THE DESIGN OF A SENSOR -BASED COMPUTERIZED TRAFFIC LIGHTS CONTROL SYSTEM.

BY

MAKUNDA, C. MALOBA

I56/7068/2001

A THESIS SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE.

KENYATTA UNIVERSITY.

NOVEMBER 2007

DECLARATION.

This thesis is my original work and has not been presented for a degree in any other university for any other a ward.

Makunda C. Maloba
I56/7068/2001.
Department of physics.
Kenyatta University.
P.o.box 43844,
Nairobi.
Kenya
Signature.........................Date 24/12/07

This thesis has been submitted for examination with our approval as University supervisors.

1) Dr. G. A. Ibitola
Department of physics.
Kenyatta University.
P. o. box 43844,
Nairobi.
Kenya
For Signature.........................Date.......................
08/01/2008

CHAIRMAN
PHYSICS DEPARTMENT

2) DR.J Odote.
Department of physics.
Kenyatta University.
P. o. box 43844,
Nairobi.
Kenya.
Signature.........................Date 07/01/2008

DEDICATION.

This thesis is dedicated to my parents Dixon and Maria for having brought me up to be what i am today.

ACKNOWLEDGEMENTS.

I take this opportunity to specifically thank all my lecturers in the department of physics especially my supervisors, Dr. Ibitola and Dr. Odote, who guided me through this research.

I also thank my colleagues Mwenda, Okinda, Mapesa, Rume and Musembi who were always there to encourage me and provided peer review to my work and helped improve the idea. Mrs. Nawiri for constantly giving me moral support. My sisters Lydia and Redempta for financial and moral support. My children Levy and Ivy for giving me time to study.

ABSTRACT

Automatic control systems permeate life in all advanced societies today. Such systems act as catalysts in promoting progress and development. They are integral components of any industrial society and are necessary for the production of standardized and quality goods. Suitable signal plans at each road intersections can be used to solve the problem of traffic flow. These plans could be fixed-time, time varying, or traffic actuated. Traffic control systems have evolved during the 20th century in response to the growth in automobile density. Hence, traffic systems must be organized and controlled carefully in order to operate efficiently and smoothly and meet the planned traffic control requirements. The advent of computers as control elements has advanced the technology in the automation of vehicle actuated traffic control.

In this research project we focused and designed an automatic vehicle actuated traffic control system for use at intersections on the road network.Astudy of the road network in Nairobi was done and the nature of traffic density. A sensor was then developed to detect the presence of vehicles and determine the length of the vehicle queues. A computer was used to a study of the road network in Nairobi was done and the nature of traffic density. A computer was used to process the signals from the sensors and allocate more time for traffic to flow on the street with longer queue.

A program was developed so that the computer allocated the timing sequence depending on the sensor status. The system can therefore be used to optimize traffic flow by ensuring that longer queues have more time for traffic to flow.

.

TABLE OF CONTENTS.

CHAPTER 1: INTRODUCTION

1.0 BACKGROUND.

Control is one category of knowledge engineering. A control system adaptively governs the overall behavior of a system. It must repeatedly interpret the current situation, predict the future, diagnose the causes of anticipated problems, formulate remedial plans and monitor their execution to ensure success. Problems addressed by the control system include air traffic control, antilock brakes in automobiles, and road traffic control [1].

The purpose of automatic control on a system is to produce a desired output when inputs to the system are changed. Considering for instance the flow of traffic in a typical city, and in the absence of a policeman and traffic lights, the traffic would be considered as uncontrolled even though there would be inherent control by factors such as the density of traffic, type of road network, number of intersections, the speed of the vehicles and the courtesy and training of the drivers [2]. If a policeman were placed at the intersection, the traffic would be said to be manually controlled. If replaced by traffic lights, it is said to be automatically controlled [2, 3].

The control theory developed in the 1950's may be categorized as conventional control theory. The introduction of the microcomputer as a control element has had enormous impact on the design of feedback control systems in that desired control specifications are met in the aircraft control for example, the development of digital control systems has greatly improved to incorporate motion sensors. All pilot commands are transmitted to the control surface actuators through electric wires by use of a fly-by wire control system [4, 5]. With the great developments in technology (computers and transducers) we have proposed a system that will ensure smooth traffic flow at the junctions.

1.1 CONTROL SYSTEMS.

Modern control can be classified into two major groups: open-loop control in which the signal controlling the output is independent of the output and closed loop control in which the signal controlling the output depends also on the output, hence it has a forward path and a feedback path [6, 7].

1.1.1 Open-loop or scheduling control

In open-loop control the signal controlling the output is independent of the output as shown in Fig. 1.1 Open loop controllers apply a single corrective effort when so commanded and assume that the desired results will be achieved [7].

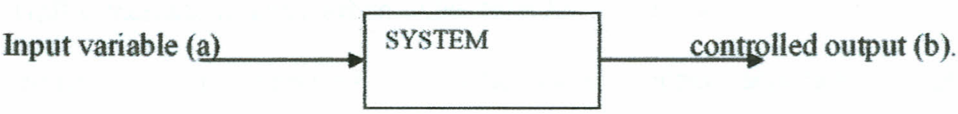


Fig.1.1: Block diagram of open-loop/scheduling control system.

On the basis of the knowledge about the system and of past experience, a prediction is made of what the input should be to give the desired output, the input is then adjusted accordingly, e.g. automatic toaster and the programmable washing machine. Such systems are unsatisfactory because any unexpected disturbances to the system can cause deviation in the output of the desired value, e.g. the cleanliness of the clothes may depend on a correct assessment of the amount of soap and length of washing cycles required [7].

1.1.2 Closed-loop or feedback control.

In this type of control, the system output is measured and compared with the desired value, the system continuously attempts to reduce the error between the two, e.g. thermostatic control and speed regulation by use of speed governors. Frequently the loop is closed through a human being at the control room. Fig.1.2. shows the general form of feedback control system [7, 8].

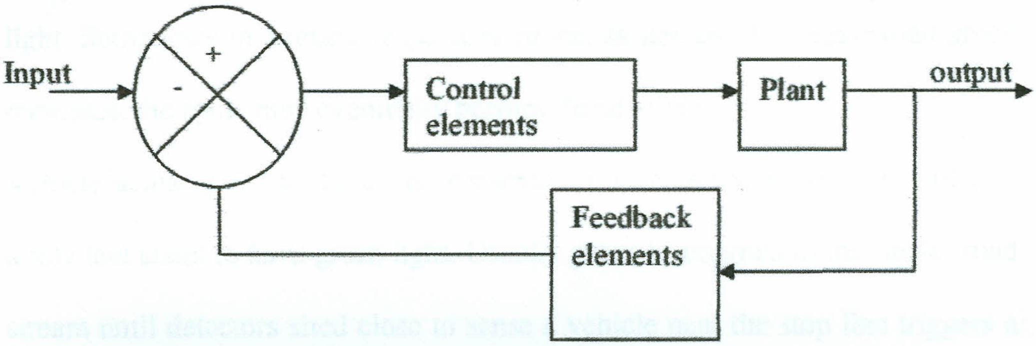


Fig 1.2: Block diagram for closed loop control.

1.2 TRAFFIC CONTROL.

Traffic increase in large urban cities calls for better management methods. New requirements are imposed on traffic lights control schemes. Congestion avoidance through better flow control is perhaps the most challenging goal for the traffic engineer. This demands fast and efficient traffic responsive systems, which can only be achieved by a closed-loop approach [11].

Traffic control can be achieved by use of sequential circuits in which the output is a function of both present input and the past outputs [8]. To facilitate the systematic design of sequential circuits, it is necessary to obtain a clear picture of the problem at hand. The most widely known sequence with variable delays is the traffic lights sequence. In its simple form, it consists of two sets of lights to govern a simple cross roads junction. But more complex systems exist which govern many other types of junctions [8, 9].

1.2.1 Modes of traffic control.

Traffic control systems are operated in three different modes: fixed-time, vehicle actuated, or demand responsive:

Fixed –time mode: In this mode, both roads are given equal times for green light. Sometimes in demand responsive mode, as demand for minor road green increases, the cycle may eventually become fixed –time.

Vehicle actuated mode: here, the presence of a car activates the controller to allow that street to have green light. Usually green is assigned to the 'main' road stream until detectors sited close to sense a vehicle near the stop line triggers a demand for 'minor' road green.

Demand responsive mode: this mode considers the different streets present and allocates more time to the street with more vehicles. It has sensors placed at intervals along the road to detect the length of the queue of vehicles awaiting green light. The individual signals optimization programs specify green for each state of the junctions in the network [8, 9, 10].

1.3 COMPONENTS OF AUTOMATED TRAFFIC CONTROL.

Various things must be considered in designing an automatic traffic lights control. The main components to be put in place include transducers (sensors) to detect the presence of traffic, the computer to control the lights and the relevant programming language.

1.3.1 Transducers.

Many of the microcomputers in service are used to control a wide range of process machines or appliances. All of these applications involve close and continuous interaction between the microcomputer and its environment. Since the nature of the signals that the microcomputer can emit and receive is clearly specified, information from the outside world needs a substantial degree of processing before it can be accepted by the processor or vice versa [12, 13].

A transducer must be matched to the system with which it is designed to work. There must be maximum voltage transfer. In all cases, the impedance of the transducer is a critical factor. Loop-detectors have been used in many detection situations for traffic control. No single sensor system meets the requirement of all applications. Many factors must be considered in the selection of a sensor such as accuracy, precision, linearity, and temperature stability specifications [13, 14]. To measure non-electrical quantities, they must be detected by a sensor and then converted into electrical quantity, e.g. voltage, which after conditioning can be employed for control purposes.

1.3.2 Computers.

Computers are generally taking over the majority of control functions in the industrialized world. They perform their functions by sending binary codes to interface circuits that can turn various actuators on and off to control the devices to which they are connected [15]. Microcomputers are used for control applications in every facet of the transport industry. They are used to control the operations of the vehicles themselves – engines, brakes, and air surfaces, and can also be used in wide area applications such as traffic lights control [16].

The main function of the microcomputers is to fetch, decode and execute instructions resident in the memory and the results obtained are used to control the sequence and time for light [17]. A personal computer has therefore been used in this research to control the flow by processing the signals from the sensors and controlling the lights output to meet the waiting car proceed and hence optimize traffic flow [16,17,18].

Interfacing enables a computer to interact with the outside world in a real-time mode [19]. Ports are the doors at the CPU system for receiving and providing data. Hence, most of data that goes out from or comes into the CPU system is routed through the ports. The basic unit of the registers in a port is the D flip-flop.

1.3.2.1 Classification of ports.

Digital computer interface ports can be classified as either parallel / serial or as input/output ports. In parallel interface ports, n bits are transferred through n wires in one cycle. While in serial interface one wire is used to transfer n bits in n cycles [20, 21]. In computers, parallel ports are more commonly used to connect to peripherals such as Visual display units (VDU), printers' and external circuits, while serial ports are used for data transmission to communications networks. When interfacing any device to the parallel port, a buffer is required to cater for current sourcing and sinking. Most parallel ports can source and sink around 12mA to 15mA [22].

Depending on the direction of information flow the ports can be classified as I/O. Input ports are used for transmitting information to the CPU, e.g., the keyboard is connected to an input port. Output ports are used for transmitting information from the central processing unit (CPU) to the outlying device e.g., a printer is connected to an output port [21, 22].

1.3.2.2 Computer programming languages.

A digital computer accepts instructions in binary codes only. However, the instructions can be written indifferent ways and the converted to binary codes for providing to the computer for execution. Hence many programming languages have been developed which can be divided into three main categories: Machine language, assembly language and high level languages [21, 23]. Machine language (ML) refers to a program is written in binary digits 0's and 1's only it is said to be in machine language. In the initial periods of digital computer development, programs were written in ML; it was a tedious process as it was depending on chains of O's and 1's.It was difficult to memorize all the instructions codes and chances of making mistakes in writing 0's and 1's were also very high [23, 24]. This problem was alleviated by the introduction of the Assembly language (AL) where Standard binary codes for computer instructions are substituted by 2, 3, or 4, letter words so that memorizing these codes is easier. These letter codes are known as mnemonics (aid to memory) and are usually initials or shortened forms of English language words. These mnemonics are used to write computer programs make programming more convenient. Assembly language also uses symbolic referencing of storage addresses. The programmer can assign the data items to the variable [24]. Assembly language expects the programmer to be conversant with the computer hardware. To eliminate this requirement, the third category High-level languages (HLL) have been developed. HLL language programming instructions are easy to write and memorize. Each instruction in the HLL may contain one or many ML operations. An interpreter program or compiler is used to translate a HLL program to machine language codes. Some popular high level languages include C++, Basic, UNIX among others [24, 25].

CHAPTER 2: LITERATURE REVIEW.

2.1. TRAFFIC LIGHTS CONTROL.

Traffic lights control strategies have developed greatly through research to march the growth in science and technology. Many systems have been designed by different researchers. Traffic control systems can either be analogue, digital, and microprocessor-based or computer- based. This chapter looks at some of these systems.

2.1.1. Analogue control.

Many research works have been done in the field of traffic lights control. In the early days, a policeman would be placed at the crossroads to control the vehicles flow to the different directions. With the introduction of traffic lights, the lights followed affixed cycle of events, allowing equal periods for the flow of traffic on either side of the crossing roads. Such lights used a simple multivibrator circuit to generate waveforms to apply this to a binary counter of sufficient capacity that it returns to zero at the end of the cycle of events [19]. However, due to increase in the number of vehicles and technological advances there was need to develop more reliable traffic lights control systems [25]. Hence, traffic lights are no longer designed by analogue circuits.

2.1.2. Digital control.

With growth in technology, digital methods of traffic control were further developed. For example besides allowing vehicles to move in a given direction there was need to include pedestrian push button, which if pressed, the lights turn red at the end of the current cycle, stay red for two minutes then resume operation as if the interruption had not occurred. Hence, the controller should act as a synchronous sequential circuit. Such a circuit may consist of a clock, JK flip-flop, RS flip-flop and the lights among other components as shown in Fig.2.1 [26].

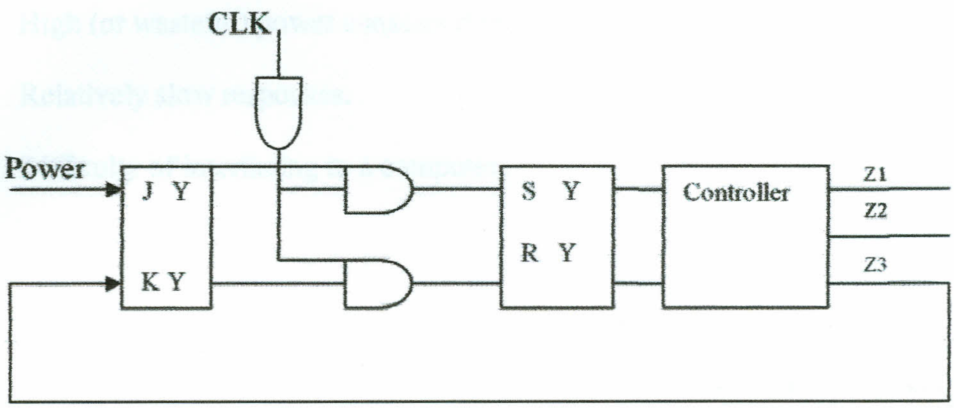


Fig.2.1: A simple traffic controller.

With the increase of traffic in urban cities more systems were developed. Work was done by Bahhadur Singh and Subbha Singh using IC 7479 decade counter in the biquinary mode to improve the fixed timing mode in which time allowed for RED, YELLOW and GREEN was in the ratio 4:1:5 [27]. Also lights of different colors on opposite roads were wired in series as they would be ON or OFF at the same times [28]. Research has also been done by Rob Paisley (2002) using IC 74145 decoder. He assumes that traffic on either side of the roads does not interact at all (it is a simple crossroad). He points out clearly that these circuits cannot cover every situation on every layout [29]. Generally, digital control systems take the form shown in Fig 2.2.

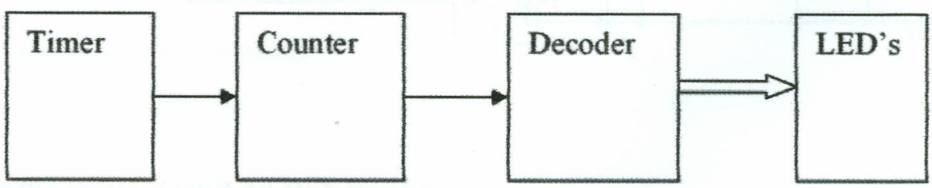


Fig. 2.2: The block diagram for digital traffic lights control.

An example of a traffic light controller used in urban cities is shown in the fig.2.3. It has been observed that this type of traffic lights controller has the following drawbacks:

i. High (or wasteful) power consumption.

ii. Relatively slow responses.

iii. Difficulty of interfacing to a computer.

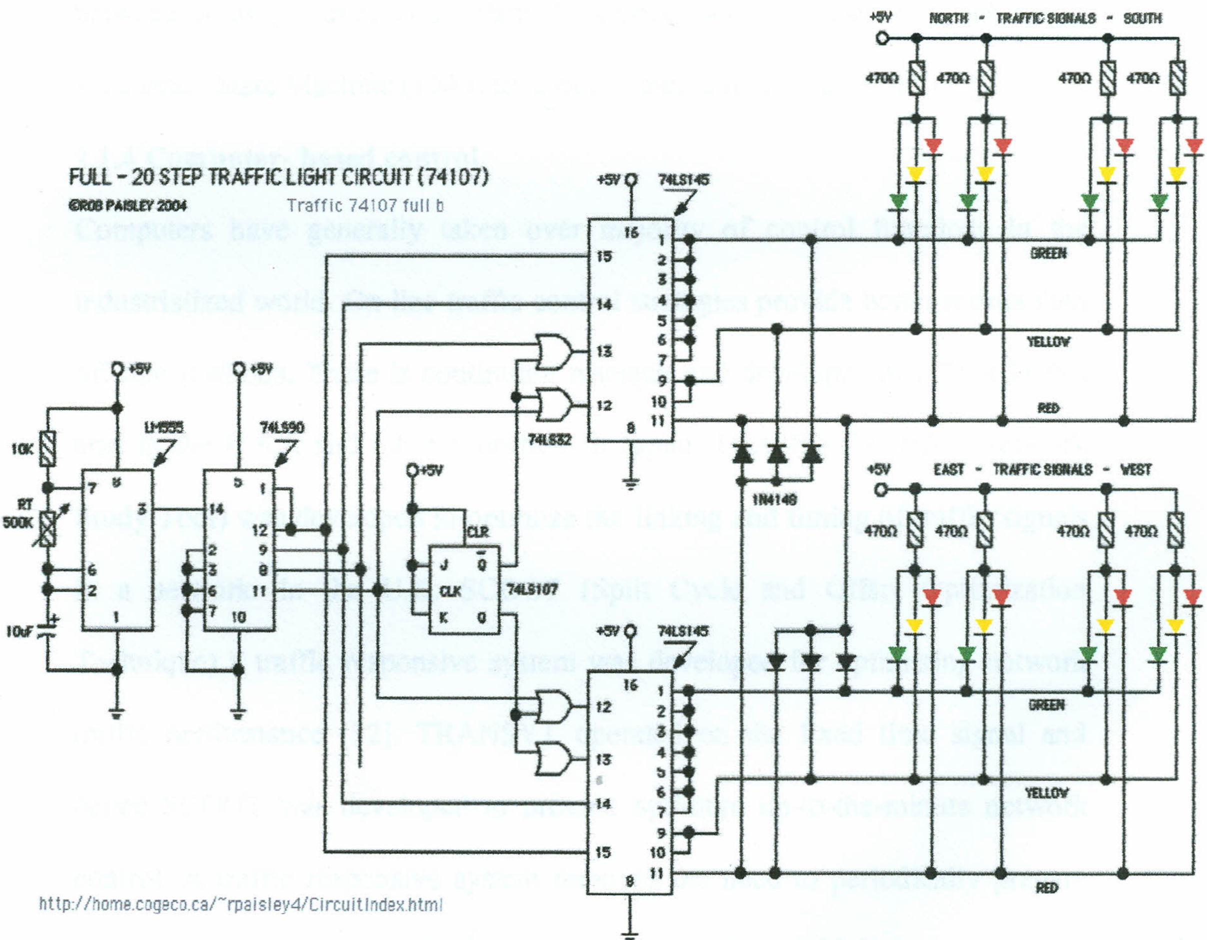


Fig. 2.3: Traffic lights circuit by Rob Paisley.

2.1.3 Microprocessor –based control.

James Fernstron& Thomas Weinich did research for the intersection of Rt.1 & 138, in Electronics projects ELE 447 in which they incorporated the use of sensors. They used a Programable LogicArray (PLA) to determine the next state of events. Using a Microprocessor, they were able to process input from sensors and actuate the relevant lights using the PLA [30].

Research was done by Fall 2002 on "a traffic light controller" using aspects of both sequential and combinational logic. His study is based on an intersection between a busy street and relatively empty one. He uses a synchronous Frequency State Machine (FSM) to control other parts of the circuit [31].

2.1.4 Computer- based control.

Computers have generally taken over majority of control functions in the industrialized world. On line traffic control strategies provide better results than off-line methods. There is continuing research and development efforts in this area in the U.S.A and other countries. In Japan, TRANSYT (Traffic Network Study Tool) was developed to optimize the linking and timing of traffic signals in a network. In the U.K. SCOOT (Spilt Cycle and Offset Optimization Technique) a traffic responsive system was developed for optimizing network traffic performance [32]. TRANSYT operated on the fixed time signal and hence SCOOT was developed to provide optimum up-to-the-minute network control. A traffic responsive system removes the need to periodically prepare new fixed time plans, which if not undertaken leads to a high degree of delay [32].

The disadvantages of the fixed time modes include:

1) The systems are not vehicle actuated and hence maintain the fixed cycle of events with or without need resulting in unnecessary delays.

2) The delay caused by these systems lead to increased costs of travel in both time and money- fuel consumption is higher and pollution is increased.

Therefore there is need to create a more reliable control system that is real-time cheap and cost effective. Many people have ideas about how to make traffic conditions better, but except in the simplest cases, they often disagree about what they mean by better, and even if they agree about this, they still disagree on how to achieve it or how much it is worth spending to do so. However, traffic systems vary widely over the road systems and they are often far from satisfactory [28,29].

The advent of microcomputers as an aid to control systems has provided motivation to the traffic control engineer. The implementation of efficient microcomputer based systems requires that the engineer is aware of the capabilities of the microcomputer technology and the possibility of achieving the required results [33, 34].

Computer based systems have replaced the conventional analogue and digital systems because of the following advantages [35, 36]:

a. Decision-making and logic capability of the systems components.
b. Increased flexibility of the control programs.
c. The programs can be modified to accommodate design changes or adaptive performance without variations in the hardware.
d. They are able to monitor the situation and display the results.
e. Storage of information for future use or reference.

Computer-based traffic control systems have been developed in the highly developed countries. In this research project, we have used a computer to analyze the sensors output and control the lights accordingly.

2.2 Mathematical model of traffic flow.

Research on traffic flow has been widely carried out by mathematicians. Traffic flow theories seek to describe in a precise mathematical way the interactions among vehicle, drivers and the infrastructure. These theories are an indispensable element of all traffic models being used in the designed and operation of streets and highways. In the fluid flow analogy the traffic stream is treated as a one- dimensional compressible fluid. The flow observes the equation of continuity and the equation of state. It has a one to one relationship between the speed and density [37].

Renyi [38] describes two mathematical models of the flow of traffic on a divided highway by supposing that a vehicle enters the highway at the same entrance so that the instants at which a vehicle enters the highway form a homogeneous Poisson process with density λ. He assumes that there are no junctions or exits i.e. the highway extents in one direction to infinity [38].

Another simple mathematical model was proposed by a Japanese physicist. Bando[39]. He considered a single lane road without any cross-section. The Bando model assumes that the acceleration of the vehicle is determined by the equation

$$X_n = 1/T[V_{OP}(\Delta X_n) - X_n]$$

Where Xn represents the vehicle positions and n is the vehicle index such that

$$\Delta Xn = X_{n+1} - X_n$$

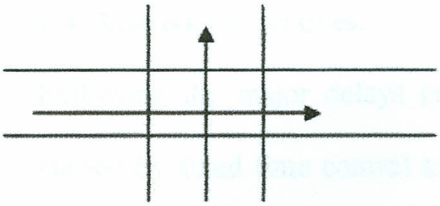
V_{OP} is the optimal velocity for the given vehivle spacing as vehicles slow down or accelerate up so that, $V_{OP}(\Delta X) = \tan h\,(\delta x - x_{safe} - 2.) + \tanh 2$ [39].

Stochastic models have been successfully in many different interdisciplinary problems e.g. modeling of traffic flow using cellular automata (CA). Cellular automata models are discrete in space, time and state variables. The street is divided into cells of length 7.5m (car length + distance to preceding car). The cells can be occupied by one car or remain empty [40].

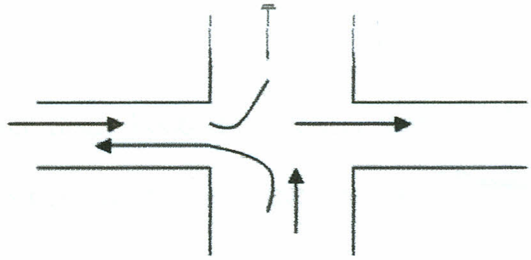
It follows that if the density is high enough, a car that brakes at one point has the probability to break again to avoid a crash. This produces a chain reaction and hence a spontaneous jam. For small densities, the flow is proportional to the density because there is almost no interaction between the cars. The simulation shows the typical spontaneous jams [40].

2.3 Road networks.

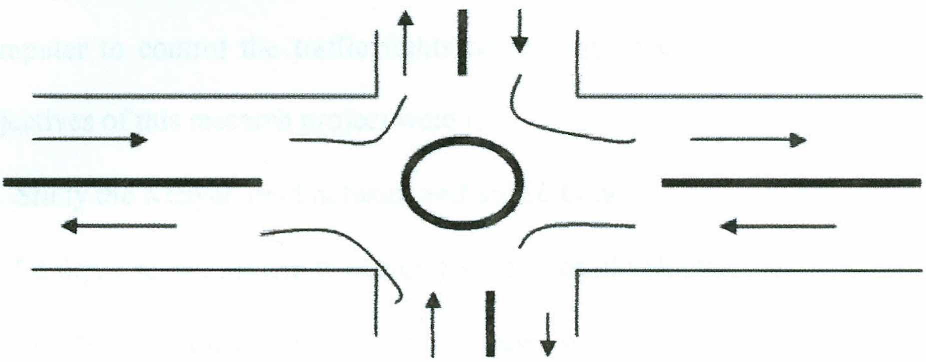
Traffic control systems vary widely depending on the nature of the road network to which it is applied. In the developed countries, the road networks are highly developed such that once you get onto a lane leading out of town for example you have to leave town. In Kenya, the different types of network that we have are shown in fig.2.4. Traffic controllers are developed with a particular layout in mind and cannot be applicable to all other layouts.



a. Simple crossroad



b. interactive crossroad.



c Round about

Fig. 2.4: Types of road intersections

Based on the above, we have developed a controller for the round about on our roads. The controller operates according to the software developed by the traffic engineer. We have developed the software for controlling the traffic lights in C++ high level programming language. The software purely controls the delay between the lights. The void functions (subroutines) are called upon depending on the active sensor that calls a certain timing sequence.

We have therefore developed a traffic controller based on the Kenyan road network. The advantages of the developed system are:

a. It ensures smooth flow without causing much delay.

b. It results in reduction of journey time spent by motorists on the way.

c. Reduction in fuel costs in terms of fuel consumption.

d. The system allows for flexibility since the output is software oriented.

2.4. Research objectives.

Following the major delays experience on our roads because of traffic jam caused by fixed time control techniques, the general objective of this research project was to design and construct a working prototype using a personal computer to control the traffic lights depending on the sensors. The specific objectives of this research project were to:

- Study the Kenyan road network and traffic flow.
- Design a sensor system to detect the presence of vehicles.
- Interface the data input from the sensors with a PC to relay to the output (make the proper signal control.).
- Develop and implement software for monitoring the signals from the sensors and controlling the traffic lights.
- Design a suitable traffic control system adaptable on the Kenyan roads.

2.5. Rationale/justification of the study.

Road traffic monitoring has become so sophisticated with new developments in t technology. Computer control and traffic sensor technology has made several forms of automatic vehicle classification possible for traffic monitoring and control [41]. Vehicle detection is a basic element in real-time traffic control systems. It is the ability to sense and indicate the presence and passage of a vehicle. New development in sensor technology has led to the designs of several transducers. Photo detectors are a predominant means of vehicle detection due to their simplicity in design [42].

During periods of low traffic such as at night and early morning, the system may monitor the presence of a car at an intersection and immediately switch the lights to let it pass. During periods of high-density traffic, the system allocates more time for the longer traffic queues than the shorter ones. Many vehicle actuated systems exist but may not be directly applicable on the Kenyan network.Applicability depends on the nature of the road network,traffic rules and other factors such as the number of lanes & technological developments of the given country.

A visit to the Nairobi city council traffic lights control unit at the city hall revealed that the lights on Kenyan streets are controlled by the fixed time mode determined by the city engineer on a demand responsive schedule. The control room was vandalized many years ago and they are forced to fix controllers at every juction (i.e ST 800 or smartic S7 GSM controllers.). These controllers are preprogrammed for the specific time requirements of the given juction after a traffic flow survey is carried out. There are manuals on the use of the controllers and the council technicians are trained by the suplier before they fix them. The lights follow the same sequence irrespective of whether the vehicles are many or not. Sometimes the traffic police are forced to control the lights manually especially when an accident takes place.

The proposed system will be designed with the Kenyan road network in mind. We hope to reduce the time spent by motorists at intersections during high traffic periods. The sensor will triger the controller to give more time to the street that requires traffic to flow. The project is a prototype using LED's to demonstrate the ability to control lights using a computer. We have developed the required software for the control purpose. A computer in the control panel will be used to monitor the functioning of the entire system.

CHAPTER 3: RESEARCH METHODOLOGY.

3.1 GENERAL OVERVIEW.

Driving on Kenyan roads has become a nightmare, with the high rate of growth in automobile travel in the recent years. Traffic control is still done using mediocre traffic lights that are not vehicle actuated [43]. This results in a lot of time spent by motorists on the roads, which in turn increases the costs of travel and air pollution. Sometimes motorists travel better with the aid of policemen than with the use of traffic lights (on 24th July 2003, the city mayor ordered the lights that had been installed at the Westland's roundabout to be switched off as they had caused more harm than good) [44]. Since automobile travel has greatly increased there is need to study the Kenyan road network and come up with proper vehicle actuated traffic controllers.

This research project focused on developing the traffic controller for the Nairobi roads. Since the existing controllers are designed with a specific network in mind, a proper study of the intersections was done, the number of lights properly designed and a way of sensing the presence of vehicles developed. Hence the software was developed to control the time allowed for the delays on either side of the road. The research methodology adopted for this research project was as shown in Fig 3.1.

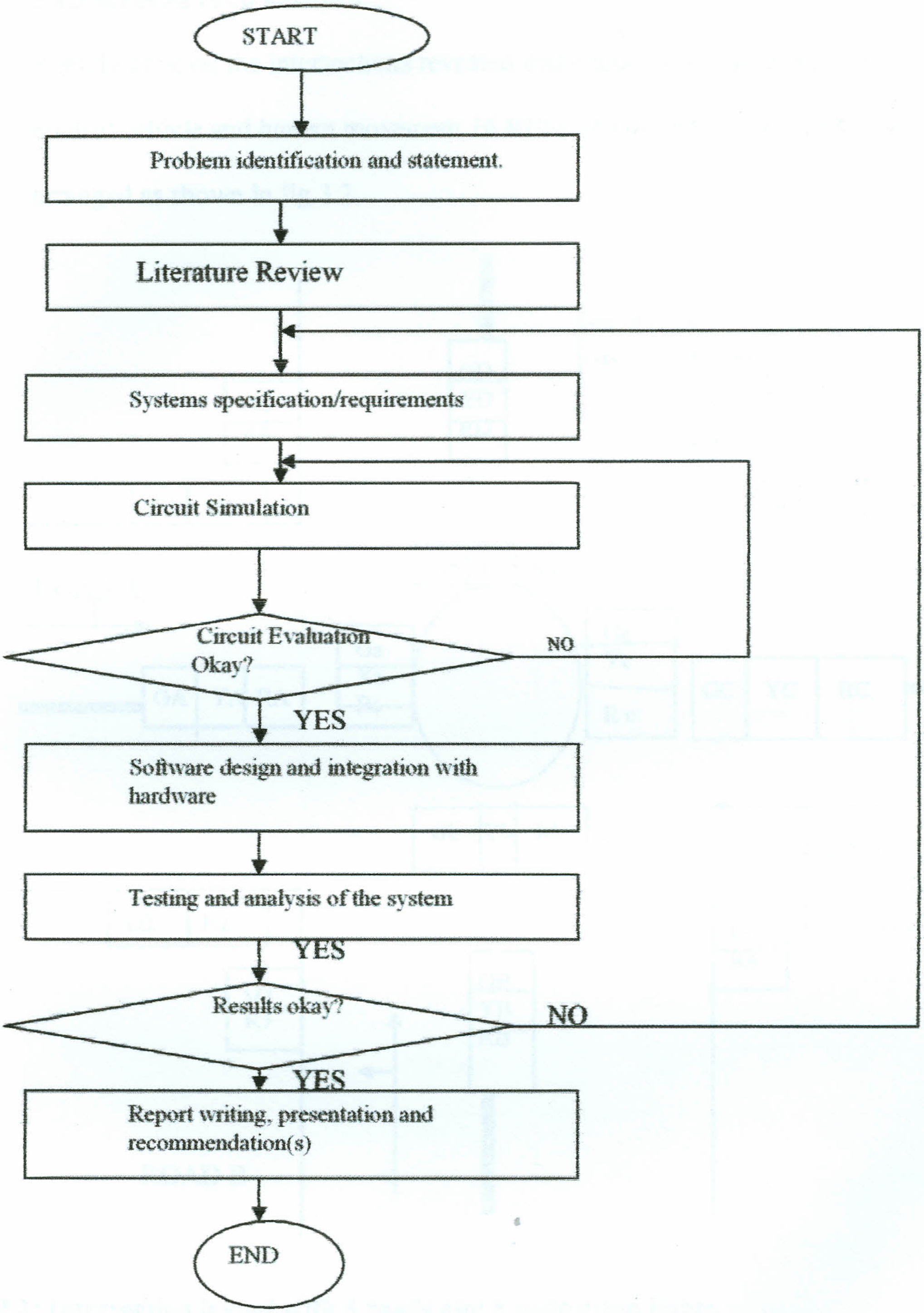


Fig. 3.1: The Research methodology flow chart.

3.1.1 Kenyan road network.

A study done on the intersections revealed that a total of 40 lights are required to control vehicle and human movement 16 RED, 16 GREEN and 8 YELLOW arranged as shown in fig.3.2.

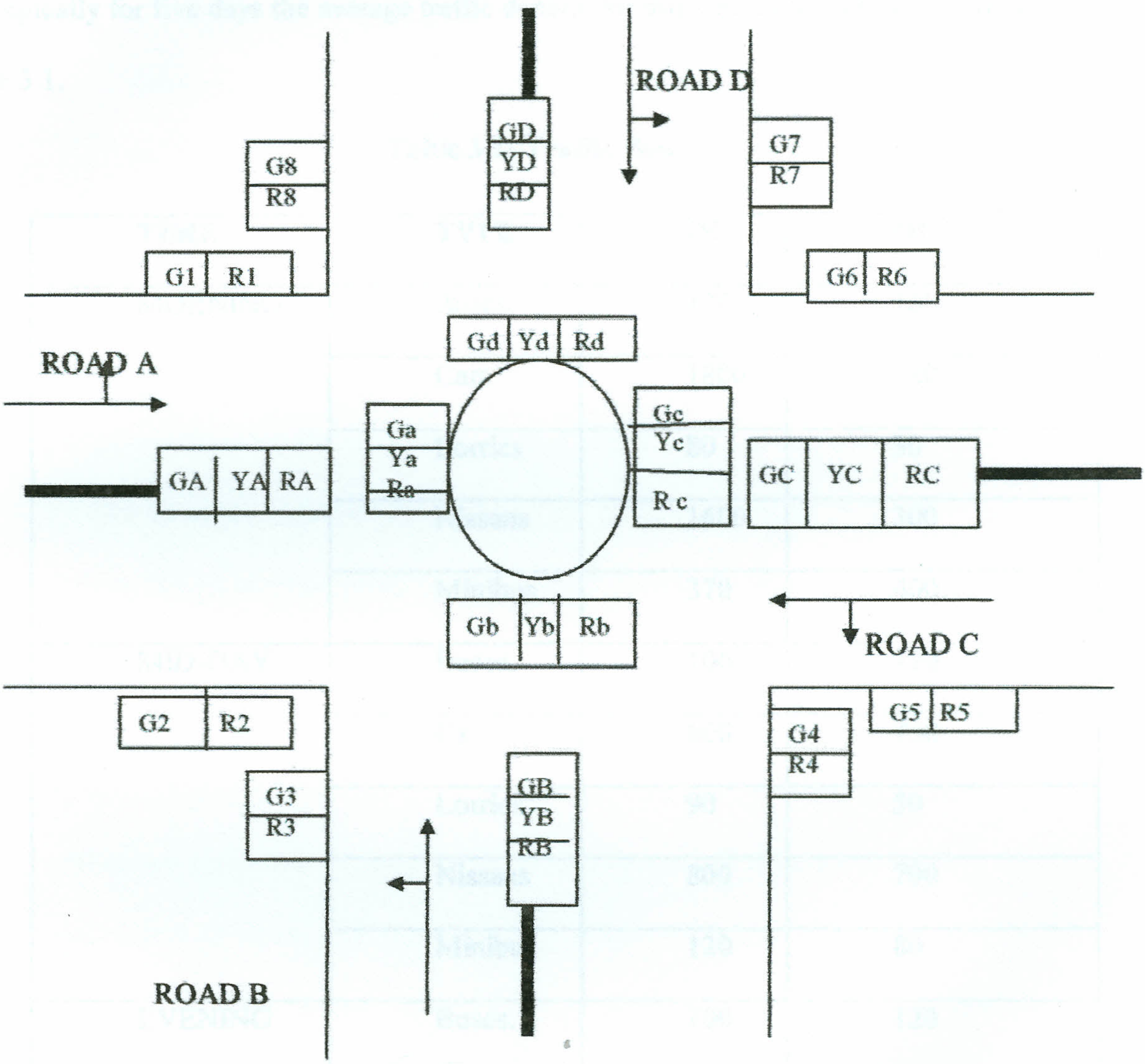


Fig.3.2: Intersection layout with 4 roads and 8 pedestrian lights.

The lighting sequence for the main roads is as shown *appendix1* such that only one main road has green light for the vehicles to move at any time period.

A survey carried out between August 2004 and December 2004 at the Kenyatta Avenue /Uhuru highway rounder about revealed that the traffic density varies according to the time of the day. In the morning there is heavy traffic towards the city center as opposed to the evening when heavier flow is away from the city center. After counting strategically for five days the average traffic density for different times was as shown in table 3.1.

Table 3.1: Traffic flow.

TIME	TYPE	IN	OUT
MORNING	Buses.	170	160
	Cars	1800	220
	Lorries	80	30
	Nissans	1600	300
	Minibus	370	400
MID-DAY	Buses.	100	120
	Cars	660	750
	Lorries	90	50
	Nissans	800	700
	Minibus	120	80
EVENING	Buses.	100	120
	Cars	200	1250
	Lorries	50	180
	Nissans	300	1300
	Minibus	200	180

A summarized table clearly shows the difference in the traffic density (i.e. vehicles per hour) at the different times of the day.

Table 3.2: Traffic flow summary Nairobi city centre.

	IN	OUT
MORNING	4020	1010
MIDDAY	1770	1700
EVENING	850	3130
TOTALS	6040	5840

Following these results we concluded that traffic lights at our intersections can be well managed if proper timing plans are designed to allow traffic flow in to the city during morning hours to have greater delays as compared to the out flow at the same time. During mid-day hours the timing should be the same since the traffic seems to be balanced. In the evening traffic flow out of the city should be allowed greater delays than those to the city. Properly managed such a system will ensure optimization of the traffic reduce delays and ensure smooth flow for motorists at all times.

The project is a prototype using LED's to represent the lights. When electric lamps replace LEDs in real life situation, current amplifiers are needed to drive them. The driver circuit is implemented using IC's that are compatible with the TTL logic circuit requirements as described in Section 4.1. The software is developed using visual C++ high-level language.

3.1.2 Sequential circuit design.

The flow chart in Fig. 3.3 shows the lighting sequence of the traffic lights. To start the system, all the roads have the red lights ON.

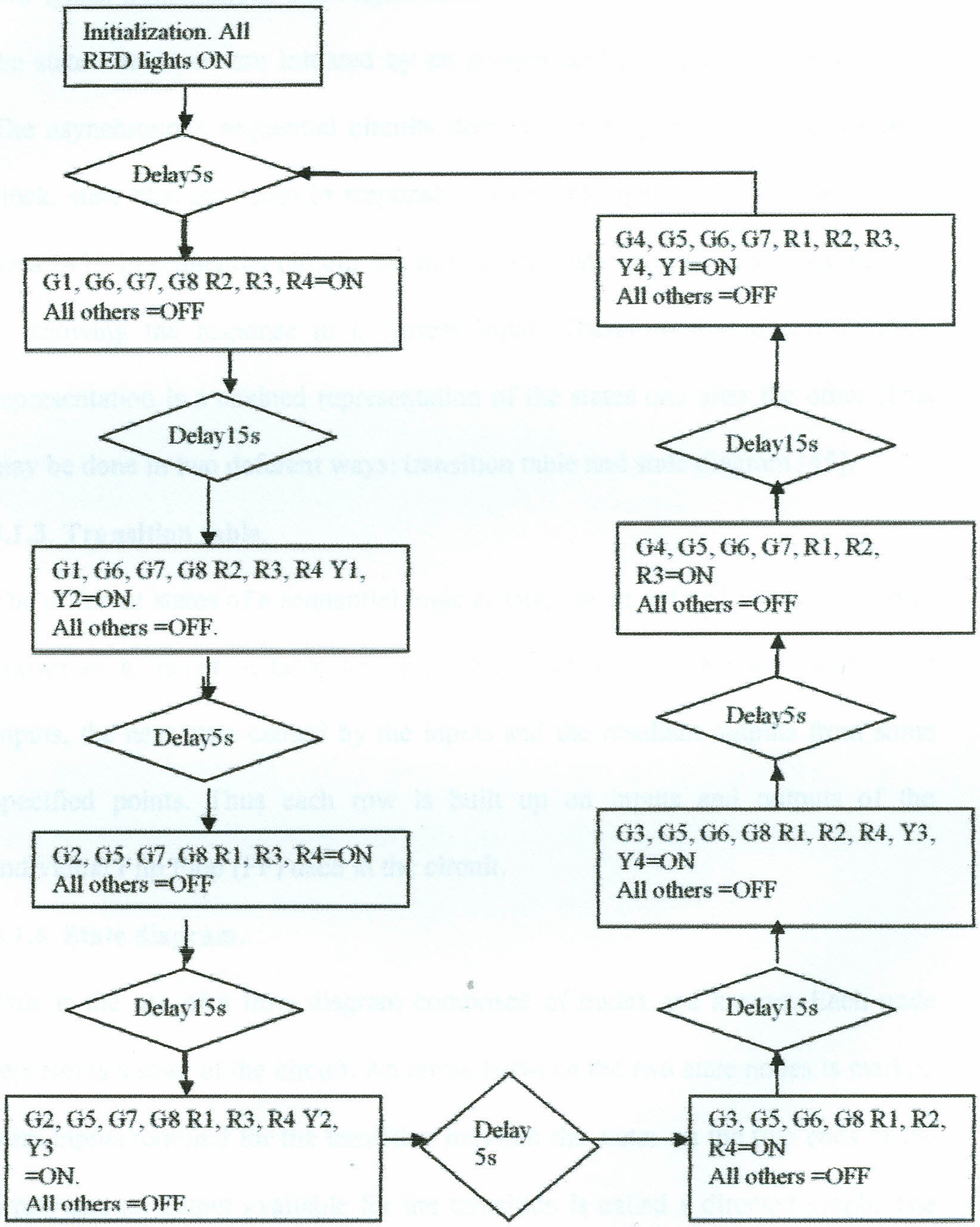


Fig.3.3: Lighting sequence flow chart.

The most widely known sequence with variable delays is the traffic lights control sequence. It employs a fully closed loop control mechanism. Sequential circuits possess true memory capability and can issue time dependent sequences of logic signals controlled by the past information. Sequential circuits, are of two types: Synchronous and Asynchronous. In synchronous sequential circuits the state transitions are initiated by an independent periodic signal (CLOCK). The asynchronous sequential circuits does not employ the use of an external clock, state changes occur in response to signal changes on primary input data lines [45]. In sequential circuits the immediate past condition is always relevant in knowing the response to a current input. Therefore the sequential logic representation is a chained representation of the states one after the other .This may be done in two deferent ways: transition table and state diagram [45].

3.1.3. Transition table.

The different states of a sequential logic circuit can be defined in a tabular form known as a transition table or state table. Each row shows possible present inputs, the next state caused by the inputs and the resultant outputs from some specified points. Thus each row is built up on inputs and outputs of the individual Flip Flop (FF) used in the circuit.

3.1.4. State diagram.

This is the use of a flow diagram composed of nodes and arrows. Each node represents a state of the circuit. An arrow between the two state nodes is marked with inputs required for the transition between the states on the two ends of the arrow and an output available for the transition is called a directed graph. The number of nodes in a state diagram represents the number of all possible states of the circuit such that n $FF=2^n$.

The state diagram for the green light on the four main streets would be as shown in Fig.3.4.

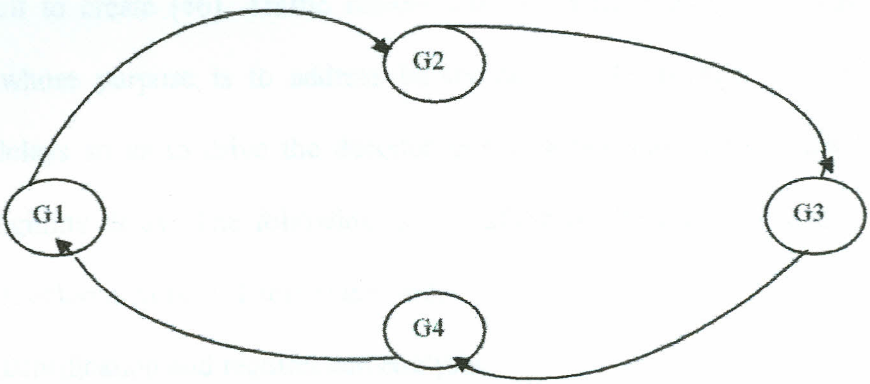


Fig.3.4: Traffic lights state diagram.

In this research project, however, we have used the state table to determine the next state of events. The state table for the traffic lights at a Kenyan road intersection is as shown in *appendix2.*

By considering the lights that are on and OFF at the same time, i.e., have the same output for the entire time, it can be observed that YA= Ya, GA=G8, YB= Yb, GB=G2, YC=Yc, GC=G4, YD=Yd, GD=G6, G1=Ga, G3= Gb, the table reduces to that shown in appendix 3, the reduced truth table for the traffic lights.

3.2 The hardware system design.

The traffic controller was made from three different circuits in this research project namely, the traffic control circuit, sensor circuit and the interface circuit. Each of these circuits was set up on its own and tested for the required outputs. Finally, the three were merged together to produce one traffic controller using a specified program. The hardware system design is described in chapter 4.

3.3. Systems software design.

Traffic lights are usually located at rounder about of busy road junctions near or within the city center where the traffic density is high and requires proper control techniques. Software oriented techniques are easier to modify though a bit difficult to create [46]. Traffic control can be easily achieved through a program whose purpose is to address binary data to the parallel port with suitable delays so as to drive the decoder in the driven circuit to obtain the relevant lighting delay. The following is an outline of the different stages in software development [47]. These stages are:

a) Problem identification and requirement analysis.

b) Design of the software algorithm.

c) Coding i.e. implementation of the software into a suitable language.

d) Software testing.

3.3.1 Problem identification and requirement analysis.

Identification of a problem is the first stage of the software lifecycle process. The definition phase is designed to define overall project objectives that must be satisfied, determine project feasibility, develop a strategy for accomplishing the objectives and establish the project resource costs & schedule.

The definition phase begins with the evaluation of the overall systems requirements (hardware and software.). Systems evaluation defines the interrelationships that must be understood prior to software planning- the first step in the definition of the software. The problem definition is done without consideration of the language to be used [48]. Software requirements detail the necessary computer systems, related hardware and software tools required to solve the problem.

In this research project the problem entails developing a lighting sequence that will ensure optimum traffic flow at different times of the day to cater for the heavy traffic in and out of town. This requires software capable of changing the lighting sequence delay such that different streets are allowed more time for the traffic to flow. An output port must be available for the program to be effective.

3.3.2 Design of the software algorithm.

A sequence of precise instructions that leads to a solution is called an algorithm. A computer program is simply an algorithm expressed in a language that the computer can understand. The basic elements of an algorithm are- information specification, input procedures, manipulations (actions and calculations) and output procedures.

Information specification encompasses data control and status type depending on how it is used. This information can be in the form of a circuit, a voltage pressure, paper numbers names or codes. The input procedure depends on the process and its implementation. It may involve normal movement of control of a transducer data input; the manipulative operation can involve the numerical solution of a differential equation for control or the updating of an instruments data file. The output procedure translates the output into a proper output form such as feedback control transfer [49]. The algorithm for the traffic lights control is as follows.

i. To begin set all the lights to RED.

ii. Check the active sensor i.e. LED ON.

iii. If both sensors on road 1 and 2 are on, then select the sequence for mid-day hours. Else,

iv. If only sensor 1 is ON then select the sequence for morning hours to allow more time for in-flow. Else

v. If only sensor 2 is ON then select sequence for evening hours to allow more time for out-flow. Else

vi. If no sensor is ON then select the sequence for mid-day during the day and that for night after mid-night.

This may be represented in form of a flow chart as shown in Fig. 3.5 below.

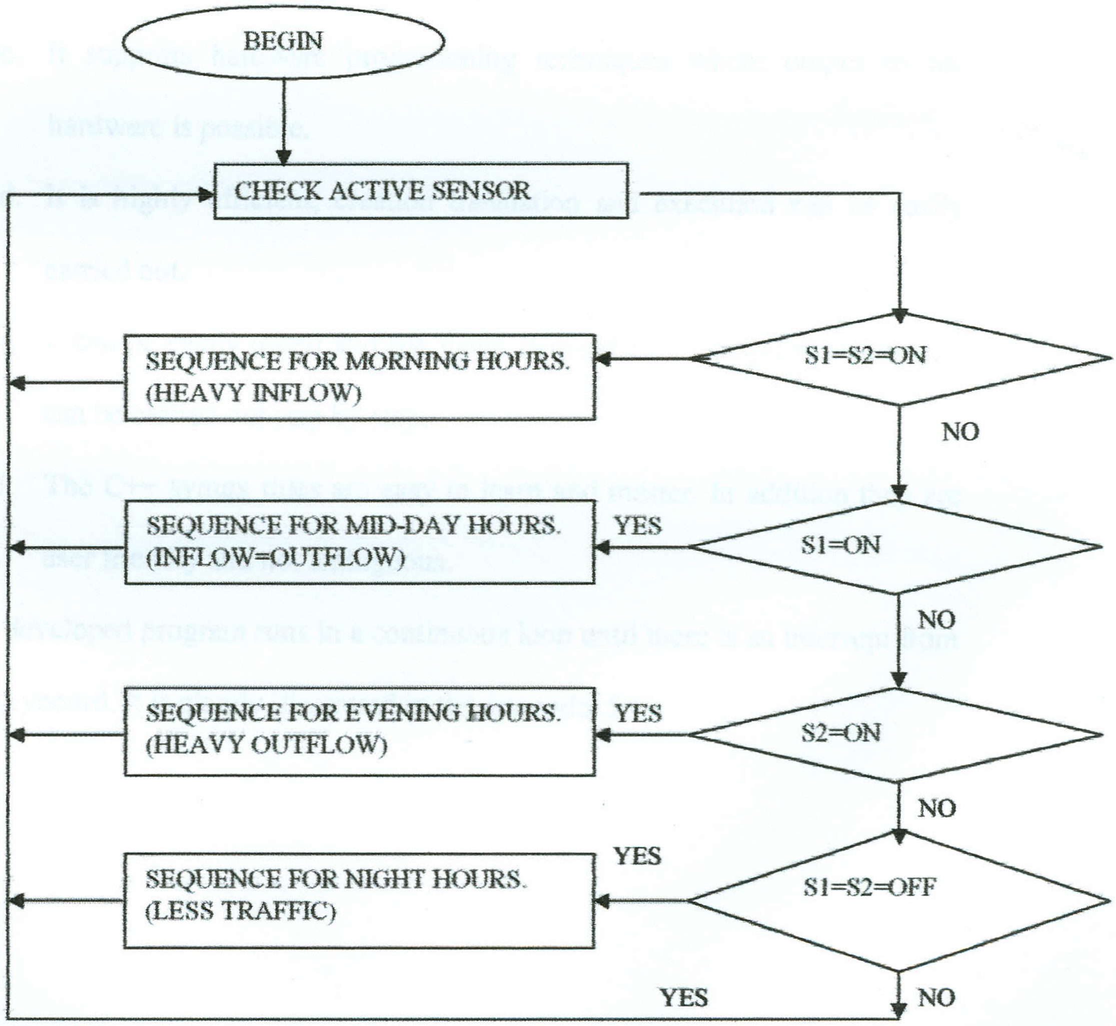


Fig.3.5: Traffic lights controller Software flow chart

3.3.3 Software implementation.

An algorithm that has been realized into computer software is called a computer program. The above algorithm is therefore translated into a C++ programming language [*appendix 5*] such that the streets with heavier traffic density receive more time for traffic to flow. The C++ programming language is chosen because [50]:

a. It has a clear program structure. One can easily follow the individual steps in the program.

b. It is easier to extend and edit using the \\ operator.

c. It supports hardware programming techniques where output to the hardware is possible.

d. It is highly efficient; creation translation and execution can be easily carried out.

e. It can be easily tested and run using Borland C++ compiler. Debugging can be carried out step by step.

f. The C++ syntax rules are easy to learn and master. In addition they are user friendly and not ambiguous.

The developed program runs in a continuous loop until there is an interrupt from the keyboard. It is clearly illustrated in the *appendix 5*.

3.3.4 Software testing.

Testing demonstrates the presence of errors and not their absence. Evaluation at all phases of the software life cycle must be an integral part of any software engineering approach. Debugging or software error correction must also proceed in an orderly and structured fashion. The testing of the software program follows from the structured requirements that define the performance. The syntax errors are easier to correct by following the laid down structures. However, the logical errors which many times occur during runtime are difficult to detect and debugging accordingly [51]. In this research project, expertise knowledge was sought from other members of the department. It was the most difficult process yet the core of the entire research. It took a long time for the program to run. An algorithm that has been realized into computer software is called a computer program. The ease of effectively translating an algorithm into a computer language depends upon the existence of a suitable language [52]. The above algorithm is therefore translated into a C++ programming language *[appendix5]* such that the streets with heavier traffic density receive more time for traffic flow.

CHAPTER 4: SYSTEMS HARDWARE DESCRIPTION.

4.1 INTRODUCTION.

The design of any electronic circuit is done after a proper analysis of the components to determine the basic voltage and current requirements amongst other properties of the individual chips used. Most circuits use the common TTL logic based chips that are easier and safer to handle since the current / voltage requirements can be easily met in the laboratory [53]. In this research work components were carefully chosen to fit the TTL requirements. The circuits to control the traffic lights with sensors consisted of two main circuits. A description of each of these circuits is done in this chapter.

4.2. The hardware system architecture.

The hardware system consists of the different chips that are connected to actualize the traffic lights control circuit. The two main circuits i.e. the lights control circuits and the sensor circuits were interfaced via a computer

4.2.1. The lights control circuit.

The lights are represented using LED's, they were connected to the printer port across resistors and a driver to limit the current and protect both the LED and the printer port. The components of the lights circuit included:

i. The light emitting diode (LED).

To simulate the traffic lights, 5mm 5V LED's were used and for practical purposes they were selected for compatibility with the other TTL circuit components. They were used to represent the traffic lights on either side of the road. RED is used for STOP, YELLOW for PREPARE/wait and GREEN for GO.

a) LED **b) circuit symbol.**

Fig. 4.1. The LED.

LED's emit light when an electric current passes through them. LED's must be connected the correct way round, the diagram may be labeled **a** or + for anode and **k** or - for cathode. The cathode is the short lead and there may be a slight flat on the body of round LED's. LED's can be damaged by heat when soldering, but the risk is small unless you are very slow. It should never be connected directly to a battery or power supply! It will be destroyed almost instantly because too much current will pass through and burn it out. LED's must have a resistor in series each to limit the current to a safe value

ii. The resistors.

An LED must have a resistor connected in series to limit the current through the LED, otherwise it will burn out almost instantly. The resistor value, R is given by:

$$R = (V_S - V_L) / I$$

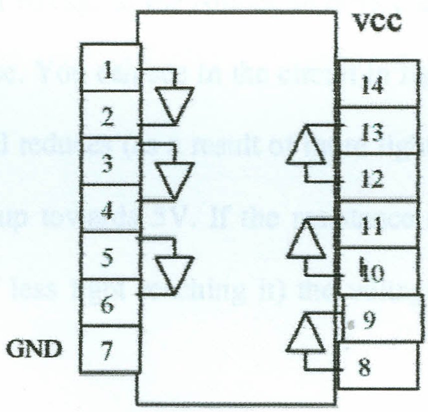
V_S = supply voltage

V_L = LED voltage (usually 2-5V, but 4V for blue and white LED's)

I = LED current (e.g. 20mA), this must be less than the maximum permitted.

If the calculated value is not available choose the nearest standard resistor value which is greater.

iii. The 7405 LED driver.



The pin assignment of the LED driver is as shown in fig.4.4. It ensures smooth flow of current in the LED and protects it from noise.

Fig. 4.2. The 7405 LED driver.

4.2.2 The sensor circuit.

The sensor is used to detect the presence or absence of a vehicle. In this research work, the photo detector was preferred for sensing the presence of vehicles and hence the queue. A photocell is a type of resistor. When light strikes the cell, it allows current to flow more freely. When dark, its resistance increases dramatically.

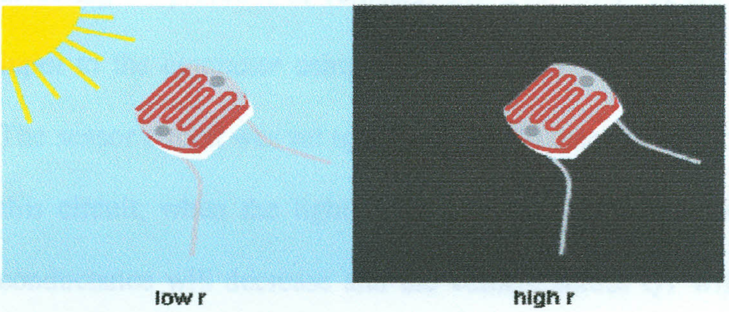


Fig.4.3: Photocell.

The circuit used is a plain voltage divider. It is used since we need to present a voltage to the Analog In device and the photocell merely changes resistance. You can see in the circuit in fig.4.7 that as the resistance of the photocell reduces (as a result of more light reaching it) the voltage on Ain will go up towards 5V. If the resistance of the photocell increases (as a result of less light reaching it) the voltage on Ain will fall towards Gnd (0V).

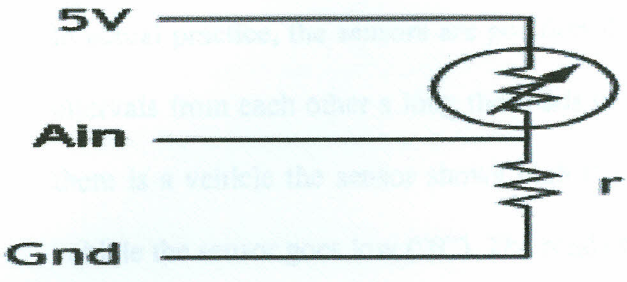


Fig.4.4: Illustrating sensor operation.

For the traffic lights to be vehicle actuated, it is necessary to have a way of detecting the presence of vehicles [54]. The output of the sensor is used as input to the computer using software to control the output of the lights. The sensor circuit was set up using a photo detector as shown in fig.4.5. In this circuit, when the light falling on the photo detector is blocked, its conductance will decrease and the voltage across Q1 will rise. When the voltage rises above ½ of the supply voltage, the output of the comparator will turn ON and the LED lights up.

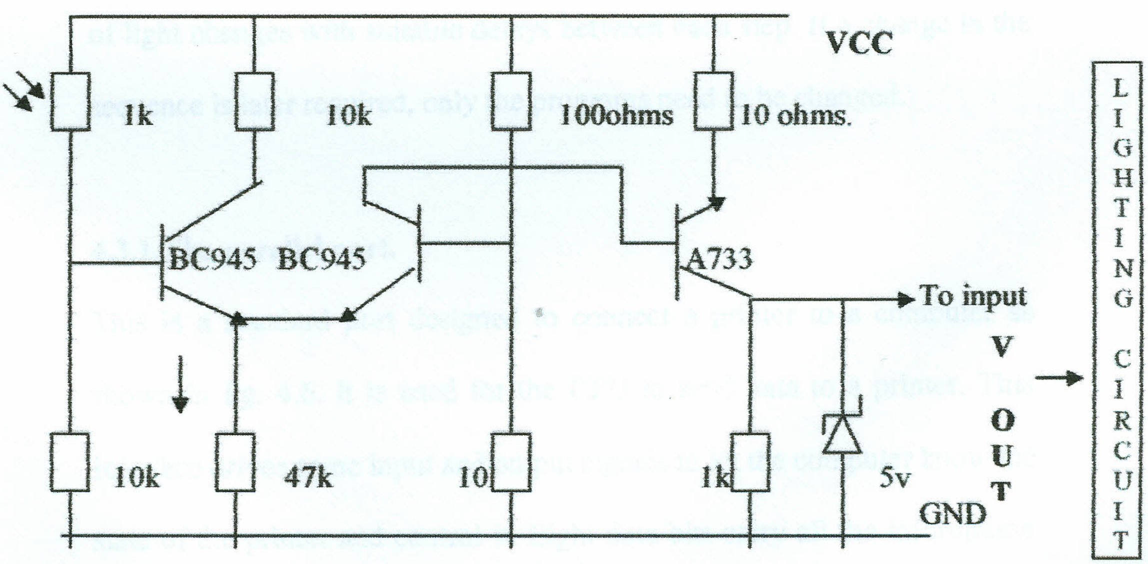


Fig.4.5 sensor circuit set-up.

In actual practice, the sensors are positioned at equal measurable distance intervals from each other a long the roads or streets at the junction. When there is a vehicle the sensor shows high ("1"), while in the absence of a vehicle the sensor goes low ("0"). The road or street with the longest string of 1's (which translates to greatest length) is given more time for traffic flow.

4.3. The computer.

A computer system has four main parts- C.P.U, memory, input and output as shown in Fig 4.1 A computer is more reliable in process control and automation due to its ability to monitor the process and trigger the relevant signals besides analyzing the incoming signals. A good example of this is the traffic lights controller [55]. The lights must change according to some predefined sequence. The computer should perform the required sequence of light changes with suitable delays between each step. If a change in the sequence is later required, only the programs need to be changed.

4.3.1.The parallel port.

This is a standard port designed to connect a printer to a computer as shown in fig. 4.6. It is used for the CPU to send data to a printer. This interface drives some input and output signals to let the computer know the state of the printer and control it. Eight data bits carry all the information sent with each clock pulse [56].

The original IBM-PC's Parallel Printer Port had a total of 12 digital outputs and 5 digital inputs accessed via 3 consecutive 8-bit ports in the processor's I/O space.

- 8 output pins accessed via the **DATA Port**
- 5 input pins (one inverted) accessed via the **STATUS Port**
- 4 output pins (three inverted) accessed via the **CONTROL Port**
- The remaining 8 pins are grounded

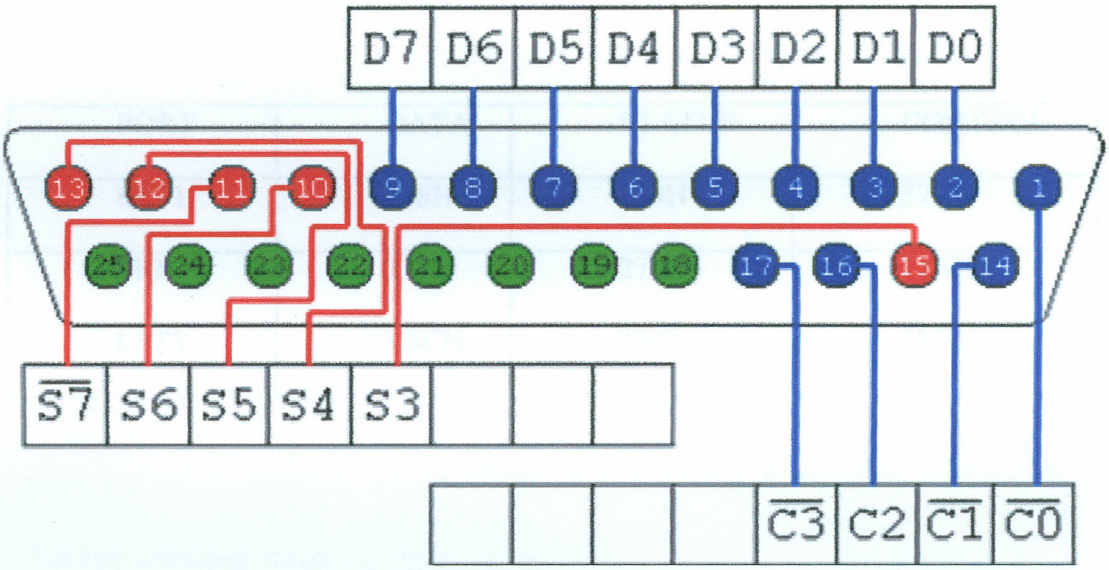


Fig. 4.6. Parallel port pin assignment.

The input/output signals are made available at the back of the adapter through a right-angled PCB mounted 25 pin D-type female connector which protrudes at the rear panel of the system, where a cable may be attached. Each port uses 3 addressees of the I/O map [56].

For LPT1 these addressees are 378H, 379H &37AH.

378H is the output port and 379H is the input port; these signals are used to know the status of the printer. 37AH is a control port on which the computer writes signals to control the printer. The table below shows the addresses of the LPTn ports.

TABLE 4.1 : Addresses of the LPT-n ports.

PORT	DATA	STATUS	CONTROL
LPT1	378H	379H	37AH
LPT2	278H	279H	27AH
LPT3	3BCH	2BDH	3BEH

Output to the parallel port can be accomplished in C++ with the help of the header library *conio.h*. This library contains a function _outp, that can write data to a given port and _inp, which receives data from an external source [56]. By programming this port appropriately, the required traffic lights sequence as obtained.

4.3.2. The printer cable.

This is the link between the printer port on computer and the printer. It has the DB –25 male connector on one end to connect to the port on the computer and wires that carry information in form of bits to & from the printer. The no. of wires in the cable corresponds to the no. of pins on either end.

4.3.3 DB-25 connector.

The connector is used to interface the computer to the LED's and sensors to implement the traffic lights controller. It has 25 pins similar to the pins found on the printer to receive the cable from the computer. The pin assignment of the connector is similar to that of the printer port, there are 5 input status pins, 5 output control pins and 8 output data pins [57,58].

To link the sensors to the computer, the input lines are used to send signals from the sensors to the computer. The computer processes the signals and sends the information as signals to the output pins to implement the required sequence of traffic lights output. The connector is therefore the main link between the computer, sensors and the LED's.

4.4. The interface circuit design.

This is the circuit used to link the computer to the traffic lights. The interface is done via the PC parallel printer port. The parallel port is specifically designed to attach the printer to the computer, but it can be used as a general input/output port for any device or application that matches its input/output capabilities. It has 12 TTL buffer output points which are latched and can be written and read under program control using the processor in or out instruction [58]. The interface circuit is made of the parallel printer port, a printer cable, DB-25 connector and a buffer as described in CH. 4. The Fig. 4.7 shows the block diagram of the system interface circuit. It is through this circuit that the software is able to operate the traffic lights in a predefined sequence. Actual pin-assignment and logic-gates within the chips is as shown in appendix3.

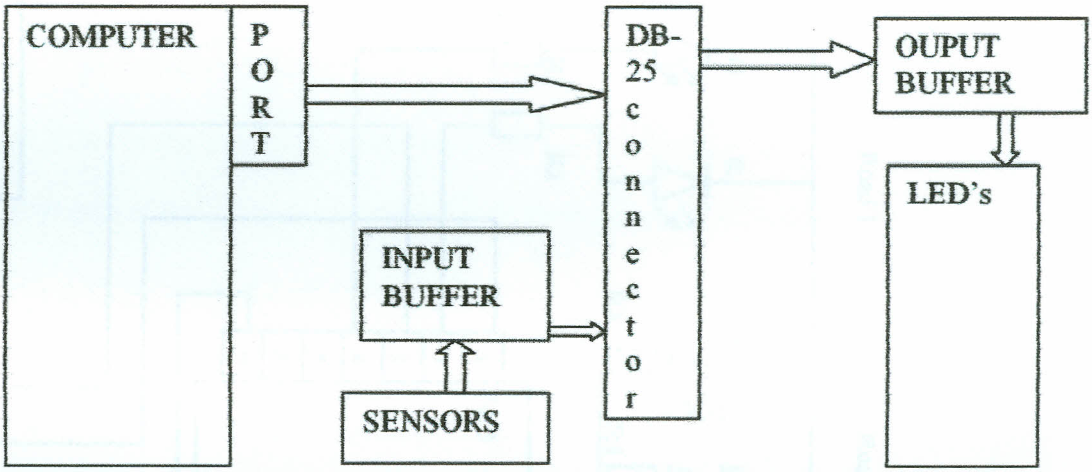


Fig 4.7: Interface circuit Block diagram.

Using the interface the lights and the sensor were connected to the computer for the software to control the lights according to the sensor output. The complete systems set up was as shown in Fig. 4.8 below.

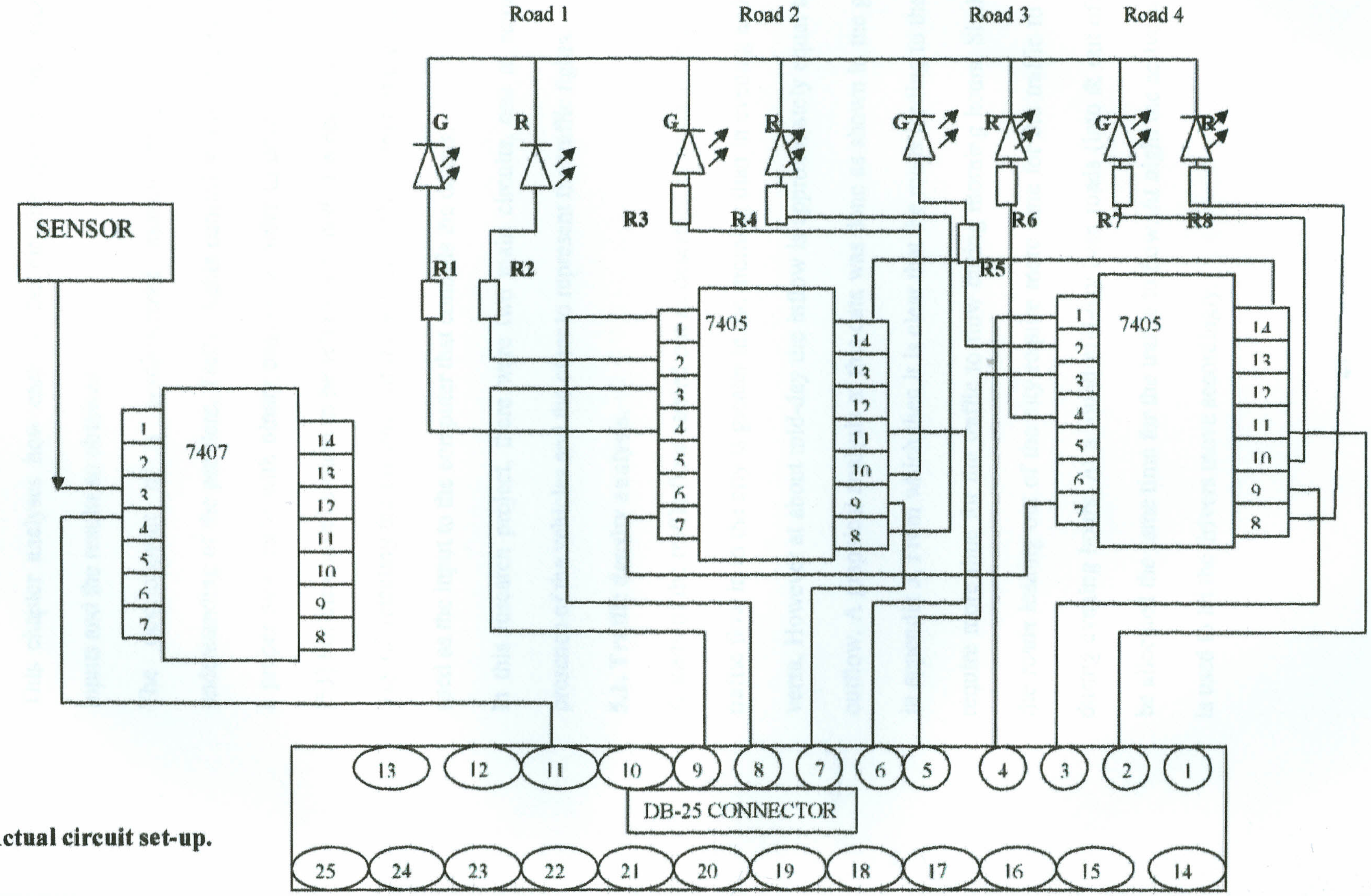


Fig. 4.8 Actual circuit set-up.

CHAPTER 5: RESULTS AND DISCUSSION.

5.0 Introduction.

This chapter analyses how each of the circuits operate, the necessary inputs and the results so obtained.

The development of a computerized system requires a proper understanding of the problem. Traffic lights control requires the design of a proper state table with binary outputs in order to develop the software [53] For the control system to be vehicle actuated, it is necessary to have a way of detecting the presence of the vehicles. The output of the sensor is used as the input to the computer that controls the delays.

In this research project, there were two main circuits, one to sense the presence of the vehicles and the other to represent the traffic lights.

5.1. Traffic density analysis.

A study of the traffic flow carried out as described in chapter 3 shows that traffic flow in to the city is greater in the morning than in evening and vice versa. However at about mid-day the inflow is approximately equal to the outflow. A graphical analysis of this data was done as shown in the graph in appendix 3. From which then it is clear that the roads leading to the city require more time for the traffic to flow during morning hours. Similarly the roads leading out of the city require more time for the traffic to flow during evening hours. At a round mid-day both roads (into & out of) can be allocated the same time for the traffic to flow. At night the active sensor is used to let the drivers move accordingly.

5.2. Sensor circuit analysis.

The sensor to detect the presence or absence of vehicles on a given road was developed using a photocell as described in chapter 4. When the vehicle hinders light from reaching the photocell, the conductance decreases and the voltage across the phototransistor rises hence lighting the LED. Blocking light from reaching the photocell tested this circuit, each time the light reached the photocell the LED was off. When the light was blocked from reaching the photocell, the LED turned on. This is sent as a signal to the computer and the system responds by changing the lights according to the software.

The system is such that when sensor 1 is ON it implies that the road leading to the city has more vehicles and should thus be allowed more time for the traffic to flow. When sensor 2 is on, the road leading out of the city has more vehicles and should now be allowed more time for the vehicles to move. However, when both sensors are On then both roads have many vehicles and should be allowed equal times for the traffic to flow as in the mid morning hours. When no sensor is on, equal delays should be implemented, i.e., the two sensors are used to indicate whether inflow >or < outflow for the relevant sequence to be implemented.

The sensor is also used to detect the length of the vehicles waiting on the queue of a given street. By putting a number of sensors on one road and writing a program that keeps checking how many of those sensors are active, the controller may give more time when all the sensors are high and less time when one or two of the sensors are on. The table below is a summary of how the sensor functions.

Table 5.1. Sensor effect.

SENSOR	EFFECT
Sensor1 = ON	Allows road leading to the city more time
Sensor1 = Sensor2 ON	Allows equal time intervals.
Sensor2 = ON	Allows road leading out of the city more time.
Sensor1 = Sensor2 OFF	Allows equal time intervals (checks all the streets).

5.3 Traffic lights analysis.

To simulate the lights the components described in chapter 4 were used a suitable software was developed to control the lights in the desired sequence. The state table in ***appendix2*** gives the logic used in implementing the lighting circuit. For practical purposes, only a few lights are used to implement a working controller. The system was tested using the software and the lights could be seen to be changing from green to red from one road to the other depending on the state of the sensor.

5.4 The software design.

The program to control the traffic lights was written in visual C++ programming language with the following considerations.

i) The time allowed for the green light on a given road depended on the time of the day and the sensor that is ON.

ii) When a given sensor is on, the delay program would call a given delay that allows that road more time according to the time of the day. The program to control the traffic lights is as shown in ***appendix5.***

It follows that when sensor 1 is ON then the first road leading to the city has more vehicles and requires more time for traffic to move. Therefore a sequence that allows more time for inflow is called upon (i.e. void-morning).the table 5.2 below shows the effect of the sensor on the timing sequence.

Table 5.2. Effect of Sensor on active sub-routine.

ACTIVE SENSOR	IMPLICATION	ACTIVE SUB-ROUTINE
S1=S2=OFF (during the day)	No major queues i.e. allow equal time.	Void – mid-day
S1=ON S2=OFF	A longer queue on road 1 leading to the city i.e. allow more time to in-flow.	Void – morning
S1=S2=ON	Both roads have equal queues i.e. Allow equal time.	Void – mid-day
S1=OFF S2=ON	A longer queue on road 2 leading out of the city i.e. allows more time to out-flow.	Void – Evening
LATE NIGHT	Beyond midnight.	Void- night

CHAPTER 6: CONCLUSION AND RECOMMENDATIONS.

6.1 Summary.

This research project was carried out with great consultation with other members of the department. We successfully designed: a circuit to detect the presence of vehicles using a photo detector, a circuit to control traffic flow using red and green LEDs, a software program in C++ that controlled the lights according to the active sensor and interfaced the circuits to the computer using a DB-25 connector. The programming aspect was difficult and took along time to be learnt. It was eventually successful but the major limitation was the use of 10 lights since other outputs of the printer port are grounded.

6.2 Recommendations.

In view of the above research we recommend its adaptability specifically in Nairobi round about that involve four different roads. It's easily adaptable to simpler junctions though it needs modification to fit on complicated ones.We recommend further research to

i) Consider a proper plan to manage the roundabouts such that when a vehicle moves from one junction it should not stop at the next junction.

ii) Develop a system that can take care of more complicated road networks.

iii) Consider the use of CCTV monitoring in detecting the vehicles and the length of the queue.

REFERENCES

[1] Hayes Roth. F, "*Building expert systems*",Adison Wesley, Massachusets , 1983.

[2] Fransis J.Hale, "*Introduction to control system analysis and design.*"Prentice Hall inc. U.S.A 1973.

[3] Tom Duncan, "*Electronics for Today and Tomorrow*", John Murray, 1982.

[4] John J. D'Azzo and Constantine H. Houpis, " *Linear Control Systems Analysis and Design: conventional and modern*", 4th Edition, McGraw Hill Book company, Singapore, 1995.

[5]Donald Fink and Alexander McKenzie, "*Electronic Engineers Handbook*", McGraw Hill, New York, 1975.

[6] J. Schwarzenbach and K.F.Gill, "*System modelling and control*", 2nd Edition, Edward Arnold ,London 1978.

[7] Douglas Lewin, "Logical *design of switching circuits*" 2nd Edition Nelson, Great Britain, 1974.

[8] Tietze.U, "*Electronics circuits design and applications*" Sprimger Verlag, New York 1991.

[9] David Turner, "*Practical exercises in microelectronics*", Stanley Thornes, England 1992.

[10] Markos Papageorgion, "*Concise encyclopedia of traffic and transportation systems*",Pergamon press 1991.

[11] Chilley J.C,"*Transducers for microprocessor systems*" Mcmillian, London, 1985.

[12] Peter Lawrence and Konrad Mauch, "*Real-time Microcomputer System design an introduction*",Mc.Graw Hill, U.S.A.,1987.

[13] N.J.Fliege "*Multirate digital signal processing*" John Wiley and sons Ltd,Great Britain, 1994.

[14] Michael F. Hordeski "*Control technology and personal computers system design and implementation.*" Van Norstrand Reinhold, New York 1992.

[15] David Jiles, "*Introduction to the electronic properties of materials*" Chapman and Hall, Great Britain, 1994.

[16] Lipovski G.J " *Microcomputer interfacing*", Lexington Books, Massachusetts ,1980.

[17] Richard Dorf, "*Electrical Engineering handbook*", CRC press, Florida, 1993.

[18] Donald Fink and Wayne Beaty ,"*Standard Handbook for Electrical Engineers*",12th edition,Mc.Graw Hill U.S.A. 1987.

[19] Be love, " *Handbook of Modern Electronics and Electronics Engineering*", Willey interscience, New York, 1986.

[20] Howard Austerlitz, "*Data acquisition techniques using personal computers*" , Academic press inc. uk.1991.

[21] Ram Bansal, " *logic and microcomputer technology*" , VIGYACHARYA. India 1999.

[22] http: // www.hut.fi/misc/Electronics/circuits/parrallel_output.html. simple "*circuit and program to show how to use the PC parallel port output capabilities*" Tom Englal.-2005.

[23] Terrence W. Pratt, "*Programming languages design and implementation*",Prentice hall inc. USA, 1975.

[24] J.E. Cooling, "*Software design for real time systems*", International Thompsons computer press. London. 1995.

[25] Edwards J. Gorburn, " *Microcomputers hardware software and programming*", Bob Merrill educational publishing, USA, 1984.

[26] Arthur D. Friedman and Pramachandran R. Menon, " *Theory and design of switching circuits*''. Prentice hall international ltd.,U.S.A,2003.

[27] Subbha Singh, "*Automatic traffic lights control*", Electronics projects vol.8, India 1986.

[28] Bahadur Singh, "*The traffic controller*", Electronics projects vol.9 India 1987.

[29] Rob Paisely http://home.cogeco.ca/~rpaisely4/circuit index.html, "Basic 10 step traffic controller.2004.

[30] http:// www.electronics // ELE 447-Traffic Controller.html 2002-08-05

[31]http:// www.cs.duke .edu /~ola/ book.html.ola @cs.duke.edu.-traffic control.

[32] Rudolf F. Graf, "*Encyclopedia of electronic circuits vol* 3", Mc.Grawhil,USA, 1991.

[33] D Lewin and D. Noaks , " *Theory and design of digital computer systems*'', Chapman and hall London, 1992.

[34] Adisson wisley, "*Software engineering, 4th edition*", Ian Sommervile, USA, 1992.

[35] Ralf J. Smith, "*Electronics circuits and devices*", John Wiley and sons, USA 1973.

[36] Collin Wecler, "*Microprocessors and industrial control*", Hutchison computer studies series, Great Britain, 1988.

[37]Henry Lieu http:// www.tfhrc.gov/pubrds /janfeb 99 /traffic htm. " traffic flow theory"

[38] A. Renyi.http:// www.jsto.org/view/00219002/sp050002/05x1165u/o.two mathematical modelaof the traffic on a divided highway

[39] http:// www.thp.uni_koeln.de/~as/mypage/verkehr.html.. " amathematical model of traffic flow".

[40] http:// www.thp.uni_koeln.de/~as /mypage/traffic html. "traffic flow modelling"

[41] George Meyerlle "Microcomputer technology", Computers and Electronics, Vol 21 No.3. March. 1983.

[42]] http://www.howstuffworks.com/question 234.html. " How do the lights know that a vehicle is waitng?"-2005.

[43] East African Standard , " *Traffic lights at Westlands are switched off* " ,25th July 2003.

[44] The Daily nation, "*Westlands traffic lights switched off*", 25th July 2003

[45]. Allan Johnson, " *Electronics, A systems approach*", Hodder and Stoghton Ltd. Bangalore, India 1987.

[46] Michael A Lynch, "*Microprogrammed state machine design*", CRC Press, USA, 1993.

[47] Anne Kaldewaiji, " *Programming, the derivation of algorithms*", Prentice hall international ltd., Great Britain, 1990.

[48] http://support.intel.com/support/controller/peripheral/7547.htm, "port programming made easier''.-2006.

[49] Boris Beizer, " *Software testing techniques second edition*", International Thompson's computer press, USA, 1990.

[50] Boris Beizer " *Guide to good programming practice*" 2nd edition, International Thompson's computers press. USA 1990.

[51] Walter Savitch, ''*problem solving with C++ the object of programming* 2nd edition''.Addison Wesley. USA 1999.

[52]James Roberge and George Smith , ''*Introduction to programming in C++*'',

[53] "*TTL data book for design engineers*," Texas instruments inc. USA 1986.

[54] A.C Downton, " *Computers and microprocessors, components and systems*", Chapman and hall 2nd edition, UK, 1992.

[55] Dr. Thomas Pabst, " *High performance PC secrets: Tom's Hardware guide*", SAMS Publishing, USA, 1998.

[56] http://home.alt.net/~n7vk/howit.htm, " programming the Parallel port''. -2006.

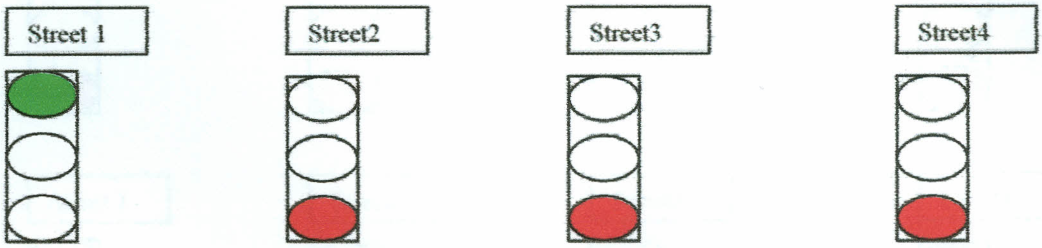
[57] M.D.BEER, " *Microcomputer interfacing*'', William Collins and Sons company ltd. Great Britain.1985.

[58]http://www.mech.northwester.ed/~design/mechatronics/2000/team25/circuits.html-2006.

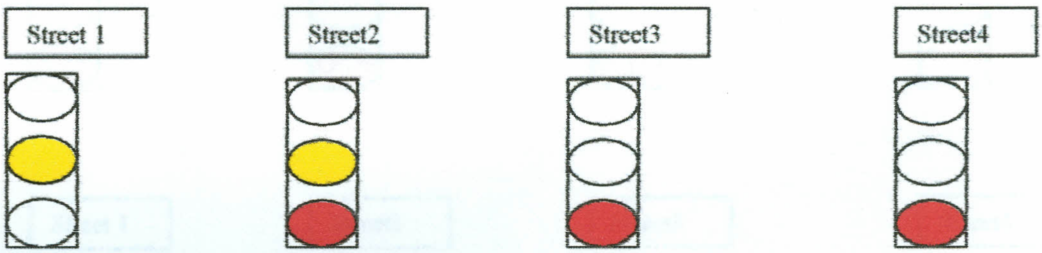
APPENDIX 1.

LIGHTING SEQUENCE OF THE LIGHTS.

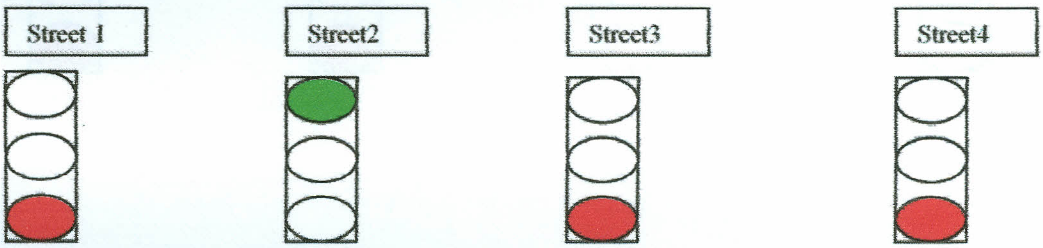
1.



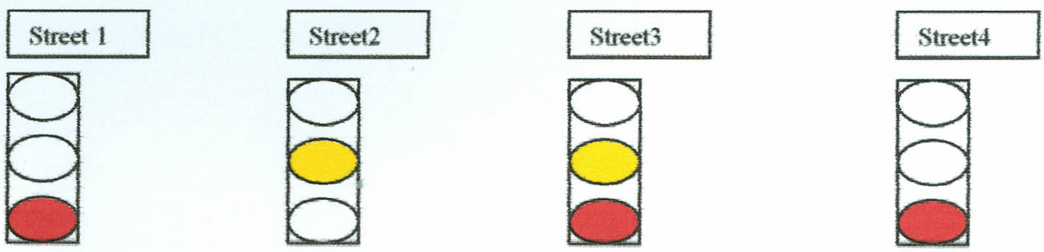
2.



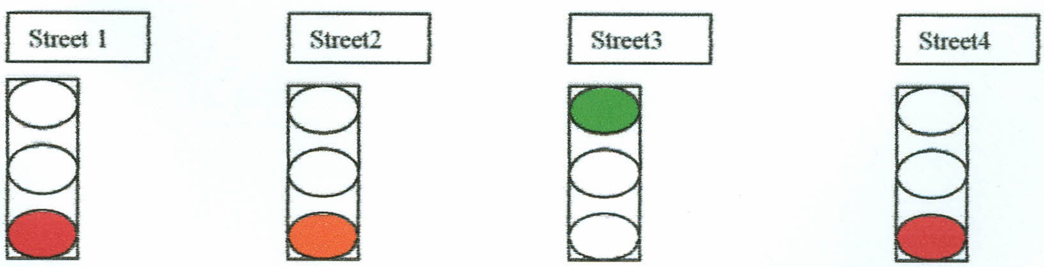
3.



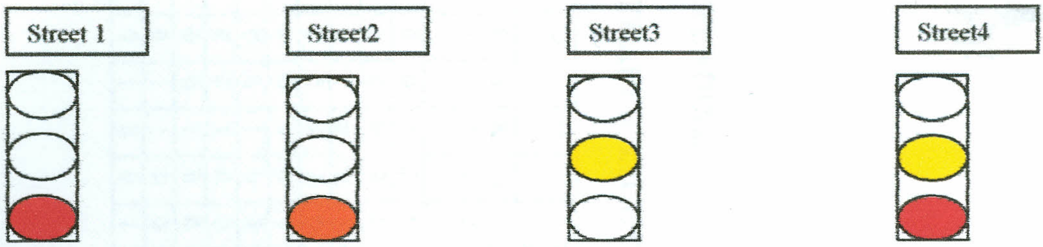
4.



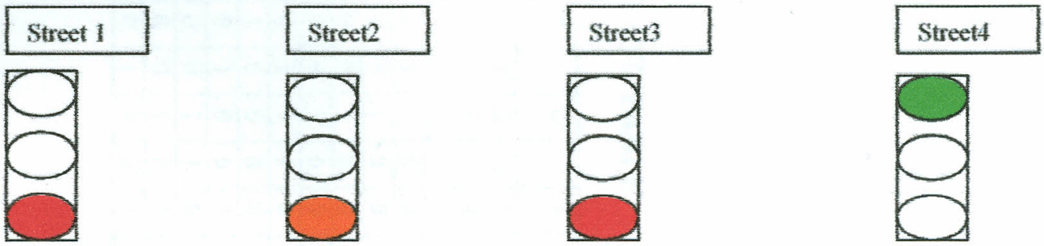
5.



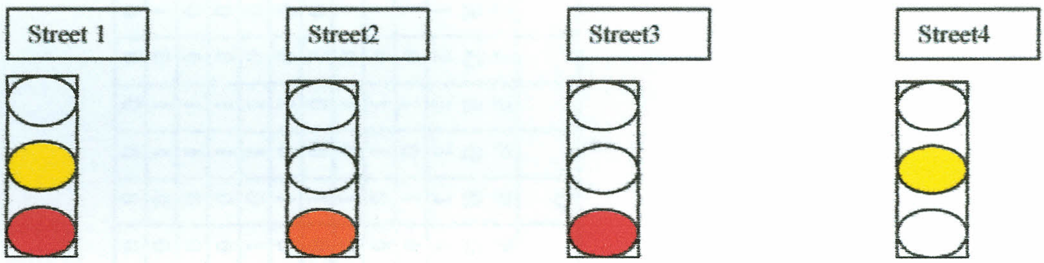
6.


Street 1
Street2
Street3
Street4

7.


Street 1
Street2
Street3
Street4

8.


Street 1
Street2
Street3
Street4

APPENDIX2.

a) TRAFFIC LIGHTS STATE TABLE

I N	P U	T S		O		U		T		P		U		T		S						U		T		P		U		T		P							
A	B	C	D	G1	Y1	R1	G2	Y2	R2	G3	Y3	R3	G4	Y4	R4	PG1	PR1	PG2	PR2	PG3	PR3	PG4	PR4	PG5	PR5	PG6	PR6	PG7	PR7	PG8	PR8	G5	R5	G6	R6	G7	R7	G8	R8
0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	1	0	1	0	1	1	0	0	1	1	0	0	1	1	0	1	0	0	1	1	0	1	0	1	0
0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	1	1	0	0	1	1	0	0	1	0	1	0	0	1	0	1	0
0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0
0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0	0	1	0	1	1	0	0	1	1	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	1	0	0	1	1	0	1	1	0	1	0	1	1	0	0	1	1	0	1	0	0	1	1	0
0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	1	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	1	0	1	0	0	1	0	0
0	1	1	1	0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	1	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	0	1	0	0	1
1	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0	0	1	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

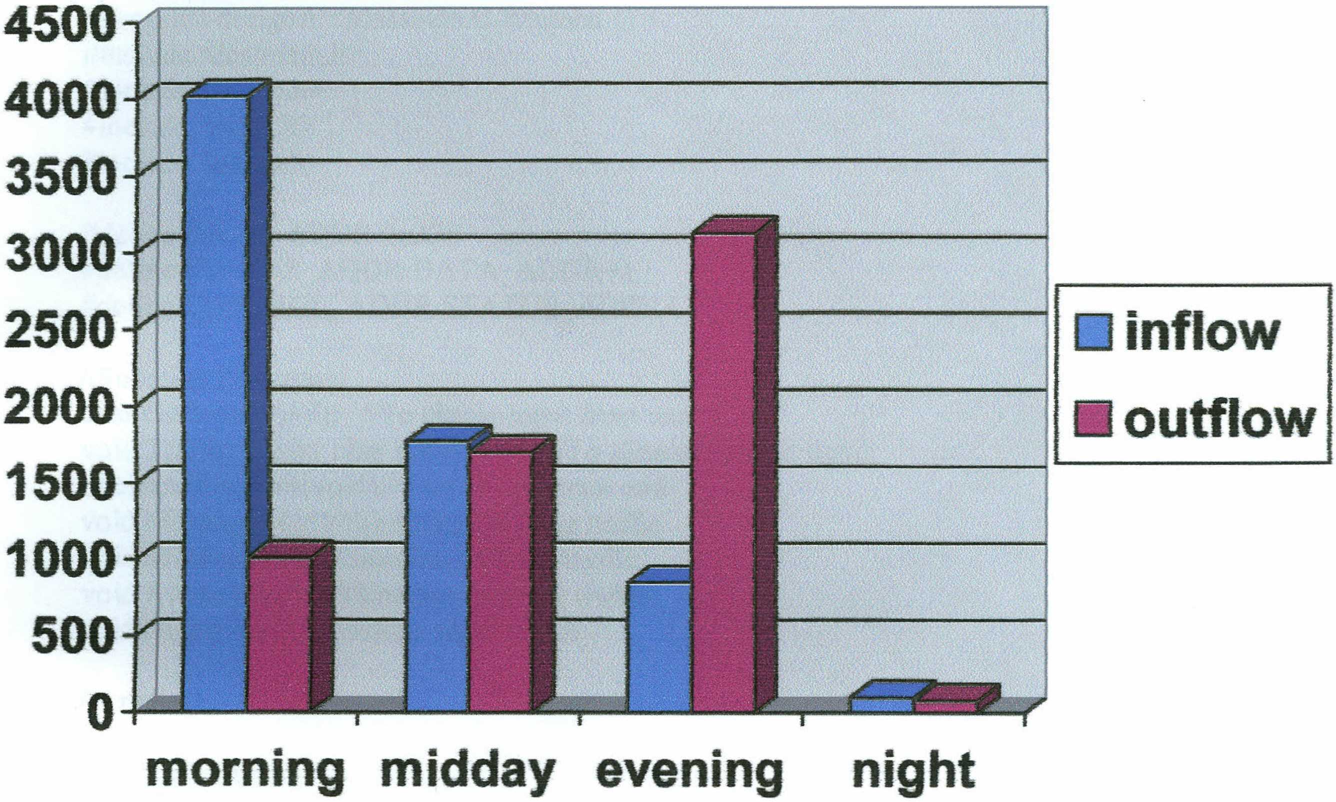
b) REDUCED TRAFFIC LIGHTS STATE TABLE.

INPUTS				OUTPUTS																											
A	B	C	D	G1=PG8	Y1	R1	G2=PG2	Y2	R2	G3=PG4	Y3	R3	G4=PG6	Y4	R4	PR1	PR2	PR3	PR4	PR5	PR6	PR7	PR8	G5=PG1	R5	G6=PG3	R6	G7=PG5	R7	G8=PG7	R8
0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1
0	0	0	1	1	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1	0	1	0	0	0	1	1	0	1	0	1	0
0	0	1	0	0	1	0	0	1	1	0	0	1	0	0	1	1	1	1	1	0	1	0	1	0	1	0	0	1	0	1	0
0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1	0	0	1	1	0	1	0	1	1	0	0	1	1	0	1	0
0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	0	0	1	1	0
0	1	1	0	0	0	1	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	0	1	0	0	1	0	0
0	1	1	1	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	1	0	0	1	1	1	0	1	0	1	0	0	1
1	0	0	0	0	1	1	0	0	1	0	0	1	0	1	0	1	1	0	0	0	1	1	1	0	0	1	0	1	0	0	1
1	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

c) Simulated state table

STATUS	ROAD		ROAD2		ROAD 3		ROAD 4	
	G1	R1	G2	R2	G3	R3	G4	R4
ALL RED	0	1	0	1	0	1	0	1
ROAD1	1	0	0	1	0	1	0	1
CLEAR	0	1	0	1	0	1	0	1
ROAD2	0	1	1	0	0	1	0	1
CLEAR	0	1	0	1	0	1	0	1
ROAD3	0	1	0	1	1	0	0	1
CLEAR	0	1	0	1	0	1	0	1
ROAD4	0	1	0	1	0	1	1	0
CLEAR	0	1	0	1	0	1	0	1
ROAD1	1	0	0	1	0	1	0	1

Appendix3 **TRAFFIC DENSITY (IN/OUT OF THE CITY).**



APPENDIX 4. TRAFFIC LIGHTS CONTROL SOFTWARE.

```
//This program is meant to simulate traffic lights
//For the purpose of demonstration, the real times for each
//road have been reduced by a factor of 1/10
//Program designer : Makunda C. Maloba.
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>

#define DATA_ADDR 0x378
#define STATUS_ADDR DATA_ADDR+1
#define CONTROL_ADDR STATUS_ADDR+1

//Function Prototypes
char GetInput(void);   //To obtain input from user
void Traffic_Light( char time);        //To simulate traffic lights
bool EndProgram(void);//Test for program end
void Morning(void);//Simulate morning traffic
void Midday(void);//Simulate midday traffic
void Evening(void);//Simulate evening traffic
void Night(void);//Simulate night traffic

int main()
{
        char time;

        do
        {
                //time=GetInput();
                Traffic_Light(time);
        }
        while(!EndProgram());
        /*{
                time=GetInput();
                Traffic_Light(time);
        }*/
        //cout<<"End of program"<<endl;

        return 0;
}

//Function definitions
char GetInput(void)
```

```
{
        char session;
        cout<<"To simulate the traffic lights,enter:"<<endl;
        cout<<"M for morning session"<<endl;
        cout<<"A for midday session"<<endl;
        cout<<"E for evening session"<<endl;
        cout<<"N for night session"<<endl;

        cin>>session;
        return session;
}

void Traffic_Light( char time)
{

        for(int counter=0;counter<=2;counter++)
        {
        if(_inp(STATUS_ADDR)==255)
        {
                Morning();
        }
        else if(_inp(STATUS_ADDR)==127)
        {
                Midday();
        }
        }
        Traffic_Light(time);

        return;
}

bool EndProgram(void)
{
        char reply;
        cout<<"Enter Y to end the program and N to continue with";
        cout<<"the simulation:"<<endl;
        cin>>reply;
        if(reply=='Y')
                return true;
        else if(reply=='N')
                return false;
        else
        {
                cout<<"Invalid Input! The program is ending!"<<endl;
                return 1;
```

```
        }
}

void Morning(void)
{
        while(!kbhit())
        {
                //Turn all red to begin the session
                for(int i=0;i<1000;i++)
                {
                        _outp(0x378,170);
                        cout<<"All set to red, begin the morning session"<<endl;
                }
                for(int j=0;j<10000;j++) //Allow traffic into the city/town
longer(road 1)
                {
                        _outp(0x378,106);
                        cout<<"Road leading to city open"<<endl;
                }

                for(int k=0;k<10000;k++) //Allow traffic transition(Amber for road
1) to avoid collission
                {
                        _outp(0x378,170);
                        cout<<"Clearing the highway for road 2"<<endl;
                }
                for(int l=0;l<5000;l++) //Allow traffic into the city/town longer(road
1)
                {
                        _outp(0x378,154);//open road 2
                        cout<<"Road 2 open"<<endl;
                }
                for(int f=0;f<1000;f++) //Transition (Amber for road 2)
                {
                        _outp(0x378,170);
                        cout<<"Clearing highway for road 3"<<endl;
                }
                for(int s=0;s<5000;s++) //Open road 3
                {
                        _outp(0x378,166);
                        cout<<"Road 3 open"<<endl;
                }
                for(int h=0;h<1000;h++) //Clear highway for road 4
                {
                        _outp(0x378,170);
                        cout<<"Clearing highway for Road 4"<<endl;
```

```
		}
		for(int a=0;a<10000;a++) //Open road 4 for longer since
		{						// it is leading to the
city
			_outp(0x378,169);
			cout<<"Road 4 open"<<endl;
		}
		return;
	}

}

void Midday(void)
{

	while(!kbhit())
	{
		//Turn all red to begin the session
		for(int i=0;i<1000;i++)
		{
			_outp(0x378,170);
			cout<<"All set to red, midday session"<<endl;
		}
		for(int j=0;j<5000;j++) //Allow traffic into the city/town longer(road
1)
		{
			_outp(0x378,106);
			cout<<"Road leading to city open"<<endl;
		}

		for(int k=0;k<1000;k++) //Allow traffic transition(Amber for road 1)
to avoid collission
		{
			_outp(0x378,170);
			cout<<"Clearing the highway for road 2"<<endl;
		}
		for(int l=0;l<5000;l++) //Allow traffic into the city/town longer(road
1)
		{
			_outp(0x378,154);//open road 2
			cout<<"Road 2 open"<<endl;
		}
		for(int f=0;f<1000;f++) //Transition (Amber for road 2)
		{
			_outp(0x378,170);
			cout<<"Clearing highway for road 3"<<endl;
```

```
		}
		for(int s=0;s<5000;s++) //Open road 3
		{
			_outp(0x378,166);
			cout<<"Road 3 open"<<endl;
		}
		for(int h=0;h<1000;h++) //Clear highway for road 4
		{
			_outp(0x378,170);
			cout<<"Clearing highway for Road 4"<<endl;
		}
		for(int a=0;a<5000;a++) //Open road 4 for longer since
		{							// it is leading to the
city
			_outp(0x378,169);
			cout<<"Road 4 open"<<endl;
		}
	return;
	}

}

void Evening(void)
{

	while(!kbhit())
	{
		//Turn all red to begin the session
		for(int i=0;i<10000;i++)
		{
			_outp(0x378,170);
			cout<<"All set to red, begin the evening session"<<endl;
		}
		for(int j=0;j<50000;j++) //Allow traffic into the city/town
longer(road 1)
		{
			_outp(0x378,106);
			cout<<"Road leading to city open"<<endl;
		}

		for(int k=0;k<10000;k++) //Allow traffic transition(Amber for road
1) to avoid collission
		{
			_outp(0x378,170);
			cout<<"Clearing the highway for road 2"<<endl;
```

```
            }
            for(int l=0;l<100000;l++) //Allow traffic into the city/town
longer(road 1)
            {
                    _outp(0x378,154);//open road 2
                    cout<<"Road 2 open"<<endl;
            }
            for(int f=0;f<10000;f++) //Transition (Amber for road 2)
            {
                    _outp(0x378,170);
                    cout<<"Clearing highway for road 3"<<endl;
            }
            for(int s=0;s<100000;s++) //Open road 3
            {
                    _outp(0x378,166);
                    cout<<"Road 3 open"<<endl;
            }
            for(int h=0;h<10000;h++) //Clear highway for road 4
            {
                    _outp(0x378,170);
                    cout<<"Clearing highway for Road 4"<<endl;
            }
            for(int a=0;a<50000;a++) //Open road 4 for longer since
            {                                               // it is leading to the
city
                    _outp(0x378,169);
                    cout<<"Road 4 open"<<endl;
            }

        }


}

void Night(void)
{
        cout<<"Just use common sense, dear driver."<<endl;

}
```

***APPENDIX 5.* ACTUAL CIRCUIT PICTURE.**