**PROJECT TITLE: FOUR WAYS TRAFFIC LIGHT CONTROL USING GSM**

**ABSTRACT:**

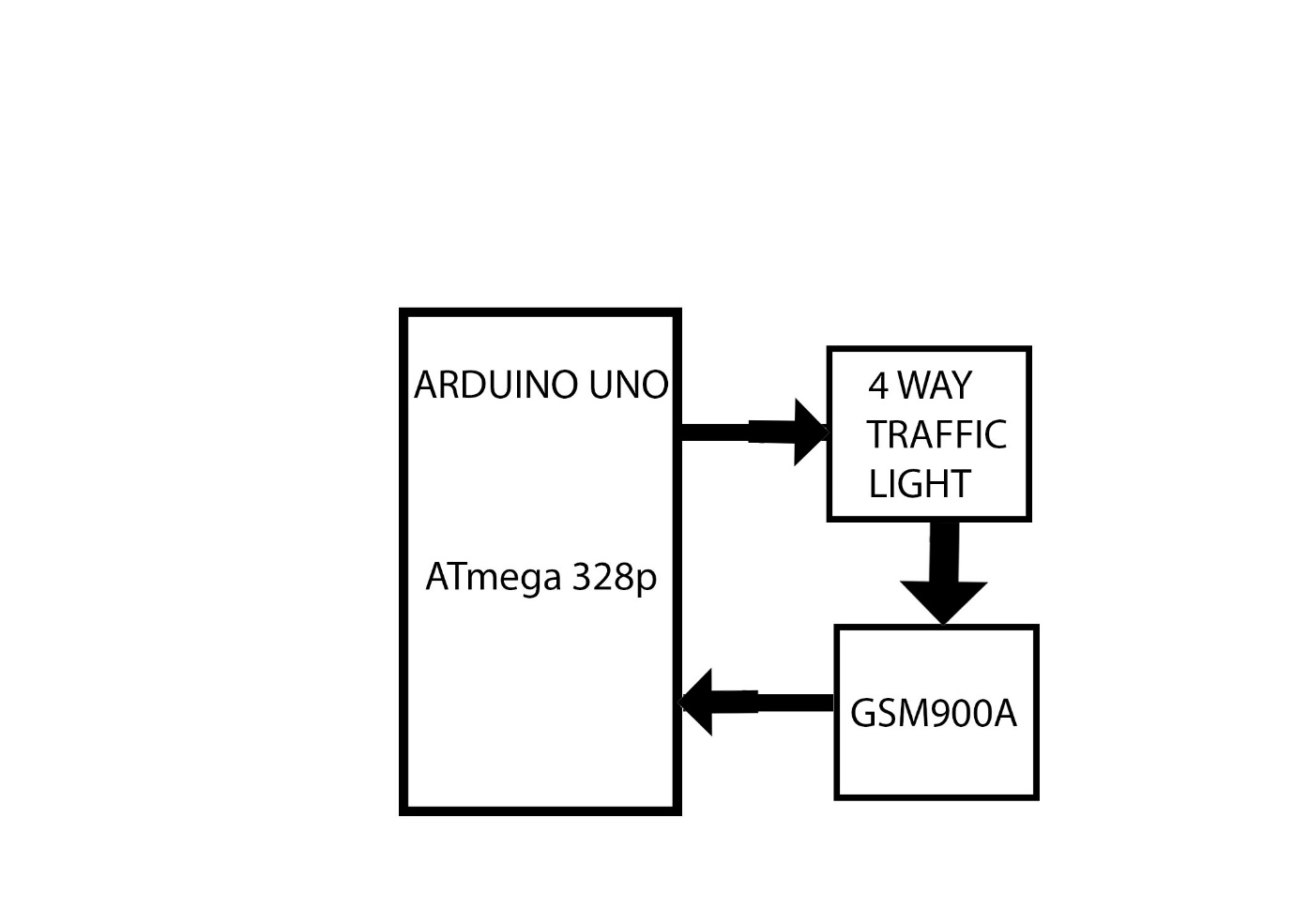
The increasing population growth, urbanization, and motorization have resulted in increased road traffic congestions worldwide. Forecasts imply that road traffic will grow faster than road capacity within the next years, leading to a worsening of the traffic situation. Requirements to improve traffic congestions include efficient traffic management, using smart traffic control systems. This system not only allows traffic management but also traffic reporting advice to road users.The occurrence of traffic congestion at the intersections in cities is one of major problem. Thepresent traffic light controlling system is not sufficient to tackle the traffic congestion due to the lack of efficiency in designing. Presently the automated systems have fewer manual operations, but high flexibility, reliability and accuracy. Due to these advantages every field prefers automated control systems. The monitoring and control of city traffic is becoming a major problem in many countries. With the ever-increasing number of vehicles on the road, the Traffic Monitoring Authority has to find new methods of overcoming such a problem. Simulation 0f the proposed design was performed using proteus software. *The function of a traffic light controller requires sophisticated control and coordination to ensure that traffic moves as smoothly and safely as possible*

This paper also aims to design a safe and efficient traffic flow, to assign the right way and minimizes the delay or waiting time at road. This system also provides GSM phone interface to the user, with SMS facility to those who wish to obtain the latest traffic information on congested roads. The map of the traffic signal will be provided to the users on their request.

PROBLEM STATEMENT

A Traffic light controller plays a very important role to control the vehicles on the busy roads. Nowadays there are occurring many accidents because of the lack of properly maintained traffic signals and design. Besides, it creates a huge traffic jam. the flow of vehicles in most of the crossways is not properly maintained. This traffic light controller can be used to solve the traffic problem.

To solve these congestion problems, we have to build new facilities and infrastructure. The only disadvantage of making new roads on facilities is that it makes the surroundings more congested. So, for that reason we need to change the system rather than making new infrastructure twice. The Main goals of this paper are improving safety, minimizing travel time and increasing the capacity of Infrastructures. Such improvements are beneficial to health, economy and the environment. Measurably improve traffic flow

**BLOCK DIAGREM** 

**DISCRIPTION**

ARDUINO UNO it has microcontroller which have code that control traffic light, that traffic light it has three **led** **yellow, red** and **green** so it has the resistor of 1k ohm which used to protect the led, the traffic light it will receiver signal from Arduino uno so when **RED COLOR LED** which is parallel **north** and **south** it shows that road it is closed so the **GSM** its receiver the message which tell that road **north** and **south** is closed another road is open even road **EAST** and **WEST** is closed another open , **YELLOW COLOR LED** which is **north** and **south** it give the passenger signal to be read to pass so the **GSM** its receiver the message which tell road **north** and **south** is closed another road is open even **east** and **west** is closed another road is open ,**GREEN COLOR RED** which is **north** and **south** it give the passenger to pass so that the **GSM** its receiver the message which tell that road **north** and **south** is open another closed even **east** and **west** is open another is closed .

**SOURCE CODE**

**#include <SoftwareSerial.h>**

**char inchar; // Will hold the incoming character from the GSM shield**

**SoftwareSerial SIM900(0, 1); // gsm module connected here.**

**const int GN=2;**

**const int YN=3;**

**const int RN=4;**

**const int GS=5;**

**const int YS=6;**

**const int RS=7;**

**const int GE=8;**

**const int YE=9;**

**const int RE=10;**

**const int GW=11;**

**const int YW=12;**

**const int RW=13;**

**void setup()**

**{**

**Serial.begin(9600);**

**SIM900.begin(9600); // original 19200**

**// delay(10000); // give time to log on to network.**

**randomSeed(analogRead(0));**

**SIM900.print("AT+CMGF=1\r"); // set SMS mode to text**

**delay(1000);**

**SIM900.print("AT+CNMI=2,2,0,0,0\r");**

**// blurt out contents of new SMS upon receipt to the GSM shield's serial out**

**delay(1000);**

**SIM900.println("AT+CMGD=1,4"); // delete all SMS**

**delay(5000);**

**Serial.println("Ready...");**

**}**

**void sendSMS(String message)**

**{**

**SIM900.println("AT+CMGF=1\r"); // AT command to send SMS message**

**delay(1000);**

**SIM900.println("AT+CMGS = \"+923339218213\""); // recipient's mobile number, in international format**

**delay(1000);**

**SIM900.println(message); // message to send**

**delay(1000);**

**SIM900.println((char)26); // End AT command with a ^Z, ASCII code 26**

**delay(1000);**

**SIM900.println();**

**delay(1000); // give module time to send SMS**

**// turn off module**

**}**

**void loop() {**

**if(SIM900.available() == 0)**

**{**

**// Wait a few seconds between measurements.**

**delay(2000);**

**pinMode(GN,OUTPUT);**

**pinMode(YN,OUTPUT);**

**pinMode(RN,OUTPUT);**

**pinMode(GS,OUTPUT);**

**pinMode(YS,OUTPUT);**

**pinMode(RS,OUTPUT);**

**pinMode(GE,OUTPUT);**

**pinMode(YE,OUTPUT);**

**pinMode(RE,OUTPUT);**

**pinMode(GW,OUTPUT);**

**pinMode(YW,OUTPUT);**

**pinMode(RW,OUTPUT);**

**}**

**if(SIM900.available() == 0)**

**{**

**// Wait a few seconds between measurements.**

**delay(2000);**

**}**

**digitalWrite(GN,HIGH);**

**digitalWrite(GS,HIGH);**

**digitalWrite(RE,HIGH);**

**digitalWrite(RW,HIGH);**

**digitalWrite(YN,LOW);**

**digitalWrite(RN,LOW);**

**digitalWrite(YS,LOW);**

**digitalWrite(RS,LOW);**

**digitalWrite(GE,LOW);**

**digitalWrite(YE,LOW);**

**digitalWrite(GW,LOW);**

**digitalWrite(YW,LOW);**

**Serial.println("Road North and Road South Open others Closed");**

**SIM900.println("Road North and Road South Open others Closed");**

**delay(2000);**

**digitalWrite(YN,HIGH);**

**digitalWrite(YS,HIGH);**

**digitalWrite(RE,HIGH);**

**digitalWrite(RW,HIGH);**

**digitalWrite(GN,LOW);**

**digitalWrite(RN,LOW);**

**digitalWrite(GS,LOW);**

**digitalWrite(RS,LOW);**

**digitalWrite(GE,LOW);**

**digitalWrite(YE,LOW);**

**digitalWrite(GW,LOW);**

**digitalWrite(YW,LOW);**

**Serial.println("Road North and Road South Open others Closed");**

**delay(2000);**

**digitalWrite(RN,HIGH);**

**digitalWrite(RS,HIGH);**

**digitalWrite(GE,HIGH);**

**digitalWrite(GW,HIGH);**

**digitalWrite(GN,LOW);**

**digitalWrite(YN,LOW);**

**digitalWrite(GS,LOW);**

**digitalWrite(YS,LOW);**

**digitalWrite(YE,LOW);**

**digitalWrite(RE,LOW);**

**digitalWrite(YW,LOW);**

**digitalWrite(RW,LOW);**

**Serial.println("Road West and Road East Open others Closed");**

**delay(2000);**

**digitalWrite(RN,HIGH);**

**digitalWrite(RS,HIGH);**

**digitalWrite(YE,HIGH);**

**digitalWrite(YW,HIGH);**

**digitalWrite(GN,LOW);**

**digitalWrite(YN,LOW);**

**digitalWrite(GS,LOW);**

**digitalWrite(YS,LOW);**

**digitalWrite(GE,LOW);**

**digitalWrite(RE,LOW);**

**digitalWrite(GW,LOW);**

**digitalWrite(RW,LOW);**

**Serial.println("Road West and Road East Open others Closed");**

**delay(2000);**

**if(SIM900.available() >0)**

**{**

**inchar=SIM900.read();**

**Serial.println(inchar);**

**delay(20);**

**if (inchar=='v')**

**{**

**delay(10);**

**Serial.println(inchar);**

**// Wait a few seconds between measurements.**

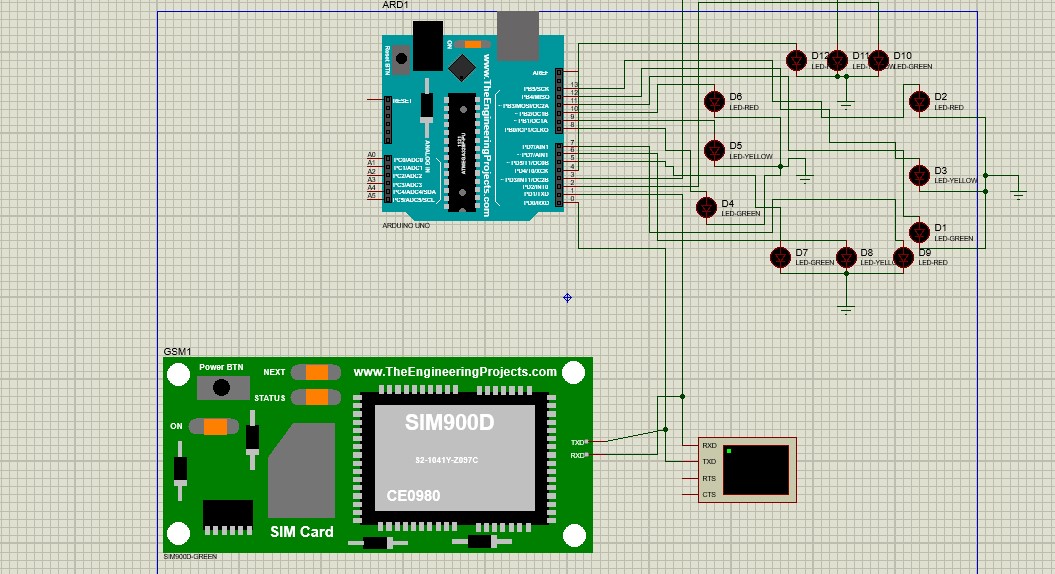
**delay(2000);**

**}**

**}**

**}**

**CIRCUIT IN PROTEUS**

****