ABSTRACT

ABSTRACT

The project involves a traffic light system at an intersection, with automatic switching based on the infrared sensors output. The sensors are placed at specific lane intersections and are interfaced to a microcontroller whereby they measure the density of traffic at a particular lane and only opens (traffic light goes green) that lane with highest density, while closing the rest (traffic light goes red). It also features a timer which regulates the amount of time a lane stays open, preventing 'starvation' of the other lanes in case the number of cars are too excessive. Thus this system is a multistate input-timed system whereby it controls the traffic based on the density which is inputted by the sensor and a timer which regulates the amount of time between each state. The microcontroller makes the decisions for the system, whereas the sensors are used for object detection.

DEDICATION

DEDICATION

This report is dedicated to our fathers, who taught us that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to our mothers, who taught us that even the largest task can be accomplished if it is done one step at a time.

Also we devote the work of this traffic light project to the respectable and honorable lecturers at Maseno University, who taught and supported us in developing our personality as competent professionals and finally to all the students at the School of Computing and Informatics in Maseno University!

ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

This project was possible thanks to the cooperation and support from a number of people, who have enabled us to gain much more than what the scholastic or project aspects of the program could have given. We are grateful to them all, and would like to express our appreciation to the following people:

- Mr. John Alwala (Our Supervisor), the chairman of department of computer science, from Maseno University, for supervising our progress with the project and for the valuable guidance he has given us in the writing of this report and for his sage advice, encouragement, suggestion and very insightful criticism have immensely aided us to work hard in the project
- Thank you to all the staffs and colleagues in the School of Computing and Informatics for their full support, assistance and training offered during the attachment, particularly Mr. Anthony Obongo, the Senior Lab Technician, for his steadfast support during the semester project work which was greatly needed and deeply appreciated.

Finally we would also like to thank God for the continuous grace He has shown us all through our University Education and having seen us through the successful completion of our degree.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION

INTRODUCTION

Nowadays, controlling the traffic in the metropolitan areas has become a major issue because of rapid increase in automobiles and also because of large time delays between traffic lights. So, in order to rectify this problem, we opt for a density based traffic lights system, that controls the traffic based on the density (number of vehicles) and timing.

OBJECTIVE

In this system, we will use IR (Infrared) sensors to measure the traffic density. We have to arrange one IR sensor one for each road; these sensors always sense the traffic on that particular road. All these sensors are interfaced to a microcontroller. Based on these sensors, controller detects the traffic and controls the traffic system. The important considerations in the development of traffic light control systems are Automation, Power consumption and Cost Effectiveness.

BACKGROUND INFORMATION

In the recent past that is before this invention of traffic lights the roads were pretty chaotic. Later after the inventions, during the olden days Traffic lights were controlled manually, these was really time consuming and was encountered with many errors. But these days' automation of traffic lights has emerged, which has played vital roles when it comes to safety in our everyday lives. Automatic traffic lights control system provides various benefits as opposed to the manual system like safer movement for cars to help them avoid collisions with cars and people. They help movement and help conduct an orderly flow by giving right of way to some cars and not others which helps in reducing the number of accidents and make collisions at intersections a lot less frequent. Nowadays some lights don't have detectors for example in large cities, the traffic lights may simply operate on timers! Since no matter what time of the day it is, there is going to be a lot of traffic. Whereas in the suburbs and country roads, however, detectors are common. Which may detect when cars arrive at an intersection or when too many cars are stacked up at an intersection to control the lights. These detectors may be composed of many technologies like lasers to rubber hoses filled with air, or even an inductive loop (most common) which detects changes in the inductance at the road surface to know if there is a car waiting.

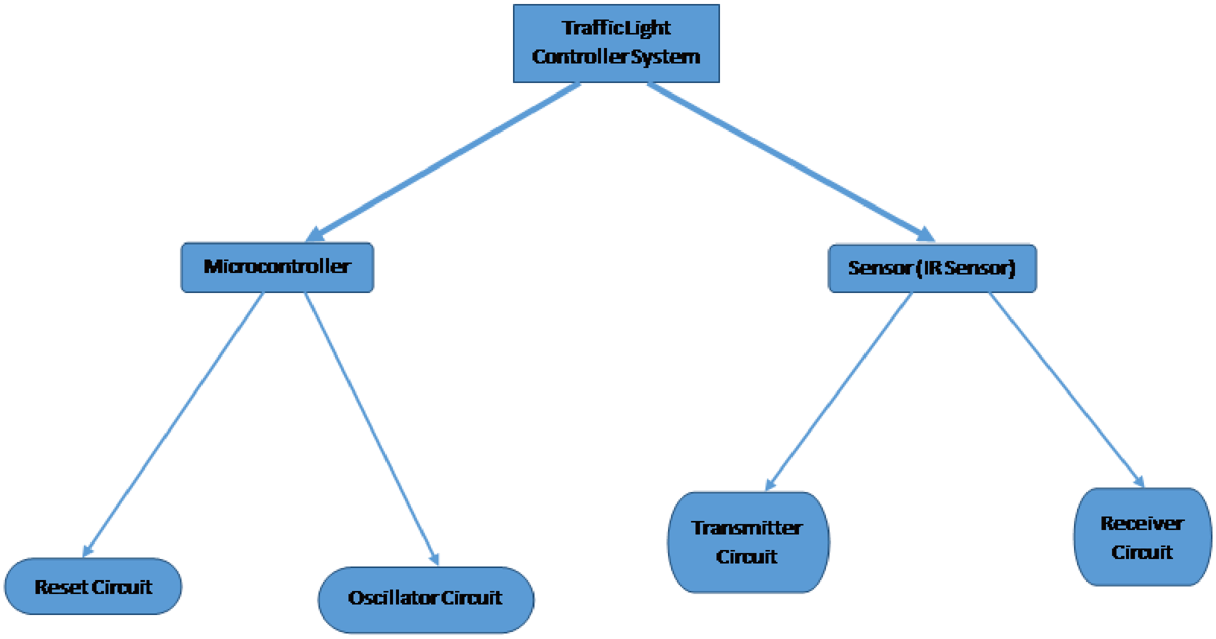
PROJECT DISCUSSIONS

PROJECT DISCUSSIONS

In our recent project report we discussed on the components that made up the circuit system, now we are going to base our discussion on two major subsystem the sensor and the microcontroller

The Circuit System subsystems are the 8051 microcontroller and The Infrared Sensor.

SYSTEM OPERATION HIERARCHY



The SENSOR (Infrared Sensor)

The infrared sensor is made up of the transmitter and the receiver circuit:

THE TRANSMITTER CIRCUIT

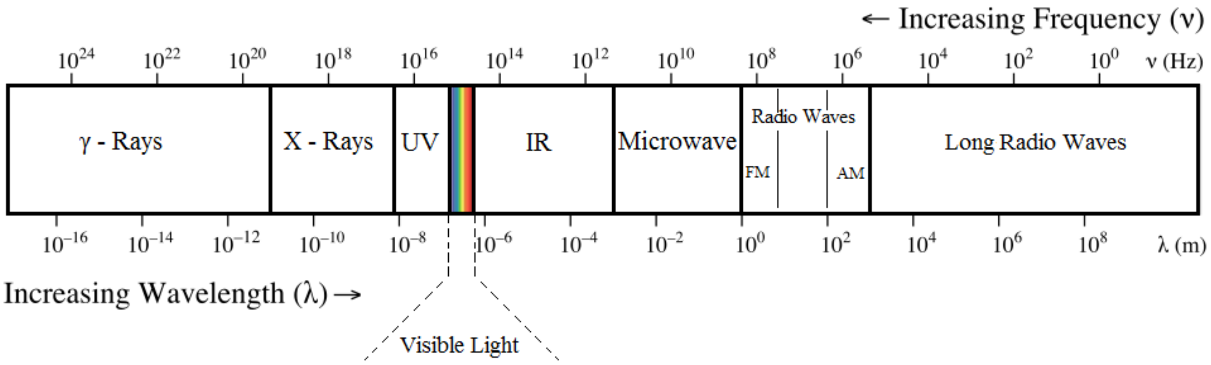
The infrared transmitter circuit is made up of the following components:

- NE555 Timer IC
- 3 Resistors*9
- 2 Ceramic Capacitors
- An Infrared LED 3mm T-1 HIR205C/H0
- Switch

- Connecting Wires
- Power Source
- Ground terminal

The IR Led

IR LED is a special type of LED that emits Infrared rays of the Electromagnetic Spectrum. The wavelength of Infrared Rays (700nm – 1mm) is greater than that of Visible light and hence they are invisible to human eye.



A typical IR LED emits infrared rays in a wavelength range of 740 – 760 nm. There are many sources of infrared light like sun, light bulbs, all hot items and even human body.

So, in order to prevent interference and false triggering, we will modulate the infrared light. The modulated signal can only be demodulated by the appropriate IR Receiver.

Features of the IR LED

- High reliability
- 2.54mm lead spacing
- Low forward voltage
- Good spectral matching to Si photodetector
- High radiant intensity

Datasheet for the LED

Parameter	Symbol	Min.	Typ.	Max	Unit	Condition
Radiant Intensity	IE	11	-	48	mW/sr	I_f=20mA

PROJECT DISCUSSIONS

Peak Wavelength	λ_P	-	850	-	nm	I_f=20mA
Spectral Bandwidth	Δλ	-	45	-	nm	I_f=20mA
Forward Voltage	V_F	-	1.45	1.65	V	I_f=20mA
		-	1.8	2.4		I_f=100mA
Reverse Current	IR	-	-	10	μA	VR=5V
View angle	2Θ1/2	-	40	-	Degree	If=20mA

555 TIMER IC

They are of two kinds:

- NE temp 0deg to 70deg Celsius
- SE temp -55deg to 125deg Celsius

The 555 IC is used to create a free running Astable oscillator to continuously produce square wave pulses.

It can be connected in either:

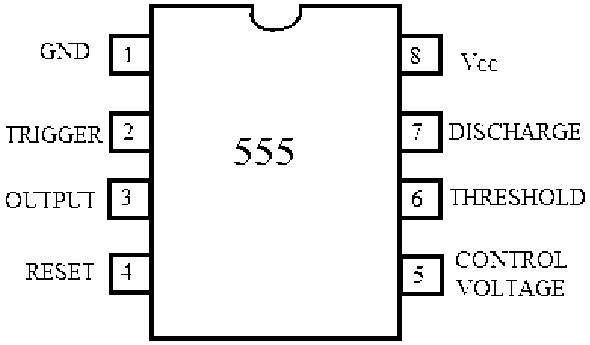
- Monostable mode-producing a precision timer for a fixed time duration
- Bistable mode – to produce flip-flop type switching action
- Astable mode- to produce a very stable 555 oscillator circuit 4 generating highly accurate free running waveforms whose output frequency can be adjusted by means of an externally connected RC tank circuit consisting of just two resistors and a capacitor.

Features of the 555

- 3 modes of operation Astable Mode (for Oscillator Circuit or Pulse Generator), Monostable mode (as a timer) and Bistable mode (as a flip flop)
- Wide varieties of power supplies from 5V to 18V
- High output current
- Adjustable duty cycle

PROJECT DISCUSSIONS

PIN OUT DIAGRAM



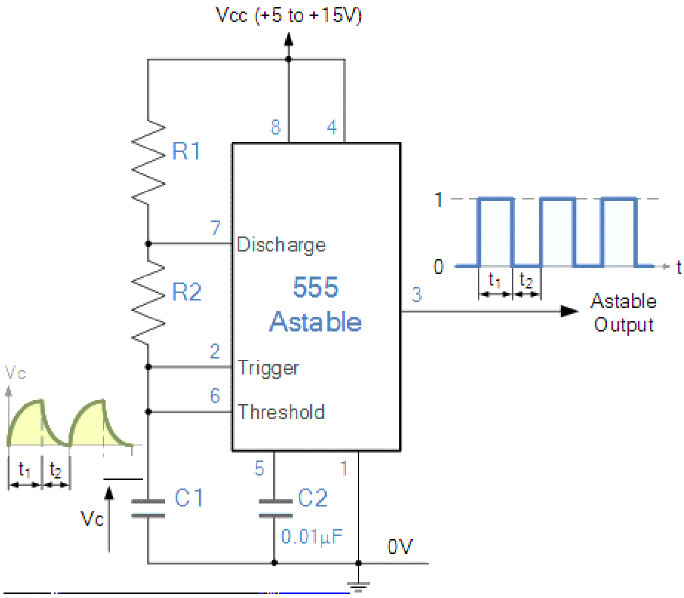
8-Pin DIP

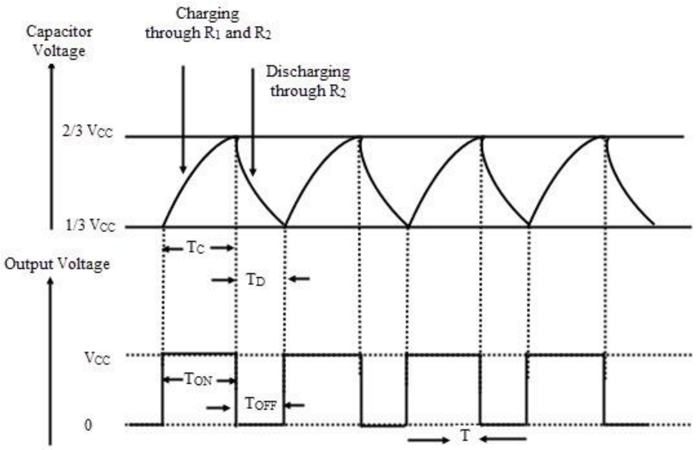
Descriptions of the Pinout

PIN		IO	DESCRIPTION
NO	**NAME**		
1	GND	O	Ground Reference Voltage
2	Trigger	I	Responsible for the transition of SR flip-flop
3	Output	O	Output Driven Waveform
4	Reset	I	A negative pulse on reset will disable or reset the timer
5	Control Voltage	I	Controls the width of the output pulse by controlling the threshold and trigger levels
6	Threshold	I	Compares the voltage applied at the terminal with reference voltage of 2/3
7	Discharge	I	Connected to open collector of the transistor which discharges a capacitor between intervals
8	VCC Supply	I	Supply Voltage

PROJECT DISCUSSIONS

THE ASTABLE OSCILLATOR CIRCUIT





In order to get the 555 oscillator to operate as an Astable multivibrator, it is necessary to continuously retrigger the 555 IC after each and every timing cycle.

The retriggering is achieved by connecting the trigger input (pin 2) and the threshold input (pin 6) together. Thus allowing the device to re-trigger itself on each and every cycle allowing it to operate as a free running oscillator, therefore it has no stable states as it continuously switched from one state to another.

Also the single timing resistor of the Monostable circuit should be split into two separate resistors R1 and R2 with their discharge connected to *discharge* input (pin 7) as shown above.

During each cycle capacitor, C1 charges up through both timing resistors, R1 and R2, but discharges itself only through resistor, R2 as the other side of R2 is connected to the *discharge* terminal, pin 7. Then the capacitor charges up to 2/3Vcc (the upper comparator limit) which is determined by the 0.693(R1+R2) C1 combination and discharges itself down to 1/3Vcc (the lower comparator limit) determined by the 0.693(R2*C1) combination. This results in an output waveform whose voltage level is approximately equal to VCC – 1.5V and whose output “ON” and “OFF” time periods are determined by the capacitor and resistors combinations. The individual times required to complete one charge and discharge cycle of the output is therefore given as:

Astable 555 oscillator charge and discharge times:

PROJECT DISCUSSIONS

$t1=0.693(R_1+R_2).C1$ ~ High

And

$t2=0.693*R_2*C1$ ~ Low

Where R is in Ohms

C is in Farads

NB: Vdiode=0.693V, which can be determined from datasheet or by testing

Therefore When connected as an Astable multivibrator, the output from the **555 Oscillator** will continue indefinitely charging and discharging between 2/3Vcc and 1/3Vcc until the power supply is removed.
The duration of one full timing cycle is therefore equal to the sum of the two individual times that the capacitor charges and discharges added together and is given as:

555 Oscillator cycle time

$T=t1+t2=0.693(R_1+2R_2)*C$

The output frequency of oscillations can be found by inverting the equation above for the total cycle time giving a final equation for the output frequency of an Astable 555 Oscillator as:

555 Oscillator Frequency Equation:

$$F=\frac{1}{T}=\frac{1.44}{(R1+2R2)*C}$$

By altering the time constant of just one of the RC combinations, the **Duty Cycle** (Mark-to-Space) ratio of the output waveform can be accurately set and is given as the ratio of resistor R2 to resistor R1. The Duty Cycle for the 555 Oscillator, which is the ratio of the "ON" time divided by the "OFF" time is given by:
555 Oscillator Duty Cycle

$$Duty\ Cycle=\frac{TON}{TOFF+TON}=\frac{R1+R2}{R1+2R2}\%$$

If both timing resistors, R1 and R2 are equal in value, then the output duty cycle will be 2:1 that is, 66% ON time and 33% OFF time with respect to the period.

PROJECT DISCUSSIONS

As the timing capacitor, C charges through resistors R1 and R2 but only discharges through resistor R2 the output duty cycle can be varied between 50 and 100% by changing the value of resistor R2. By decreasing the value of R2 the duty cycle increases towards 100% and by increasing R2 the duty cycle reduces towards 50%. If resistor, R2 is very large relative to resistor R1 the output frequency of the 555 Astable circuit will determined by R2 x C only.
The problem with this basic Astable 555 oscillator configuration is that the duty cycle, the "mark to-space" ratio will never go below 50% as the presence of resistor R2 prevents this. In other words we cannot make the outputs "ON" time shorter than the "OFF" time, as (R1 + R2)C will always be greater than the value of R1 x C. But one way to overcome this problem is to connect a signal bypassing diode in parallel with resistor R2, which we will not do in this project.

Lastly, a 50% duty cycle means the high time is equal to the low time. If an LED is placed at the output of this Astable circuit, it will turn on at the same span of time as it is turned off. But getting an exact 50% duty cycle is impossible with this circuit.

CHOOSING TIMING COMPONENTS FOR RC CIRCUIT IN TIMER

Choosing the right values for resistors and capacitors is necessary as the 555 timer can provide delays from microsecond to g depending on the values of R and C in the charging circuit.

When the 555 timer is operating in Astable mode, then it requires an RC circuit consisting of two resistors and a capacitor. R1 & R2 should be in the range of 1 Kilo Ohms to 1 Mega Ohms, as discussed below:

Timing Capacitor

- Choosing capacitors with large capacitances will be a problem. This is because electrolyte capacitors with large capacitances often tend to have wider tolerance limits. So the actual values and the marked values may have a significant difference.
- Large capacitance electrolyte capacitors will have high leakage currents which can affect the timing accuracy as the capacitor charges.
- It is better to avoid electrolyte capacitors that have a high working voltage rating as they do not work efficiently when operated at a voltage 10% less than their rated voltage.
- Hence, capacitors with working voltage greater than the VCC of the 555 timer should be chosen.
- Timing capacitors with capacitance less than 100pF in order to produce short output pulses may also cause problems.
- For capacitors with such low values, stray capacitance around the circuit might affect the capacitance of the timing capacitor.

PROJECT DISCUSSIONS

Timing Resistor

- When operating the 555 timer as an Astable multivibrator, the value of the timing resistor should be at least 1 Kilo Ohms. If the idea is to build a low power consumption circuit, then it is better to have higher values for the timing resistors.
- The disadvantage in choosing resistors with higher resistances as they lead to inaccuracies in timing. In order to minimize these inaccuracies, the value of the timing resistor shouldn’t be more than 1 Mega Ohms.
- Using high values of resistance > 1 Mega Ohms can increase the error between calculated and actual frequency. So a maximum of 1 Mega Ohms is recommended.
- If the value of R1 is less than 1 Kilo Ohms, there is a danger that the trigger input (pin2) may not be able to reach a low enough voltage to trigger the comparator in the 555 and so oscillations cannot take place.

Trigger Pulses

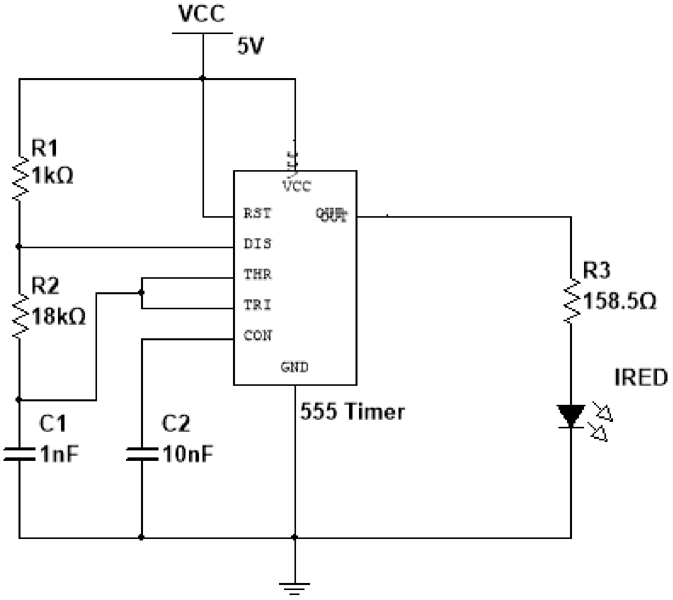
- The Pin 2 in the 555 timer is a trigger input. When the trigger input goes below the reference voltage i.e. 1/3 VCC, the output of the timer is high and the timing interval begins.
- The trigger pulse should momentarily go below the reference voltage and the duration is important as it should not be longer than the output pulse.
- Trigger pulses are generally identified by a narrow negative going spike. A differentiator circuit made from a capacitor and a resistor will produce two symmetrical spikes but a diode is used to eliminate the positive going spike.
- The duration of the pulse is determined by the differentiator circuit (i.e. it depends on the capacitor and resistor).

`

TRANSMITTER CIRCUIT CALCULATIONS

Taking the value:

- C1=1nF
- C2=10nF
- R1=1kOhms
- And we want it to generate a frequency of approximately 38kHz



Thus finding the value of R2, from the formula:

$$f = \frac{1}{T} = \frac{1.44}{(R1 + 2R2) * C}$$

$$R2 = \frac{1.44}{FC} - R1 * \frac{1}{2}$$

Thus, $R2 = \frac{1.44}{38kHZ * 1nF} - 1k\Omega * \frac{1}{2}$

$$R2 = 18\ approx\ 18k\Omega$$

Value of the current limiting resistor for the IRLED in the transmitter is given by:

$$Vf = 1.83V\ \&\ I = 20mA\ from\ the\ atasheet,$$

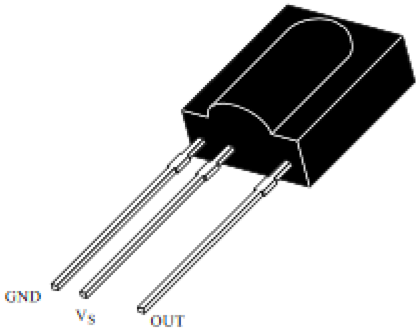
$$thus{:}\ \frac{V - Vf}{0.02} = \frac{5 - 1.83}{0.02} = 158.8ohms$$

THE RECEIVER CIRCUIT

It consists of an IR phototransistor, a diode, a MOSFET, a potentiometer and an LED. When the phototransistor receives any infrared radiation, current flows through it and MOSFET turns on. This in turn lights up the LED which acts as a load. The potentiometer is used to control the sensitivity of the phototransistor.

TSOP1738 Features:

- Photo detector and preamplifier in one package
- Output active low
- Internal filter for PCM frequency
- High immunity against ambient light
- Improved shielding against electric field disturbance
- 5 Volt supply voltage, low power consumption
- TTL and CMOS compatibility

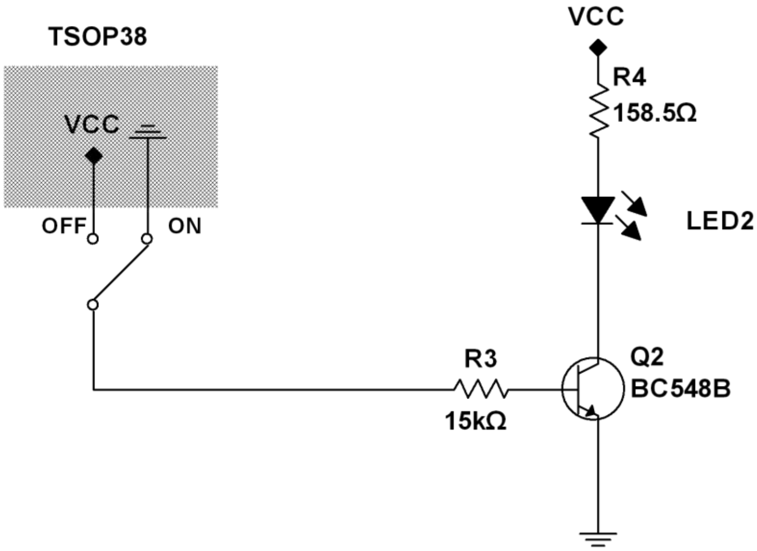


PROJECT DISCUSSIONS

- Continuous transmission possible

Circuit Components

- TSOP 1738
- LED
- 2 Resistors
- BC548B Transistor
- Breadboard



- For a RED Led:
  - Vf=1.83V &
  - I=20mA
- Vcc, supply Voltage=5V
- $R = \frac{Vs - Vf}{I}$
- $\frac{5-1.83}{0.02} = 158.5\Omega$

From the BC548 Datasheet,

Hfe(max)	150
Hfe(Min)	90
IC(Max)	0.1
Vbe(Sat)	0.7
Vce(Sat)	0.09

Geometric Average of hfe is given by:

$$hfe = \beta = \sqrt{hfe(\mathrm{max}\quad) * \mathrm{hfe}(\mathrm{min}\quad)} = \sqrt{90 * 150} = 116.19$$

When Saturated,

$$Vbe = 0.7V$$

$Collector\ Curent\ is$:

$$IC = \frac{Vcc}{Rc} = \frac{5}{158.5} = 0.0315A$$

And Since:

$$hfe = \frac{IC}{IB} \quad then\ \ IB = \frac{0.0315}{116.19} = 0.0002715A$$

And $IBRB = VCC - VBE$

$RB = \frac{VCC-VBE}{IB} = \frac{5-0.7}{0.0002715} = 15k\Omega$, value of the base resistor of the transistor.

So when TSOP 38 is LOW (**IR Signal Detected**) the transistor will be functional and it will drag the transistors output to 0V that is logic 0. Otherwise when TSOP1738 is HIGH (**No IR Signal in Range**) the output will be at 5V which is logic 1. Here is a mini table of the states:

State(IR Signal)	TSOP	Output LED	TRANSISTOR
Present	0	off	off
Absent	1	On	on

The Microcontroller will take action upon receiving a logic 0 from the receivers output.

EXPECTED TRANSMISSION RANGE FOR THE IR SENSOR

The maximum transmission distance for the IR, depends on several factors (primary factors) are:

- Radiant Intensity of the Emitter (I.e.=11Mw/sr)
- Sensitivity of the Receiver (Ee=0.5Mw/M^2)

Max transmission distance also depends on:

- Reflective conditions of the test room
- The optical transmittance of winders or light guide in front of the receiver.
- Disturbance conditions

The relation between intensity of a source and the resulting irradiance in the distance r is given by the basic square root rule law. An emitted intensity I.e. generates in a distance r the irradiance Ee = Ie/r2. This relationship is not valid under near field conditions and should be used not below a distance d smaller than 5 times the emitter source diameter.

Using a single radiation point source, one gets the following relation between the parameter E_e, ϕ_e, and r

$$Ee = \frac{d\phi_e}{dA}\left[\frac{W}{m^2}\right]$$

$$use$$

$$I_e = \frac{d\phi_e}{d\Omega}, \Omega = \frac{A}{r^2} and\ get$$

$$E_e = \frac{d\phi_e}{dA} = I_e\left(\frac{d\Omega}{dA}\right) = \frac{I_e}{r^2}\left[\frac{W}{m^2}\right]$$

$$The\ distance\ r\ resulting\ as$$

$$r = \sqrt{\frac{I_e}{E_e}} = \sqrt{\frac{Intensity}{senstivity}}$$

And since our transmitter's 3mm LED has $I_e\ of\ 11Mw/sr$ and our receivers TSOP1738 has an $E_e\ of\ O.5mW/m^2$ at a frequency of 30-40 kHz, these were obtained from their respective datasheet, thus

$$r = \sqrt{\frac{11}{0.5}} = 4.69Meters$$

Transmission distance increases as the intensity increases and as the receiver becomes more sensitive.

Where Ie is the IR-LED radiant intensity in W-sr^1 and d is the source-to-eye distance in meters.

FIGURE 8: RADIANT INTENSITY VS. FORWARD CURRENT

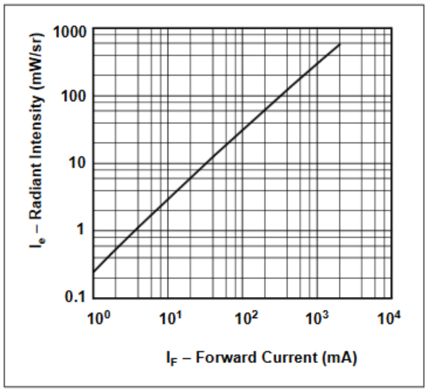
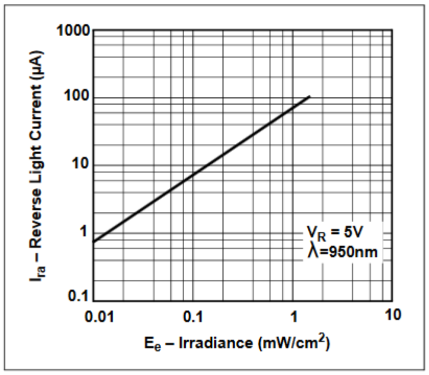


FIGURE 12: IRRADIANCE VS. CURRENT

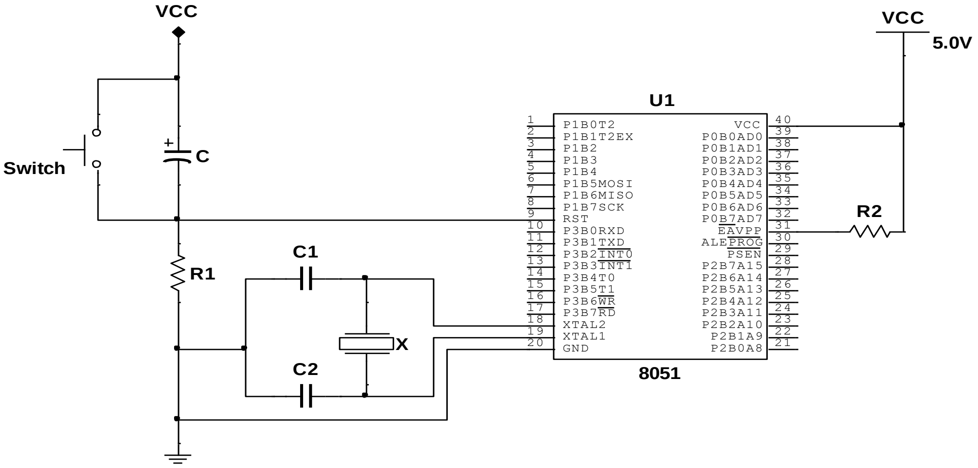


The total irradiance, Ee, can be determined from IR-LED datasheet specifications.

Sensitivity Analysis of The Infrared Sensor (i.e. between the Transmitter and the Receiver)

THE MICROCONTROLLER

Basic Circuit of the Microcontroller

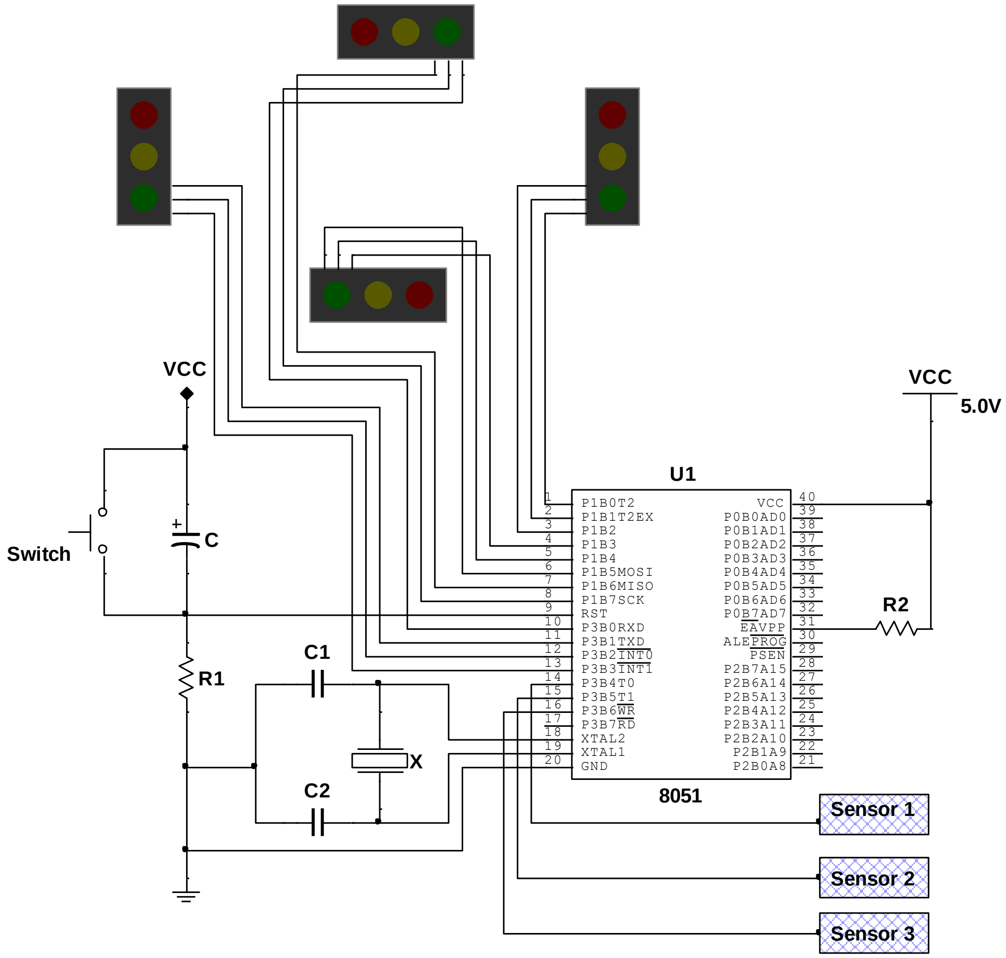


- This basic circuit of 8051 microcontrollers is the minimal interface required for it to work. The basic circuit includes a Reset Circuit, the oscillator circuit and power supply.
- First is the power supply. Pins 40 and 20 (VCC and GND) of the 8051 Microcontroller are connected to +5V and GND respectively.

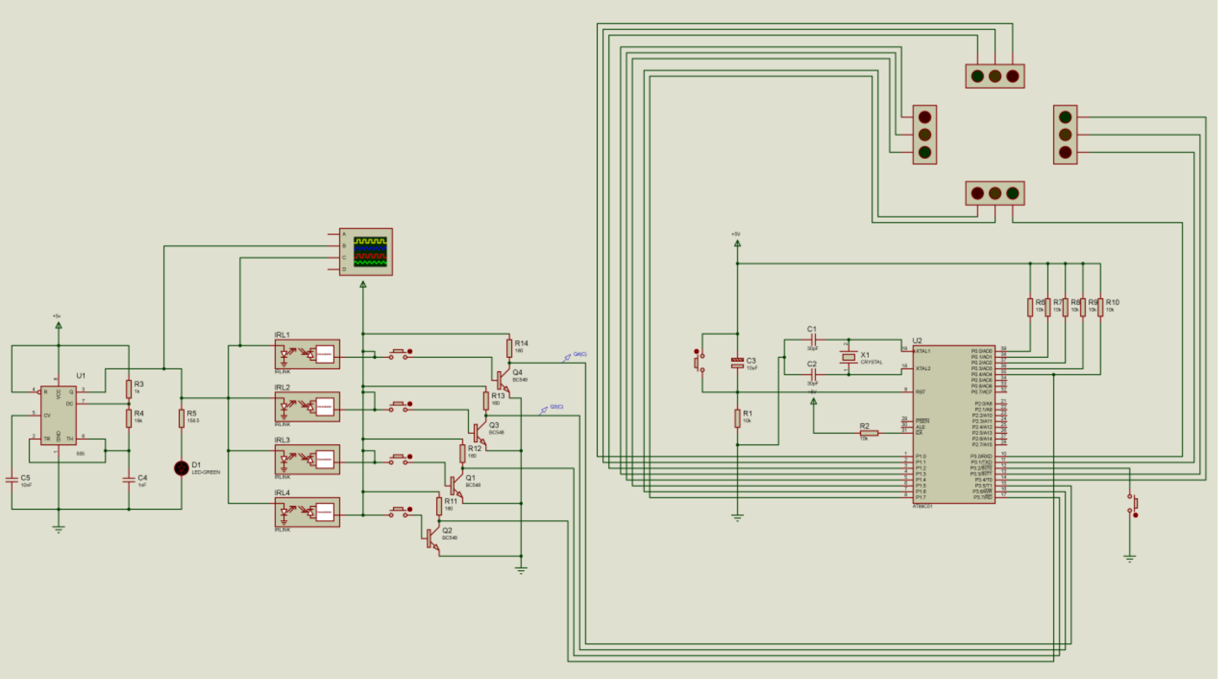
PROJECT DISCUSSIONS

- Next is the Reset Circuit. A logic HIGH (+5V) on Reset Pin for a minimum of two machine cycles (24 clock cycles) will reset the 8051 Microcontroller. The reset circuit of the 8051 Microcontroller consists of a capacitor, a resistor and a push button and this type of reset circuit provides a Manual Reset Option. If you remove the push button, then the reset circuit becomes a Power-On Reset Circuit.
- The next part of the basic circuit of the 8051 Microcontroller is the Oscillator Circuit or the Clock Circuit. A Quartz Crystal Oscillator is connected across XTAL1 and XTAL2 pins i.e. Pins 19 and 18. The capacitors C1 and C2 can be selected in the range of 20pF to 40pF.
- 8051 Microcontroller Pin Description, PORTS 1, 2 and 3, all have internal pull – ups and hence can be directly used as Bidirectional I/O Ports. But, we need to add external Pull – ups for PORT 0 Pins in order to use it as an I/O Port.

Microcontroller Circuit Diagram

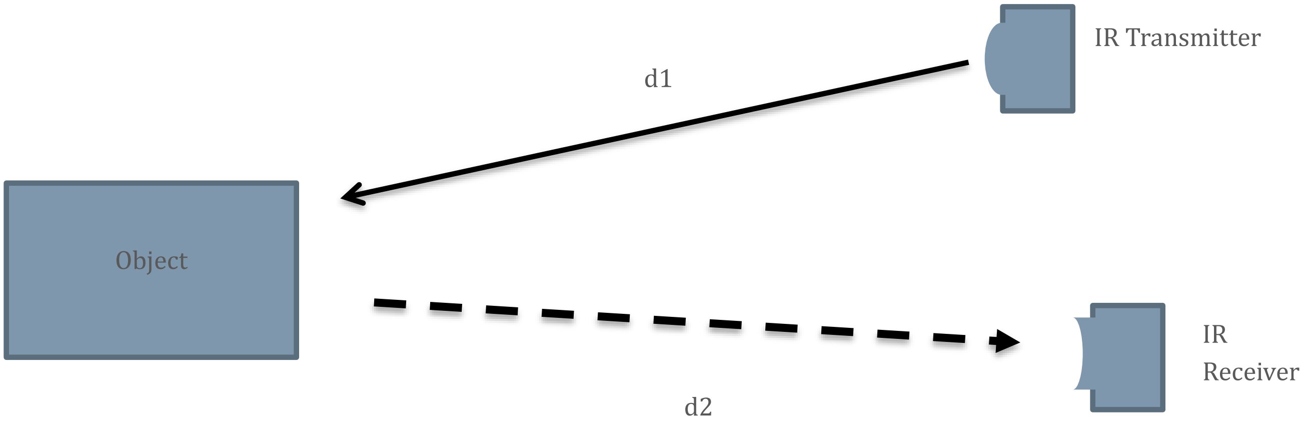


PROJECT DISCUSSIONS



Time of Flight

Time of flight is a method for measuring the distance between a sensor and an object, based on the time difference between the emissions of a signal and its return to the sensor, after being reflected by an object.



$$Total\ Distance = d1 + d2$$

PROJECT DISCUSSIONS

Time of Flight works on the principle of measuring time between the emission of a signal and its return home after being reflected by an object.

Signal travels at a constant velocity which allows the calculation of the distance.

Light is preferred to sound because of its travel speed allowing higher measuring frequency.

IR light is typically chosen because it ensures less disturbance from natural ambient light.

$$Speed\ of\ Infrared\ Light$$

$$C = 3.0 * 10^8 m/s$$

Assuming the object (a car) is moving at a speed of 50km/hour, Will our system be able to detect it in time?

Given the length of normal car, basically a saloon car is 4.5m, and its initial speed is 50km/hr. For the car to pass by without being detected it will have to cover its body length, where:

$Speed = 50kmhr$ Which is equivalent to? $50 * \frac{5}{18} = 13.8889ms$

$$timeToEscape = \frac{Distance}{Speed} = \frac{4.5meters}{13.8889} = 0.3240S \sim\ 324ms$$

$$Finding\ the\ Max\ Distance\ between\ the\ Transmitter\ and\ the\ Receiver::$$

$$d_{max} = \sqrt{\frac{I_e}{E_e}} = \sqrt{\frac{Intensity}{senstivity}}$$

And since our transmitter's 3mm LED has $I_e\ of\ 11Mw/sr$ and our receivers TSOP1738 has an $E_e\ of\ O.5mW/m^2$ at a frequency of 30-40 kHz, these were obtained from their respective datasheet, thus

$d = \sqrt{\frac{11}{0.5}} = 4.69Meters$

Since the transmitter is constantly firing the IR rays, we do not need to find the time for the light to propagate from the transmitter (T_x) to the object. The time we need to calculate is d2 that is the time for the IR to bounce off from the object to the receiver (Rx).

$$Tof1 = \frac{Object \rightarrow Receiver_{Distance}}{Velocity} = \frac{2.3}{3.0 * 10^8} = 7.6667uS$$

PROJECT DISCUSSIONS

Finding the response time of our microcontroller, since the MCU is able to execute 1Million Instructions after every second:

$$Internal\ Oscillator\ Frequency = 12MHz$$

$$External\ Oscillator\ Frequecy = 11.0592MHz$$

$$Frequency\ to\ Exec\ a\ single\ Instruction = \frac{1}{12} * 11.0592 = 0.9216MHz$$

$$Period(Time\ for\ One\ Instruction) = \frac{1}{F} = \frac{1}{0.9216} = 1.085uS$$

$$Tof2 = 1.085uS$$

Since for a full switch, each full transition takes approximately 2ms (Since our simulation system was a bit slow) that is:

GREEN → AMBERRED or → RED AMBER → GREEN →

Thus time to switch is $2ms\ \sim 2 * 1000 = 2000uS$

$Tof3 = 2000uS$

Thus the time of flight is given by:

$Tof\ = tof1 + tof2 + tof3$

$= 7.667 + 1.085 + 2000 = 2008.7517uS \sim 2.0087mS$

Hence our system is effective since:

$$System\ time < Car\ time\ i.e$$

$$2.0087mS < 324mS$$

PROJECT DISCUSSIONS

Failsafe Systems

A Fail-Safe engineering is a design feature or practice that in the event of a specific type of failure, inherently responds in a way that will cause no or minimal harm to other equipment, the environment or to people.

A system being fail-safe does not mean that failure is impossible or improbable but rather that the systems design prevents or mitigates unsafe consequences of the systems failure.

To achieve a fail-safe engineering, whereby in the event of a specific type of failure it inherently responds in a way that will cause minimal harm to the equipment, the environment or people, we have incorporated the following into our design:

- Hardware traps
- Efficient codes, capable of diagnosing the system or troubleshooting for errors. Catching and masking errors before it spreads in the entire system.
- Prevention, Detection and Correction of interlocks (undesired states) which may cause system failure.
- Failsafe engineering can be employed by redundancy, fault tolerance or recovery situations.

Traffic Light controllers use a conflict monitor unit to detect fault or conflicting signals and switch an intersection to an all flashing error signal, rather than displaying potentially dangerous conflicting signals for example, blinking amber on and off continuously.

States Changes in Traffic Light System.

- Green Light allows traffic to proceed in the direction denoted, if it is safe to do so and there is room on the other side of the intersection.
- The amber (yellow) light warns that the signal; is about to change to red. A phase where red and yellow are displayed together indicates that the signal is about to change to green.
  - A flashing amber indication is a warning signal.
- Red prohibits traffic from preceding in the given direction.
  - A flashing red indication is treated as a stop sign.
- Our system is a Multism input /timed system in which transition between the states (& behavior in each state) will depend both on the passage of time and system inputs.
- For example our system might only move to state A & B if a particular input is received with Xb seconds of system output being generate.

PROJECT DISCUSSIONS

Interrupts

An interrupt is an external or internal event that interrupts the microcontroller to inform it that a device needs its service

Interrupts can be saved in two ways:

Interrupts

- Whenever any device needs its service, the device notifies the microcontroller by sending it an interrupt signal
- Upon receiving an interrupt signal, the microcontroller interrupts whatever it is doing and serves the device
- The program which is associated with the interrupt is called the interrupt service routine (ISR) or interrupt handler

Polling

- The microcontroller continuously monitors the status of a given device
- When the conditions met, it performs the service
- After that, it moves on to monitor the next device until everyone is serviced

Polling can monitor the status of several devices and serve each of them as certain conditions are met

- The polling method is not efficient, since it wastes much of the microcontroller's time by polling devices that do not need service

The advantage of interrupts is that the microcontroller can serve many devices (not all at the same time)

- Each devices can get the attention of the microcontroller based on the assigned priority
- For the polling method, it is not possible to assign priority since it checks all devices in a round-robin fashion

Upon activation of an interrupt, the microcontroller goes through the following steps

1. It finishes the instruction it is executing and saves the address of the next instruction (PC) on the stack.
2. It also saves the current status of all the interrupts internally (i.e.: not on the stack)
3. It jumps to a fixed location in memory, called the interrupt vector table, that holds the address of the ISR
4. The microcontroller gets the address of the ISR from the interrupt vector table and jumps to it

    - It starts to execute the interrupt service subroutine until it reaches the last instruction of the subroutine which is RETI (return from interrupt)

5. Upon executing the RETI instruction, the microcontroller returns to the place where it was interrupted
    - First, it gets the program counter (PC) address from the stack by popping the top two bytes of the stack into the PC
    - Then it starts to execute from that address

WORKING OF THE SYSTEM

WORKING OF THE SYSTEM

EXPLANATIONS OF THE SYSTEM WORKING

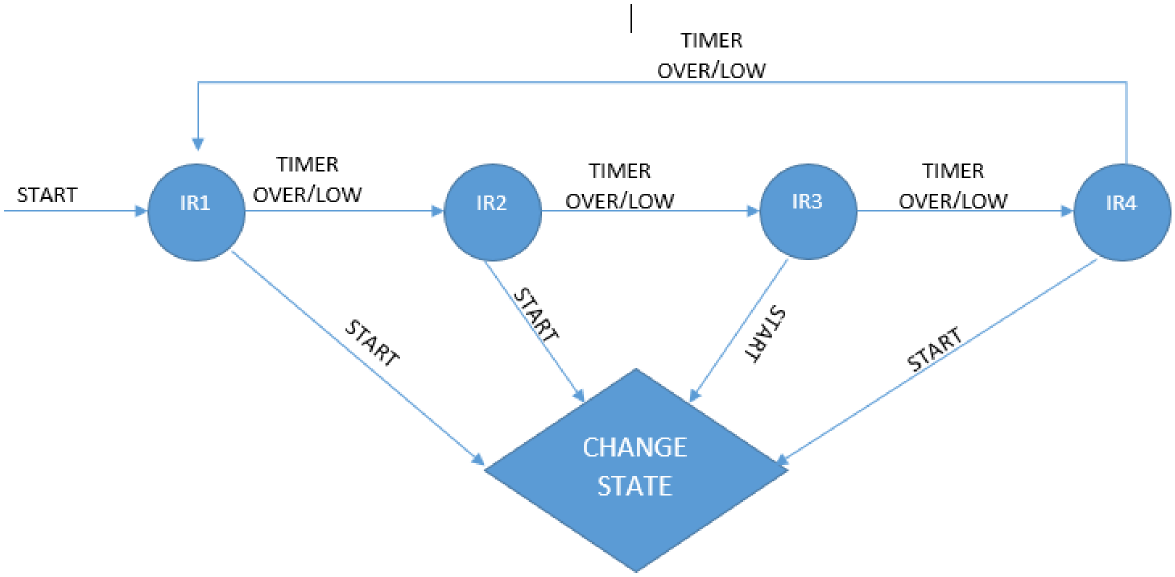
The system consist of four sensors IR1, IR2, IR3, and IR4 placed at an intersection, which are interfaced to the microcontroller.

1. Firstly the microcontroller checks the state of IR1, if it's HIGH, meaning it has detected an object, the MCU turns GREEN the traffic lanes corresponding to that sensor that will be lane 1 and 2, while disabling the rest of the lights (turning red) for lane 3 and 4. Then a timer for 30 seconds count is initiated.
2. The microcontroller cannot proceed to check the states of the other sensors unless the timer has ran out, that is, it has finished counting up to 30 seconds, this is so as to prevent rapid switching to and fro between the states, as several sensors can be HIGH at once.
3. After the timer runs out the MCU goes to check the state of IR2, if it happens to be HIGH then lane 3 and 4 Lights go green, while lane 1 and 2 lights go red. 30s timer is then initiated,
4. If any of the consecutive sensor happens to be LOW then that sensor is ignored then the next sensor following it is checked whether it's HIGH,
5. After timer for IR2 runs out, the MCU checks IR3, if HIGH, lane 1 and 2 goes green while lane 3 and 4 goes red. Then its 30seconds timer is initiated.
6. After the timer, it checks IR4, if HIGH lane 3 and 4 goes green while lane 1 and 2 goes red. Then initiate the timer. After timer runs out it again begins from step 1.

   NOTE:

   The sequence of execution is from IR1 to IR4, and also the pedestrians lights are only active provided that the traffic light for that particular lane if RED (traffic is not flowing pedestrians lights go GREEN)

WORKING OF THE SYSTEM

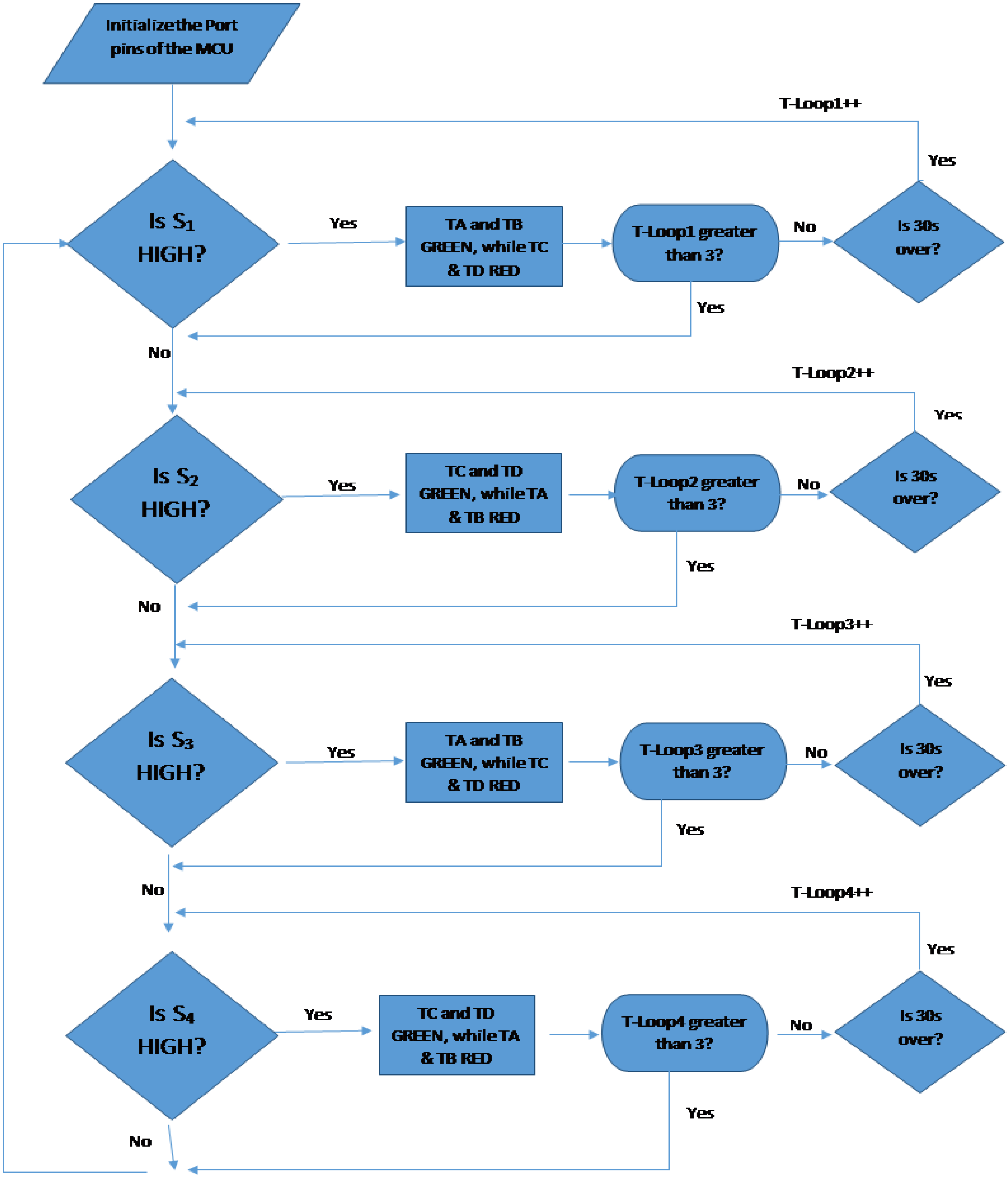


FLOWCHART SHOWING THE DECISION FLOW

Variables	Representation	Sensor
Traffic Light 1	T_A	S_1
Traffic Light 2	T_B	S_2
Traffic Light 3	T_C	S_3
Traffic Light 4	T_D	S_4

Table of variables

WORKING OF THE SYSTEM



WORKING OF THE SYSTEM

DIAGRAM OF THE REAL WORKING SYSTEM.

EXPLANATION

-This is the default virtual arrangement of the systems components. Here There is no power going into the system.

-The **ERROR** button acts as a failsafe detector whereby if it's pressed causes all the traffic lights to blink amber ON and OFF.

-**RESET** Button is used to reset the whole system.

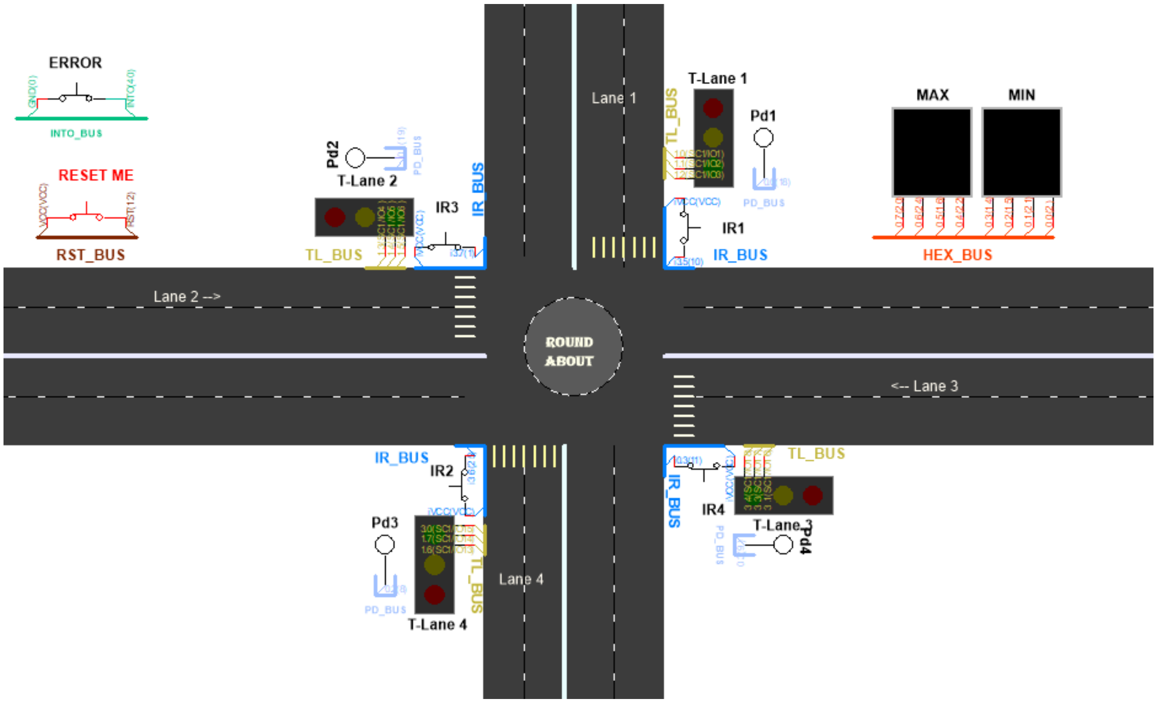
-Vehicle Traffic Lights:

- T-Lane1 is traffic light for lane 1,
- T-Lane2 is for lane 2,
- T-Lane3 for lane 3, and
- T-Lane4 for Lane4.

-Pedestrians Traffic Light:

- Pd1 is light for lane 1,
- Pd2 is light for lane 2,
- Pd3 is light for lane 4,and

DIAGRAM



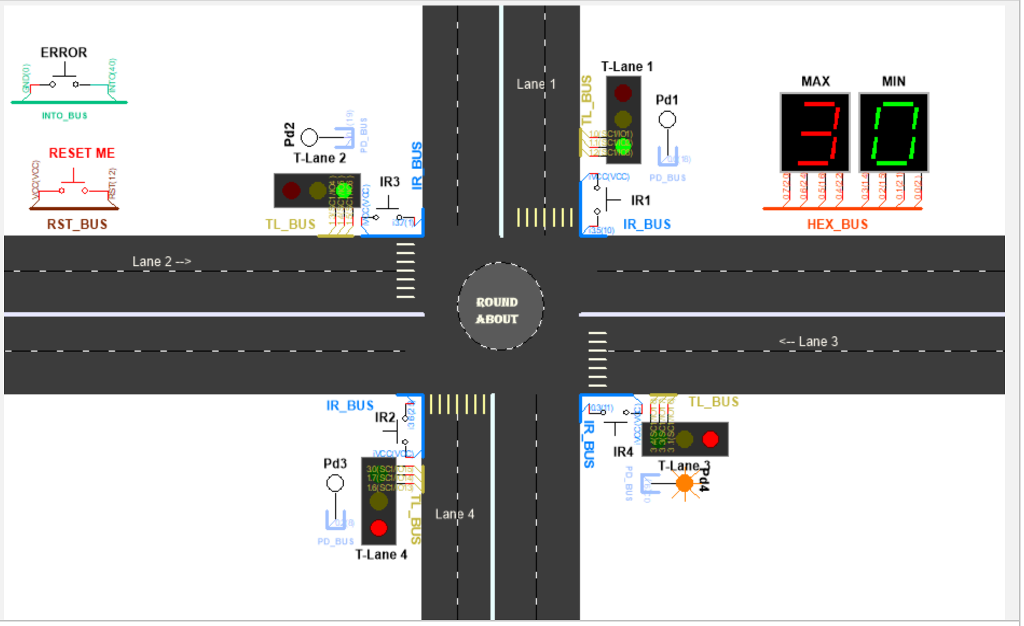
The two digital hex displays labelled **MAX** and **MIN are** used to indicate the time allocated for each lane after switching, since our system does not rely only on the sensors but also on the timer, this is to prevent the lights from staying on too long in a particular lane, provided there are excessive number of cars in that lane.

-Sensors:

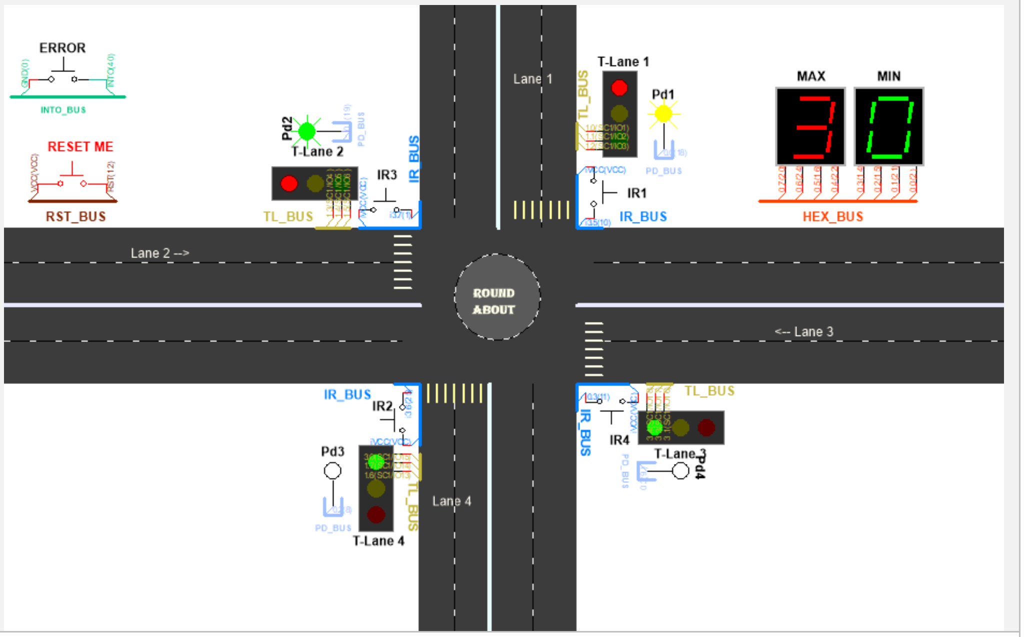
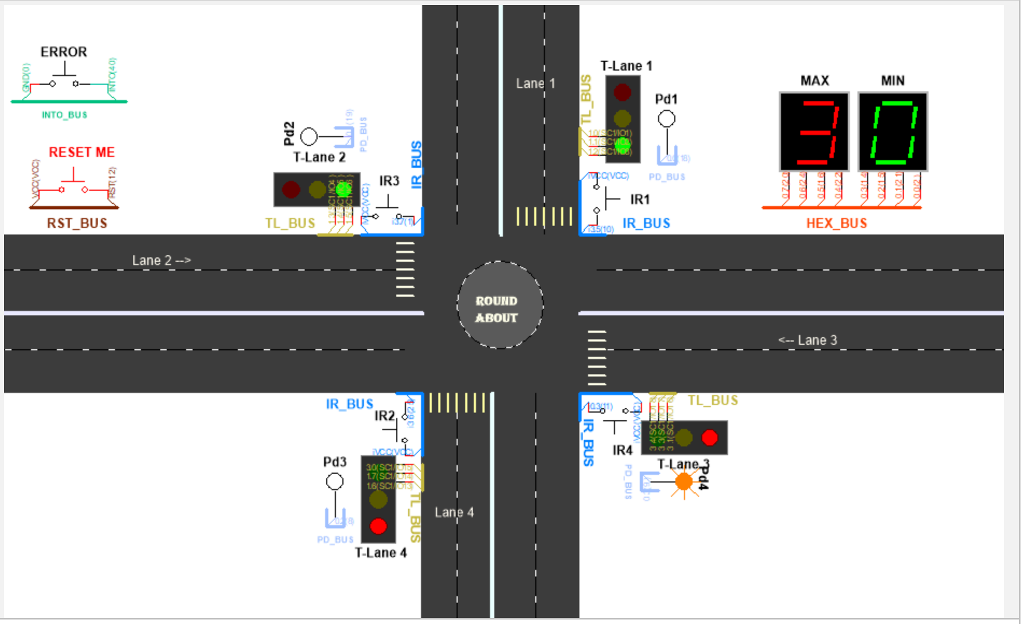
- IR1 is detects at lane 1
- IR2 detects at lane 4
- IR3 detects at lane 2
- IR4 detects at lane 3

• Pd4 is light for lane 4,	

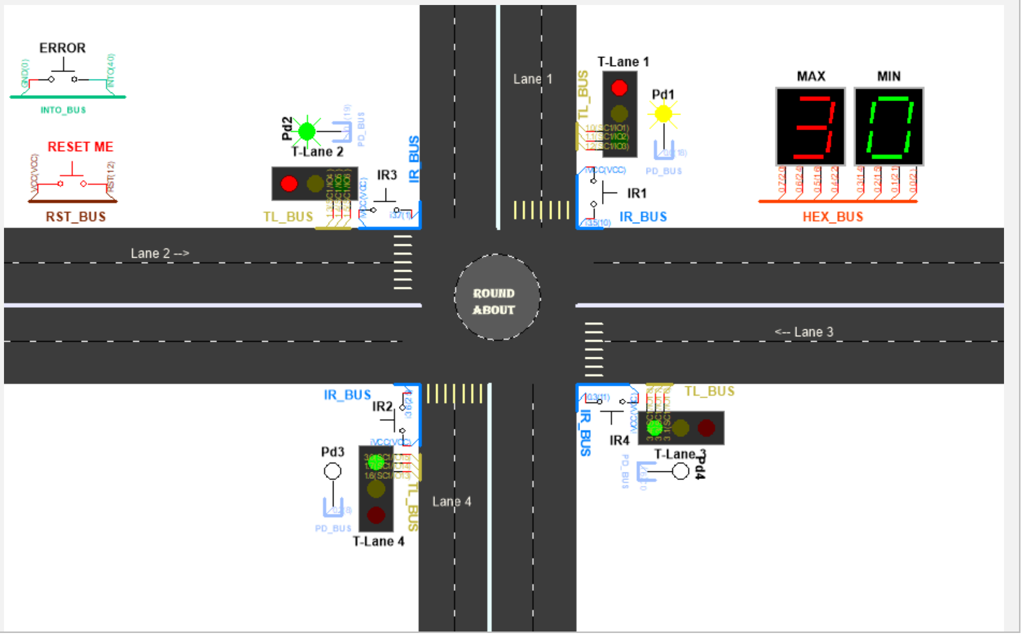
WORKING OF THE SIMULATED SYSTEM STATES

EXPLANATIONS	DIAGRAM
-Sensor IR1 is current HIGH(Detecting) Therefore Traffic Lights at Lane 1 and 2 Goes GREEN, to free up the traffic while that for lane 3 and 4 goes RED. -Until the timer runs out is when we can listen for the states of the other sensors, that is IR2,IR3 and IR4	 ERROR INTO_BUS RESET ME RST_BUS Pd2 T-Lane 2 PD_BUS IR3 TL_BUS IR_BUS Lane 1 T-Lane 1 TL_BUS Pd1 PD_BUS IR1 IR_BUS MAX MIN 3 0 HEX_BUS Lane 2 --> ROUND ABOUT <-- Lane 3 IR_BUS IR2 Pd3 PD_BUS T-Lane 4 TL_BUS Lane 4 IR_BUS IR4 TL_BUS T-Lane 3 Pd4 PD_BUS

WORKING OF THE SYSTEM

-Sensor IR2 is current HIGH(Detecting) Therefore Traffic Lights at Lane 4 and 3 Goes GREEN, to free up the traffic while that for lane 1 and 2 goes RED. -Until the timer runs out is when we can listen for the states of the other sensors, that is IR3,IR4 and IR1	 ERROR INTO_BUS RESET ME RST_BUS Pd2 T-Lane 2 IR3 IR BUS TL_BUS Lane 1 T-Lane 1 Pd1 PD_BUS IR1 IR_BUS MAX MIN 30 HEX_BUS Lane 2 --> ROUND ABOUT <-- Lane 3 IR_BUS IR2 Pd3 PD_BUS T-Lane 4 Lane 4 TL_BUS IR4 T-Lane 3 Pd4
-Sensor IR3 is current HIGH(Detecting) Therefore Traffic Lights at Lane 1 and 2 Goes GREEN, to free up the traffic while that for lane 3 and 4 goes RED. -Until the timer runs out is when we can listen for the states of the other sensors, that is IR4,IR1 and IR2	 ERROR INTO_BUS RESET ME RST_BUS Pd2 T-Lane 2 IR3 IR BUS TL_BUS Lane 1 T-Lane 1 Pd1 PD_BUS IR1 IR_BUS MAX MIN 30 HEX_BUS Lane 2 --> ROUND ABOUT <-- Lane 3 IR_BUS IR2 Pd3 PD_BUS T-Lane 4 Lane 4 TL_BUS IR4 T-Lane 3 Pd4

WORKING OF THE SYSTEM

-Sensor IR4 is current HIGH(Detecting) Therefore Traffic Lights at Lane 3 and 6 Goes GREEN, to free up the traffic while that for lane 1 and 2 goes RED. -Until the timer runs out is when we can listen for the states of the other sensors, that is IR1,IR2 and IR3	 ERROR INTO_BUS RESET ME RST_BUS Pd2 T-Lane 2 TL_BUS IR3 IR_BUS Lane 1 T-Lane 1 TL_BUS Pd1 PD_BUS IR1 IR_BUS MAX MIN HEX_BUS Lane 2 --> ROUND ABOUT <-- Lane 3 IR_BUS IR2 Pd3 PD_BUS TL_BUS T-Lane 4 Lane 4 IR_BUS IR4 TL_BUS T-Lane 3 Pd4

C CODE OF THE SYSTEM

```
/* Main.c file generated by New Project wizard
 * Created:   Fri Aug 31 2018
 * Processor: AT89C51
 * Compiler:  Keil for 8051
*/
#include <REG51.H> /* define 8051 registers */

void initINT();
unsigned char changeTrafficL1(void);
unsigned char changeTrafficL2(void);
unsigned char changeTrafficL3(void);
unsigned char changeTrafficL4(void);
void execTLightChange(unsigned char , unsigned char );

/*Declaring the functions prototypes for the delay functions*/
void delay_1ms(void);
void delay_100us(void);
void delay_50us(void);
void delay_1us(void);
```

```
/*----------------------------------------------*/

void time_Alloc(void);
void init_Traffic(void);
#define HEX P2

/*-----------------------------------------------
;P1.0-P1.2 are connected as OUTPUT pins for Traffic Light 1(For Lane 1)
;P1.0 is connected to the RED Light of Traffic Light
;P1.1 is connected to the AMBER/YELLOW Light of Traffic Light
;P1.2 is connected to the GREEN Light of Traffic Light
-----------------------------------------------*/
sbit A0=P1^0;
sbit A1=P1^1;
sbit A2=P1^2;

/*-----------------------------------------------
;P1.3-P1.5 are connected as OUTPUT pins for Traffic Light 2(For Lane 2)
;P1.3 is connected to the RED Light of Traffic Light
;P1.4 is connected to the AMBER/YELLOW Light of Traffic Light
;P1.5 is connected to the GREEN Light of Traffic Light
-----------------------------------------------*/
sbit B0=P1^3;
sbit B1=P1^4;
sbit B2=P1^5;

/*-----------------------------------------------
;P1.6-P3.0 are connected as OUTPUT pins for Traffic Light 3(For Lane 3)
;P1.6 is connected to the RED Light of Traffic Light
;P1.7 is connected to the AMBER/YELLOW Light of Traffic Light
;P3.0 is connected to the GREEN Light of Traffic Light
-----------------------------------------------*/
sbit C0=P1^6;
sbit C1=P1^7;
sbit C2=P3^0;

/*-----------------------------------------------
;P3.1-P3.4 are connected as OUTPUT pins for Traffic Light 4(For Lane 4)
```

```
;P3.1 is connected to the RED Light of Traffic Light
;P3.3 is connected to the AMBER/YELLOW Light of Traffic Light
;P3.4 is connected to the GREEN Light of Traffic Light
-----------------------------------------------*/
sbit D0=P3^1;
sbit D1=P3^3;
sbit D2=P3^4;

/*----------------------------------------------
;P3.5-P0.4 are connected as INPUT pins for the infrared Sensors Rx
;P3.5 is connected to the 1st infrared sensor receiver
;P3.6 is connected to the 2nd infrared sensor receiver
;P3.7 is connected to the 3red infrared sensor receiver
;P0.4 is connected to the 1st infrared sensor receiver
-----------------------------------------------*/
sbit ir1=P3^5;
sbit ir2=P3^6;
sbit ir3=P3^7;
sbit ir4=P0^4;

/*----------------------------------------------
;P0.0-P0.5 are connected as OUTPUT pins for Pedestrains traffic control
;P0.0 is connected to Pd1
;P0.1 is connected to Pd2
;P0.3 is connected to Pd4
;P0.5 is connected to Pd3
-----------------------------------------------*/
sbit pd1=P0^0;
sbit pd2=P0^1;
sbit pd3=P0^5;
sbit pd4=P0^3;

//These are all the Possible states of the system
typedef enum {RED, RED_AND_AMBER, GREEN, AMBER} eLight_State;

/*==========================================================================
======
```

WORKING OF THE SYSTEM

```
DECLARATION OF THE INTERRUPT HANDLER (THE INTERRUPT SERVICE ROUTINE)
FUNCTION
THIS WILL BE EXCUTED IN CASE OF AN ERROR IN THE SYSTEM AND IT WILL CAUZE THE
LIGHTS TO BLINK AMBER ON-OFF CONSECTIVELY FOR A NUMBER OF TIME
=====================================================================
====*/
unsigned char ex0_isr_counter = 0;//Keeping Records of the No. of Interrupts
unsigned char ex0_isr_exit=0;//initialise the interrupt count;
void ex0_isr (void) interrupt 0
{
ex0_isr_exit=20;                                //Re intiailise to value 20
ex0_isr_counter++;   // Increment the count(No of interrupts)

        HEX=0;                                                  //Turn of the Display time(HEX)
        pd1=0;pd2=0;pd4=0;   //Turn of all the pedestrains Lights except 1 P3
        A0=0;                                                           //Turn of all Traffiv
Lights
        A2=0;
        B0=0;
        B2=0;
        C0=0;
        C2=0;
        D0=0;
        D2=0;

        while(ex0_isr_exit>0){ //Exit loop ex0_isr_exit is <0 & thus exit the interrupt
                /*This turns ON and OFF the amber parts of the traffic lights, while the other
lights remain ON*/
                A1=0;
                B1=0;
                C1=0;
                D1=0;
                delay_100us();

                A1=1;
                B1=1;
                C1=1;
                D1=1;
                ex0_isr_exit--;
```

WORKING OF THE SYSTEM

```
    }
}

/*======================================================================
======
The main C function.  Program execution starts
here after stack initialization.
=======================================================================
====*/
void main(){
    ir1=ir2=ir3=ir4=0;              //Clear the Infrared Sensor Rx inputs bits
    pd1=0;pd2=0;pd4=0;pd3=0; //Clear the Pedestrain traffic output bits

    init_Traffic();

/*-----------------------------------------------
Wait forever/Loop Forever.
Note that an embedded program never exits (because
there is no operating system to return to).  It
must loop and execute forever.
-----------------------------------------------*/
    while(1){
        if(ir1==1){
                ir1=changeTrafficL1();
        }
        else if(ir2==1){
                ir2=changeTrafficL2();
        }
        else if(ir3==1){
                ir3=changeTrafficL3();
        }
        else if(ir4==1){
                ir4=changeTrafficL4();
        }

    };
}
/*-----------------------------------------------
```

```
Function thats responsible for initialising the
System and configuring the variables
-----------------------------------------------*/
void init_Traffic(void){

/*-----------------------------------------------
Configure INT0 (external interrupt 0) to generate
an interrupt on the falling-edge of /INT0 (P3.2).
Enable the EX0 interrupt and then enable the
global interrupt flag.
-----------------------------------------------*/
IT0 = 1;   // Configure interrupt 0 for falling edge on /INT0 (P3.2)
EX0 = 1;   // Enable EX0 Interrupt
EA = 1;    // Enable Global Interrupt Flag

        changeTrafficL1();
}
unsigned char changeTrafficL1(void){
        unsigned char x;

        delay_1ms();
        execTLightChange('A',1);

        delay_1ms();
        execTLightChange('A',2);

        pd1=0;pd2=0;pd4=1;
        time_Alloc();
        for(x=0;x<2;x++){
                if(ir1==1){
                        time_Alloc();
                }else
                        break;
        }
        return 0;
}
unsigned char changeTrafficL2(void){
        unsigned char x;
        delay_1ms();
        execTLightChange('C',1);

        delay_1ms();
        execTLightChange('C',2);

        pd1=1;pd2=1;pd4=0;
        time_Alloc();
        for(x=0;x<2;x++){
                if(ir2==1){
                        time_Alloc();
                }else
                        break;
        }
        return 0;
}
unsigned char changeTrafficL3(void){
        unsigned char x;

        delay_1ms();
        execTLightChange('B',1);

        delay_1ms();
        execTLightChange('B',2);

        pd1=0;pd2=0;pd4=1;
        time_Alloc();
        for(x=0;x<2;x++){
                if(ir3==1){
                        time_Alloc();
                }else
                        break;
        }
        return 0;
}

unsigned char changeTrafficL4(void){
        unsigned char x;

        delay_1ms();
        execTLightChange('D',1);

        delay_1ms();
        execTLightChange('D',2);

        pd1=1;pd2=1;pd4=0;
        time_Alloc();
        for(x=0;x<2;x++){
                if(ir4==1){
                        time_Alloc();
                }else
                        break;
        }
        return 0;
}
void execTLightChange(unsigned char TrafficLight, unsigned char Light_State ){


        if(TrafficLight=='A' || TrafficLight=='B'){
                switch(Light_State)
                {
                        case RED:
                            A0=1;
                            A1=0;
                            A2=0;
                            B0=1;
                            B1=0;
                            B2=0;

                            C0=0;
                            C1=0;
                            C2=1;
                            D0=0;
                            D1=0;
                            D2=1;
```

WORKING OF THE SYSTEM

```
            break;

        case RED_AND_AMBER:
            A0=1;
            A1=1;
            A2=0;
            B0=1;
            B1=1;
            B2=0;

            C0=0;
            C1=1;
            C2=0;
            D0=0;
            D1=1;
            D2=0;

        case GREEN:
            A0=0;
            A1=0;
            A2=1;
            B0=0;
            B1=0;
            B2=1;

            C0=1;
            C1=0;
            C2=0;
            D0=1;
            D1=0;
            D2=0;

            break;

        case AMBER:
            A0=0;
            A1=1;
            A2=0;
            B0=0;
            B1=1;
            B2=0;

            C0=0;
            C1=1;
            C2=0;
            D0=0;
            D1=1;
            D2=0;
            break;

        default:
                A0=0;
                A1=0;
                A2=0;
                C0=0;
                C1=0;
                C2=0;
            while(1){
                A1=1;
                C1=1;
                A1=0;
                C1=0;
        }
    }
}
else if(TrafficLight=='C' || TrafficLight=='D' ){
    switch(Light_State)
    {
        case RED:
            D0=1;
            D1=0;
            D2=0;
            C0=1;
            C1=0;
            C2=0;
```

```
                A0=0;
                A1=0;
                A2=1;
                B0=0;
                B1=0;
                B2=1;
                break;

        case RED_AND_AMBER:
                D0=1;
                D1=1;
                D2=0;
                C0=1;
                C1=1;
                C2=0;

                A0=0;
                A1=1;
                A2=0;
                B0=0;
                B1=1;
                B2=0;

        case GREEN:
                D0=0;
                D1=0;
                D2=1;
                C0=0;
                C1=0;
                C2=1;

                A0=1;
                A1=0;
                A2=0;

                B0=1;
                B1=0;
                B2=0;
                break;

            case AMBER:
                D0=0;
                D1=1;
                D2=0;
                C0=0;
                C1=1;
                C2=0;

                A0=0;
                A1=1;
                A2=0;
                B0=0;
                B1=1;
                B2=0;
                break;

            default:
                    D0=0;
                    D1=0;
                    A2=0;
                    C0=0;
                    C1=0;
                    C2=0;
                while(1){
                    D1=1;
                    C1=1;
                    D1=0;
                    C1=0;
            }
        }
    }
}
void time_Alloc(void){
    unsigned char x;
    for(x=0;x<49;x++){
        HEX=x;
```

```
                        delay_100us();
            }
}


/*=========================================================================
======
            CIRCUIT DELAYS, FOR GENERATING DELAYS IN THE CIRCUIT, EACH FUNCTION
DELAYS,
            ARE 1mS, 100uS, 50uS & 1uS RESPECTIVELY, MAX DELAY ACHIAVABLE FOR 8051
OPERATING,
            AT TMOD 1 16BIT IS 65mS AND MIN DELAY IS 1us
===========================================================================
====*/
void delay_1ms(void){

/*Timer 0, mode 1(16-bit mode no Auto-reload) Selected*/
            TMOD&=0x0F; //CLEAR all Timer 0(T0) bits (T1 Unchanged)
            TMOD|=0x01;  //SET required bits (i.e. 1111, en mode 1)

            /* Values for 1mS Delay*/
            TH0=0xFC;       //the high byte
            TL0=0x66; //the low byte

            TF0=0; //initialize by clearing timer 0 flag
            TR0=1; //start the timer 0

            while(TF0==0);              //Loop until timer 0 overflows/rolls over(TF0==1)
            TR0=0; //stop timer 0
            TF0=0; //clear timer 0 flag
}
void delay_100us(void){

            /*Timer 0, mode 1(16-bit mode no Auto-reload) Selected*/
            TMOD&=0x0F; //CLEAR all Timer 0(T0) bits (T1 Unchanged)
            TMOD|=0x01;  //SET required bits (i.e. 1111, en mode 1)

            /* Values for 100uS Delay*/
            TH0=0xFF;       //the high byte
```

```
        TL0=0xA4; //the low byte

        TF0=0; //initialize by clearing timer 0 flag
        TR0=1; //start the timer 0

        while(TF0==0);          //Loop until timer 0 overflows/rolls over(TF0==1)
        TR0=0; //stop timer 0
        TF0=0; //clear timer 0 flag
}
void delay_50us(void){

        /*Timer 0, mode 1(16-bit mode no Auto-reload) Selected*/
        TMOD&=0x0F; //CLEAR all Timer 0(T0) bits (T1 Unchanged)
        TMOD|=0x01;  //SET required bits (i.e. 1111, en mode 1)

        /* Values for 50uS Delay*/
        TH0=0xFF;       //the high byte
        TL0=0xD2; //the low byte

        TF0=0; //initialize by clearing timer 0 flag
        TR0=1; //start the timer 0

        while(TF0==0);          //Loop until timer 0 overflows/rolls over(TF0==1)
        TR0=0; //stop timer 0
        TF0=0; //clear timer 0 flag
}
void delay_1us(void){

        /*Timer 0, mode 1(16-bit mode no Auto-reload) Selected*/
        TMOD&=0x0F; //CLEAR all Timer 0(T0) bits (T1 Unchanged)
        TMOD|=0x01;  //SET required bits (i.e. 1111, en mode 1)

        /* Values for 1uS Delay*/
        TH0=0xFF;       //the high byte
        TL0=0xFF; //the low byte

        TF0=0; //initialize by clearing timer 0 flag
        TR0=1; //start the timer 0
```

```
            while(TF0==0);            //Loop until timer 0 overflows/rolls over(TF0==1)
            TR0=0; //stop timer 0
            TF0=0; //clear timer 0 flag
}

/*=========================================================================
======
===========================================================================
====*/
```

EVALUATION OF THE PROJECT

EVALUATION OF THE PROJECT

ADVANTAGES OF THIS TRAFFIC LIGHT SYSTEM

- Easy installation
- Low maintenance
- Compact elegant design
- Consumes very little energy (<10V)

LIMITATION OF THE SYSTEMS PERFORMANCE

The traffic light has been successful in controlling the traffic but it also has certain limitations as well. As stated below:

- Performance of IR sensors has been limited by their poor tolerance to light reflections such as ambient light or bright object colors.
- IR sensors work only for a fewer distances hence no object recognition at the dead zone area, for example our IR sensor dead zone was limited to a minimum of 2.3 m.
- IR sensors also give inaccurate detection result with transparent or bright color materials.
- Detection results also depend on the weather conditions and the sensing reliability of IR sensors decreases with moisture and humidity.
- Furthermore, IR sensors can sense IR radiation from the sunlight, which can cause correctable or non-correctable errors at output.
- Besides that, if analogue IR sensor is used, signal losses will occur at the amplifier circuit.
- We have to arrange IR sensors in accurate manner otherwise they may not detect the traffic density.

HOW THE SYSTEM CAN BE IMPROVED

- Further improvement can be achieved by adding more sensors along the road intersection. Besides that, speed traps and cameras can be implemented for monitoring applications to deal with traffic crimes.
- Use of IR and PIR sensors for object detection:
  - Infrared sensors detect the object's distance with infrared radiation. When the beam detects an object, the light beam returns to the receiver with an angle after reflection. The method of triangulation is as shown in Figure 4 below.
  - PIR sensors are also known as Pyroelectric Infrared sensor, Passive Infrared sensor or IR motion sensor, which detect the difference in temperature, thermal radiation, human body or an animal. PIR sensor operates with the radiation of

body heat as shown in Figure 5 below. The hotter the detected object, there will be more emission occurs in PIR sensor.

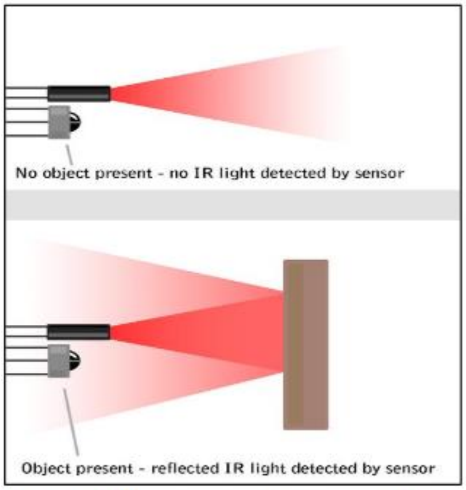

No object present - no IR light detected by sensor
Object present - reflected IR light detected by sensor

Figure 4. IR sensor working principle [15]

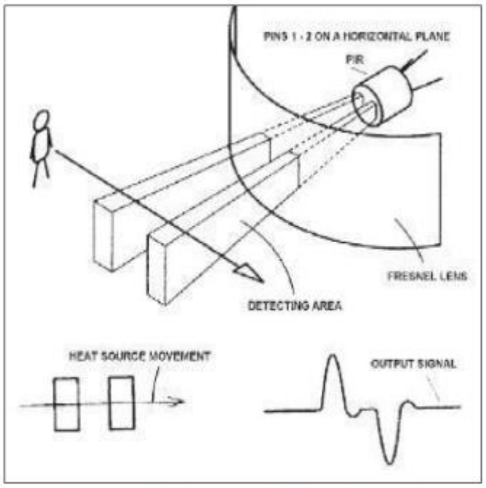


Figure 5. PIR sensor object detection

Object temperature calculation is based on Stefan-Boltzmann Law.

$$Te = \sqrt{T_S^4 + \frac{\Phi}{A\sigma\varepsilon\varepsilon_S}}$$

$$Ts = Sensor's\ surface\ temperature\ Tc = objects\ temperature\ in\ Kelvin,$$

$$\Phi = magnitude\ of\ net\ thermal\ radiation\ flux\ \phi, \varepsilon = emissivity\ of\ the\ object$$

PIR sensor relatively has a lower power consumption compared to IR sensor. PIR sensor also senses accurate detection in narrow areas and is compatible to work in microcontrollers. The IR and PIR sensors can act as a transducer since they use infra-red signal as the input and convert it to analogue electrical output signal

BIBILIOGRAPHY/REFERENCES

BIBILIOGRAPHY/REFERENCES

From the website http://www.electronicshub.org where we researched on:

http://www.electronicshub.org/density-based-traffic-signal-system-using-microcontroller/
http://www.electronicshub.org/different-types-sensors/
http://www.electronicshub.org/led-light-emitting-diode/
http://www.electronicshub.org/basic-electronic-components/

Books:

Warwick A. Smith, C Programming for Embedded Microcontrollers
Sedra and Smith, Microelectronic Circuits, fourth edition, Oxford University Press, 1998
R.S. Sedha, 2002. A Text Book of Applied Electronics, S. Chand and Company Ltd., New Delhi
Michael J. Pont, Embedded C

Others:

Embedded Systems/ 8051 Microcontroller Available at:
http://en.wikibooks.org/wiki/Embedded_Systems/8051Microcontroller
Programming and Customizing the AVR Microcontroller by Dhananjay Gadre
Keil website: http: //www.keil.com/c51/. C51 Documentation/ Tutorials and Examples
The Features of ATmega8: https://handhua.wordpress.com/2011/04/27/the-features-of-atmega8/>
Wikipedia https://en.wikipedia.org/wiki/Atmel_AVR

APPENDIX

APPENDIX

A-ampere

SI unit of electrical current

C capacitance

Unit: F (farad) = C/V

Ee, **E irradiance** (at a point of a surface)

F **frequency**

Unit: s^{-1}, Hz (Hertz)

I.e., I **radiant intensity** (of a source, in a given direction

M **meter**

SI unit of length

IC **collector current**

LED and IRLED

Light Emitting Diode LED:

Solid state device embodying a p-n junction, emitting optical radiation when excited by an electric current.

IR Abbreviation for infrared

T **period of tim**e (duration)

λ **Wavelength**, general

Sr **steradian** (sr)

SI unit of solid angle : Solid angle that Ω, having its vertex at the center of a sphere, cuts off an area of the surface of the sphere equal to that of a square with sides of length equal to the radius of the sphere.

Radiation and Light

Visible radiation

Any optical radiation capable of causing a visual sensation directly.

Radiation and Light

IR infrared radiation

Optical radiation for which the wavelengths are longer than those for visible radiation. Note: The range between 780 nm and 1 mm

VCC supply voltage (positive)

V **volt**

VCE_{sat} **collector-emitter saturation voltage**

The saturation voltage is the dc voltage between collector and emitter for specified (saturation) conditions, i.e., IC and EV (Ee or IB), whereas the operating point is within the saturation region

V_F **forward voltage**

The voltage across the diode terminals which results from the flow of current in the forward direction

APPENDIX

VS **supply voltage**

$\phi_e; \phi, P$

Radiant flux; radiant power

Power emitted, transmitted or received in the form of radiation. Unit: W = Watt

MCU

-Microcontroller