INTELLIGENT TRAFFIC LIGHT CONTROLLER

A PROJECT REPORT

Submitted by

BL.EN.U4ECE13166 SHARATH R NAIK

BL.EN.U4ECE13167 SHEKHAR M

BL.EN.U4ECE13157 ROHITH P V

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

May-2017

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, BANGALORE, 560035

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"INTELLIGENT TRAFFIC LIGHT CONTROLLER"** submitted by

SHARATH R NAIK	BL.EN.U4ECE13166
SHEKHAR M	BL.EN.U4ECE13167
ROHITH P V	BL.EN.U4ECE13157

in partial fulfilment of the requirements for the award of the **Degree Bachelor of Technology** in **ELECTRONICS AND COMMUNICATION ENGINEERING** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Bangalore.

Signature	Signature
Ms. Kavita Baliger	Dr. Shikha Tripathi
Assistant Professor (Sr. Gr)	Professor and Chairman
Dept. of ECE	Dept. of ECE

This project report was evaluated by us on ……

EXAMINER 1 EXAMINER 2

ACKNOWLEDGEMENT

The development of technology that we see today is a result of Engineering endeavours and perseverance. We take pride in being an Engineer and feel privileged to make our contribution to this world of technology.

We would like to offer our sincere regards to **Amrita Vishwa Vidyapeetham** and our beloved chancellor, **Mata Amritanandamayi**, without whose grace we would not have reached this stage. We would like to thank **Dr. Rakesh S.G**, Associate Dean of Amrita School of Engineering for providing the opportunity to undertake this project.

We extend our gratitude to **Kavita Baliger**, our mentor and Assistant Professor of Electronics and Communication Department, Amrita School of Engineering Bangalore, for her guidance and support throughout the course of this project.

We are also thankful to our project examiners, **Dr. T.K Ramesh**, Professor and Chairperson of Department of Electronics and Communication Engineering, **Mr. Nandivardhan H.R**, Assistant Professor and **Mr. Giriraja**, Assistant Professor, Dept. of Electronics and Communication Engineering, for their encouragement and suggestions.

Last, but not least, we would like to offer our gratitude to our family, friends and all others who have contributed directly or indirectly to the success of this project.

ABSTRACT

In the era of ever increasing population and the subsequent increase of vehicles on the road, pile-ups at traffic lights are a very common sight in all major cities. This increases the travel time of every individual consequently make travel a very cumbersome task. With automation expanding into various domains, it has made its way into traffic controls as well. However, because traffic control is a new domain to automation, the systems are not as efficient as the manual traffic control. As the number of road users constantly increases, and resources provided by current infrastructures are limited, intelligent control of traffic will become a very important issue in the future.

The basic objective of this project is to improve the efficiency of the present traffic light control systems in arterial roads, thereby decreasing the waiting time of the vehicles at a set of traffic lights. This is accomplished by dynamically assigning the interval of the green light depending on the presence of the traffic pile-up in cross roads and on the main roads or arterial roads. The present systems have been able to accomplish this to a certain extent, they are left as independent traffic lights. Much research has not been done into the field of inter-communication between the traffic lights on a major road to provide a smooth flow of traffic along the main road, thereby forming a road network. This project aims to accomplish this network of Arterial Traffic Light (ATL) control system using the Intelligent Traffic Light Controller (ITLC).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
ATL	Arterial Traffic Light
AVR	Alf and Vegard's RISC
CISC	Complex Instruction Set Computing
EEPROM	Electrically Erasable Programmable Read-Only Memory
Ga As	Gallium Arsenide
IR	Infra-Red
ISP FLASH	In-System Programmable Flash
ITLC	Intelligent Traffic Light Controller
JTAG	Joint Test Action Group
LED	Light Emitting Diode
MIPS	Million Instruction Per Second
RAM	Random Access Memory
SP	Stack Pointer
SRAM	Static Random Access Memory
USART	Universal Synchronous/Asynchronous Receiver/Transmitter

1 INTRODUCTION

1.1 Traffic Light Controllers

Present Traffic Light Controllers (TLC) are based on microcontroller and microprocessor. These TLC have limitations because it uses the pre-defined hardware, which is functioning according to the program that does not have the flexibility of modification on real time basis. Due to the fixed time intervals of green, orange and red signals the waiting time is more and car uses more fuel.

Fast transportation systems and rapid transit systems are nerves of economic developments for any nation. All developed nations have a well-developed transportation system with efficient traffic control on road, rail and air. Transportation of goods, industrial products, manpower and machinery are the key factors which influence the industrial development of any country. Mismanagement and traffic congestion results in long waiting times, loss of fuel and money. It is therefore utmost necessary to have a fast, economical and efficient traffic control system for national development.

One way to improve traffic flow and safety of the current transportation system is to apply automation and intelligent control methods to roadside infrastructure and vehicles.

Intelligent Traffic Light Controller (ITLC) is a controller which helps in dynamically assigning the green light time depending on the parameters such as presence of the vehicles at each side of the intersection. Infrared sensors, mounted on the sides of the road, at regular intervals are used to detect the presence of vehicles which in turn helps in calculating the amount of time to be assigned.

In present system, each traffic light generates its schedule without considering the schedules of neighbouring traffic light reports. The ATL controlling algorithm is an adaptive variant of the ITLC algorithm, in which the phases scheduled for the surrounding traffic lights are considered in addition to the real- time traffic characteristics of competing traffic flows. The ITLC at each road intersection

reports the lighting sequence to neighbouring signalized intersections. Each traffic light will consider the received schedule to neighbouring traffic lights during the next timing cycle. Generally, the closer the traffic signals, the more important it is to coordinate the location of traffic lights.

The main objective of the open network is to indicate priority for the continuous traffic flow along arterial roads.

1.2 Main Objective of ITLC

The main objective of this project is to design and implement a basic traffic light controlling system which is capable of the following tasks:

- Manage an independent intersection by assigning passage time to vehicles depending on the presence of the vehicles coming into the intersection.
- Giving priority to those vehicles that are approaching along the main road to that of those approaching along the cross roads.
- Communicating with adjacent intersection so that the platoon of vehicles moving along the main road can ideally travel without any waiting time along the whole stretch of the designed network of roads.

The design consists of both hardware and software. Based on the selection of the features in the first phase of implementation, we would try to integrate the modules of selected sensors using microcontroller. Based the on the microcontroller, the firmware will be written to integrate the features.

1.3 Organisation of this report

Rest of the content of the report are as follows:

Chapter 2: The various literature survey undertaken during the course of implementation of this project has been discussed here.

Chapter 3: A detailed explanation of the traffic light system has been given in this chapter.

Chapter 4: The hardware components used in the demonstration of the simulation is discussed in detail throughout this chapter.

Chapter 5: This chapter consists of the actual system that we wish to achieve in this project.

Chapter 6: A few screenshots of various scenarios tried in the simulation has been shown here along with the explanation of each one.

Chapter 7: This chapter concludes this report and the observations of made throughout the implementation of various test scenarios along with the explanation of the future scope of the designed system.

1.4 Outline of Work

One way to improve traffic flow and safety of the current transportation system is to apply automation and intelligent control methods to roadside infrastructure and vehicles. Transportation research has the goal to optimize transportation flow of people and goods. As the number of road users constantly increases, and resources provided by current infrastructures are limited, intelligent control of traffic will become a very important issue in the future. In this project, we have introduced an ITL scheduling algorithm (ITLC) for isolated traffic light scenarios. Furthermore, we have designed an ATL controlling algorithm for open network scenarios. The ITLC algorithm with the help of IR sensors kept on the sides of junctions, schedule the traffic lights by giving more priority for the vehicles in the arterial road. The flow with the vehicle presence is scheduled to pass the road intersection first while the traffic light phases are being set. On the other hand, the ATL algorithm aims to guarantee high traffic fluency for the arterial flows. From the experimental results, we could infer that the ITLC algorithm achieves a better performance compared with previously introduced algorithms in this field. In terms of the average delay experienced by each vehicle when crossing the signalized road intersection, ITLC decreases the delay by 25%. At the same time, ITLC increases the throughput of each road intersection by 30%.

2 LITERATURE SURVEY

The objective of this chapter is to provide an overview of the literature survey performed on the present traffic light systems.

Traffic Management on the road has become a severe problem of today's society. An efficient traffic management techniques are needed to reduce waiting and traveling times, save fuel and money. In order to alleviate the problem, a large number of methods and approaches have been suggested in the literature. It includes rule based learning to the modern fuzzy and neural network approaches.

2.1 Problems of Typical Conventional Traffic Light Controller

The main problems of typical conventional traffic light controller are given below:

- Heavy traffic jams
- No traffic, but still need to wait
- Emergency car stuck in traffic jam
- Lack of Traffic Information from neighbouring traffic signals

2.1.1 Heavy Traffic Jams

With increasing number of vehicles on road, heavy traffic congestion has substantially increased in major cities. This happened usually at the main junctions commonly in the morning, before office hour and in the evening, after office hours. The main effect of this matter is increased time wasting of the people on the road. The solution for this problem is by developing a program which configures different delays timings for different junctions. The delay for junctions that have high volume of traffic should be setting longer than the delay for the junction that has low of traffic.

2.1.2 No traffic, but still need to wait

At certain junctions, sometimes even if there is no traffic, people have to wait. Because the traffic light remains red for the pre-set time period, the road users

should wait until the light turn to green. If they run the red light, they have to pay fine. The solution of this problem is by developing a system which detects traffic flow on each road and set timings of signals accordingly. Moreover, synchronization of traffic signals in adjacent junctions is also necessary.

2.1.3 Emergency car stuck in traffic jam

Usually, during traffic jam, the emergency vehicle, such as ambulance, fire brigade and police will be stuck especially at the traffic light junction. This is because the road users waiting for the traffic light turn to green. This is very critical problem because it can cause the emergency case become complicated and involving life.

2.1.4 Lack of Traffic Information from neighbouring traffic signals

Present traffic systems fail to provide traffic information including congested roads and alternate routes available in case of congestion. Measurements taken at a particular signal intersection or the traffic condition at a particular signal is not taken into consideration for scheduling the traffic at next signal.

2.2 Review of Existing Traffic Light Controlling Algorithm

Several algorithms have been introduced to develop an ITLC algorithm for an isolated traffic light or to design an ITL system for an entire signalized road network. In these algorithms, ITLs consider the characteristics of each direction of traffic flow that intends to cross each road intersection. The majority of the previous ITLC algorithms has been designed for isolated traffic light scenarios (i.e., a traffic light that schedules the traffic timing cycles without considering neighbouring signalized intersections). These algorithms aim to reduce the waiting delay time at the road intersections.

3 INTELLIGENT TRAFFIC LIGHT

3.1 Design of Intelligent Traffic Light Controller Using Embedded System

This project presents an improvement over the fixed time traffic controller by introducing the concept of ITLC wherein the interval of the green light is allocated dynamically to each intersection individually. This is achieved using a microcontroller based system. The algorithm used here gives priority gives importance to that direction which has the maximum pile-up. This system upon implementation is compared to the fixed time traffic light controller and experimental results show that the average waiting time is decreased because of the increase in the switching frequency.

However, in the event of heavy traffic from the same direction, there may arise an issue of time allocation to other directions with lesser priority resulting in a prolonged red-light interval for certain signals.

3.2 Arterial Traffic Light Control System

Arterial Traffic Light (ATL) controller is implemented by connecting the adjacent traffic lights to form an inter-communicating network of traffic light. The basic concept used is that of the ITLC. It is further developed to integrate it with ATL such that there is a smooth non-stop flow of traffic from one end to the other of the arterial street. ATL ensures zero waiting time ideally when travelling across a main road which is connected by a network of smaller crossroads at intersections. The algorithm used here gives higher priority to the traffic on the arterial street. The passage of traffic through the intersection is done based on cycles.

3.3 Ready Area

A virtual square area around each signalized road intersection is configured as the ready area by the local traffic light. The distance from the traffic light to the

boundary of the ready area is set based on the average speed of all competing traffic flows and based on the maximum allowed green time of the traffic light. Vehicles inside this area are ready to cross the intersection in the case that a green phase is scheduled for the traffic flow. However, in the case where the traffic light schedules a red light to a certain flow of traffic, vehicles at such a traffic flow begin to decrease their speed as they prepare to stop. If these vehicles stopped inside the configured ready area, they would be considered in the next scheduled platoon of the traffic flow.

Ready Area is the area from the traffic signal to a fixed distance which encloses the vehicles that will pass through the intersection upon initiation of a green light.

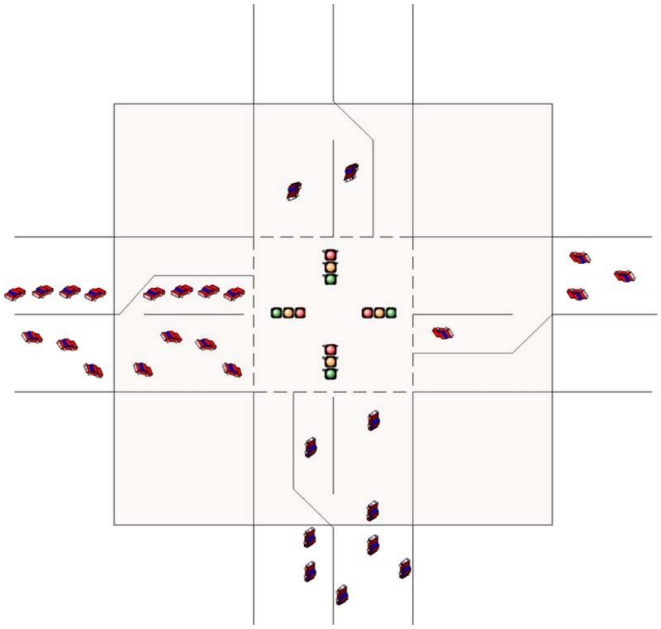

Fig. (3.3.1) Ready Area

Fig. (3.3.1) Ready Area shows an example of a ready area at a signalized road intersection. The ready area divides the successive platoons within each flow of traffic. Thus, it guarantees fair sharing of the road intersection among the competing flows of traffic. For each flow of traffic, all vehicles located inside the ready area during the data-gathering phase are considered at the same platoon of traffic that should pass through the road intersection at the same.

4 PROPOSED SYSTEM

4.1 Proposed Model

The Infrared Sensors to detect vehicles are mounted along the side the road. The presence or absence of a vehicle is sensed by a sensor assembly mounted near the beginning of the signal on each road. And the Infrared Sensors placed at the end of the ready area will give us information about the presence of vehicles on that particular road. This acts as an input to the ITLC unit. This input signal indicates the length of vehicles on each road. The ITLC unit generates output signals for Red, Green and Orange Signal and monitor their timings taking into consideration the length of vehicles on each road. The same information about the timings of the traffic lights is transmitted to the neighbouring traffic lights in order.

The proposed hardware layout is shown in Fig. (4.1.1)

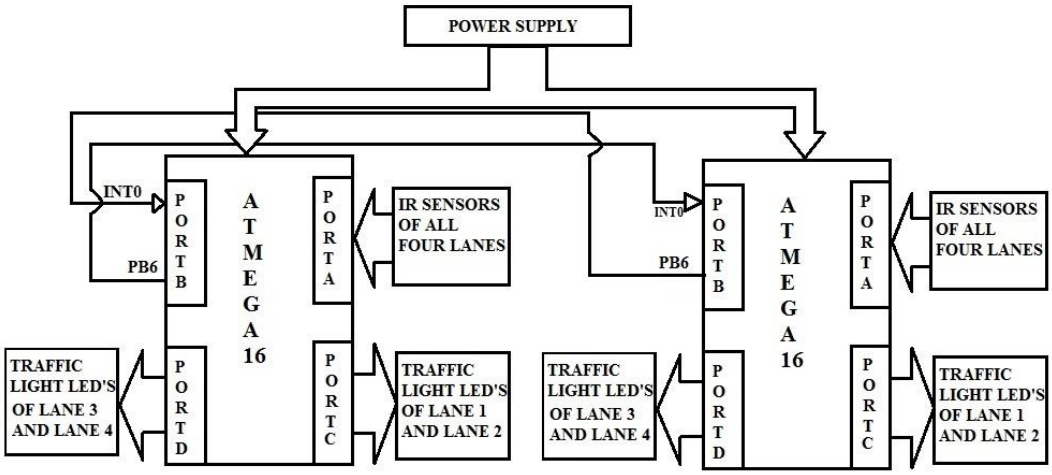

Fig. (4.1.1) Block diagram of proposed system

PIN	PERIPHERAL
PA0-PA7	IR Receiver
PC0-PC6	Traffic Light LED's
PD0, PD4-PD7	
PB2	Interrupt from adjacent microcontroller

Table [4.1.1] Connections to the microcontroller

4.2 Intelligent Scheduling Algorithm

The system proposed here is designed to work at intersections which have crossroads intersecting the main road. The main objective is to reduce the waiting time of the vehicles travelling along the main road and also to provide a smooth flow of traffic from one end to the other.

4.2.1 Isolated System

With reference to Fig. (4.2.1), lane 1 and lane 2 are considered to be the main road and lane 3 and lane 4 are considered to be the crossroads. The following algorithm for a standalone system is executed:

1. The IR sensors, placed at definite intervals along the sides of the road checks whether the vehicles are present in the lane. The placement of the sensors are such that it is able to check the Ready Area shown in Fig. (3.3.1).
2. In the event that there are vehicles waiting at one lane only, the timing of the signal is set based upon the presence of vehicles present and they are allowed to pass.
3. In the event of vehicles being detected at two different lanes at the same time the following algorithm is executed:
    a. If in case of a main road and a crossroad, higher priority is given to the main road and hence the vehicles traveling along the main road are allowed to pass followed by those along the cross roads.
    b. If vehicles are detected at both the main road lanes, or both cross road lanes, then the following priority is taken into consideration
        i. Lane 1 > Lane 2
        ii. Lane 3 > Lane 4
    c. If vehicles are detected at all the lanes the worst case scenario of heavy traffic is assumed and the passage follows the order:
        i. Lane 1
        ii. Lane 2
        iii. Lane 3

iv. Lane 4

The reading of IR sensor is done only upon the completion of the previous cycle.

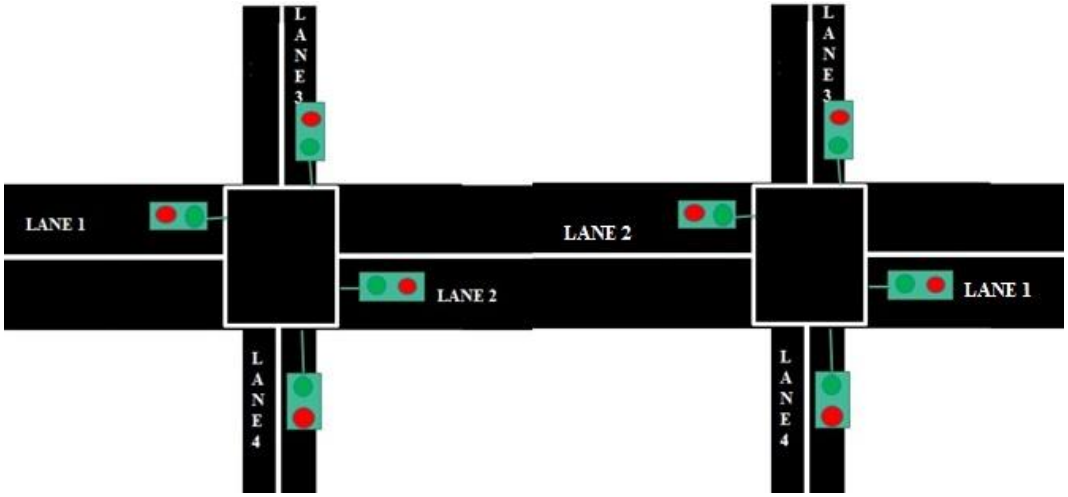

Fig. (4.2.1) Layout of road network

The Table [4.2.1] shows the time proposed in case of each scenario

Lane	Proposed time
Lane 1	40s
Lane 2	40s
Lane 3	35s
Lane 4	35s

Table [4.2.1] Proposed signal times

4.2.2 Combined System

The main intention of the combined system is to communicate with the adjacent intersections so as to ensure that the vehicles which have already passed through the initial intersection do not face any more stoppages until they have crossed the whole road section to the other end of the area.

In order for this to happen the following algorithm is used:

1. When any of the intersections allow passage of vehicles moving along the main road to continue on the same path, a signal is sent to the microcontroller of that junction which comes next in their path

2. The microcontroller upon receiving the signal calculates the time required by with the formula shown in Fig. (4.2.2)
3. When the calculated time is passed, it interrupts its own running process and allows those vehicles to pass through.
4. Once the time is up for them to pass through it goes back to whichever task it was doing.

$$\text{Delay} = \frac{\text{Distance between adjacent signals}}{\text{Average speed of cars between the signal}}$$

Fig. (4.2.2) Formula to calculate delay before interrupting

The time the microcontroller waits before interrupting the task is dependent upon the distance between the adjacent traffic signals and the average speed a vehicle can travel along that road.

5 MAJOR HARDWARE COMPONENTS

5.1 ATmega16

The ATmega16 is a low power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS allowing the system designed to optimize power consumption versus processing speed. The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

Fig. (5.1.1) Atmega16 Microcontroller

5.1.1 Features of ATmega16

The ATmega16 provides the following features: 16 Kbytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1 Kbyte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented

Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density non-volatile memory technology. The On chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

5.1.2 Architectural overview

In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 × 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle. Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash Program memory. These added function registers are the 16-bit X-register, Y-register, and Z-register, described later in this section. The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation. Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16-bit or 32-bit instruction. Program Flash memory space is divided in two sections, the Boot program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section. During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are

executed). The Stack Pointer SP is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture. The memory spaces in the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the Status Register. All interrupts have a separate interrupt vector in the interrupt vector table. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority. The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File.

5.1.3 Major Features of Atmega16

- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 16 Kbytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1 Kbyte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C(1)
  - Optional Boot Code Section with Independent Lock Bits
  - In-System Programming by On-chip Boot Program
  - True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard

- Extensive On-chip Debug Support
- Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface

❖ Peripheral Features

- Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- Real Time Counter with Separate Oscillator
- Four PWM Channels
- 8-channel, 10-bit ADC
- 8 Single-ended Channels
- 7 Differential Channels in TQFP Package Only
- 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
- Byte-oriented Two-wire Serial Interface
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator

❖ Special Microcontroller Features

- Power-on Reset and Programmable Brown-out Detection
- Internal Calibrated RC Oscillator
- External and Internal Interrupt Sources
- Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

❖ I/O and Packages

- 32 Programmable I/O Lines
- 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF

❖ Operating Voltages

- 2.7V - 5.5V for ATmega16L
- 4.5V - 5.5V for ATmega16

❖ Speed Grades

- 0 - 8 MHz for ATmega16L

- 0 - 16 MHz for ATmega16

❖ Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L

- Active: 1.1 mA
- Idle Mode: 0.35 mA
- Power-down Mode: < 1 μA

5.1.4 Layout

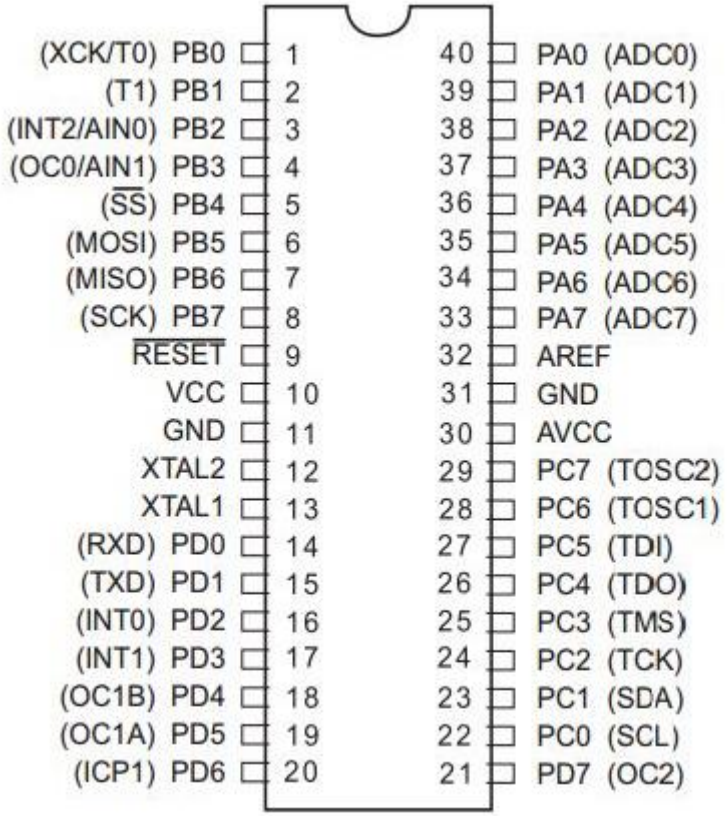

Fig. (5.1.2) Pin Diagram of Atmega16

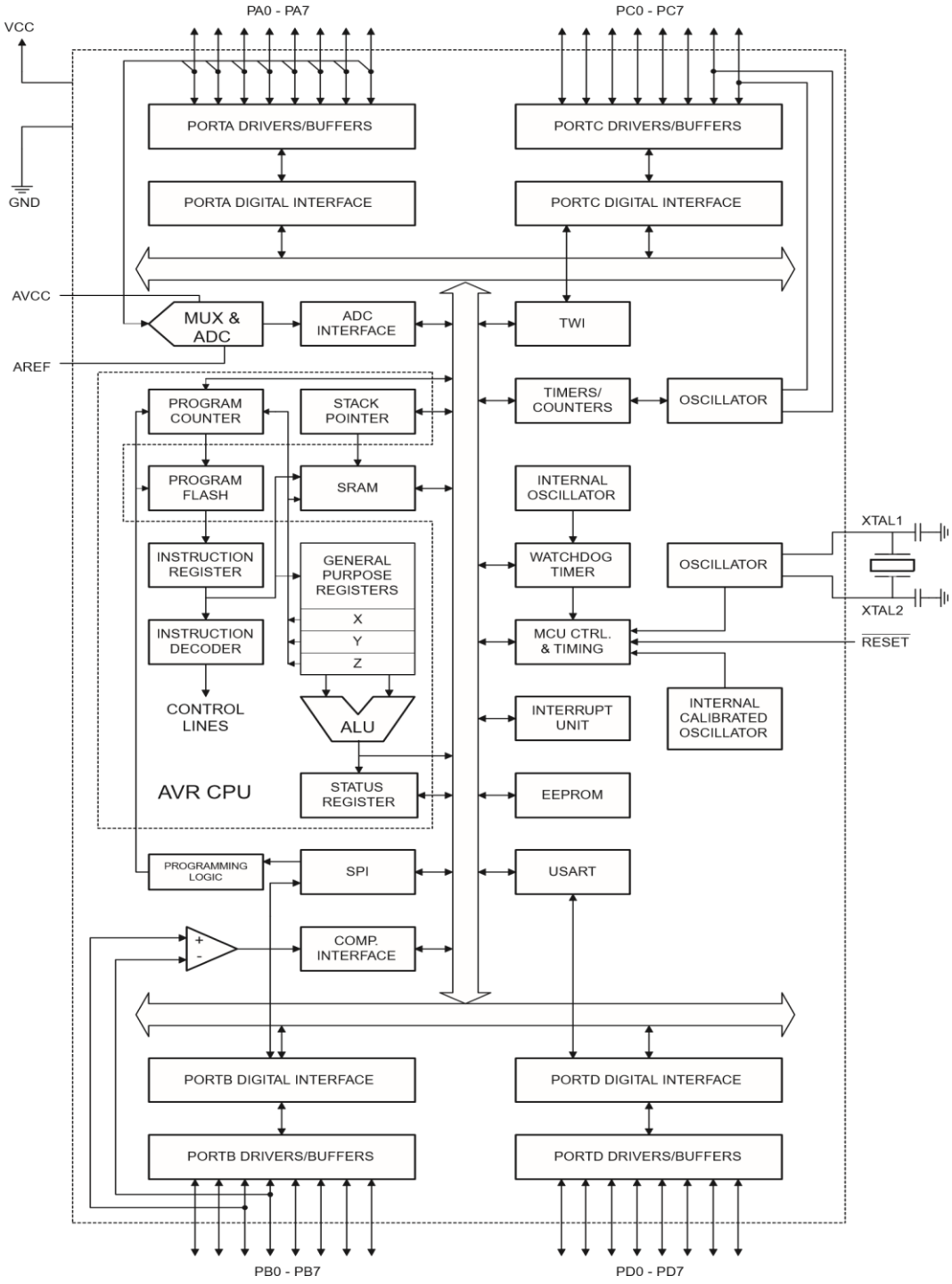

Fig. (5.1.3) Block Diagram of Atmega16

5.2 Light Emitting Diodes

It is a semiconductor diode having radioactive recombination. It requires a definite amount of energy to generate an electron-hole pair.

The same energy is released when an electron recombines with a hole. This released energy may result in the emission of photon and such a recombination. Hear the amount of energy released when the electro reverts from the conduction band to the valence band appears in the form of radiation. Alternatively the released energy may result in a series of phonons causing lattice vibration. Finally the released energy may be transferred to another electron. The recombination radiation may be lie in the infra-red and visible light spectrum. In forward is peaked around the band gap energy and the phenomenon is called injection luminescence. In a junction biased in the avalanche break down region, there results a spectrum of photons carrying much higher energies. Almost White light then gets emitted from micro-plasma breakdown region in silicon junction. Diodes having radioactive recombination are termed as Light Emitting Diode, abbreviated as LEDs.

In gallium arsenide diode, recombination is predominantly a radiation recombination and the probability of this radioactive recombination far exceeds that in either germanium or silicon. Hence Ga As LED has much higher efficiency in terms of photons emitted per carrier. The internal efficiency of Ga As LED may be very close to 100% but because of high index of refraction, only a small fraction of the internal radiation can usually come out of the device surface. In spite of this low efficiency of actually radiated light, these LEDs are efficiency used as light emitters in visual display units and in optically coupled circuits, the efficiency of light generation increases with the increase of injected current and with decreases in temperature. The light so generated is concentrated near the junction since most of the charge carriers are obtained within one diffusion length of the diode junction.

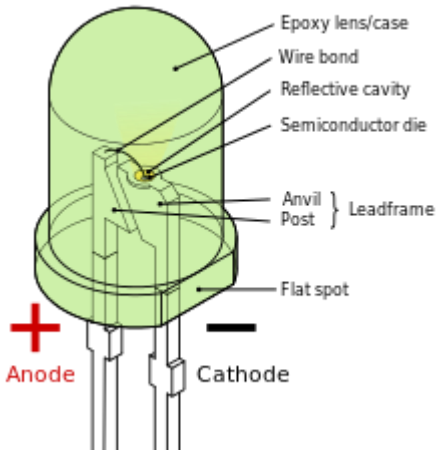

Fig. (5.2.1) Structure of an LED

5.3 IR Sensor

An infrared sensor is an electronic device, which emits in order to sense some aspects of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. These types of sensors measures only infrared radiation, rather than emitting it that is called as a passive IR sensor. Usually in the infrared spectrum, all the objects radiate some form of thermal radiations. These types of radiations are invisible to our eyes that can be detected by an infrared sensor. The emitter is simply an IR LED (Light Emitting Diode) and the detector is simply an IR photodiode which is sensitive to IR light of the same wavelength as that emitted by the IR LED. When IR light falls on the photodiode, the resistances and these output voltages, change in proportion to the magnitude of the IR light received. An infrared sensor circuit is one of the basic and popular sensor module in an electronic device. This sensor is analogous to human's visionary senses, which can be used to detect obstacles and it is one of the common applications in real time.

5.3.1 Photodiode

A photodiode is a type of photo detector capable of converting light into either current or voltage, depending upon the mode of operation.

Photodiodes are similar to regular semiconductor diodes except that they may be either exposed (to detect vacuum UV or X-rays) or packaged with a window or optical fibre connection to allow light to reach the sensitive part of the device. Many diodes designed for use specifically as a photodiode will also use a PIN junction rather than the typical PN junction.

A photodiode is a PN junction or PIN structure. When a photon of sufficient energy strikes the diode, it excites an electron thereby creating a mobile electron and a positively charged electron hole. If the absorption occurs in the junction's depletion region, or one diffusion length away from it, these carriers are swept from the junction by the built-in field of the depletion region. Thus holes move toward the anode, and electrons toward the cathode, and a photocurrent is produced.

The material used to make a photodiode is critical to defining its properties, because only photons with sufficient energy to excite electrons across the material's band gap will produce significant photocurrents.

6 SIMULATION

6.1 Simulation Layout

In the simulation layout shown in Fig. (6.1.1) below, the interconnection between the two intersections are shown.

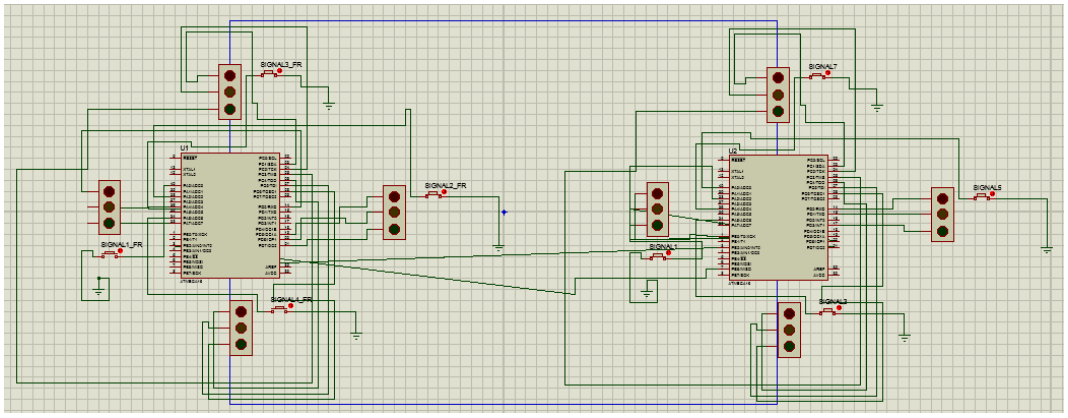

Fig. (6.1.1) Overview of simulation circuit

6.2 Working layout of simulation

Fig. (6.2.1) below shows one of the cases wherein the passage of vehicles is allowed. Specifically the case shown below is that of lane 1.

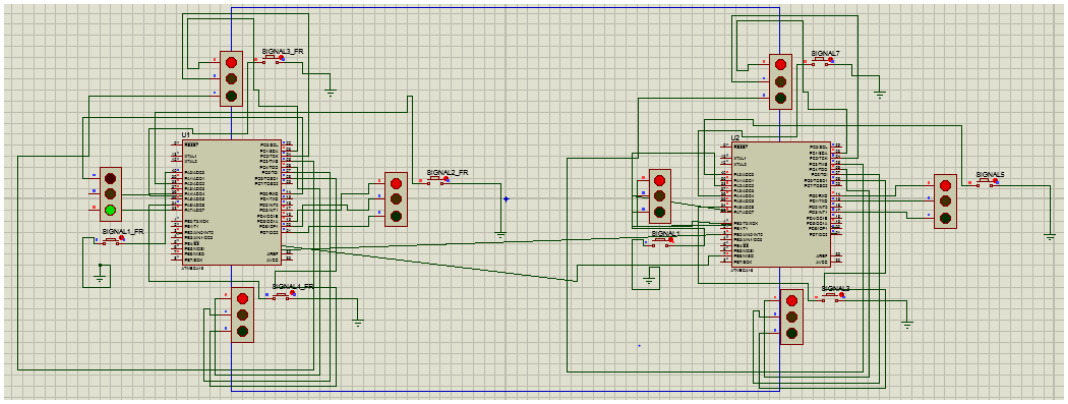

Fig. (6.2.1) Working cases of intersection 1

Fig. (6.2.2) shown below depicts the scenario wherein, after lane 1 has tuned red, a signal was sent to the adjacent intersection. Upon receiving the signal, the 2nd microcontroller calculates the delay required before it has to be interrupted.

Once the time elapses, the intersection allows passage to lane 1 and lane 2 thereby allowing a smooth flow for the vehicles coming from the previous intersection.

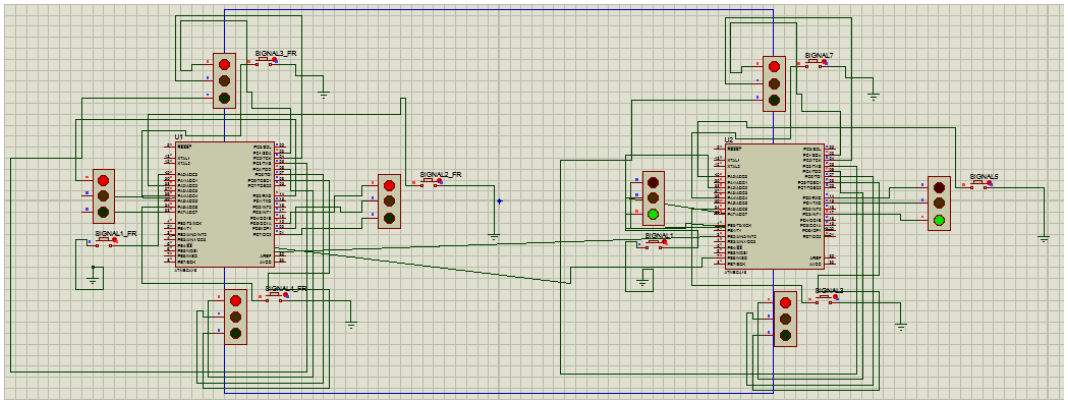

Fig. (6.2.2) Interrupt triggered in intersection 2

The Fig. (6.2.3) below shows the scenario wherein lane 3 of the 2nd intersection

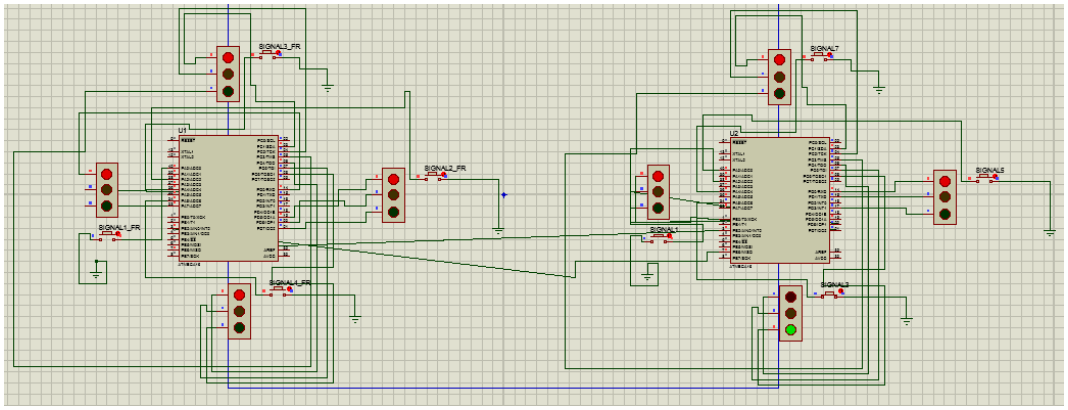

Fig. (6.2.3) Lane 3 triggered in intersection 2

Fig. (6.2.4) shows the similar scenario of Fig. (6.2.2) wherein the signal has caused the interrupt after the calculated delay.

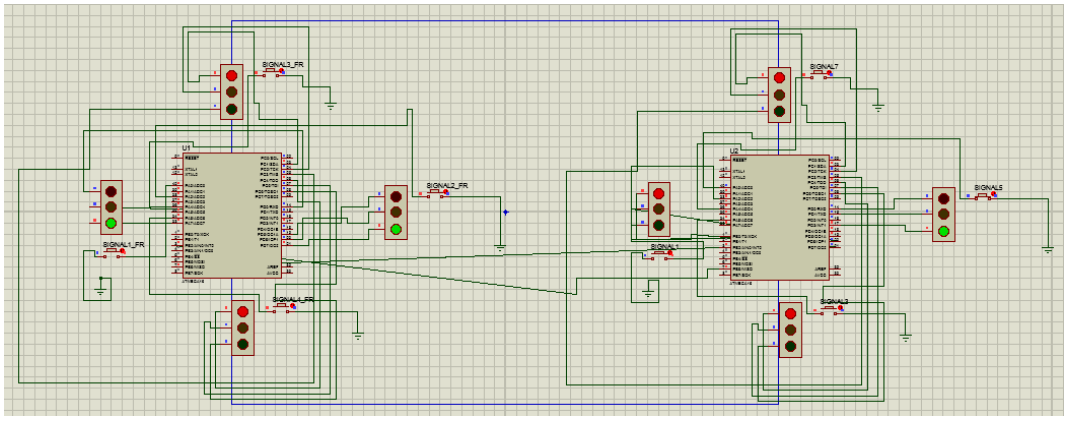

Fig. (6.2.4) Interrupt triggered on intersection 1

7 HARDWARE

7.1 Hardware Layout

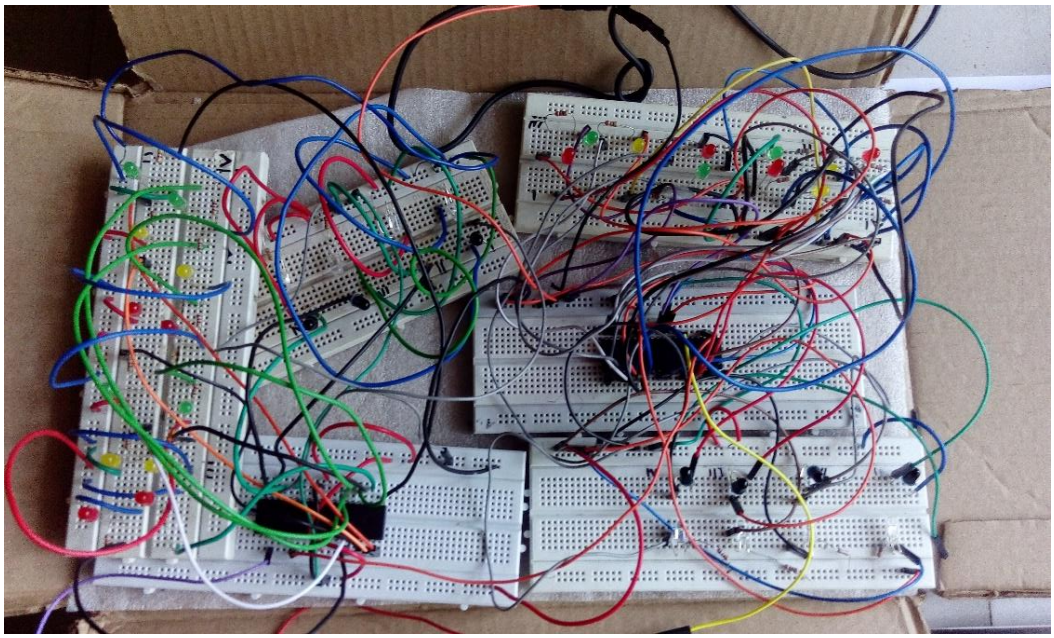

Fig. (7.1.1) Hardware layout of the circuit

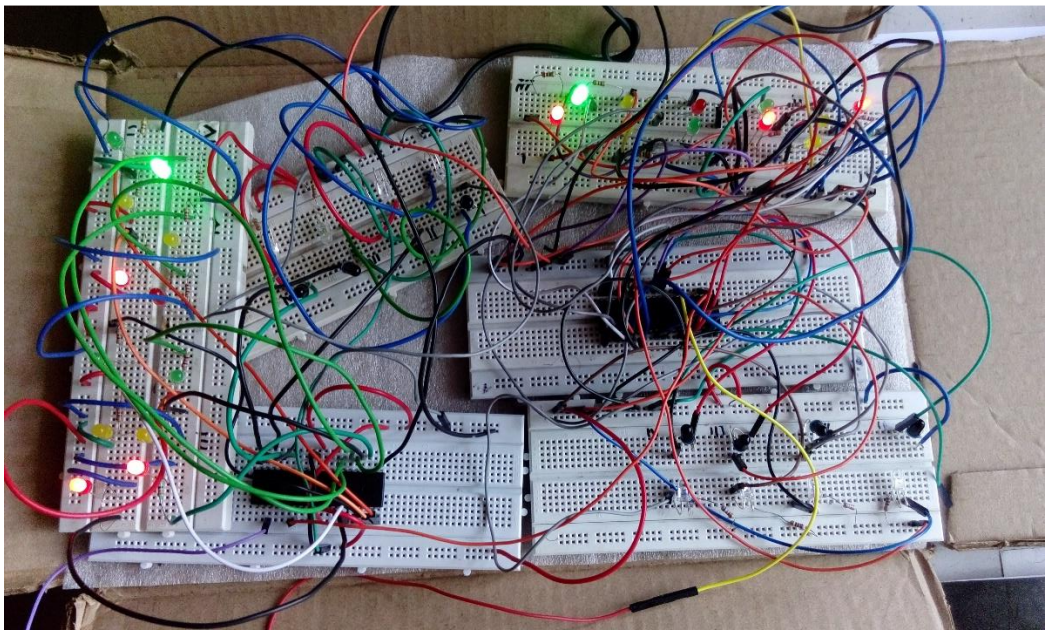

Fig. (7.1.2) Lane 3 of intersection 1 and Lane 4 of intersection 2

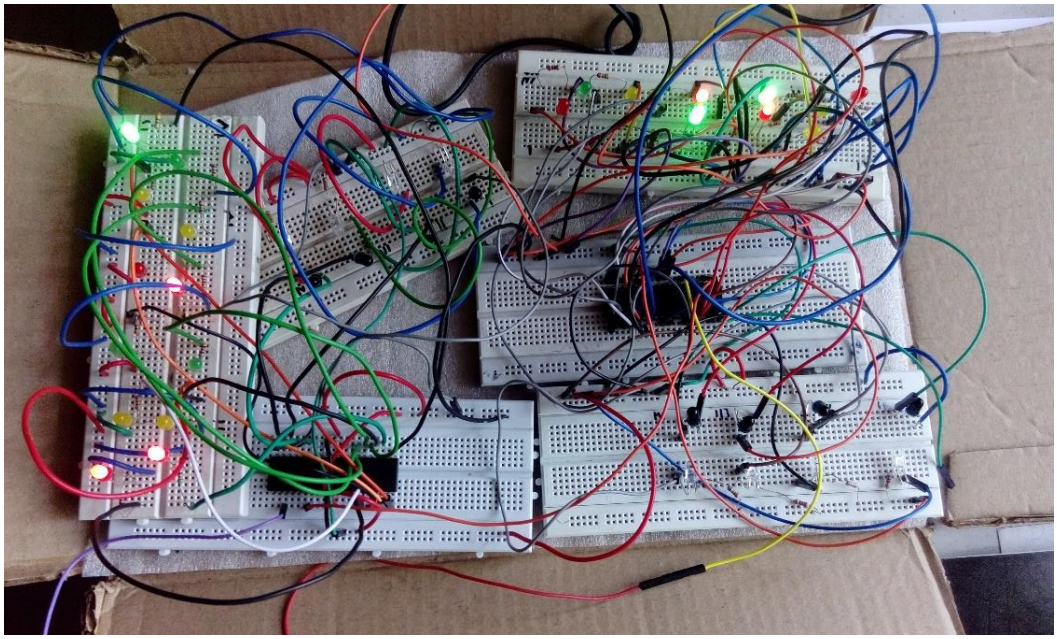

Fig. (7.1.3) Interrupt triggered in intersection 2

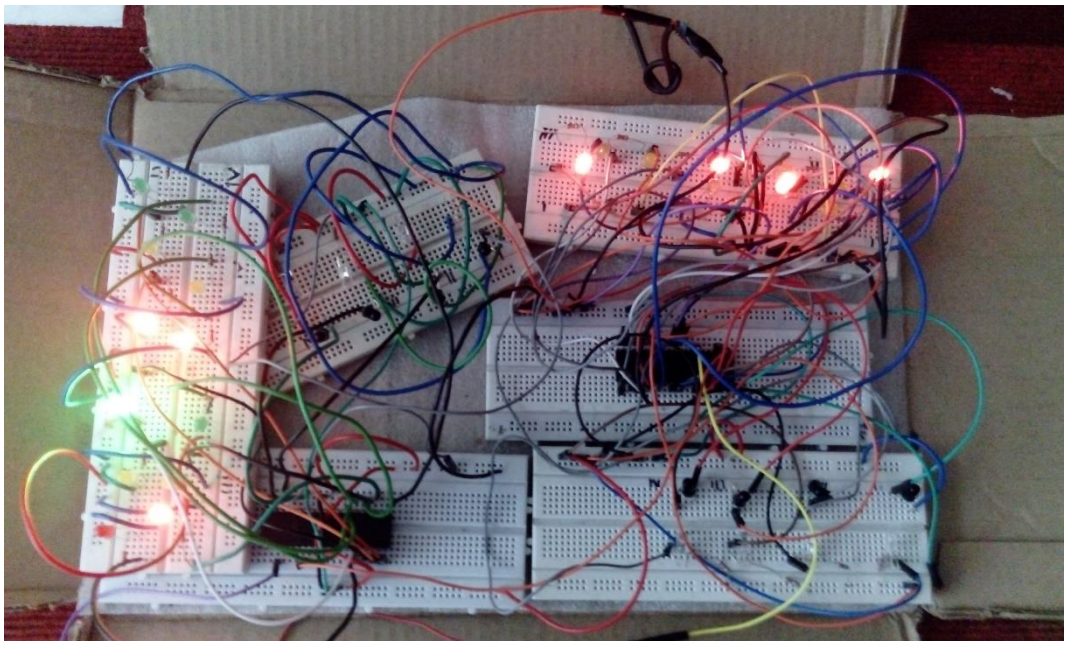

Fig. (7.1.4) Lane 1 of intersection 2 triggered

8 CONCLUSION

8.1 Conclusion

This project proposes a new and basic system to control traffic lights in an efficient manner. The system implemented here is a combination of already existing or proposed systems of Intelligent Traffic Light Controllers (ITLC) and Arterial Traffic Light (ATL) control system. Reduction of waiting time at each intersection with a traffic signal as well as communication between adjacent traffic lights so that the movement of traffic lights along the main road without any hindrance from one end to the other end of the designated road network area will make the movement of vehicles more easily and also decrease the probability of congestion and pile up at a set of traffic lights.

8.2 Future Scope

Since the project has been designed as a basic system for integration of two systems, this design can be further improved in the future according to the various needs and requirements of the road networks as well as each intersection. The additions to this design can include, but not limited to the following:

- Triggering of interrupt to make it favourable for emergency service vehicles to pass through.
- Measuring the density of vehicles at each lane and giving priority or more passage time to that particular lane.
- Integration of pedestrian traffic lights triggered by manual switch input when pedestrian crossing is necessary.
- Keeping track of vehicles prohibited from entering the road network

REFERNCES

[1] M. B. Y. Younes and A. Boukerche, "Intelligent Traffic Light Controlling Algorithms Using Vehicular Networks," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 8, pp. 5887-5899, 2016.

[2] L. Bhaskar, A. Sahai, D. Sinha, G. Varshney and T. Jain, "Intelligent Traffic Light Controller Using Inductive Loops for Vehicle Detection," in *International Conference on Next Generation Computing Technologies*, Dehradun, India, 2015.

[3] M. A. Mazidi, S. Naimi and S. Naimi, AVR Microcontroller and Embedded Systems The: Using Assembly and C, Prentce Hall, 2011.

[4] B. Ghazal, K. ElKhatib, K. Chahine and M. Kherfan, "Smart Traffic Light Control System," in *Third Conference on Electrical, Electronics, Computer Engineering and their Applications*, Beirut, Lebanon, 2016.

[5] J. Pang, "Intelligent Traffic Light Controller Design Using FPGA," in *IEEE Conference on Consumer Electronics*, Berlin, Germany, 2016.

[6] S. S. Chavan, R. Dr. Deshpande and J. Rana, "Design of Intelligent Traffic Light Controller Using Embedded System," in *Second International Conference on Emerging Trends in Engineering and Technology*, Nagpur, India, 2009.

[7] EasyCalculation.com, "AVR Timer Calculator," [Online]. Available: https://www.easycalculation.com/engineering/electrical/avr-timer-calculator.php. [Accessed May 2017].

[8] maxEmbedded, "maxEmbedded.com, a guide to robotics, embedded electronics and computer vision," August 2013. [Online]. Available: http://www.maxembedded.com. [Accessed May 2017].

APPENDIX 1

1.1 Atmega16 Pin Description

Pin No.	Pin name	Description	Alternate Function
1	(XCK/T0) PB0	I/O PORTB, Pin 0	T0: Timer0 External Counter Input. XCK : USART External Clock I/O
2	(T1) PB1	I/O PORTB, Pin 1	T1:Timer1 External Counter Input
3	(INT2/AIN0) PB2	I/O PORTB, Pin 2	AIN0: Analog Comparator Positive I/P INT2: External Interrupt 2 Input
4	(OC0/AIN1) PB3	I/O PORTB, Pin 3	AIN1: Analog Comparator Negative I/P OC0 : Timer0 Output Compare Match Output
5	(SS) PB4	I/O PORTB, Pin 4	In System Programmer (ISP) Serial Peripheral Interface (SPI)
6	(MOSI) PB5	I/O PORTB, Pin 5	
7	(MISO) PB6	I/O PORTB, Pin 6	
8	(SCK) PB7	I/O PORTB, Pin 7	
9	RESET	Reset Pin, Active Low Reset	
10	Vcc	Vcc = +5V	
11	GND	GROUND	
12	XTAL2	Output to Inverting Oscillator Amplifier	

Pin No.	Pin name	Description	
13	XTAL1	Input to Inverting Oscillator Amplifier	
14	(RXD) PD0	I/O PORTD, Pin 0	USART Serial Communication Interface
15	(TXD) PD1	I/O PORTD, Pin 1	
16	(INT0) PD2	I/O PORTD, Pin 2	External Interrupt INT0
17	(INT1) PD3	I/O PORTD, Pin 3	External Interrupt INT1
18	(OC1B) PD4	I/O PORTD, Pin 4	PWM Channel Outputs
19	(OC1A) PD5	I/O PORTD, Pin 5	
20	(ICP) PD6	I/O PORTD, Pin 6	Timer/Counter1 Input Capture Pin
21	PD7 (OC2)	I/O PORTD, Pin 7	Timer/Counter2 Output Compare Match Output
22	PC0 (SCL)	I/O PORTC, Pin 0	TWI Interface
23	PC1 (SDA)	I/O PORTC, Pin 1	
24	PC2 (TCK)	I/O PORTC, Pin 2	JTAG Interface
25	PC3 (TMS)	I/O PORTC, Pin 3	
26	PC4 (TDO)	I/O PORTC, Pin 4	
27	PC5 (TDI)	I/O PORTC, Pin 5	

Pin No.	Pin name	Description	Alternate Function
28	PC6 (TOSC1)	I/O PORTC, Pin 6	Timer Oscillator Pin 1
29	PC7 (TOSC2)	I/O PORTC, Pin 7	Timer Oscillator Pin 2
30	AVcc	Voltage Supply = Vcc for ADC	
31	GND	GROUND	
32	AREF	Analog Reference Pin for ADC	
33	PA7 (ADC7)	I/O PORTA, Pin 7	ADC Channel 7
34	PA6 (ADC6)	I/O PORTA, Pin 6	ADC Channel 6
35	PA5 (ADC5)	I/O PORTA, Pin 5	ADC Channel 5
36	PA4 (ADC4)	I/O PORTA, Pin 4	ADC Channel 4
37	PA3 (ADC3)	I/O PORTA, Pin 3	ADC Channel 3
38	PA2 (ADC2)	I/O PORTA, Pin 2	ADC Channel 2
39	PA1 (ADC1)	I/O PORTA, Pin 1	ADC Channel 1
40	PA0 (ADC0)	I/O PORTA, Pin 0	ADC Channel 0

Table [1.1.1] List of Atmega16 Pins and their functions

1.2 Atmega16 Reset and Interrupt Vectors

Sl no.	Source	Description
1	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	INT0	External Interrupt 0
3	INT1	External Interrupt Request 1
4	TIMER2_COMP	Timer/Counter2 Compare Match
5	TIMER2_OVF	Timer/Counter2 Overflow
6	TIMER1_CAPT	Timer/Counter Capture Event
7	TIMER1_COMPA	Timer/Counter1 Compare Match A
8	TIMER1_COMPB	Timer/Counter1 Compare MatchB
9	TIMER1_OVF_vect	Timer/Counter1 Overflow
10	TIMER0_OVF	Timer/Counter0 Overflow
11	SPI_STC	Serial Transfer Complete

Sl no.	Source	Description
12	USART_RXC	USART, Rx Complete
13	USART_UDRE	USART Data Register Empty
14	USART_TXC	USART, Tx Complete
15	ADC	ADC Conversion Complete
16	EE_RDY	EEPROM Ready
17	ANA_COMP	Analog Comparator
18	TWI	2-wire Serial Interface
19	INT2	External Interrupt Request 2
20	TIMER0_COMP	Timer/Counter0 Compare Match
21	SPM_RDY	Store Program Memory Ready

Table [1.2.1] Reset and Vector Interrupts

APPENDIX 2

2.1 Status Register of Atmega16

SREG – AVR Status Register

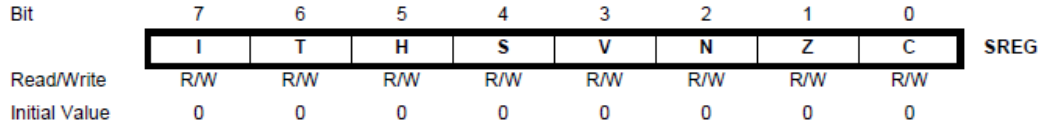

Fig. (2.1.1) Atmega16 status register

- Bit 7 – I: Global Interrupt Enable

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions.

- Bit 6 – T: Bit Copy Storage

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- Bit 5 – H: Half Carry Flag

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry is useful in BCD arithmetic.

- Bit 4 – S: Sign Bit, $S = N \oplus V$

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V.

- Bit 3 – V: Two's Complement Overflow Flag

The Two's Complement Overflow Flag V supports two's complement arithmetic's.

- Bit 2 – N: Negative Flag

The Negative Flag N indicates a negative result in an arithmetic or logic operation.

- Bit 1 – Z: Zero Flag

The Zero Flag Z indicates a zero result in an arithmetic or logic operation.

- Bit 0 – C: Carry Flag

The Carry Flag C indicates a carry in an arithmetic or logic operation.

2.2 TCCR0 – Timer/Counter 1 Control Register

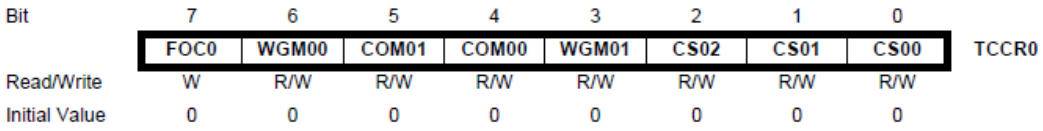

Fig. (2.2.1) TCCR0 Register Bits

- Bit 7 – FOC0: Force Output Compare

The FOC0 bit is only active when the WGM00 bit specifies a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0 is written when operating in PWM mode. When writing a logical one to the FOC0 bit, an immediate compare match is forced on the Waveform Generation unit. The OC0 output is changed according to its COM01:0 bits setting. Note that the FOC0 bit is implemented as a strobe. Therefore it is the value present in the COM01:0 bits that determines the effect of the forced compare.

A FOC0 strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0 as TOP. The FOC0 bit is always read as zero.

- Bit 3, 6 – WGM0[1:0]: Waveform Generation Mode

These bits control the counting sequence of the counter, the source for the maximum (TOP) counter value, and what type of Waveform Generation to be used. Modes of operation supported by the Timer/Counter unit are:

Normal mode, Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes.

- Bit 5:4 – COM01:0: Compare Match Output Mode

These bits control the Output Compare pin (OC0) behaviour. If one or both of the COM01:0 bits are set, the OC0 output overrides the normal port functionality of the

I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0 pin must be set in order to enable the output driver.

- Bit 2:0 – CS02:0: Clock Select

The three Clock Select bits select the clock source to be used by the Timer/Counter.

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter Stopped)
0	0	1	clk/(No prescaling)
0	1	0	clk/8 (From prescaler)
0	1	1	clk/64 (From prescaler)
1	0	0	clk/256 (From prescaler)
1	0	1	clk/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Table [2.2.1] Timer0 Clock Select Bits

2.3 TCCR1A- Timer/Counter1 Control Register A

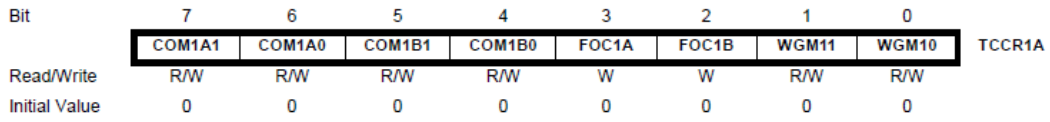

Fig. (2.3.1) TCCR1A Register Bits

- Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A
- Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B

The COM1A1:0 and COM1B1:0 control the Output Compare pins (OC1A and OC1B respectively) behaviour. If one or both of the COM1A1:0 bits are written to one, the OC1A output overrides the normal port functionality of the I/O pin it is connected to. If one or both of the COM1B1:0 bit are written to one, the OC1B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC1A or OC1B pin must be set in order to enable the output driver.

• Bit 3 – FOC1A: Force Output Compare for Channel A

- Bit 2 – FOC1B: Force Output Compare for Channel B

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. However, for ensuring compatibility with future devices, these bits must be set to zero when TCCR1A is written when operating in a PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the Waveform Generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1x1:0 bits that determine the effect of the forced compare.

A FOC1A/FOC1B strobe will not generate any interrupt nor will it clear the timer in Clear Timer on Compare match (CTC) mode using OCR1A as TOP.

The FOC1A/FOC1B bits are always read as zero.

- Bit 1:0 – WGM11:0: Waveform Generation Mode

Combined with the WGM13:2 bits found in the TCCR1B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used.

2.4 TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. (2.4.1) TCCR1B Register Bits

- Bit 7 – ICNC1: Input Capture Noise Canceler

Setting this bit (to one) activates the Input Capture Noise Canceler. When the Noise Canceler is activated, the input from the Input Capture Pin (ICP1) is filtered. The filter function requires four successive equal valued samples of the ICP1 pin for changing its output. The Input Capture is therefore delayed by four Oscillator cycles when the Noise Canceler is enabled.

- Bit 6 – ICES1: Input Capture Edge Select

This bit selects which edge on the Input Capture Pin (ICP1) that is used to trigger a capture event. When the ICES1 bit is written to zero, a falling (negative) edge is

used as trigger, and when the ICES1 bit is written to one, a rising (positive) edge will trigger the capture. When a capture is triggered according to the ICES1 setting, the counter value is copied into the Input Capture Register (ICR1). The event will also set the Input Capture Flag (ICF1), and this can be used to cause an Input Capture Interrupt, if this interrupt is enabled.

- Bit 5 – Reserved Bit

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCR1B is written.

- Bit 4:3 – WGM13:2: Waveform Generation Mode

See TCCR1A Register description.

- Bit 2:0 – CS12:0: Clock Select

The three Clock Select bits select the clock source to be used by the Timer/Counter.

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter Stopped)
0	0	1	clk/(No prescaling)
0	1	0	clk/8 (From prescaler)
0	1	1	clk/64 (From prescaler)
1	0	0	clk/256 (From prescaler)
1	0	1	clk/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Table [2.4.1] Timer1 Clock Select Bits

2.5 TIMSK – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. (2.5.1) TIMSK Register Bits

- Bit 5 – TICIE1: Timer/Counter1, Input Capture Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Input Capture Interrupt is enabled. The corresponding Interrupt Vector is executed when the ICF1 Flag, located in TIFR, is set.

- Bit 4 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A match interrupt is enabled. The corresponding Interrupt Vector is executed when the OCF1A Flag, located in TIFR, is set.

- Bit 3 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B match interrupt is enabled. The corresponding Interrupt Vector is executed when the OCF1B Flag, located in TIFR, is set.

- Bit 2 – TOIE1: Timer/Counter1, Overflow Interrupt Enable

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow Interrupt is enabled.

2.6 TIFR – Timer/Counter Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Fig. (2.6.1) TIFR Register Bits

- Bit 5 – ICF1: Timer/Counter1, Input Capture Flag

This flag is set when a capture event occurs on the ICP1 pin. When the Input Capture Register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 Flag is set when the counter reaches the TOP value. ICF1 is automatically

cleared when the Input Capture Interrupt Vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- Bit 4 – OCF1A: Timer/Counter1, Output Compare A Match Flag

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register A (OCR1A). Note that a Forced Output Compare (FOC1A) strobe will not set the OCF1A Flag. OCF1A is automatically cleared when the Output Compare Match A Interrupt Vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- Bit 3 – OCF1B: Timer/Counter1, Output Compare B Match Flag

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the Output Compare Register B (OCR1B). Note that a forced output compare (FOC1B) strobe will not set the OCF1B Flag. OCF1B is automatically cleared when the Output Compare Match B Interrupt Vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

• Bit 2 – TOV1: Timer/Counter1, Overflow Flag

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 Flag is set when the timer overflows.

2.7 Timer Loading

$$Timer\ Count = \frac{Required\ Delay}{Clock\ Time\ Period} - 1$$

Fig. (2.7.1) Formula to calculate the Upper Limit for a required delay

Timer	Resolution	Available prescalars	Maximum delay (With 1MHz clock and 1024 prescalar)
TIMER0	8 bit	1,8,64,256,1024	0.262s
TIMER1	16 bit	1,8,64,256,1024	67.1s
TIMER2	8 bit	1,8,64,256,1024	0.262s

Table [2.7.1] Timer details of Atmega16