ABSTRACT

The picture of smart city contains list of services that describes level of aspiration. To provide for the aspiration and needs of the public urban planners ideally aim to developing entire urban ecosystem, which is represented by four pillars of development – institutional, physical, social and economic infrastructure. Currently, in the whole world, enormous electric energy is consumed by the street lamps, which are automatically turn on when it becomes dark and automatically turn off when it becomes bright. This is the huge waste of energy in the whole world and should be changed. Automation plays an increasingly important role in the world economy and in daily life. Automatic systems are being preferred over manual system. The research work shows automatic control of street lights, gas and temperature sensors as a result of which power is saved to some extent, while at the same time taking care of human safety too.

The main goals of our project include a way to detect vehicle movement on highways, switch ON only a block of street lights ahead of it (vehicle), and to switch OFF the trailing lights to save energy. During the night, all the lights on the highway remain ON for the vehicles, but lots of energy is wasted when there is no vehicle movement. So, this can be used for lights in parking areas of industries, hotels, restaurants, etc.

1.INTRODUCTION

1. INTRODUCTION

Our Project “SMART CITY MANAGEMENT” defines city architected to address public issues via ICT-based solutions on the basis of a multi-stakeholder municipally based partnership. A smart-city is a socio-technical system of systems. The final aim of smart city is to support better living, create more opportunities, support stronger and more cohesive communities and improve the quality of life overall for all residents. Smart City is a developed urban area that create sustainable economic development & high quality of life by excelling in multiple key areas ; Economic , Mobility ,environment , people , living & Govt. excelling in these key areas can be done so through strong human capital ,social capital and/or ICT infrastructure

2.LITERATURE SURVEY

2. LITERATURE SURVEY

S.Suganya et al [1] have proposed about Street Light Glow on detecting vehicle movement using sensor is a system that utilizes the latest technology for sources of light as LED lamps. It is also used to control the switching of street light automatically according to the light intensity to develop flow based dynamic control statistics using infrared detection technology and maintain wireless communication among lamppost and control terminal using ZigBee Wireless protocol. It also combines various technologies: a timer, a statistics of traffic flow magnitude, photodiodes, LED, power transistors.

M.Abhishek et al [3] have implemented design of traffic flow based street light control system with effective utilization of solar energy in the year 2015. They used the renewable source of energy i.e. the solar power for street lighting. They have also used 8052 series microcontroller and is developed by replacing the normal bulbs with the LEDs due to which the power consumption is reduced by 3 times. Sensors are placed on either side of the road which senses the vehicle movement and sends the commands to the microcontroller to switch ON and OFF the lights. Here all the street lights remain switched off and it glows only when it senses the vehicle movement. Hence, because of the microcontroller, even when its night the lights are switched off.

Radhi Priyasree [4] explains a system to reduce the power consumption of street lights by avoiding inefficient lighting which wastes significant financial resources each year. This is done by dimming the lights during less traffic hours. It also aims at detecting consumption of alcohol by the driver and if it exceeds certain level it impairs the driver from entering into the Vehicle. This prevents occurrence of accidents or any fatal crashes. This initiative will help the government to save this energy and meet the domestic and industrial needs.

From this literature survey, the methods each one has implemented and used is simple and easy to understand. These papers and journals has given many ideas to further implement a much efficient system and make things automated.

3. BLOCK DIAGRAM

3.BLOCK DIAGRAM

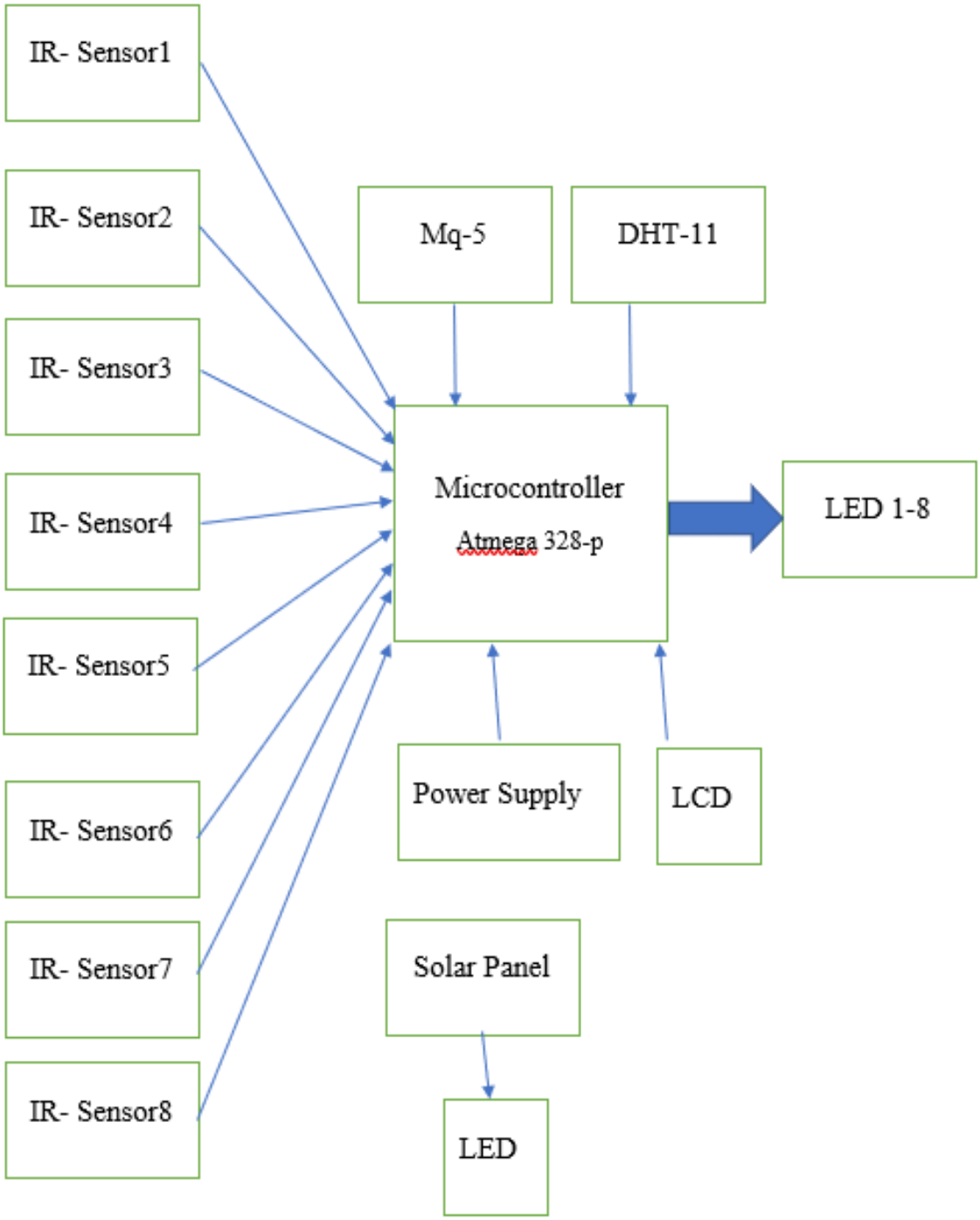

Fig. 3.1: Block Diagram Of Project

Block Description:

LDR:

LDRs or Light Dependent Resistors are very useful especially in light/dark sensor circuits. Normally the resistance of an LDR is very high, sometimes as high as 1000 000 ohms, but when they are illuminated with light resistance drops dramatically. When the light level is low the resistance of the LDR is high. This prevents current from flowing to the base of the transistors. Consequently the LED does not light. However, when light shines onto the LDR its resistance falls and current flows into the base of the first transistor and then the second transistor. The LED lights.

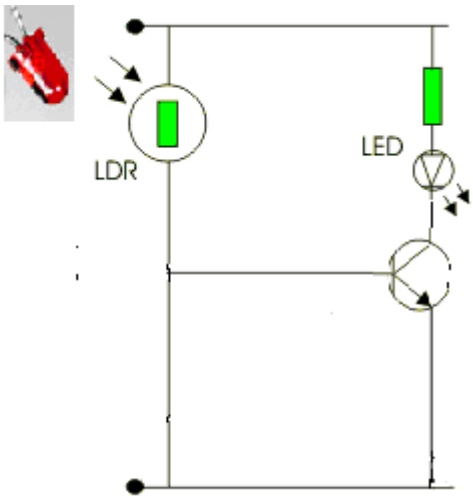

Fig. 2.2:LDR Diagram

Here in our project we place the LDR in open, so that it can take direct sunlight from sun. When the sun sets and amount of light decreased the LDR gives output which is monitored by the micro controller.

LIGHT-EMITTING DIODE (LED) :

Light-emitting diodes are elements for light signalization in electronics. They are manufactured in different shapes, colours and sizes. For their low price, low consumption and simple use, they have almost completely pushed aside other light sources- bulbs at first place. They perform similar to common diodes with the difference that they emit light when current flows through them.

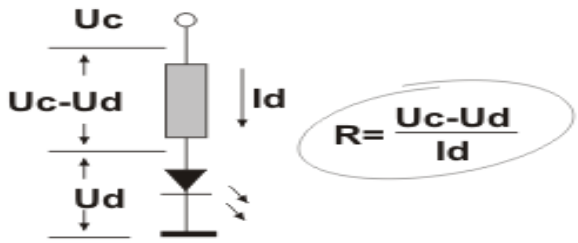

Fig. 2.3 : LED Diagram

It is important to know that each diode will be immediately destroyed unless its current is limited. This means that a conductor must be connected in parallel to a diode. In order to correctly determine value of this conductor, it is necessary to know diode's voltage drop in forward direction, which depends on what material a diode is made of and what colour it is. Values typical for the most frequently used diodes are shown in table below: As seen, there are three main types of LEDs. *Standard* ones get full brightness at current of 20mA. *Low Current* diodes get full brightness at ten times lower current while *Super Bright* diodes produce more intensive light than Standard ones.

Since the 8051 micro controllers can provide only low input current and since their pins are configured as outputs when voltage level on them is equal to 0, direct connecting to LEDs is carried out as it is shown on figure (*Low current* LED, cathode is connected to output pin).

RELAYS:

A relay is an electrically controllable switch widely used in industrial controls, automobiles and appliances.

The relay allows the isolation of two separate sections of a system with two different voltage sources i.e., a small amount of voltage/current on one side can handle a large amount of voltage/current on the other side but there is no chance that these two voltages mix up.

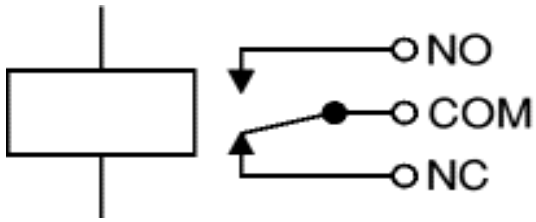

Fig 2.4: Circuit symbol of a relay

OPERATION:

When current flows through the coil, a magnetic field is created around the coil i.e., the coil is energized. This causes the armature to be attracted to the coil. The armature's contact acts like a switch and closes or opens the circuit. When the coil is not energized, a spring pulls the armature to its normal state of open or closed. There are all types of relays for all kinds of applications.

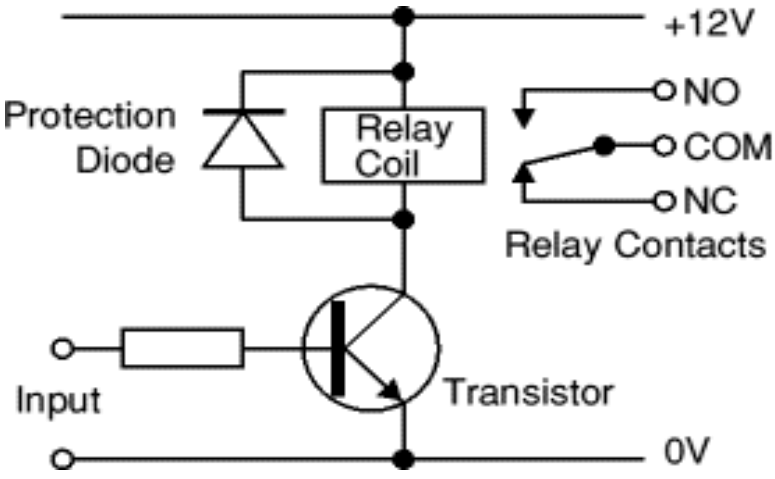

Fig 2.5: Relay Operation and use of protection diodes

Transistors and ICs must be protected from the brief high voltage 'spike' produced when the relay coil is switched off. The above diagram shows how a signal diode (eg 1N4148) is connected across the relay coil to provide this protection. The diode is connected 'backwards' so that it will normally not conduct. Conduction occurs only when the relay coil is switched off, at this moment the current tries to flow continuously through the coil and it is safely diverted through the diode. Without the

diode no current could flow and the coil would produce a damaging high voltage 'spike' in its attempt to keep the current flowing.

In choosing a relay, the following characteristics need to be considered:

1. The contacts can be normally open (NO) or normally closed (NC). In the NC type, the contacts are closed when the coil is not energized. In the NO type, the contacts are closed when the coil is energized.

2. There can be one or more contacts. i.e., different types like SPST (single pole single throw), SPDT (single pole double throw) and DPDT (double pole double throw) relays.

3. The voltage and current required to energize the coil. The voltage can vary from a few volts to 50 volts, while the current can be from a few milliamps to 20milliamps. The relay has a minimum voltage, below which the coil will not be energized. This minimum voltage is called the “pull-in” voltage.

4. The minimum DC/AC voltage and current that can be handled by the contacts. This is in the range of a few volts to hundreds of volts, while the current can be from a few amps to 40A or more, depending on the relay.

DRIVING A RELAY:

An SPDT relay consists of five pins, two for the magnetic coil, one as the common terminal and the last pins as normally connected pin and normally closed pin. When the current flows through this coil, the coil gets energized. Initially when the coil is not energized, there will be a connection between the common terminal and normally closed pin. But when the coil is energized, this connection breaks and a new connection between the common terminal and normally open pin will be established. Thus when there is an input from the micro controller to the relay, the relay will be switched on. Thus when the relay is on, it can drive the loads connected between the common terminal and normally open pin. Therefore, the relay takes 5V from the micro controller and drives the loads which consume high currents. Thus the relay acts as an isolation device.

Digital systems and micro controller pins lack sufficient current to drive the relay. While the relay’s coil needs around 10milli amps to be energized, the micro

controller's pin can provide a maximum of 1-2milli amps current. For this reason, a driver such as ULN2003 or a power transistor is placed in between the micro controller and the relay. In order to operate more than one relay, ULN2003 can be connected between relay and micro controller.

Block Diagram of Power Supply:

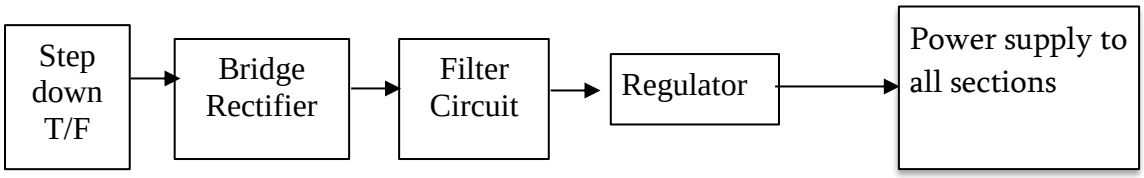

Fig 2.6: Block Diagram Of Power Supply

POWER SUPPLY:

The input to the circuit is applied from the regulated power supply. The a.c. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating d.c voltage. So in order to get a pure d.c voltage, the output voltage from the rectifier is fed to a filter to remove any a.c components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

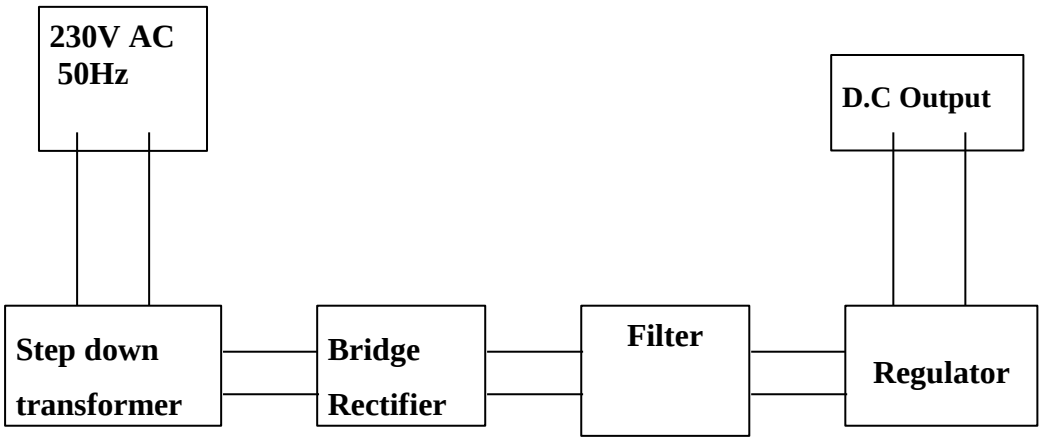

Fig 2.7: Power Supply

TRANSFORMER:

Usually, DC voltages are required to operate various electronic equipment and these voltages are 5V, 9V or 12V. But these voltages cannot be obtained directly. Thus the a.c input available at the mains supply i.e., 230V is to be brought down to the required voltage level. This is done by a transformer. Thus, a step down transformer is employed to decrease the voltage to a required level.

RECTIFIER:

The output from the transformer is fed to the rectifier. It converts A.C. into pulsating D.C. The rectifier may be a half wave or a full wave rectifier. In this project, a bridge rectifier is used because of its merits like good stability and full wave rectification.

FILTER:

Capacitive filter is used in this project. It removes the ripples from the output of rectifier and smoothens the D.C. Output received from this filter is constant until the mains voltage and load is maintained constant. However, if either of the two is varied, D.C. voltage received at this point changes. Therefore a regulator is applied at the output stage.

VOLTAGE REGULATOR:

As the name itself implies, it regulates the input applied to it. A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level. In this project, power supply of 5V and 12V are required. In order to obtain these voltage levels, 7805 and 7812 voltage regulators are to be used. The first number 78 represents positive supply and the numbers 05, 12 represent the required output voltage levels.

4.METHODOLOGY

4.METHODOLOGY

The Smart street light control system adopts a dynamic control methodology. According to the proposed plan, initially when it becomes dark, all the street lights automatically glow for a few seconds and switches off. But throughout the night, only one streetlights remains switched on for security concerns. When a vehicle passes by, a block of street lights glows and as the vehicle moves forward, the next block of lights starts glowing where the previous block switches off. In this automatic mode operation the LDR Sensor (Light Dependent Resistor) can be used for measuring light intensity for switched ON or OFF the street light using relays. The main principle of LDR is when the light intensity is low; light is going to be ON otherwise it's going to be OFF. For the efficient reduction of power wastage IR (Infrared) Sensor is integrated. If any vehicle or obstacle is detected using IR sensor at that time it will check the light intensity level using LDR sensor then light will go ON or OFF.

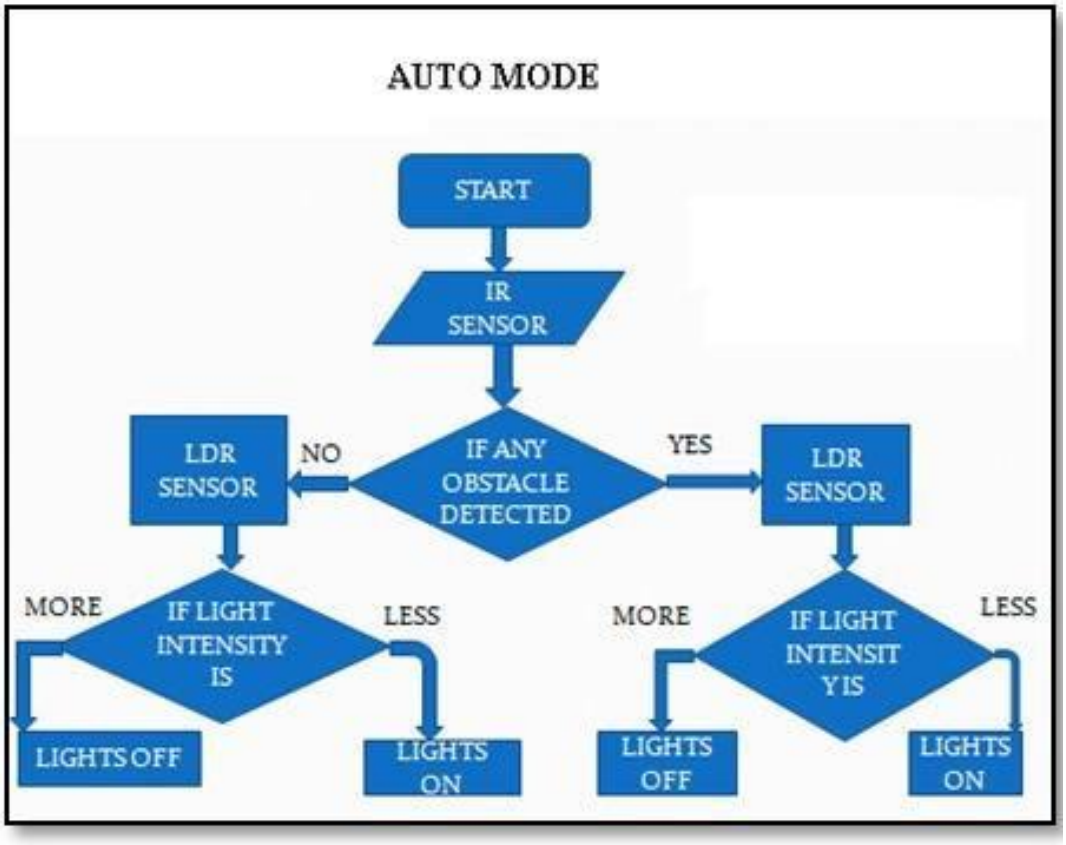

Fig : Flow Chart for Street Light Operation

5.EXISTING SYSTEM AND ITS DISADVANTAGES

5.EXISTING SYSTEM AND ITS DISADVANTAGES

EXISTING SYSTEM:

Industry of street lighting systems are growing rapidly and going to complex with rapid growth of industry and cities. Automation, Power consumption and Cost Effectiveness are the important considerations in the present field of electronics and electrical related technologies. To control and maintain complex street lighting system more economically, various street light control systems are developed. These systems are developed to control and reduce energy consumption of a town's public lighting system using different technologies. The existing work is done using HID lamps. Currently, the HID is used for urban street light based on principle of gas discharge, thus the intensity is not controlled by any voltage reduction method as the discharge path is broken.

HID lamps are a type of electrical gas discharge lamp which produces light by means of an electric arc between tungsten electrodes housed inside a translucent or transparent fused quartz or fused alumina arc tube. This tube is filled with both gas and metal salts. The gas facilitates the arc's initial strike. Once the arc is started, it heats and evaporates the metal salts forming plasma, which greatly increases the intensity of light produced by the arc and reduces its power consumption. High-intensity discharge lamps are a type of arc lamp.

DISADVANTAGES OF EXISTING SYSTEM:

- HID lamps consume more power.
- It cannot be used in all outdoor applications.
- Brightness of the lights in the rear view mirrors which causes a problem for drivers in front of your vehicle.

6.PROPOSED SYSTEM AND ITS ADVANTAGES

6.PROPOSED SYSTEM AND ITS ADVANTAGES

PROPOSED SYSTEM:

Since the HID lamps are not cost effective and not reliable, smart street light system has overcome by replacing the HID lamps with LED. Due to automation, power consumption and cost effectiveness in the present field of electronics and electrical related technologies, industry of street lighting systems are growing rapidly and going to complex with rapid growth of industry and cities. To control and maintain complex street lighting system more economically, various street light control systems are developed. These systems are developed to control and reduce energy consumption of a town's public lighting system using different technologies which uses IR motion sensors to detect the vehicle movement after which the street light begins to glow. As the vehicle moves, the street light that was glowing switches off and the following lights begins to glow.

ADVANTAGES OF PROPOSED SYSTEM:

- More saving of energy used in the street lights.
- Safe road lighting for smooth vehicular movement.
  Intelligent intensity control.
- This idea can be implemented on both small roads and busy highways.
- Security system in applications like homes, hostels, industries ,vehicles.

7.CIRCUIT DIAGRAM

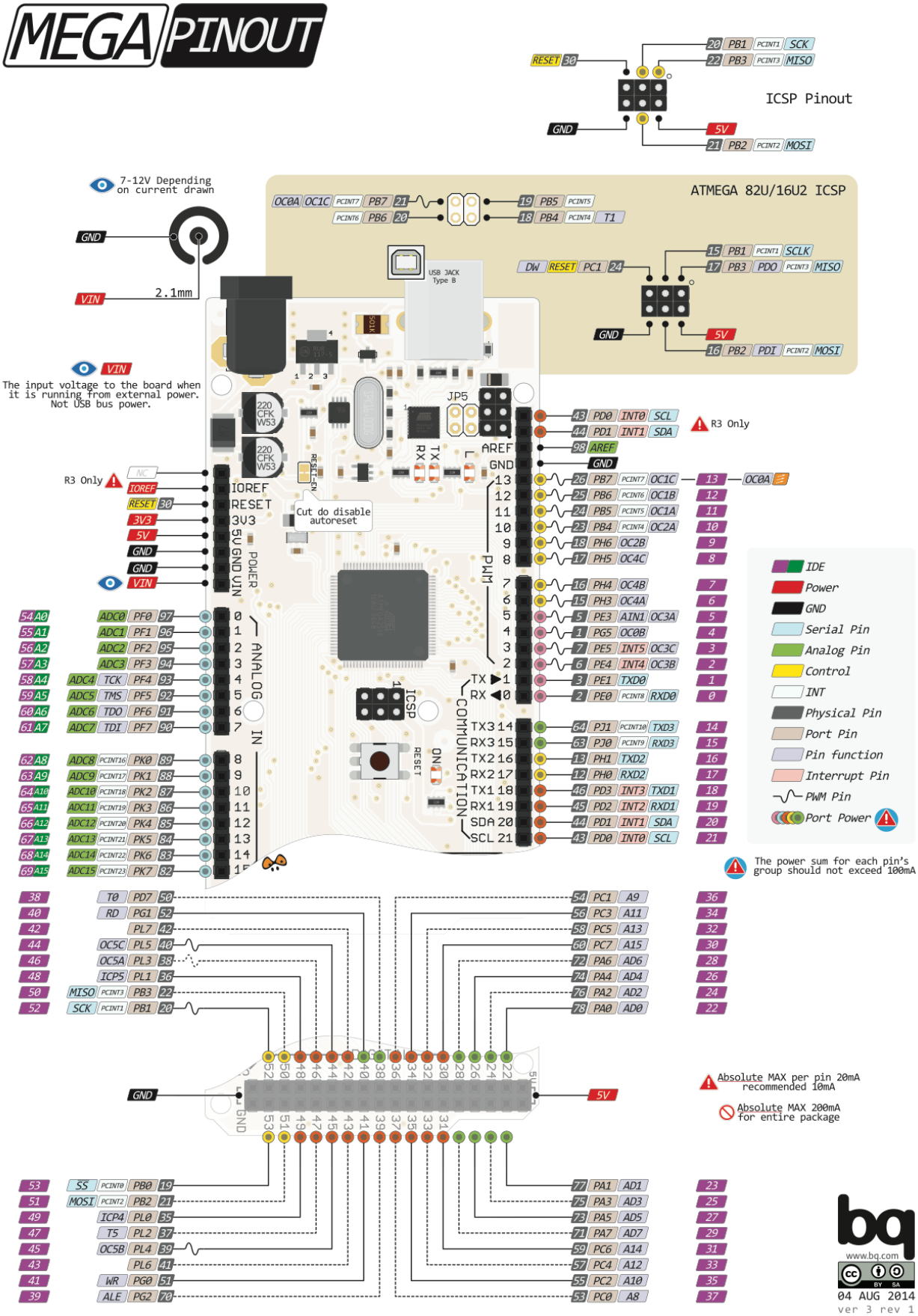

Fig. 7.1: Arduino Mega Circuit Diagram

8.HARDWARE REQUIREMENTS

DHT11 – Temperature and Humidity Sensor

This sensor is used to sense humidity. It facilitates us with analog and digital output. We are using digital output pin to connect it directly with the Arduino to Arduino's digital pin (pin 7).there is a step up register in the sensor to control the power. VCC and GND pins are also connected to Arduino.

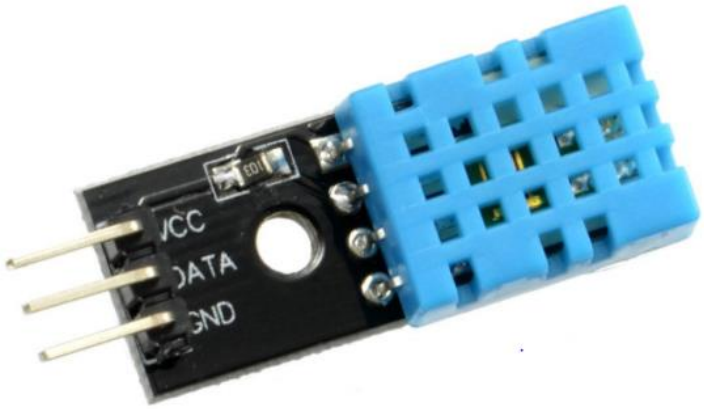

Fig. 8.1: DHT 11 Sensor Diagram

MQ-5 LPG Sensor

This is an ideal sensor which is used for the detection of the harmful LPG leakages at homes or in a work place, storage equipment as well as in vehicles in which the fuel used is LPG. This section is very easy to incorporate into a circuit with alarms, to start an alarm tone or even to show a visual view of the concentration of the gas. This sensor provides a good sensitivity along with with the good response time. Whenever input gas is present, the conductivity of the sensor is higher and the concentration also rises.

MQ-5 SensorMQ-5 gas sensor senses the toxic gases but give accurate values for LPG. It can also sense the natural gas available. It is economical.MQ-5 gas sensor detects Methane, LPG and carbon monoxide gas and sends a signal to the

microcontroller After that micro controller send an active signal to other externally connected devices .

Fig8.2: MQ-5 Sensor Diagram

Arduino Microcontroller:

Arduino is a source with open electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino is capable to sense the environment by getting an input signal from a different sensors and can accordingly control different operations. The Arduino programming language is used to the program the micro controller on the Arduino board and it also uses the Arduino development environment. Here this Arduino gets the sensed information from gas sensor. An Arduino board consists of an At mega micro controller which is of 8-bits along with complementary components in order to perform programming and its incorporation into the other circuits. A boot loader is used to preprogrammed an Arduino micro controller so that the programs to be uploaded on the on-chip flash memory is simplified to a great level in comparison with other devices that uses an external programmer.

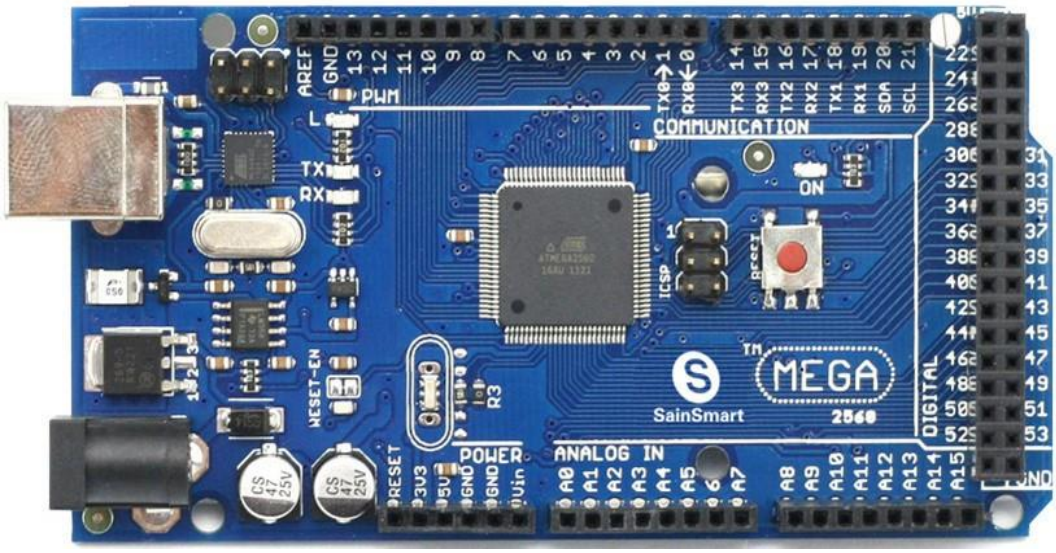

Fig. 8.3: Arduino Microcontroller

LCD Display

The LCD (Liquid Crystal Display) screen is an electronic module which is used for displaying information. It has a large number of applications in day to day life. A 16*2 LCD display which is the common module is extremely used in various electronic circuits and devices. Mostly the LCD modules are preferred in comparison with the seven segments and the different other multi segments. This is because that the LCD is economical and can easily be programmed. It has no bounding of displaying special and custom characters, animations and a lot more. The meaning of 16x2 LCD is that it can display 16 characters per line and there are 2 such lines. Here each character is displayed in 5x7 pixel matrix size. The prototype of leakage system has been tested by sensing a small amount of LPG, Methane and carbon monoxide gas near to the sensor. As a result LCD display will show a alarm for the detected gasses. When reset button is pressed, the system refreshes system regains its initial position itself.

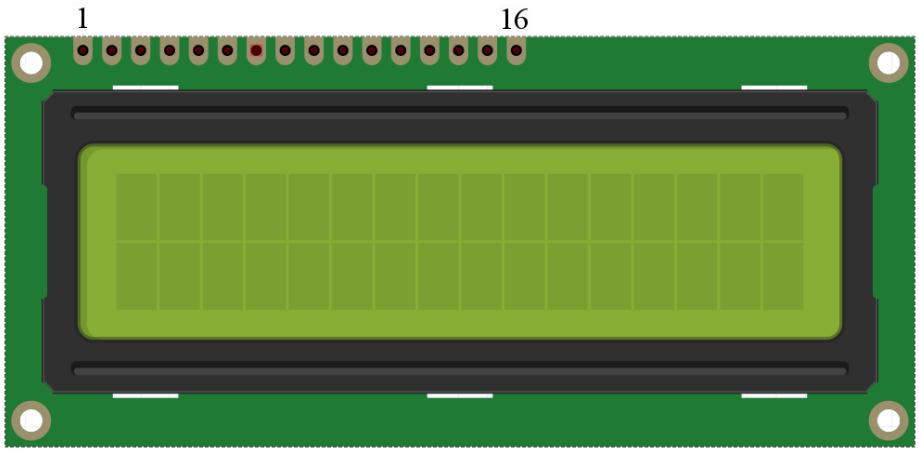

Fig. 8.4: LCD Display

LDR – Light Dependent Resistor

An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits. A light-dependent resistor (LDR) is a light-controlled variable resistor. The resistance of this decreases with increasing incident light intensity; in other words, it exhibits photo-conductivity. An LDR can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits. An LDR is made of a high resistance semiconductor. In the dark, an LDR can have a resistance as high as a few mega ohms (MΩ), while in the light, an LDR can have a resistance as low as a few hundred ohms. If incident light on an LDR exceeds a certain frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electrons (and their whole partners) conduct electricity, thereby lowering resistance. The resistance range and sensitivity of an LDR can substantially differ among dissimilar devices.

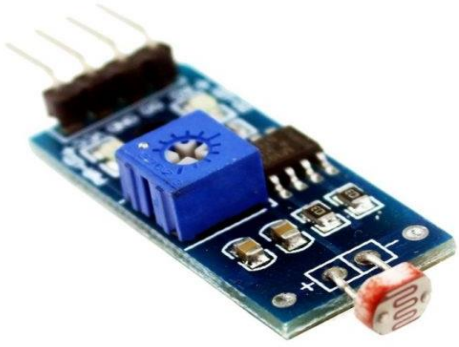

Fig. 8.5: LDR Sensor

9.SOFTWARE DETAILS

9.SOFTWARE DETAILS

[APPLICATION CODE IN ARDUINO LANGUAGE PROGRAMMING]

```
//THIS PROJECT CODE IS FOR SMART CITY
MANAGEMENT
#include <dht.h>
dht DHT;
#define DHT11_PIN 8

#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int LED1 = 9;
int LED2 = A1;
int LED3 = A2;
int LED4 = A3;
int LED5 = A4;
int LED6 = A5;
int LED7 = A6;
int LED8 = A7;
int ldrPin=7;
int buzzer = 10;
int sensor = A8;
int gas_value;
int obstaclePin1 = 14;
int obstaclePin2 = 15;
int obstaclePin3 = 16;
int obstaclePin4 = 17;
int obstaclePin5 = 18;
```

```
int obstaclePin6 = 19;
int obstaclePin7 = 20;
int obstaclePin8 = 21;
int ldrs=LOW;
int hasObstacle1 = HIGH;
int hasObstacle2 = HIGH;
int hasObstacle3 = HIGH;
int hasObstacle4 = HIGH;
int hasObstacle5 = HIGH;
int hasObstacle6 = HIGH;
int hasObstacle7 = HIGH;
int hasObstacle8 = HIGH;
void setup()
{
  lcd.begin(16, 2);
  lcd.setCursor(3,0);
  lcd.print("SMART CITY");
  delay(2000);
  lcd.clear();
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(LED4, OUTPUT);
  pinMode(LED5, OUTPUT);
  pinMode(LED6, OUTPUT);
  pinMode(LED7, OUTPUT);
  pinMode(LED8, OUTPUT);
  pinMode(buzzer,OUTPUT);
  pinMode(ldrPin,INPUT);
  pinMode(sensor,INPUT);
  pinMode(obstaclePin1, INPUT);
  pinMode(obstaclePin2, INPUT);
  pinMode(obstaclePin3, INPUT);
```

```
  pinMode(obstaclePin4, INPUT);
  pinMode(obstaclePin5, INPUT);
  pinMode(obstaclePin6, INPUT);
  pinMode(obstaclePin7, INPUT);
  pinMode(obstaclePin8, INPUT);

  Serial.begin(9600);
}
void loop() {
  hasObstacle1 = digitalRead(obstaclePin1);
  hasObstacle2 = digitalRead(obstaclePin2);
  hasObstacle3 = digitalRead(obstaclePin3);
  hasObstacle4 = digitalRead(obstaclePin4);
  hasObstacle5 = digitalRead(obstaclePin5);
  hasObstacle6 = digitalRead(obstaclePin6);
  hasObstacle7 = digitalRead(obstaclePin7);
  hasObstacle8 = digitalRead(obstaclePin8);
  gas_value=analogRead(sensor);
  lcd.setCursor(0,1);
  lcd.print("Gas: ");
  lcd.print(gas_value);
  if(gas_value >= 650)
  {
    digitalWrite(buzzer,HIGH);
    delay(2000);
  }
  else
  {
    digitalWrite(buzzer,LOW);
  }
  ldrs = digitalRead(ldrPin);
  int chk = DHT.read11(DHT11_PIN);
  lcd.setCursor(0,0);
```

```
 lcd.print("Temp: ");
 lcd.print(DHT.temperature);
 lcd.print("C");
 if (ldrs==HIGH)
 {
 if (hasObstacle1 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  analogWrite(LED1,255);
  delay(1000);//Illuminates the 13th Port LED
 }
 else
 {
  Serial.println("Path is clear");
   analogWrite(LED1,255/5);
 }
  if (hasObstacle2 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED2,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED2,LOW);
 }
if (hasObstacle3 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED3,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
```

```
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED3,LOW);
 }
 if (hasObstacle4 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED4,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED4,LOW);
 }
 if(hasObstacle5 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED5,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED5,LOW);
 }
 if (hasObstacle6 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED6,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
```

```
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED6,LOW);
 }
 if (hasObstacle7 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED7,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED7,LOW);
 }
 if (hasObstacle8 == LOW)
 {
  Serial.println("Stop something is ahead!!");
  digitalWrite(LED8,HIGH);
  delay(1000);//Illuminates the 13th Port LED
 }
 else
 {
  Serial.println("Path is clear");
  digitalWrite(LED8,LOW);
 }
 }
 else{
  analogWrite(LED1,255/5);
  analogWrite(LED2,LOW);
  analogWrite(LED3,LOW);
  analogWrite(LED4,LOW);
```

```
   analogWrite(LED5,LOW);
   digitalWrite(LED6,LOW);
   digitalWrite(LED7,LOW);
   digitalWrite(LED8,LOW);

  }
  if (ldrs==LOW)
  {
   analogWrite(LED1,LOW);
   delay(1000);
  }
  delay(100);
}
```

Library Code

```
//
//    FILE: dht22.cpp
// VERSION: 0.1.00
// PURPOSE: DHT22 Temperature & Humidity Sensor library for Arduino
//
// DATASHEET:
//
// HISTORY:
// 0.1.0 by Rob Tillaart (01/04/2011)
// inspired by DHT11 library
//

#include "dht.h"

#define TIMEOUT 10000

////////////////////////////////////////////////
//
// PUBLIC
//


// return values:
//  0 : OK
// -1 : checksum error
// -2 : timeout
int dht::read11(uint8_t pin)
{
```

```
	// READ VALUES
	int rv = read(pin);
	if (rv != 0) return rv;

	// CONVERT AND STORE
	humidity    = bits[0];  // bit[1] == 0;
	temperature = bits[2];  // bits[3] == 0;

	// TEST CHECKSUM
	uint8_t sum = bits[0] + bits[2]; // bits[1] && bits[3] both 0
	if (bits[4] != sum) return -1;

	return 0;
}

// return values:
//  0 : OK
// -1 : checksum error
// -2 : timeout
int dht::read22(uint8_t pin)
{
	// READ VALUES
	int rv = read(pin);
	if (rv != 0) return rv;

	// CONVERT AND STORE
	humidity    = word(bits[0], bits[1]) * 0.1;

	int sign = 1;
	if (bits[2] & 0x80) // negative temperature
	{
		bits[2] = bits[2] & 0x7F;
		sign = -1;
```

```
        }
        temperature = sign * word(bits[2], bits[3]) * 0.1;


        // TEST CHECKSUM
        uint8_t sum = bits[0] + bits[1] + bits[2] + bits[3];
        if (bits[4] != sum) return -1;

        return 0;
}

////////////////////////////////////////////////
//
// PRIVATE
//

// return values:
//  0 : OK
// -2 : timeout
int dht::read(uint8_t pin)
{
        // INIT BUFFERVAR TO RECEIVE DATA
        uint8_t cnt = 7;
        uint8_t idx = 0;

        // EMPTY BUFFER
        for (int i=0; i< 5; i++) bits[i] = 0;

        // REQUEST SAMPLE
        pinMode(pin, OUTPUT);
        digitalWrite(pin, LOW);
        delay(20);
        digitalWrite(pin, HIGH);
```

```
delayMicroseconds(40);
pinMode(pin, INPUT);

// GET ACKNOWLEDGE or TIMEOUT
unsigned int loopCnt = TIMEOUT;
while(digitalRead(pin) == LOW)
        if (loopCnt-- == 0) return -2;

loopCnt = TIMEOUT;
while(digitalRead(pin) == HIGH)
        if (loopCnt-- == 0) return -2;

// READ THE OUTPUT - 40 BITS => 5 BYTES
for (int i=0; i<40; i++)
{
        loopCnt = TIMEOUT;
        while(digitalRead(pin) == LOW)
                if (loopCnt-- == 0) return -2;

        unsigned long t = micros();

        loopCnt = TIMEOUT;
        while(digitalRead(pin) == HIGH)
                if (loopCnt-- == 0) return -2;

        if ((micros() - t) > 40) bits[idx] |= (1 << cnt);
        if (cnt == 0)   // next byte?
        {
                cnt = 7;
                idx++;
        }
        else cnt--;
}
```

```
        return 0;
}
//
// END OF FILE
//
```

Code for Liquid Crystal

```
#include "LiquidCrystal.h"

#include <stdio.h>
#include <string.h>
#include <inttypes.h>
#include "Arduino.h"

// When the display powers up, it is configured as follows:
//
// 1. Display clear
// 2. Function set:
//    DL = 1; 8-bit interface data
//    N = 0; 1-line display
//    F = 0; 5x8 dot character font
// 3. Display on/off control:
//    D = 0; Display off
//    C = 0; Cursor off
//    B = 0; Blinking off
// 4. Entry mode set:
//    I/D = 1; Increment by 1
//    S = 0; No shift
```

```
//
// Note, however, that resetting the Arduino doesn't reset the LCD, so we
// can't assume that its in that state when a sketch starts (and the
// LiquidCrystal constructor is called).

LiquidCrystal::LiquidCrystal(uint8_t rs, uint8_t rw, uint8_t enable,
                             uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
                             uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7)
{
  init(0, rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7);
}

LiquidCrystal::LiquidCrystal(uint8_t rs, uint8_t enable,
                             uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
                             uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7)
{
  init(0, rs, 255, enable, d0, d1, d2, d3, d4, d5, d6, d7);
}

LiquidCrystal::LiquidCrystal(uint8_t rs, uint8_t rw, uint8_t enable,
                             uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3)
{
  init(1, rs, rw, enable, d0, d1, d2, d3, 0, 0, 0, 0);
}

LiquidCrystal::LiquidCrystal(uint8_t rs,  uint8_t enable,
                             uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3)
{
  init(1, rs, 255, enable, d0, d1, d2, d3, 0, 0, 0, 0);
}

void LiquidCrystal::init(uint8_t fourbitmode, uint8_t rs, uint8_t rw, uint8_t enable,
                         uint8_t d0, uint8_t d1, uint8_t d2, uint8_t d3,
```

```
                        uint8_t d4, uint8_t d5, uint8_t d6, uint8_t d7)
{
  _rs_pin = rs;
  _rw_pin = rw;
  _enable_pin = enable;

  _data_pins[0] = d0;
  _data_pins[1] = d1;
  _data_pins[2] = d2;
  _data_pins[3] = d3;
  _data_pins[4] = d4;
  _data_pins[5] = d5;
  _data_pins[6] = d6;
  _data_pins[7] = d7;

  if (fourbitmode)
    _displayfunction = LCD_4BITMODE | LCD_1LINE | LCD_5x8DOTS;
  else
    _displayfunction = LCD_8BITMODE | LCD_1LINE | LCD_5x8DOTS;

  begin(16, 1);
}

void LiquidCrystal::begin(uint8_t cols, uint8_t lines, uint8_t dotsize) {
  if (lines > 1) {
    _displayfunction |= LCD_2LINE;
  }
  _numlines = lines;

  setRowOffsets(0x00, 0x40, 0x00 + cols, 0x40 + cols);

  // for some 1 line displays you can select a 10 pixel high font
  if ((dotsize != LCD_5x8DOTS) && (lines == 1)) {
```

```
  _displayfunction |= LCD_5x10DOTS;
 }

 pinMode(_rs_pin, OUTPUT);
 // we can save 1 pin by not using RW. Indicate by passing 255 instead of pin#
 if (_rw_pin != 255) {
  pinMode(_rw_pin, OUTPUT);
 }
 pinMode(_enable_pin, OUTPUT);

 // Do these once, instead of every time a character is drawn for speed reasons.
 for (int i=0; i<((_displayfunction & LCD_8BITMODE) ? 8 : 4); ++i)
 {
  pinMode(_data_pins[i], OUTPUT);
  }

 // SEE PAGE 45/46 FOR INITIALIZATION SPECIFICATION!
 // according to datasheet, we need at least 40ms after power rises above 2.7V
 // before sending commands. Arduino can turn on way before 4.5V so we'll wait
50
 delayMicroseconds(50000);
 // Now we pull both RS and R/W low to begin commands
 digitalWrite(_rs_pin, LOW);
 digitalWrite(_enable_pin, LOW);
 if (_rw_pin != 255) {
  digitalWrite(_rw_pin, LOW);
 }

 //put the LCD into 4 bit or 8 bit mode
 if (! (_displayfunction & LCD_8BITMODE)) {
  // this is according to the hitachi HD44780 datasheet
  // figure 24, pg 46
```

```
  // we start in 8bit mode, try to set 4 bit mode
  write4bits(0x03);
  delayMicroseconds(4500); // wait min 4.1ms

  // second try
  write4bits(0x03);
  delayMicroseconds(4500); // wait min 4.1ms

  // third go!
  write4bits(0x03);
  delayMicroseconds(150);

  // finally, set to 4-bit interface
  write4bits(0x02);
 } else {
  // this is according to the hitachi HD44780 datasheet
  // page 45 figure 23

  // Send function set command sequence
  command(LCD_FUNCTIONSET | _displayfunction);
  delayMicroseconds(4500);  // wait more than 4.1ms

  // second try
  command(LCD_FUNCTIONSET | _displayfunction);
  delayMicroseconds(150);

  // third go
  command(LCD_FUNCTIONSET | _displayfunction);
 }

 // finally, set # lines, font size, etc.
 command(LCD_FUNCTIONSET | _displayfunction);
```

```
  // turn the display on with no cursor or blinking default
  _displaycontrol = LCD_DISPLAYON | LCD_CURSOROFF | LCD_BLINKOFF;
  display();

  // clear it off
  clear();

  // Initialize to default text direction (for romance languages)
  _displaymode = LCD_ENTRYLEFT | LCD_ENTRYSHIFTDECREMENT;
  // set the entry mode
  command(LCD_ENTRYMODESET | _displaymode);

}

void LiquidCrystal::setRowOffsets(int row0, int row1, int row2, int row3)
{
  _row_offsets[0] = row0;
  _row_offsets[1] = row1;
  _row_offsets[2] = row2;
  _row_offsets[3] = row3;
}

/********** high level commands, for the user! */
void LiquidCrystal::clear()
{
  command(LCD_CLEARDISPLAY);  // clear display, set cursor position to zero
  delayMicroseconds(2000);  // this command takes a long time!
}

void LiquidCrystal::home()
{
  command(LCD_RETURNHOME);  // set cursor position to zero
  delayMicroseconds(2000);  // this command takes a long time!
```

```
}

void LiquidCrystal::setCursor(uint8_t col, uint8_t row)
{
  const size_t max_lines = sizeof(_row_offsets) / sizeof(*_row_offsets);
  if ( row >= max_lines ) {
    row = max_lines - 1;    // we count rows starting w/0
  }
  if ( row >= _numlines ) {
    row = _numlines - 1;    // we count rows starting w/0
  }

  command(LCD_SETDDRAMADDR | (col + _row_offsets[row]));
}

// Turn the display on/off (quickly)
void LiquidCrystal::noDisplay() {
  _displaycontrol &= ~LCD_DISPLAYON;
  command(LCD_DISPLAYCONTROL | _displaycontrol);
}
void LiquidCrystal::display() {
  _displaycontrol |= LCD_DISPLAYON;
  command(LCD_DISPLAYCONTROL | _displaycontrol);
}

// Turns the underline cursor on/off
void LiquidCrystal::noCursor() {
  _displaycontrol &= ~LCD_CURSORON;
  command(LCD_DISPLAYCONTROL | _displaycontrol);
}
void LiquidCrystal::cursor() {
  _displaycontrol |= LCD_CURSORON;
  command(LCD_DISPLAYCONTROL | _displaycontrol);
```

```
}

// Turn on and off the blinking cursor
void LiquidCrystal::noBlink() {
 _displaycontrol &= ~LCD_BLINKON;
 command(LCD_DISPLAYCONTROL | _displaycontrol);
}
void LiquidCrystal::blink() {
 _displaycontrol |= LCD_BLINKON;
 command(LCD_DISPLAYCONTROL | _displaycontrol);
}

// These commands scroll the display without changing the RAM
void LiquidCrystal::scrollDisplayLeft(void) {
 command(LCD_CURSORSHIFT | LCD_DISPLAYMOVE | LCD_MOVELEFT);
}
void LiquidCrystal::scrollDisplayRight(void) {
 command(LCD_CURSORSHIFT | LCD_DISPLAYMOVE |
LCD_MOVERIGHT);
}

// This is for text that flows Left to Right
void LiquidCrystal::leftToRight(void) {
 _displaymode |= LCD_ENTRYLEFT;
 command(LCD_ENTRYMODESET | _displaymode);
}

// This is for text that flows Right to Left
void LiquidCrystal::rightToLeft(void) {
 _displaymode &= ~LCD_ENTRYLEFT;
 command(LCD_ENTRYMODESET | _displaymode);
}
```

```
// This will 'right justify' text from the cursor
void LiquidCrystal::autoscroll(void) {
  _displaymode |= LCD_ENTRYSHIFTINCREMENT;
  command(LCD_ENTRYMODESET | _displaymode);
}

// This will 'left justify' text from the cursor
void LiquidCrystal::noAutoscroll(void) {
  _displaymode &= ~LCD_ENTRYSHIFTINCREMENT;
  command(LCD_ENTRYMODESET | _displaymode);
}

// Allows us to fill the first 8 CGRAM locations
// with custom characters
void LiquidCrystal::createChar(uint8_t location, uint8_t charmap[]) {
  location &= 0x7; // we only have 8 locations 0-7
  command(LCD_SETCGRAMADDR | (location << 3));
  for (int i=0; i<8; i++) {
    write(charmap[i]);
  }
}

/*********** mid level commands, for sending data/cmds */

inline void LiquidCrystal::command(uint8_t value) {
  send(value, LOW);
}

inline size_t LiquidCrystal::write(uint8_t value) {
  send(value, HIGH);
  return 1; // assume sucess
}
```

```
/************ low level data pushing commands **********/

// write either command or data, with automatic 4/8-bit selection
void LiquidCrystal::send(uint8_t value, uint8_t mode) {
  digitalWrite(_rs_pin, mode);

  // if there is a RW pin indicated, set it low to Write
  if (_rw_pin != 255) {
    digitalWrite(_rw_pin, LOW);
  }

  if (_displayfunction & LCD_8BITMODE) {
    write8bits(value);
  } else {
    write4bits(value>>4);
    write4bits(value);
  }
}

void LiquidCrystal::pulseEnable(void) {
  digitalWrite(_enable_pin, LOW);
  delayMicroseconds(1);
  digitalWrite(_enable_pin, HIGH);
  delayMicroseconds(1);    // enable pulse must be >450ns
  digitalWrite(_enable_pin, LOW);
  delayMicroseconds(100);   // commands need > 37us to settle
}

void LiquidCrystal::write4bits(uint8_t value) {
  for (int i = 0; i < 4; i++) {
    digitalWrite(_data_pins[i], (value >> i) & 0x01);
  }
```

```
  pulseEnable();
}

void LiquidCrystal::write8bits(uint8_t value) {
  for (int i = 0; i < 8; i++) {
    digitalWrite(_data_pins[i], (value >> i) & 0x01);
  }

  pulseEnable();
}
```

10.COMPONENT LIST

Sr. No.	Components	Value	Price
1	Arduino	Mega-2560	**₹ 875.03**
2	LCD	16x2	**₹ 187**
3	Temperature Sensor	DHT-11	**₹ 249**
4	Gas Sensor	MQ-5	**₹ 299**
5	IR Sensor	IR Sensor	**₹ 189**
6	LED	5mm LED	**₹ 169**
7	Solar Panel	3V	**₹ 160**
8	T1	BC547	**₹ 52**
9	T2	BC557	**₹ 20**
10	POT	10Kohm	**₹ 20**
11	R1	1Kohm	**₹ 20**
12	Connector	-	**₹ 20**
13	IC Socket	28 Pin	**₹ 120**
14	Wire	20 Meter	**₹ 40**
15	Copper Wire	10 Meter	**₹ 500**
16	Power Supply	5Volt	-
17	Buzzer	5Volt	**₹ 60**

11.FUTURE SCOPE

11.FUTURE SCOPE

Once this Smart City System is implemented, we could directly go for Wireless Power Transmission which would further reduce the maintenance costs and power thefts of the system, as cable breaking is one of the problems faced today. In addition to this, controlling the Traffic Signal lights is another feature that we could look into after successful implementation of our system. Depending on the amount of traffic in a particular direction, necessary controlling actions could be taken. Also emergency vehicles and VIP convoys can be passed efficiently. Moreover, attempts can be made to ensure that the complete system is self- sufficient on nonconventional energy resources like power, windmills, Piezo - electric crystals, etc. If the system has traffic speed sensors then this information could be used to manage traffic speed via dimming of the street lights. If the average traffic speed is too fast during evening and night hours, this could be used trigger slight dimming of the street lights. The level of dimming would be imperceptible to motorists but they would slow down, in response to the slightly diminished lighting. We hope that these advancements can make this system completely robust and totally reliable in all aspects.

12.CONCLUSION

Conclusion

This Arduino based project will provide a competent method for smart city in we are mainly focusing on road lighting systems and make the whole process of energy saving easier and efficient. With the capability to change the amount of light emitted depending upon the outside condition is no doubt an innovation with many future applications and apart from the fact that it can also be used in many present day tech such as head lights, park lights, industrial lights and many more. The usage of the smart lighting system will undoubtedly change the world that we see today. We are also providing two more functionalities that are gas sensing and temperature sensing. In gas sensing part we are monitoring atmosphere for gas leakage and poisonous gases. If sensor detects gas leakage or poisonous gases then it will give alarm.

In case of temperature monitoring system we are monitoring temperature in real time and displaying on LCD and for citizens if some where fire takes place then it will indicate people and tell them the corrective action to be take so that accident can be minimized.

As we mentioned earlier about power supplying we have also installed solar panel so that we can have optimum utilization of solar energy which can be used to power societies.

13.REFRENCES

13.REFRENCES

. [1] S. Suganya, R. Sinduja, T. S

. owmiya& S. Senthilkumar, Street light glow on detecting vehicle movement using sensor

. [2]. K.Santha Sheela,S.Padmadevi, Survey on Street Lighting System Based On Vehicle Movements

. [3] M.Abhishek, Syed ajram shah, K.Chetan, K,Arun Kumar, Design and implementation of traffic flow based street light control system with effective utilization of solar energy, International journal of Science Engineering and Advance Technology, IJSEAT, Vol 3, Issue 9, September -2015

. [4] "Wireless internet lighting control system",Budike, E.S. Lothar (Power web Technologies), US patent 7,167,777, Jan 23, 2007.

[*] Chung, H.S.H., Ho, N.M., Hui, S.Y.R. and Mai, W.Z. (2005), "Case study of a highly- reliable dimmable road lighting system with intelligent remote control", paper presented at European Conference on Power Electronics and Applications, Dresden.

[**] Costa, M.A.D., Costa, G.H., dos Santos, A.S., Schaech, L. and Pin heiro, J.R. (2009), "A high efficiency autonomous street lighting system based on solar energy and LEDs", Brazilian Power Electronics Conference (COBEP 2009), Bonito, 27 September-1 October, pp. 265-73.

ACTUAL PROJECT PICTURES