SMART CITY CHALLENGE: TRAFFIC DENSITY BASED CROSSROADS

Group Members:

- 131021 Kevin Shah
- 131025 Manthan Shah
- 131029 Netra Pathak

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to Professor Nishit Pandya and our Teaching Assistants Jaina Ma'am, Bhavika Ma'am and Ujjwal Sir for their tremendous and valuable help, support and guidance.

Without them, we wouldn't have been able to complete our project in the limited time frame, such perfectly and successfully. We have learnt a lot from them and also through this project.

1. Project Idea and Brief Application

Nowadays congestion in traffic is a serious issue. The traffic congestion can be caused because of rapid increase in automobiles, large time delays between traffic lights, etc. The delay of respective light is hard coded in the traffic light and it is independent on traffic.

As the number of road users constantly increases, and resources provided by current infrastructures are limited, intelligent control of traffic has become a very important issue.

Therefore for simulating and optimizing traffic control to better, a need to accommodate this increasing demand for density based traffic lights system using micro-controller arises. The system tries to reduce possibilities of traffic jams caused by traffic lights, to an extent. Also, there will be no blinking of lights at odd times like afternoons or late nights, reducing accidents. In general, time which is very valuable can be saved by not waiting for long; as the lanes where density is low will have green signal for shorter period of time. Every time the traffic signals will be controlled based on the density of that crossroads.

One more area this system can cover and affect is controlling the traffic lights in case of emergency (e.g. ambulance or fire brigade passing). Through an android app in the mobile phones (interfaced with the signals sent to the micro-controller), policemen can control that particular crossroads, traffic lights and hence the traffic flow in order to allow ambulance/fire brigade, etc. to pass without any delay; avoiding causalities. Instead of an android app, we can also have an RF (radio frequency) module; and through remote we can operate the traffic lights of a particular lane of the crossroads.

Thus, the idea was to implement this system with such great applications. We were motivated to take up this challenge as this advancement would make us move a step ahead towards 'Smart City'.

If AMC (Ahmedabad Municipal Corporation) implements this system in Ahmedabad, controlling the traffic lights will not be an issue anymore. Traffic light signals will be computed based on the density; and hence we are not dependent on the presence of policemen to clear traffic jams during peak hours.

2. Introduction

In this project, we have designed and implemented a smart city challenge by controlling the traffic density based on the crossroads.

For the purpose of developing this ‘adaptive’ system, the traffic density is continuously measured by IR (infrared) sensors (transmitter and receiver) connected to a micro-controller (Atmega32) based system, also performing all intersection control functions. As a result of this, the duration and relative phases of each traffic light cycle change dynamically. The LED’s are turn on and off automatically, considering the corresponding commands and computations of the port pins of the microcontroller. Also, for emergency conditions an RF module is interfaced. RF module has an edge over android app in the sense that not every person uses a smart phone. Also, it becomes dependent on the battery life of the mobile phone. Hence, a dedicated module focuses specially on the functionality of being able to tackle emergency cases.

This and other different Embedded Systems are self-contained programs, usually set to a particular task that cannot be altered without manipulating the circuitry. It has optimal efficiency as they integrate all three- Software, Firmware and Hardware.

This project finds high practical and widespread use as traffic lights have become an integral part of human’s day to day life. This system helps in saving time, measurably control the traffic flow, reduce environmental pollution caused by traffic, increase public safety on road, optimize traffic flow at network junctions, reduce accidents; etc... In general, it makes the city more attractive and improves quality of life.

We were motivated for taking up this project as living in a city like Ahmedabad, which is barely few steps away from becoming a ‘Metro City’, we need to make the city as smart as possible. In the past few years, traffic in Ahmedabad has become a major problem. To be able to handle the issues of increasing population and hence traffic density; this system befits the best. It provides sophisticated control and coordination, ensuring that traffic moves smoothly and safely.

Therefore, the project is developed to meet the requirements of solid state traffic light controller by adopting microcontroller as the main controlling element, and LED’s as the indication of light. The circuit besides being reliable and compact is also cost effective.

3. Detailed Description

The system contains 3 IR sensors mounted on the divider of the road which are equidistant from the crossroads. This IR sensor always emits IR rays from it and receives the same signal back. These IR (infra-red) rays are invisible to the human eye, but we can view these IR rays through camera. The IR sensor activates (gets disturbed) whenever there is a hindrance caused by any vehicle passing on road in front of the IR sensor and within its range. Hence, digital IR sensors are used in the system to measure the traffic density; i.e. to detect the vehicles on the road. IR sensors have 3 pins- Vcc (+5V), DO (Digital Output), GND. Also, by tuning the sensitivity threshold via the potentiometer on the sensor, the required sensing range within which we want the sensor to sense can be set. The sensor gives a 'high' output (digital 1) to the microcontroller, whenever an object is detected.

As a result, the duration and relative phases of each traffic light cycle change dynamically.

- Three IR sensors on the lanes of the crossroad
- 3 sensors would accommodate four states termed as:
    - Very Low traffic – when none of the sensors are disturbed
    - Low traffic – when the first sensor is disturbed
    - Medium traffic – when the first two sensors are disturbed
    - High traffic – when all the three sensors are disturbed

Note: Here the term 'disturbed' means the IR transmitted does get detected for significant amount of time (i.e.) the traffic is steady – not moving.

- The green signal would be displayed only for one lane at a time; remaining three lanes at that instant-red signals
- The system is conceptualized in such a way that while controlling the traffic signals based on the density, the rotation cycle (clockwise) always remains fixed
- Yellow signal starts 3 seconds before the change to green signal of the next lane
- The time period of green signal would be decided by the state at which the lane is (i.e.) Zero, Low, Medium, High traffic. Hence, this is an 'adaptive system' as it controls the traffic light timings based on the density state.
    - Very Low – 7 seconds change (of green signal) to next lane
    - Low – 10 seconds
    - Medium – 20 seconds
    - High – 30 seconds

Working:

- ✓ If the traffic doesn't clear out in specified time, the green signal would still rotate no matter what is the lane's current state.
- ✓ While the rotation, the microcontroller calculates the green signal time for the next lane 5 seconds before (last 2 seconds of green and the 3 seconds of yellow), by analyzing the density on that lane. The state of the lane is decided by the IR sensors' disturbed count.
- ✓ Also, we can distinguish passing of anything else except a car as only the car will cut all the IR sensors. A human or a dog for instance will cut only anyone IR sensor and not all the IR sensors (before it). Hence, if the first sensor and third sensor is cut continuously; only the first sensor density will be computed and the third sensor would not be considered. The reason for this is that, second sensor is not disturbed and so, the disturbance in the third sensor is because of some anomaly.
- ✓ There will also be an emergency control remote for emergency situations. If the traffic police presses any of the button on the remote given to them, all the signals will become red irrespective of their current states. This can be helpful when ambulance or fire brigades have to pass. Then, green signal for the lane in which we want traffic to clear out can be selected by the various other buttons on the remote. Once, the purpose is finished, we can resume back to our normal density based system.

Atmega32 is the brain of the entire system. It takes in all the signals and send commands. It controls the system by computing and processing according to the logic of the program loaded onto the microcontroller.

Further advancements:

Smart city-many crossroads which are linked will be managed based on their density. The timing of green signals connecting the crossroads will be managed accordingly.

Also, in future, the recorded vehicle count data can be used to analyze traffic condition at respective traffic lights connected to the system. For appropriate analysis, the recorded data can be downloaded to the computer through communication between microcontroller and the computer. Administrator sitting on computer can command system (microcontroller) to download recorded data, update light delays, erase memory, etc. Thus administrator on a central station computer can access traffic conditions on any approachable traffic lights and nearby roads to reduce traffic congestions to an extent.

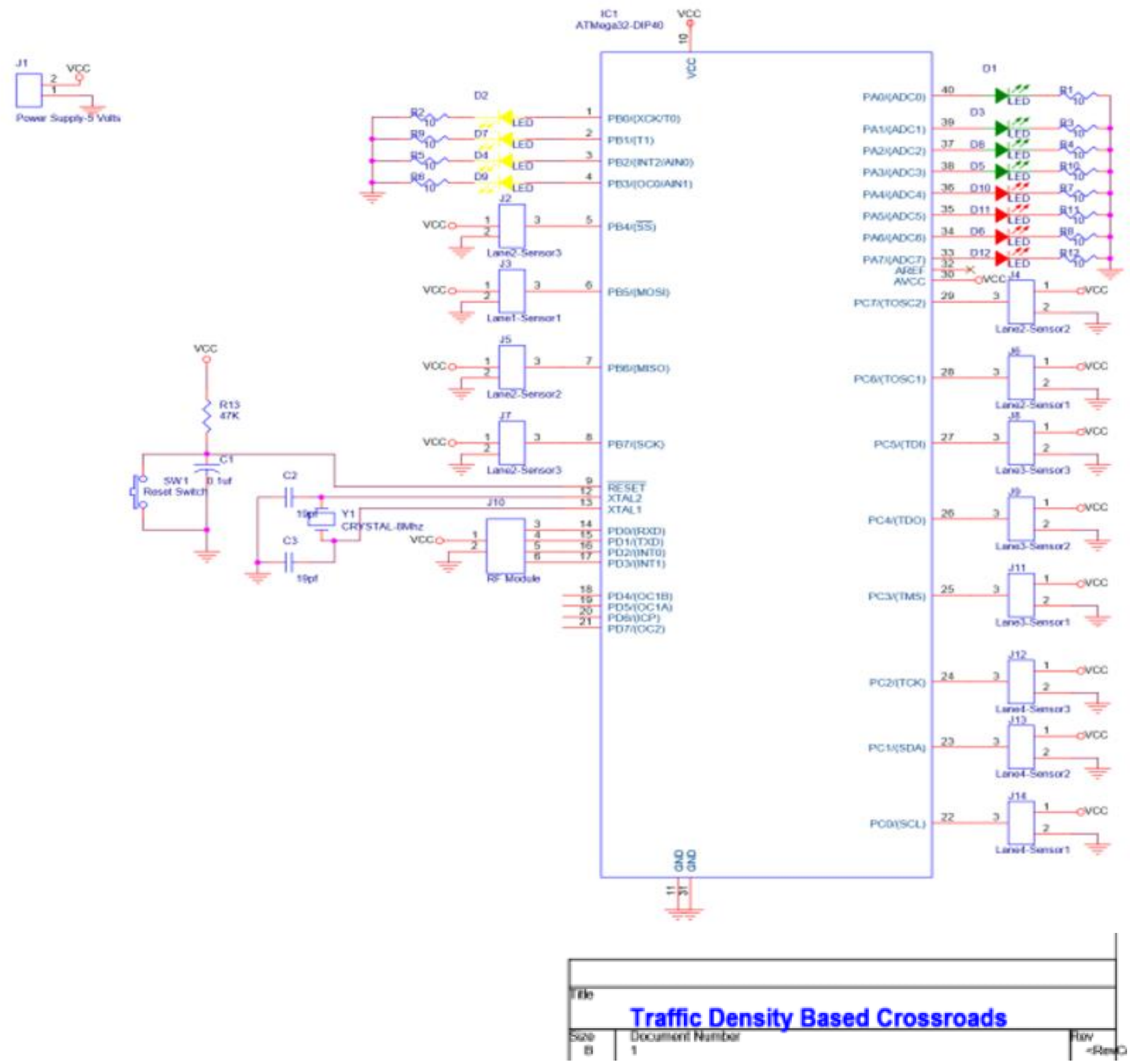
In future this system can be used to inform people about traffic condition at different places. It can also help the government on deciding the locations for development projects; making over bridges and under bridges etc.
Lastly, the data can also be used for analyzing the traffic data in different aspects such as traffic at a particular time of a day, fast lanes and slow lanes, the average traffic density, etc.

For wider and practical application on the road, metal detectors instead of IR sensors can be used.

Circuit Components:

- Microcontroller – Atmega32
- Programmer (to load the program in the microcontroller)
- PCB board
- RF module (receiver chip and remote)
- 12 IR sensors
- 12 LED’s(4-red,4-green,4-yellow)
- 12 LED holders
- 12 100 ohm Resistors
- 3 pin,4 pin and 5 pin relimate connectors
- Connecting wires, female to female connector wires
- Insulating tape
- Battery or adaptor with 5V power supply
- Serial (USB to USB) cable

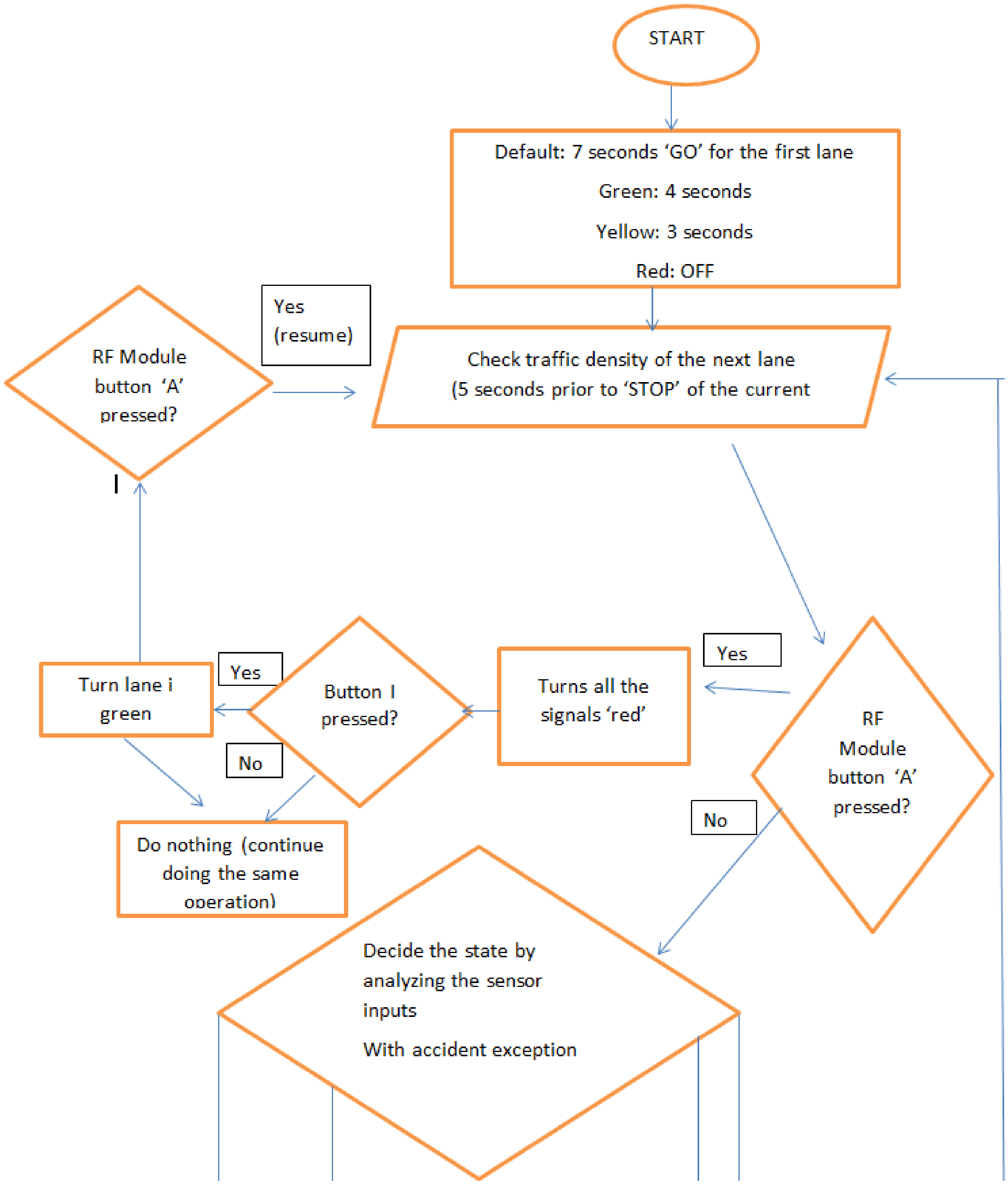
4. Schematic Diagram

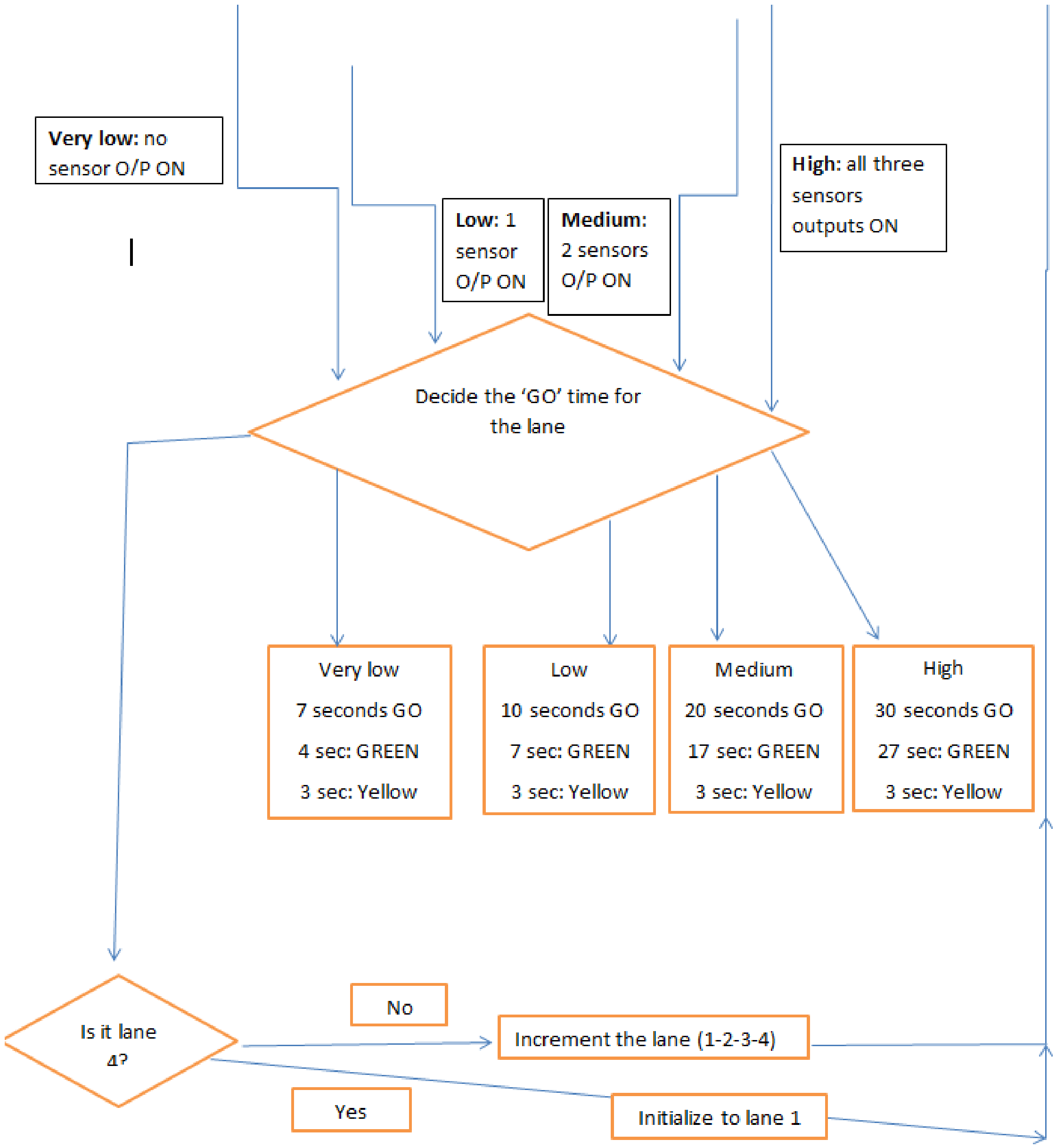


5. Bill of Materials

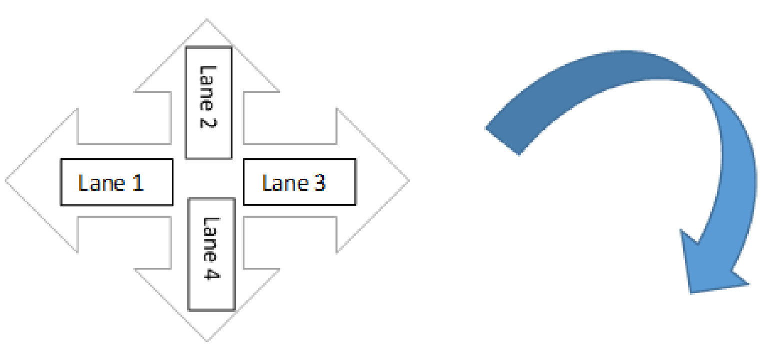
Sr. No.	Item	Price (Rs.)
1	IR Sensors (*12)	684
2	Microcontroller (ATMEGA32)	500
3	Programmer	275
4	RF Module	450
5	Electric materials (Female to Female connectors, insulating tape, PCB, relimate connectors)	511
6	Stationary (Chart Papers, Fevistick, Fevikvik, Thermocol sheets, Crap Paper, Cello tape, Acrylic Paints)	530
7	Fabrication	50
	Total	**3000**

6. Flowchart of Firmware





Very low: no sensor O/P ON
Low: 1 sensor O/P ON
Medium: 2 sensors O/P ON
High: all three sensors outputs ON
Decide the 'GO' time for the lane
Very low
7 seconds GO
4 sec: GREEN
3 sec: Yellow
Low
10 seconds GO
7 sec: GREEN
3 sec: Yellow
Medium
20 seconds GO
17 sec: GREEN
3 sec: Yellow
High
30 seconds GO
27 sec: GREEN
3 sec: Yellow
Is it lane 4?
No
Increment the lane (1-2-3-4)
Yes
Initialize to lane 1



Lane 2
Lane 1
Lane 3
Lane 4

7. C Code

```
/*
Group 6
131021 Kevin Shah
131025 Manthan Shah
131029 Netra Pathak
*/

/*Program to 'Control the traffic lights of crossroads, based on the traffic density';
  based on the measurements taken by the 3 IR sensors placed on the four lanes, and
  using Atmega32
*/

/*Variable declaration and initialization of global variables and functions*/

//Variable to count the number of iterations for each compare value (OCR0)
unsigned int counter=0;

//To initialize to start check state from second lane for the first time
unsigned int lane=1;


//To give default/initial time for green signal as 7 seconds
unsigned int delay_count=234;

//To count for the number of times sensor inputs are checked
unsigned int counter_count_check=0;

//To check for the commands by RF module (either stop/resume or select lane)
unsigned int resume_stop_check=0;

//To select when to start IR checking
unsigned int ir_check_mode=1;

//For the 3 IR sensors on each of the 4 lanes
unsigned int sensor[4][3];

/*Function that takes input from the sensors*/
```

```
void check_lane();

/*Fucntion that checks for the state of the lane*/
void check_lane_status();

/*Function that gives appropriate delay; according to the state of the lane*/
void operate(int stateCount);

/*To store PORTA and PORTB values*/
char storeA, storeB,store_resumeA;

/*Interrupt service routine for Timer0 Compare Mode*/
void TIMER0_COMP()org 0x014
{
 counter++;
}

void check_lane()
{
    /*To count the number of times IR sensors are checked for within the 5 seconds*/
    counter_count_check++;



 /*To store the IR sensors output values*/
    if(lane==0) //For lane 1
   {
    sensor[lane][0]+=PINB.B5;
    sensor[lane][1]+=PINB.B6;
    sensor[lane][2]+=PINB.B7;
   }
   else if(lane==1) //For lane 2
   {
    sensor[lane][0]+=PINC.B6;
    sensor[lane][1]+=PINC.B7;
    sensor[lane][2]+=PINB.B4;
   }
   else if(lane==2) //For lane 3
   {
    sensor[lane][0]+=PINC.B3;
```

```
    sensor[lane][1]+=PINC.B4;
    sensor[lane][2]+=PINC.B5;
   }
   Else //For lane 4
   {
    sensor[lane][0]+=PINC.B0;
    sensor[lane][1]+=PINC.B1;
    sensor[lane][2]+=PINC.B2;
   }
}

 void check_lane_status()
 {
  /*Initialization and declaration of local variables
  int j,wrong_cut_check=0,state_count=0;

  /*Logic to consider IR sensors O/P only if all consecutively disturbed;
    and avoid the case when sensor 3 cut due to other obstruction like dog or
    accident case but sensor 2 not cut
  */
  for(j=0;j<2;j++)
  {
   if(sensor[lane][j+1]>sensor[lane][j])
   {
    wrong_cut_check=j+1;
    break;
   }
   else
   {
    wrong_cut_check=3; //Default: when all consecutive sensors cut properly
   }
  }

 /*Logic to check whether steady traffic or moving traffic*/
 for(j=0;j<wrong_cut_check;j++)
  {
   sensor[lane][j]=sensor[lane][j]/(counter_count_check);
   if(sensor[lane][j]==1)
   state_count++;
  }
```

```
/*Re-initialize all values for the computation of next lane*/
counter_count_check=0;
wrong_cut_check=0;

 for(j=0;j<3;j++)
 {
  sensor[lane][j]=0;
 }

/*To start from lane 1 again after all four lanes checked in one cycle*/
 if(lane==3) //For lane 1 again after the fourth lane
 lane=0;
 else
 lane++; //For moving to the next lane In the clockwise pattern

 operate(state_count); //Calling the function
}

void operate(int stateCount)
{
 int state_lane_operate = stateCount;

 /*Switch case*/
 switch(state_lane_operate)
 {
 case 1:
 delay_count=334; //For 10 seconds time, when state='low'
 break;

 case 2:
 delay_count=668; //For 20 seconds time, when state='medium'
 break;

 case 3:
 delay_count=1002; //For 30 seconds time, when state='high'
 break;

 default:
 delay_count=234; //For 7 seconds time, when state='very low'
```

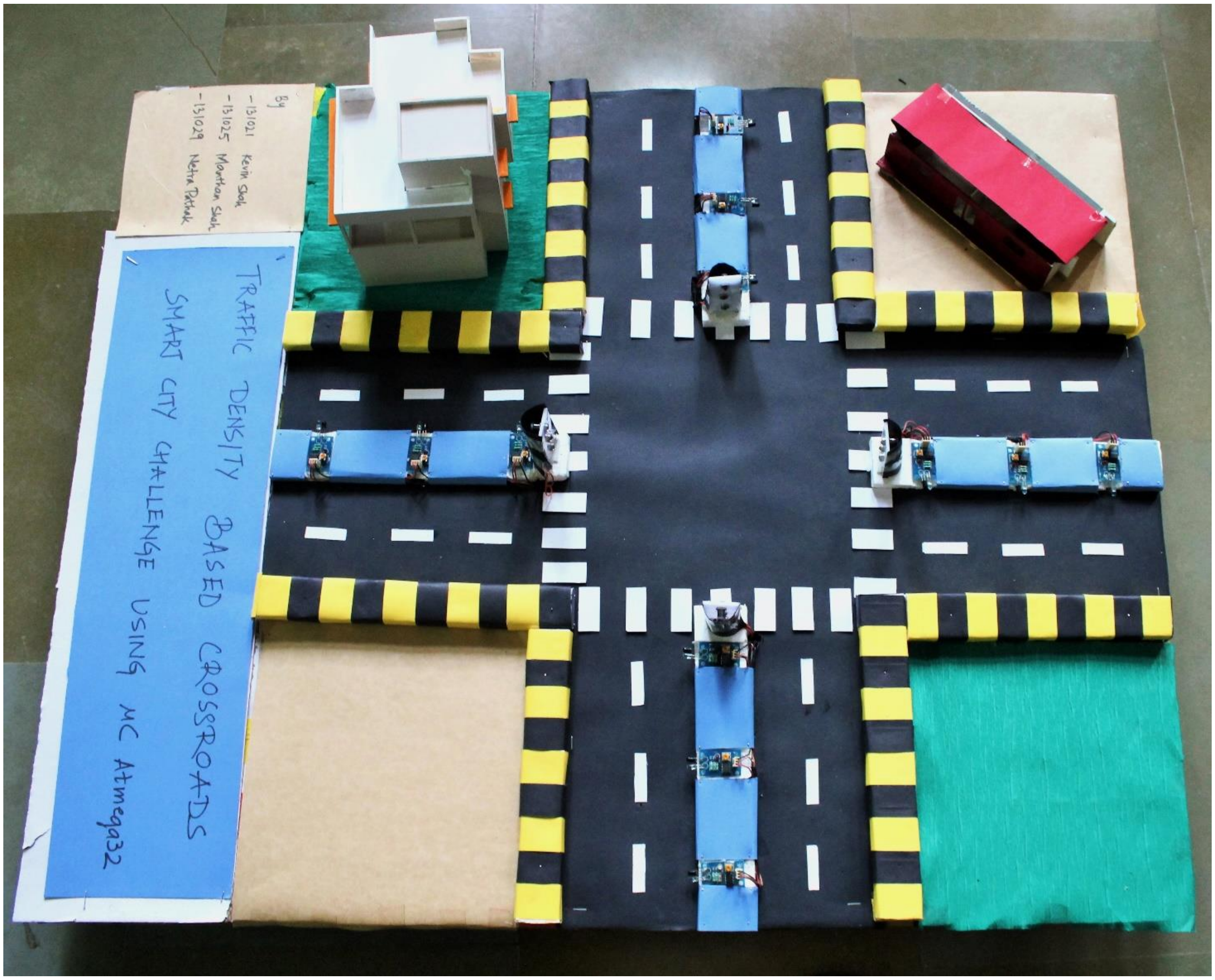
```
 break;
 }
}

void main()
{

 int i,j; //Variable declaration

 DDRA=0XFF; //Assigning PORTA as output (green LED-lower nibble & red LED-higher)
 PORTA=0XE1; //Initially start with lane 1 green signal and other 3 lanes red signal

 //Assigning PORTB lower nibble as O/P for yellow LED & higher nibble as I/P for
   IR sensors
 DDRB=0X0F;
 PORTB=0X00; //Sensors are pulled down

 DDRC=0X00; //Assigning PORTC as input for the other 9 IR sensors
 PORTC=0X00; //Pull down

 DDRD=0X00; //Assigning PORTD lower nibble as input for RF module
 PORTD=0X00;

/*Assigning initial values to the variables*/
 storeA=0xE1;
 storeB=0x00;

 /*Initialize*/
 for(i=0;i<3;i++)
 {
  for(j=0;j<3;j++)
      sensor[i][j]=0;
 }


/**********************************************************************/
 Setting up TCCR0

 For,
  Compare Mode:  WGM00=0 and WGM01=1
  External Clk/1024 Source  CS0[2:0] = 101
```

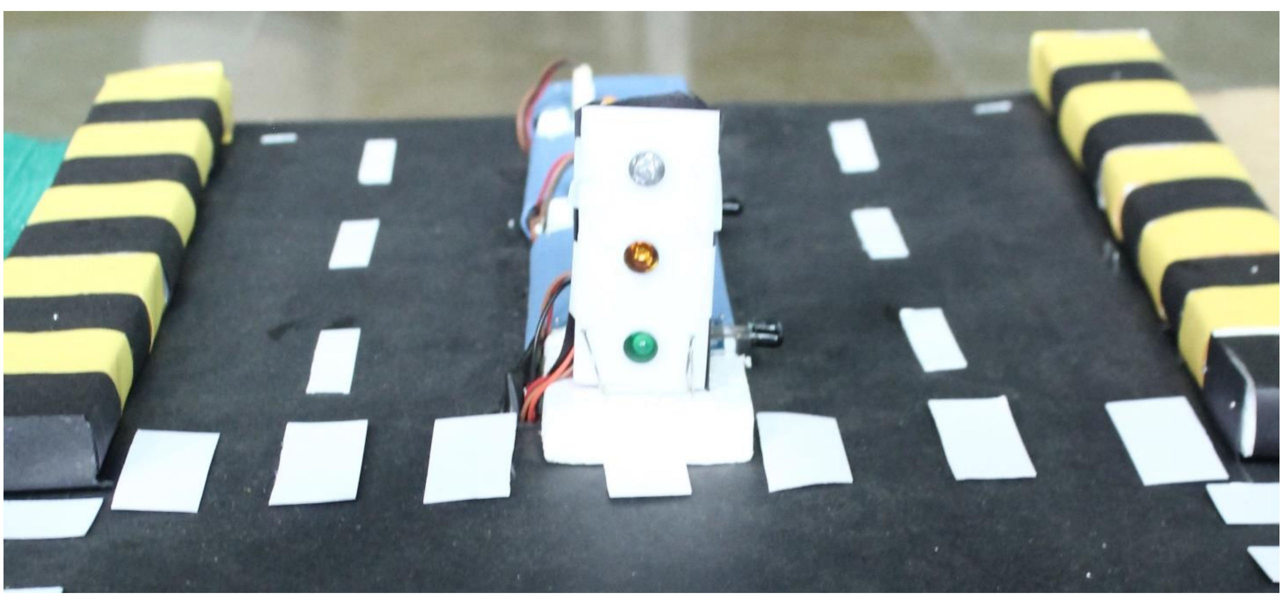
```
FOC0 WGM00 COM01 COM00 WGM01 CS02 CS01 CS00
 0      0         0         0        1          1     0     1

 TCCR0 load value = 0x0D
***************************************************************************/
 TCCR0=0X0D;
 OCR0=0XE9;

 TIMSK.B1=1; //TOIE1 Enable
 SREG.B7=1; // Global interrupt enable

 while(1)
 {
   /*For RF module- first time A button for stop- i.e. all signals Red,
      then A,B,C,D for lanes 1,2,3,4 respectively.
      Last A button for resume to control via traffic density*/

   if((PIND.B0==1) && !(resume_stop_check==2))
   {
     if(resume_stop_check==1)    //To resume to normal density based
     {
      TCCR0=0X0D;
      PORTA=store_resumeA;
      resume_stop_check=0;
      ir_check_mode=1;    //To resume IR sensor, density checking mode
     }

     //To control in case of emergency cases; for making all Red signals
     else if(resume_stop_check==0)
     {
       TCCR0=0X00; //Turn timer OFF
       PORTA=0xF0; //All lanes- RED signals
       PORTB=0X00; //Turn of yellow signals
       resume_stop_check=2;
       ir_check_mode=0; //To stop IR sensor, density checking mode
     }
     while(PIND.B0==1);
   }
```

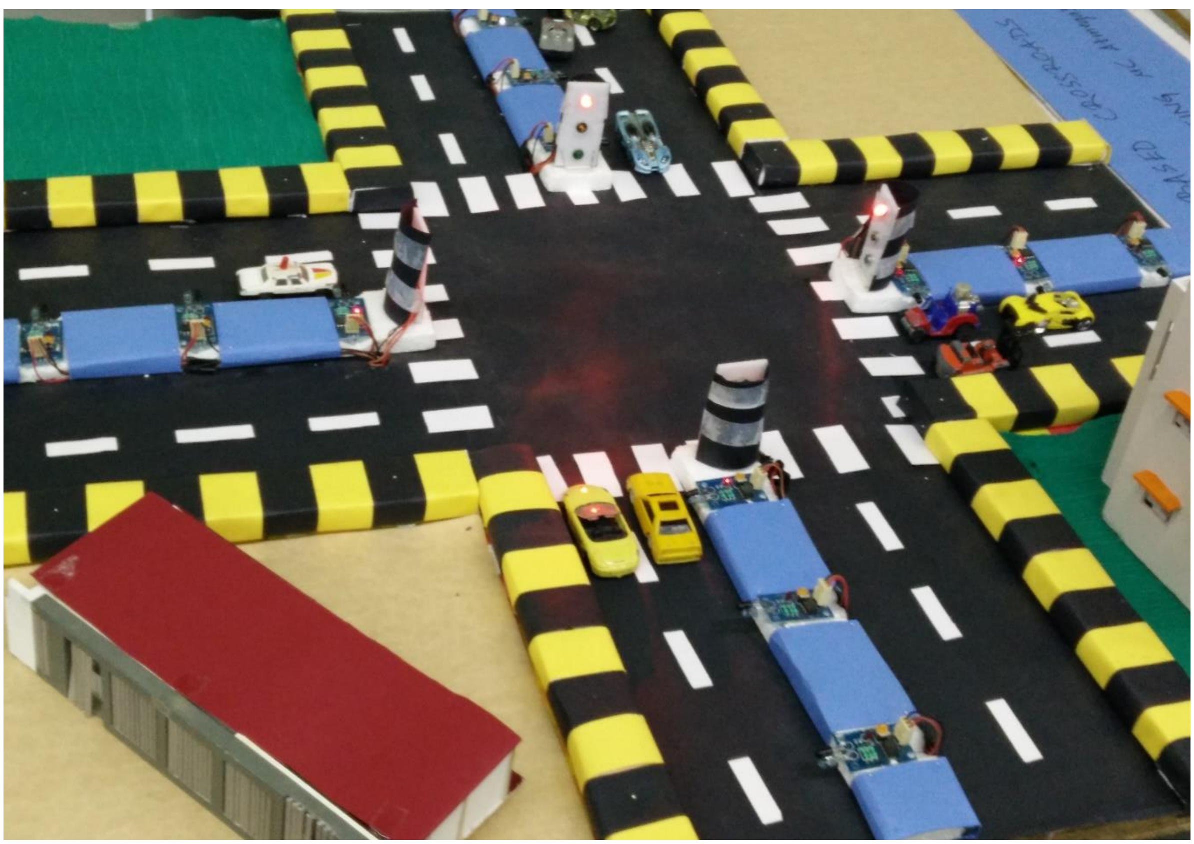
```
else if(resume_stop_check==2) //For selecting the lane to make green signal
{
    while((PIND.B0==0)&(PIND.B1==0)&(PIND.B2==0)&(PIND.B3==0));
    if(PIND.B0==1) //Remote button A for lane 1
    {
      PORTA=0XE1;
      while(PIND.B0==1);
    }

    else if(PIND.B1==1) //Remote button B for lane 2
    {
      PORTA=0XD2;
      while(PIND.B1==1);
    }

    else if(PIND.B2==1) //Remote button C for lane 3
    {
      PORTA=0XB4;
      while(PIND.B2==1);
    }

    else if(PIND.B3==1) //Remote button D for lane 4
    {
      PORTA=0X78;
      while(PIND.B3==1);
    }

    resume_stop_check=1;

}

/*To start IR checking to compute the traffic density when RF module not used*/
else if(ir_check_mode==1)
{

  //To start checking the state, 5 seconds before changing to the next lane
  if( (counter>=(delay_count-167)) && (counter<=delay_count) )
  {
  check_lane(); //Calling the function to check the state of the next lane by
                  computing the output values of the 3 IR sensors for the lane
  }
```

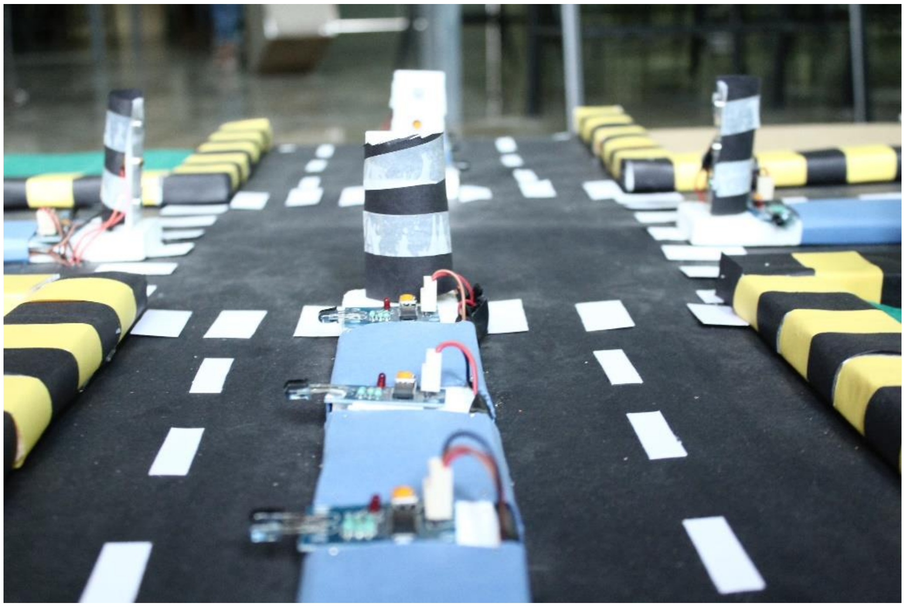
```
/*Logic for Yellow lights- between the transit of green to red;
  yellow signal starts 3 seconds before the signal is going to be red for
  the ongoing lane
*/
if((counter>=(delay_count-100)) && (counter<=(delay_count+5-100)) )
{

/*To retain the values of the green signals and pass the same to yellow;
   not considering the value in higher nibble */
 storeB = storeA & 0x0F; //Pass the green signal to yellow signal of the ongoing lane
 PORTB = storeB;
 PORTA = 0XF0 & storeA;
 store_resumeA=PORTA;
}

//Logic for changing the Red and Green lights for the next lane
if(counter>=delay_count)
{
 /*To check the state of the lane and give the time delay for green signal
    of the next lane accordingly
 */
 check_lane_status();
 PORTB=0x00;
 storeA = storeA << 1; //Shift to the next lane
 storeA = storeA + 0X10;
 PORTA = storeA;

  /*After all 4 lanes covered in the cycle, to start from the first lane again*/
  if(storeA == 0X00)
  {
         PORTA=0XE1;
         storeA=0xE1;
  }
 counter=0;
 store_resumeA=PORTA;
}
  }
 }
}
```

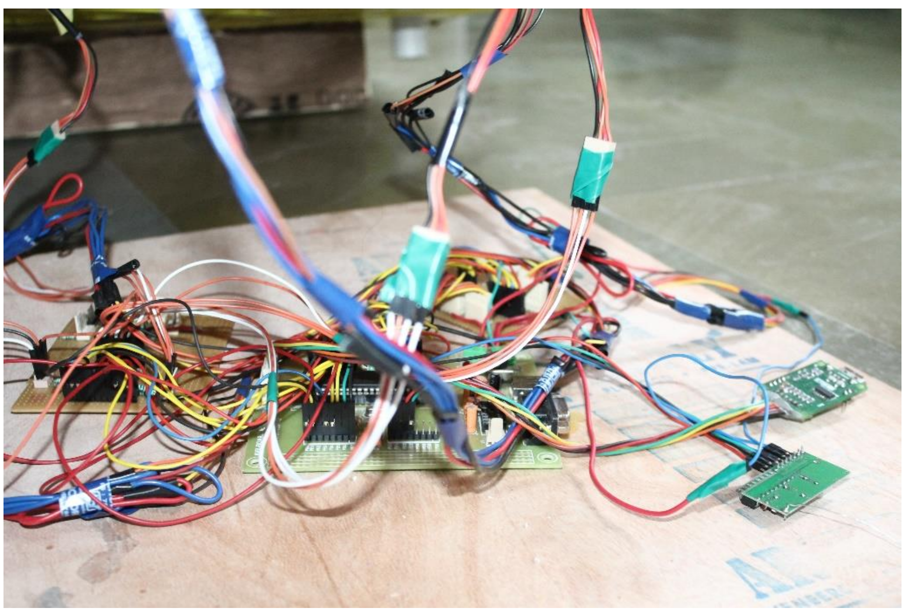
8. Photos


By
- 131021 Kevin Shah
- 131025 Manthan Shah
- 131029 Netra Pathak
TRAFFIC DENSITY BASED CROSSROADS
SMART CITY CHALLENGE USING µC Atmega32









9. Learning, Failures and Outcomes from the Project

Through this project we have learnt sensor interfacing (IR sensors) and wireless communication-transfer data/signals wirelessly; connecting these two to the embedded system consisting of the microcontroller and solving a real life problem. We also learnt to implement the RF module and form a logic for 6 different functionalities using only the 4 available buttons on the remote.

On the successful completion of this project, it gave us immense joy and happiness; as we learnt all the steps of making a good working Embedded System through this project.

Starting from designing to component identification and selection; implementing the logic on hardware, solving all the problems (debugging) faced while making the hardware; we could develop the firmware of our system. While doing this project we had an amazing experience as we learnt the key concepts in making of any Embedded System.

There were many challenges that we faced while in the process of trying to make the system work perfect!

From problems like failure of LM358 chip on the IR sensor, to burning of LED lights; loose improper connections, open resistor, etc., we learnt a lot while overcoming these. Also, we had to DE solder the general PCB of traffic lights twice.

Even for the logic part, we had to try several times and observe the wrong output many times to figure out the correct reason for the error. We developed the logic on paper and after writing the code; we had to debug each module several times, evaluating it for all possible cases before moving ahead to the next module. At times the code ran fine and sometimes it gave error. Such problems were difficult but, after continuous iterative number of try, trying to figure out the mis-happenings and solving those, we could finally get a perfect working model. At the end, we also tried to make our logic smarter by considering anomaly cases and other mis-functionalities, etc.

At the end, we have a working system that checks the traffic density for the lane and gives the time for green signal based on that. Also, we could solve the emergency case situation and blocking of IR sensors' due to random other reasons; but not due to the vehicles on the road.

REFERENCES

- http://grietinfo.in/projects/MINI/EEE/DOC-A.7-TLC%20report.pdf
- http://www.ijater.com/Files/IJATER_03_06.pdf
- http://www.ijarcsms.com/docs/paper/volume1/issue6/v1i6-0012.pdf
- http://www.slideshare.net/MuzzamilShaikh/smart-traffic-congestion-control-system
- http://www.slideshare.net/rajeevk281/traffic-signalproject
- http://www.slideshare.net/nskprasad/density-based-trafiic-control