Issue 6, Vol. 6 (November 2016)
ISSN 2249-6149

ADAPTIVE TRAFFIC CONTROL

Rohit Agrawal1, Avinash Sharma1
Hridaynarayan Vishwakarma1
and Ms.Sharmila Barwe2
1 Students TSEC, 2 Associate Professor, TSEC

Abstract – ***In this project we propose a method in which we are using a Bluetooth based mobile phone application to give command to the controller for clearance of the road for emergency vehicles. Also, we are interfacing IR modules with the controller which will have the count of vehicle in a particular lane and the traffic light will be toggled accordingly if the count crosses a predefined threshold value.***

I. Introduction

1.1 Existing Problem

Traffic lights play a very significant role in traffic control and regulation on a daily basis. The traffic lights that are used nowadays comprise of three lights: Red for stop, Yellow for wait and Green for go.

Users are made to wait for the signal to change from red to yellow and then from yellow to green. The time that a commuter has to wait for is decided by the traffic signals. The traffic lights used nowadays are hardwired at the time of installation.

They are pre-programmed to wait for a fixed duration of time after every change in signal. It is independent of the traffic on the roads and remains constant during its operation. Sometimes there is a situation where one particular road is very crowded as compared to others.

In such scenario's problems faced by emergency vehicle is significant, particularly Ambulance which is being stucked in a congested lane. It can even cost a life of a person who needs emergency medication.

1.2 Problem Solution

The system should be intelligent enough to decide the priority on a daily basis. In this system, basically, the waiting time for the motorists on road with higher density is reduced. In doing so, the density for each lane measured and a decision is passed as to which lane should be given how much amount of time and which should be the highest priority. The lane which is overcrowded should be released for a longer time compared to other lanes, in such case it becomes implicit to take a decision dynamically based on density and allocate a time to hardwired traffic signals simultaneously.

To achieve this we must have a count of vehicle in any particular lane waiting for the clearance of the lane. This can be done using IR sensors which will detect each vehicle passing through it, data will be sent to microcontroller unit which will have a vector to count the number of vehicles in that particular lane, accordingly, when the count reaches a predefined threshold, the lane will be cleared.

To handle the vehicles which need urgent clearance we can have a separate method to handle them here we use a Bluetooth operating application, which will be installed in the driver's phone, to send the request for clearing the line to Bluetooth module attached to microcontroller unit, to toggle the traffic signal to green.

II. Proposed Solution

Our project implements the proposed solution which interfaces app, IR sensors and MCU to control traffic signal as shown in block diagram.

A. Block Diagram & Working

As shown in the block diagram, we use Atmega 8 to control the traffic signal an IR module is interfaced with the MCU at port c the IR module is being used to inform the MCU about the density on any particular lane .For a Four lane junction we are using 4 IR modules to provide the density information to the MCU. In the absence of any information from the IR module the traffic signal works as usual.

If the microcontroller comes to know from the IR module that the threshold in any particular lane is exceeded, than that particular lane's green signal timing will be increased by pre-defined time unit .For example, let the default time for the green signal of any lane is 30 seconds Threshold count=10 vehicle and a particular lane, say lane2 has 12 vehicles waiting for the clearance so when the lane2 gets the green signal it will persist for 40 seconds the extra 10 seconds added to the signal is predefined time unit which got added due to the density in lane2.

To handle the emergency vehicle we are using BT module along with IR module. The receiver of the BT module is connected to transmitter of the MCU and vice versa. The Bluetooth module is operated using a android phone application. The driver of any emergency vehicle must have this app installed in his/her mobile phone .If the driver needs to clear a particular lane he has to send a request through this application to the BT module attached to MCU .The request is passed to the microcontroller as soon as the MCU receives this request, it gives the red signal to all the other lanes and green signal to the lane being requested to be clear.

If the application can be downloaded by any person then there will be big traffic chaos as everyone will try to clear his/her lane first .To overcome this problem the application must be provided to the legitimate users only with a unique registration ID and password ,so that any other intruder can't have access to application thus avoiding the mentioned chaos.

Suppose a situation in which lane1 has more no of vehicles than threshold count and at the same time an emergency vehicle, say an ambulance has arrived at lane lane2 and requested for the clearance for lane2 in such occasion even if the lane1 is being served it will be pre-empted by lane2 .Once the ambulance has passed lane2, the normal operation will resume.

This way the lane having emergency vehicle is given more priority than the lane having a density which has exceeded the threshold.

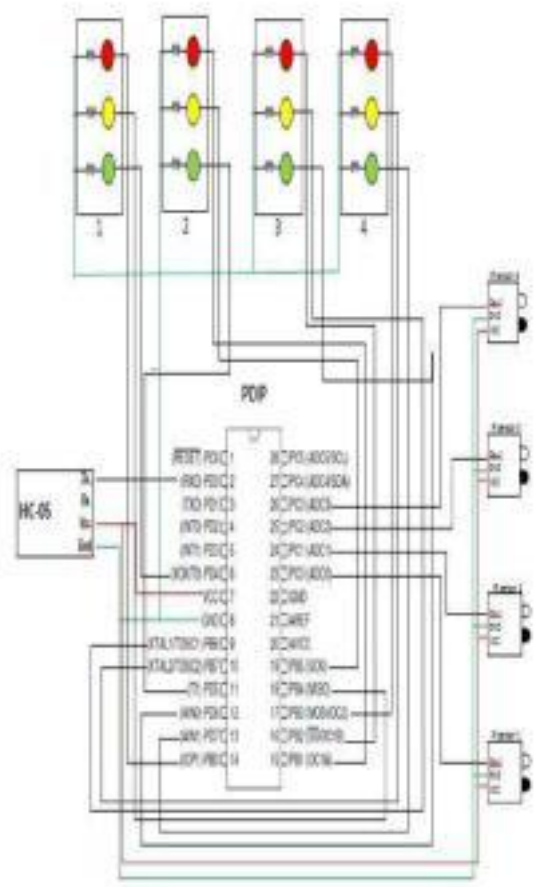
Take an another case, in which lane1 is currently being served two other lanes say 2 & 4 send request for clearance of their respective lane at the same time in such case the system will work in predefined cyclic order that is lane2 will be cleared prior to lane4.

B. Circuit Diagram

The circuit Diagram Shown below shows the detailed connection of different modules with the microcontroller unit. We are using four IR modules in this project. One Bluetooth module is used here for the handseling of emergency vehicles. All the modules are interfaced with the microcontroller using ribbon wires. The

Issue 6, Vol. 6 (November 2016)
ISSN 2249-6149

modules use serial communication to work with the microcontroller unit.



C. About the application

Bluetooth terminal emulator for communicating with any serial device using a bluetooth serial adapter. The RFCOMM/SPP protocol emulates serial communications over bluetooth. You need a bluetooth serial adapter like the HC-05 or HC-06.

You can use this to control your Arduino or Raspberry Pi, if you have connected a bluetooth serial adapter to its serial port.

This is an updated version to https://play.google.com/store/apps/details?id=es.pymasde.blueterm. In the original version it was impossible to use commands like 'ctrl-c' on newer devices with just a few buttons. Both volume keys are set to be the special key for those commands now. Simply press one of them together with e.g. 'c' on your android keyboard and the output will be 'ctrl-c'. This is an open source application. You can get more info and the source code from link presented in reference section.

D. Code for Microcontroller

```
#include<avr/io.h>//basic headder for IO

#include<util/delay.h>// delay headder

#include<avr/serial.h>// serial communication headder

#include<avr/clock.h>// standart 1sec clock headder


#define setbit(ADD,BIT) (ADD|=(1<<BIT)) //set bit to 1

#define clrbit(ADD,BIT) (ADD&=(1<<BIT)) //clear bit to 0


unsigned char ltime=5;// line timer5, 20

unsigned char current_line=1;//current line

unsigned char old_line=4;//old line

unsigned char emv=0;// emergency vehicle identifier from UART serial


void line1(void);// line selection function L1

void line2(void);// line selection function L2

void line3(void);// line selection function L3

void line4(void);// line selection function L4

void change(void);// change trafic light state

void astop(void);//all signals stop red

void sencheck(void);//update vehicle count with sensor check


void main(void)

{ //LED poles

//      1          2          3          4


setbit(DDRB,0);setbit(DDRB,1);setbit(DDRB,2);setbit(DDRB,3);// RED
```

```
setbit(DDRB,4);setbit(DDRB,5);setbit(DDR
B,6);setbit(DDRB,7);// YELLOW

setbit(DDRD,4);setbit(DDRD,5);setbit(DDR
D,6);setbit(DDRD,7);//GREEN

PORTB=0x00;PORTD=0x00;//PORTC=0x0 0; //leds
off
 //============================
 //sensors
 //

clrbit(DDRC,0);clrbit(DDRC,1);clrbit(DDR
C,2);clrbit(DDRC,3);To clear the the bits for the next
input

setbit(PORTC,0);setbit(PORTC,1);setbit(P
ORTC,2);setbit(PORTC,3); To trigger the IR sensors
when the object passes
 //============================

 //UART to BLUETOOTH to indicate emergency
vehecle

 uart_init();//configs serial communication with
bluetooth app
 clock_init(0,0,0);//configuring system clock
 _delay_ms(10);
 line1();//starts with line one

 while(1)
 {change();// changes as per the given
signal cases and timing     }
}//end of main

//sequential flow
void line1(void)// line selection function L1
RYG
{   astop();
clrbit(PORTB,0);setbit(PORTB,4);clrbit(P
ORTD,4);
 current_line=1; }

void line2(void)// line selection function L2
{astop();

clrbit(PORTB,1);setbit(PORTB,5);clrbit(P
ORTD,5);
 current_line=2; }

void line3(void)// line selection function L3
{astop();

clrbit(PORTB,2);setbit(PORTB,6);clrbit(P
ORTD,6);
 current_line=3; }

void line4(void)// line selection function L4
{   astop();

clrbit(PORTB,3);setbit(PORTB,7);clrbit(P
ORTD,7);
 current_line=4; }

//================================
================================
void change(void)// change trafic light state
when ambulance is detected
{      if(ltime==sec-2)
       {
          sec=0;old_line=current_line;//store
line value
          switch(emv)// ambulance check
           {
            case '1': {   line1(); emv=0;   } break;
            case '2': {   line2(); emv=0;   } break;
```

```
            case  '3': {     line3(); emv=0;     }
break;
            case  '4': {     line4(); emv=0;     }
break;
          default:       // regular check
                {
          if(current_line==1)
                {         line2();        }
                else if(current_line==2)
                {         line3();        }
                else if(current_line==3)
                {         line4();        }
              else
                    {         line1();        }

                        }//default end
        } // switch end
      }//if end
  }// change end

void astop(void)// all red lights turn on
{PORTB=0x0f;PORTD&=0x0f;
              }

ISR(TIMER1_COMPA_vect)// sec counting vector from clock headder file to perform the blinking of the LEDs every 2 sec
{
  sec++;
  if(sec==1)// 2 by default tocontrol yellow time to green
  {
  if(current_line==1)
  {setbit(PORTD,4);clrbit(PORTB,4);}// to turn on the LED for 2 sec and then turn it off
  else if(current_line==2)
  {setbit(PORTD,5);clrbit(PORTB,5);}
  else if(current_line==3)
  {setbit(PORTD,6);clrbit(PORTB,6);}
  else if(current_line==4)
  {setbit(PORTD,7);clrbit(PORTB,7);}
  }
}

ISR(USART_RXC_vect)// To use UART receive interrupt routine and to symbolise that is a vector name.
{         emv=UDR;                    }
```

III. CONCLUSION

This technique can be effective to combat the growing pressure of traffic on Indian roads for ambulances. It uses an app which works on bluetooth to detect the ambulances on roads and regulates the traffic accordingly.

Adaptive traffic system enhances the priority of lanes based on density of vehicles from traffic junction respectively. By doing so, it not only passes the ambulances in emergency situation but also serves regulating congestion at peak hours.

It is cost efficient and does not require the installation of complex machinery to monitor the traffic.

Deploying this system will not only save the time consumed in waiting at traffic junctions, but will also conserve a lot of lives.

IV. ACKNOWLEDGEMENT

We are highly indebted to THADOMAL SHAHANI ENGINEERING COLLEGE for its guidance and constant supervision as well as for providing necessary information regarding the project & also for supporting us.

We would significantly like to thank our guide Prof. SHARMILA BARWE for her guidance throughout the project work.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

Issue 6, Vol. 6 (November 2016)
ISSN 2249-6149

V. REFERENCES

1. http://pymasde.es/blueterm/

2. http://www.atmel.com/avr.

3.https://www.youtube.com/watch?v=j7ymvQY8zPE

4. www.atmel.com/images/atmel-8159-8-bit-avr-microcontroller-atmega8a_datasheet

5. https://en.wikipedia.org/wiki/Serial_comm unication

6. https://www.youtube.com/watch?v=-z_0aU8VHzk

7. http://www.microcontrollerboard.com/anal og-to-digital-converter.html

8. https://www.youtube.com/watch?v=FnIBP aViRbM