possible hosts of an embedded system. Embedded systems that are programmable are provided with a programming interface, and embedded systems programming is a specialized occupation. Certain operating systems or language platforms are tailored for the embedded market, such as Embedded Java and Windows XP Embedded. However, some low-end consumer products use very inexpensive microprocessors and limited storage, with the application and operating system both part of a single program. The program is written permanently into the system's memory in this case, rather than being loaded into RAM (Random Access Memory), as programs on a personal computer. In recent days, you are showered with variety of information about these embedded controllers in many places. All kinds of magazines and journals regularly dish out details about latest technologies, new devices; fast applications which make you believe that your basic survival is controlled by these embedded products. Now you can agree to the fact that these embedded products have successfully invaded into our world. You must be wondering about these embedded controllers or systems. The computer you use to compose your mails, or create a document or analyze the database is known as the standard desktop computer. These desktop computers are manufactured to serve many purposes and applications. You need to install the relevant software to get the required processing facility. So, these desktop computers can do many things. In contrast, embedded controllers carryout a specific work for which they are designed. Most of the time, engineers design these embedded controllers with a specific goal in mind. So these controllers cannot be used in any other place. Theoretically, an embedded controller is a combination of a piece of microprocessor based hardware and the suitable software to undertake a specific task. These days designers have many choices in microprocessors/microcontrollers. Especially, in 8 bit and 32 bit, the available variety really may overwhelm even an experienced designer. Selecting a right microprocessor may turn out as a most difficult first step and it is getting complicated as new devices continue to pop-up very often.

1.4 Basic Blocks of Embedded System:

Now, the details of the various building blocks of the hardware of an embedded system as shown in Fig 1.1 are

. Central Processing Unit (CPU)

· Memory (Read-only Memory and Random Access Memory)

· Input Devices

· Output devices

· Communication interfaces

· Application-specific circuitry

Central Processing Unit (CPU):

The Central Processing Unit (processor, in short) can be any of the following: microcontroller, microprocessor or Digital Signal Processor (DSP). A micro-controller is a low-cost processor. Its main attraction is that on the chip itself, there will be many other components such as memory, serial communication interface, analog-to digital converter etc. So, for small applications, a micro-controller is the best choice as the number of external components required will be very less. On the other hand, microprocessors are more powerful, but you need to use many external components with them. DSP is used mainly for applications in which signal processing is involved such as audio and video processing.

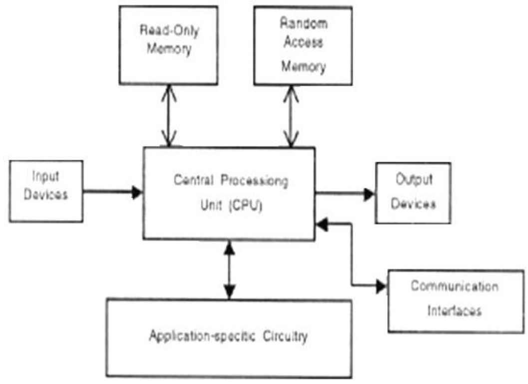

Figure.: Block Diagram of Embedded System

Memory:

The memory is categorized as Random Access Memory (RAM) and Read Only Memory (ROM). The contents of the RAM will be erased if power is switched off to the chip, whereas ROM retains the contents even if the power is switched off. So, the firmware is stored in the ROM. When power is switched on, the processor reads the ROM; the program is executed.

Input devices:

Unlike the desktops, the input devices to an embedded system have very limited capability. There will be no keyboard or a mouse, and hence interacting with the embedded system is not an easy task. Many embedded systems will have a small keypad-you press one key to give a specific command. A keypad may be used to input only the digits. Many embedded systems used in process control do not have any input device for user interaction; they take inputs from sensors or transducers and produce electrical signals that are in turn fed to other systems.

Output devices:

The output devices of the embedded systems also have very limited capability. Some embedded systems will have a few Light Emitting Diodes (LEDs) to indicate the health status of the system modules, or for visual indication of alarms. A small Liquid Crystal Display (LCD) may also be used to display some important parameters.

Communication interfaces:

The embedded systems may need to, interact with other embedded systems at they may have to transmit data to a desktop. To facilitate this, the embedded systems are provided with one or a few communication interfaces such as RS232, RS422, RS485, Universal Serial Bus (USB), and IEEE 1394, Ethernet etc.

Application-specific circuitry:

Sensors, transducers, special processing and control circuitry may be required for an embedded system, depending on its application. This circuitry interacts with the processor to carry out the necessary work. The entire hardware has to be given power supply either through the 230 volts main supply or through a battery. The hardware has to design in such a way that the power consumption is minimized. Security is the condition of being protected against danger or loss. In the general sense, security is a concept similar to safety. The nuance between the two is an added emphasis on being protected from dangers that originate from outside. Individuals or actions that encroach upon the condition of protection are responsible for the breach of security. The word "security" in general usage is synonymous with "safety," but as a technical term "security" means that something not only is secure but that it has been secured. One of the best options for providing good security is by using a technology named EMBEDDED SYSTEMS.

Microcontroller:

In the Literature discussing microprocessors, we often see the term Embedded System. Microprocessors and Microcontrollers are widely used in embedded system products. An embedded system product uses a microprocessor (or Microcontroller) to do one task only. A printer is an example of embedded system since the processor inside it performs one task only, namely getting the data and printing it. Contrast this with a Pentium based PC can be used for any number of applications such as word processor, print-server, bank teller terminal, Video game, network server, or Internet terminal. Software for a variety of applications can be loaded and run. Of course the reason a pc can perform myriad tasks is that it has RAM memory and an operating system that loads the application software into RAM memory and lets the CPU run it.

In an Embedded system, there is only one application software that is typically burned into ROM. An x86 PC contains or is connected to various embedded products such as keyboard, printer, modem, disk controller, sound card, CD-ROM drives, mouse, and so on. Each one of these peripherals has a Microcontroller inside it that performs only one task. For example, inside every mouse there is a Microcontroller to perform the task of finding the mouse position and sending it to the PC.

Application Areas

Nearly 99 per cent of the processors manufactured end up in embedded systems. The embedded system market is one of the highest growth areas as these systems are used in very market segment- consumer electronics, office automation, industrial automation, biomedical engineering, wireless communication, data communication, telecommunications, transportation, military and so on. Consumer appliances: At home we use a number of embedded systems which include digital camera, digital diary, DVD player, video recorders etc. Office automation: The office automation products using embedded systems are copying machine, fax machine, key telephone, modem, printer, scanner etc. Industrial automation: Today a lot of industries use embedded systems for process control. In hazardous industrial environment, where human presence has to be avoided, robots are used, which are programmed to do specific jobs. Medical electronics: Almost every medical equipment in the hospital is an embedded system. These equipments include diagnostic aids such as ECG, EEG, blood pressure measuring devices, radiation, colonoscopy, endoscopy etc. Computer networking: Computer networking products such as bridges, routers, Integrated Services Digital Networks (ISDN), Asynchronous Transfer Mode (ATM), X.25 and frame relay switches are embedded systems which implement the necessary data communication protocols Telecommunications: In the field of telecommunications, the embedded systems can be categorized as subscriber terminals and network equipment. The subscriber terminals such as key telephones, ISDN phones, terminal adapters, web cameras are embedded systems. The network equipment includes multiplexers, multiple access systems, Packet Assemblers Dissemblers (PADs), sate11ite modems etc.

Security: Security of persons and information has always been a major issue. We need to protect our homes and offices; and also the information we transmit and store. Developing embedded systems for devices at homes, offices, airports etc. for authentication and verification are embedded systems security applications is one of the most lucrative businesses nowadays.

Microprocessors Vs Microcontrollers:

- Microprocessors are single-chip CPUs used in microcomputers.
- Microcontrollers and microprocessors are different in 3 main aspects: hardware architecture, applications, and instruction set features.
- Hardware architecture: A microprocessor is a single chip CPU while a microcontroller is a single IC contains a CPU and much of remaining circuitry of a complete computer (e.g., RAM, ROM, serial interface, parallel interface, timer, interrupt handling circuit).
- Applications: Microprocessors are commonly used as a CPU in computers while microcontrollers are found in small, minimum component designs performing control oriented activities.
- Microprocessor instruction sets are processing Intensive.
- Their instructions operate on nibbles, bytes, words, or even double words.
- Addressing modes provide access to large arrays of data using pointers and offsets.
- They have instructions to set and clear individual bits and perform bit operations.
- They have instructions for input/output operations, event timing, enabling and setting priority levels for interrupts caused by external stimuli.
- Processing power of a microcontroller is much less than a microprocessor.

II. SOFTWARE DESCRIPTION

KEIL COMPILER

Keil compiler is software used where the machine language code is written and compiled. After compilation, the machine source code is converted into hex code which is to be dumped into the microcontroller for further processing. Keil compiler also supports C language code. It's important that you know C language for microcontroller which is commonly known as Embedded C. As we are going to use Keil C51 Compiler, hence we also call it Keil C. Keil C is not much different from a normal C program. If you know assembly, writing a C program is not a crisis. In keil, we will have a main function, in which all your application specific work will be defined. In case of embedded C, you do not have any operating system running in there. So you have to make sure that your program or main file should never exit. This can be done with the help of simple while (1) or for (;;) loop as they are going to run infinitely. We have to add header file for controller you are using, otherwise you will not be able to access registers related to peripherals.

#include <REG51.h> //header file for 89C51

To create a project, write and test the previous example source code, follow the following steps:

1. Open Keil and start a new project as shown in figure 5.1

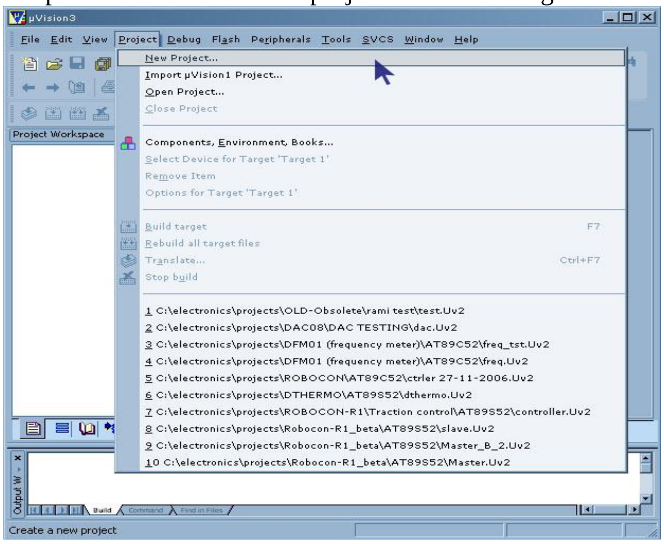

Figure Creating new project

2. You will be prompted to choose a name for your new project, Create a separate folder where all the files of your project will be stored, choose a name and click save. The following figure 5.2 will appear where you will be asked to select a device for Target 'Target 1'

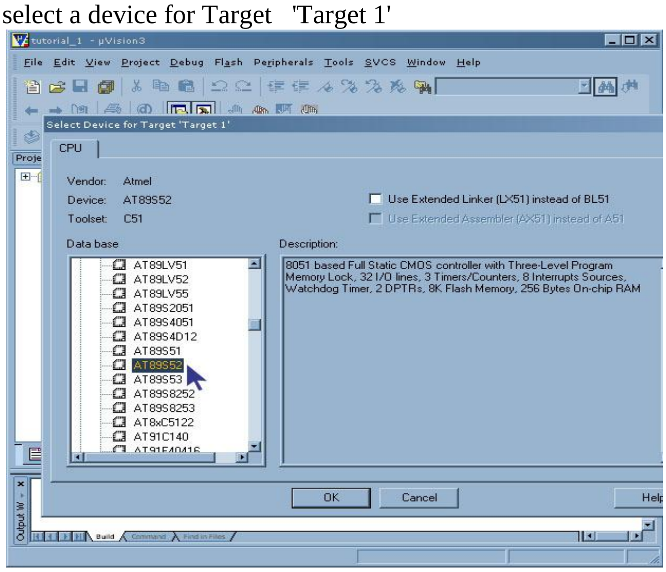

Figure Creating target

3. From the list at the left, seek for the brand name ATMEL, then under ATMEL, select AT89S52. You will notice that a brief description of the device appears on the right. Leave the two upper check boxes unchecked and click OK. The AT89S52 will be called your 'Target device', which is the final destination of your source code. You will be asked whether to 'copy standard 8051 startup code' click No.

4. Click File, New, and something similar to the following figure 5.3 should appear. The box named 'Text1' is where your code should be written later.

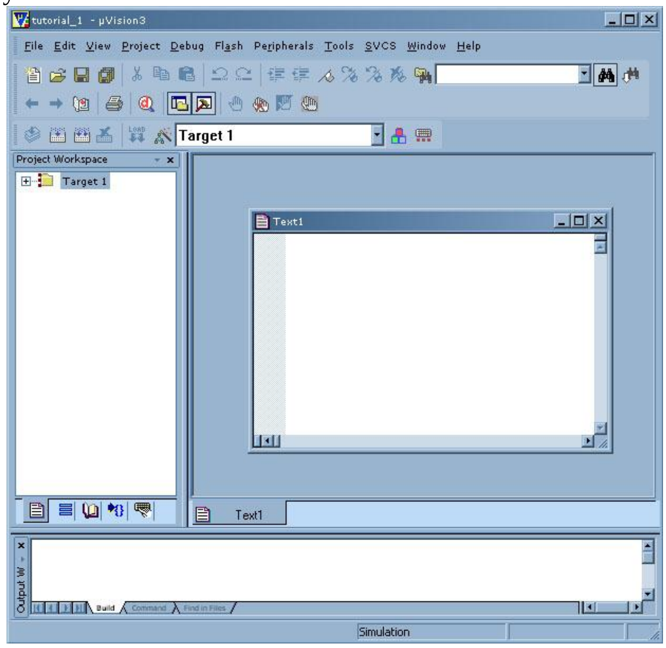

Figure Creating a file

5. Now you have to click 'File, Save as' and choose a file name for your source code ending with the letter '.c'. You can name as 'code.c' for example and click save. Then you have to add this file to your project work space at the left as shown in the following figure5.4.

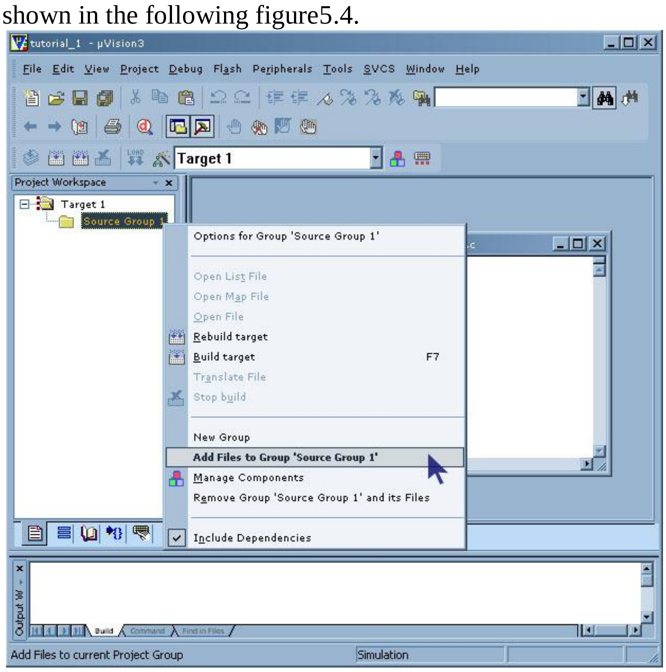

Fig Adding files to the source group

6.After right-clicking on 'source group 1', click on 'Add files to group...', then you will be prompted to browse the file to add to 'source group 1', choose the file that you just saved, eventually 'code.c' and add it to the source group. You will notice that the file is added to the project.

7. In some versions of this software you have to turn ON manually the option to generate HEX files. make sure it is turned ON, by right-clicking on target 1, Options for target 'target 1', then under the 'output' tab, by checking the box 'generate HEX file'. This step is very important as the HEX file is the compiled output of your project that is going to be transferred to the microcontroller.

8. You can then start to write the source code in the figure5.5 titled 'code.c' then before testing your source code; you have to compile your source code, and correct eventual syntax errors. In KEIL IDE, this step is called 'rebuild all targets' and has this icon.

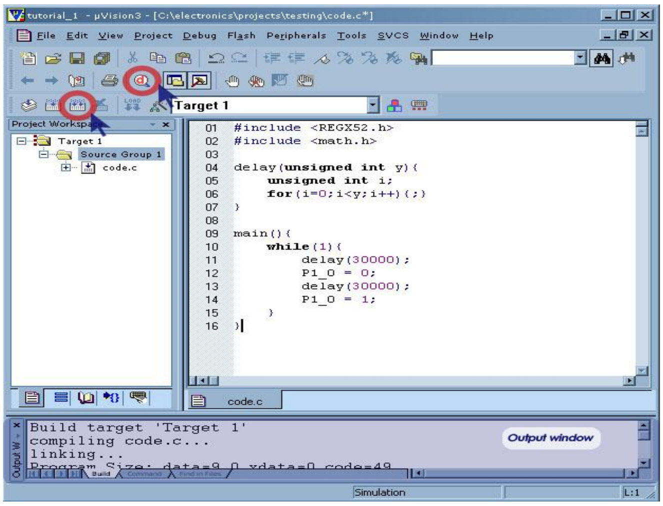

Fig Rebuilding targets

9. If after rebuilding the targets, the 'output window' shows that there is 0 errors, then you are ready to test the performance of your code. In keil, like in most development environment, this step is called Debugging, and has this icon. After clicking on the debug icon, you will notice that some part of the user interface will change; some new icons will appear, like the run icon circled in the following figure 5.6.

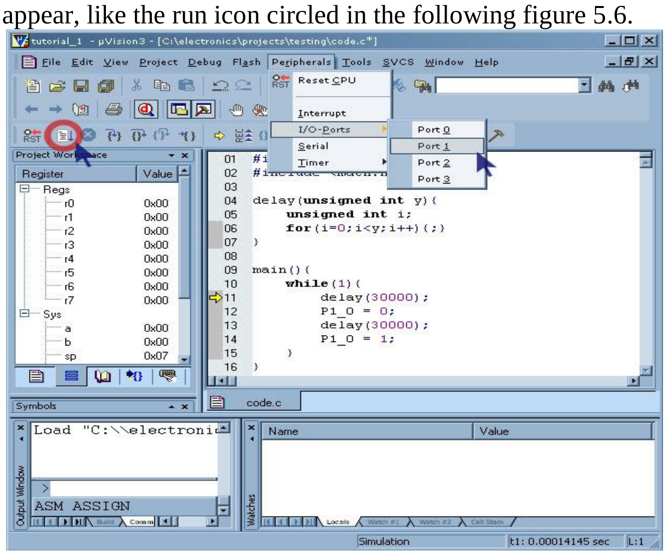

Figure Debugging

10. You can click on the 'Run' icon and the execution of the program will start. In our example, you can see the behavior of the pin 0 or port one, but clicking on 'peripherals, I/O ports, Port 1'. You can always stop the execution of the program by clicking on the stop button and you can simulate a reset by clicking on the 'reset' button. You can also control the execution of the program using the following icons: which allows you to follow the execution step by step. Then, when you're finished with the debugging, you can always return to the programming interface by clicking again on debug button

5.2 PROLOAD

Proload is software which accepts only hex files. Once the machine code is converted into hex code, that hex code has to be dumped into the microcontroller and this is done by the Proload. Proload is a programmer which itself contains a microcontroller in it other than the one which is to be programmed. This microcontroller has a program in it written in such a way that it accepts the hex file from the keil compiler and dumps this hex file into the microcontroller which is to be programmed. As the proload programmer kit requires power supply to be operated, this power supply is given from the power supply circuit designed above. It should be noted that this programmer kit contains a power supply section in the board itself but in order to switch on that power supply, a source is required. Thus this is accomplished from the power supply board with an output of 12volts.

III. RESULT ANALYSIS

Results include the successful operation of the traffic control and monitoring system. The system contains three IR transmitter and IR receiver for traffic density measurement which are mounted on the either sides of roads respectively. The IR system gets activated whenever any vehicle passes on road between IR sensors. When one sensor will be ON at that time density will be less when two sensors will be ON at that traffic density is medium when all 3 sensor will be ON at that time density will be high. Microcontroller controls the IR system and counts number of vehicles passing on road. Based on different vehicles count, the microcontroller takes decision and updates the traffic light delays as a result.

Case 1:In the first case road one has density and green light appears for road 1

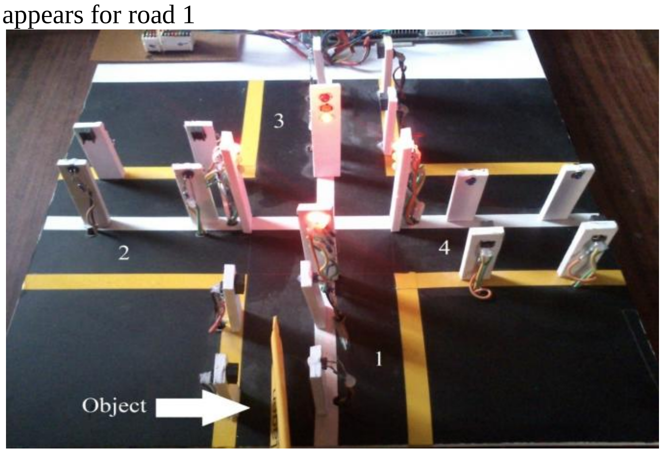

Figure Road 1 analysis

Case 2: In this case density occurs at road 2 and remaining road 1, road 3, road 4 has red lights.

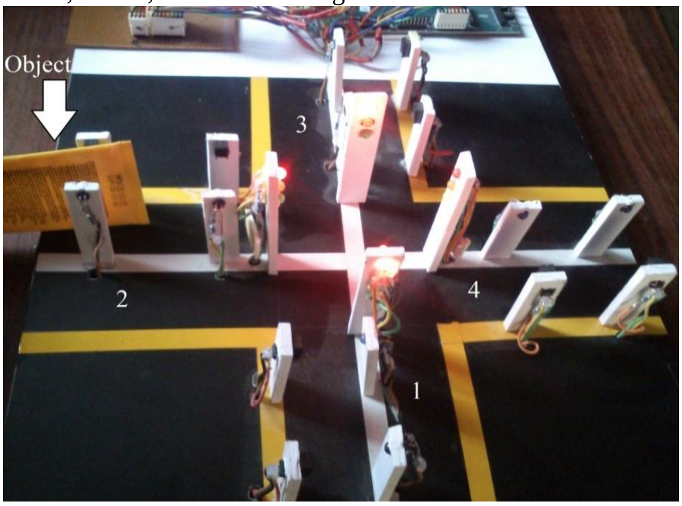

Figure Road 2 analysis

Case 3:In this case density occurs at road 3 and red light occurs for road1,road2,road4.

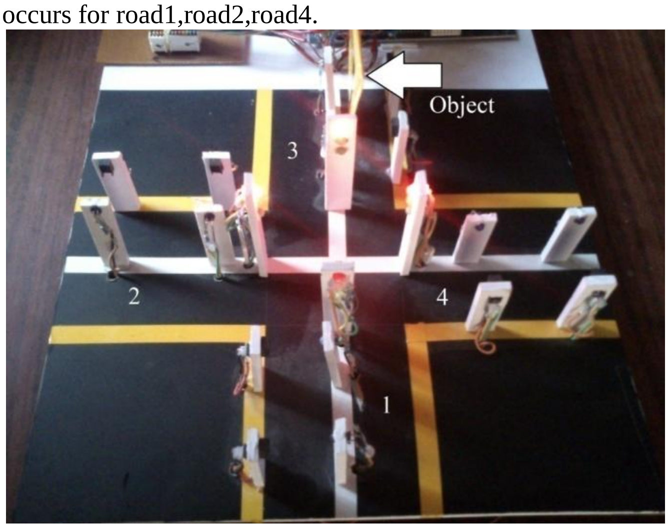

Figure Road 3 analysis

Case4:In this case density occurs at road 4 and red light occurs for road 1,road2 road3.

Figure Road 4 analysis

IV. CONCLUSIONS AND FUTURE SCOPE

In this paper we have studied the optimization of traffic light controller in a City using 8051 MICROCONTROLLER and microcontroller. The 8051 MICROCONTROLLER based traffic control system works on traffic related problems such as traffic jam; unreasonable latency time of stoppage of vehicle, emergency vehicles or forcibly passing, etc can be solved. By using this system configuration we try to reduce the possibilities of traffic jams, caused by traffic lights. Number of passing vehicle in the fixed time slot on the road decide the density range of traffics and on the basis of vehicle density calculation, microcontroller decide the traffic light delays.

BIBILIOGRAPHY

[1] Fundamentals of Micro processors and Micro computers -B.Ram
[2] Electronic Components -D.V. Prasad
[3] Wireless Communications - Theodore S .Rappaport
[4] Embedded System By Raj Kamal

Web References:
[1] http://www.national.com
[2] http://www.nxp.com
[3] http://www.microsoftsearch.com
[4] http://www.geocities.com
[5] http://www.Electronic projects.com

