Microcontroller Based Light Control System

Project Report submitted in partial
fulfillment of the requirement for the degree
of

Bachelor of Technology.

in

Computer Science & Engineering

under the Supervision of

Miss Reema Aswani

By

Deep Agarwal-111222

to

Jaypee University of Information and
Technology

Waknaghat, Solan – 173234, Himachal
Pradesh

Certificate

This is to certify that project report entitled “Microcontroller Based Light Control System ”, submitted by Deep Agarwal in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Miss Reema Aswani

Assistant Professor

Signature

Date:

Acknowledgement

There are many people who are associated with this project directly or indirectly whose help and timely suggestions are highly appreciable for completion of this project. First of all, I would like to thank Prof. Dr. SP Ghrera, Head, Department of Computer Science Engineering for his kind support and constant encouragements, valuable discussions which is highly commendable.

I would like to express my sincere gratitude to my supervisor Miss Reema Aswani, for her super vision, encouragement, and support which has been instrumental for the success of this project. It was an invaluable experience for me to be one of his students. Because of him, I have gained a careful research attitude.

Thanks to those who are also the part of this project whose names could have not been mentioned here.

Date:

Deep Agarwal
Roll No. 111222

Abstract

This project aim to develop a light control system in which In any conference/seminar hall, whenever a speaker wants to switch over from projected presentation to white maker board due to some audience query, someone has to physically monitor the speaker activity and then control the lights accordingly. This project tries to automate this activity by using Projector Light Controller Module which consists of continuous rotating sensors that senses the presence of the speaker near the white marker board and send this information via RF wireless communication to the light control system for decision making to automatically turn ON the light. As the speaker moves away from the white marker board, lights will be automatically turned off. The project uses an Arduino, and outlines how to use motion sensors to provide the input. This is about as simple as Arduino projects get, and will provide a great introduction to the fundamentals of working with Arduino: wiring input devices, locating pins, and set up and simple coding in the Arduino development environment.

Table of Content

S No.	Title	Page No.
1.	**Inroduction**	
	1.1 About microcontrollers	1
	1.2 Motivation and Background	2
2.	**Literature Review**	
	2.1Application of the light control system	3
	2.2Scope of the project	3
	Requirements for the project	
	2.3.Arduino	4
	2.3.1 Arduino Board	5
	2.3.2 Arduino Software	9
	2.4 LDR sensor	
	2.4.1 Working Principle of the LDR sensor	16
	2.4.2 Characteristics of the LDR sensor	16
	2.4.2 Implementation of a circuit using LDR sensor	21
	2.5 Relay Circuit	24
3.	**Implentation**	
	3.1)Implementation of circuit on Board	26
	3.2)Coding in the software	27
	3.3)Uploading of code	28
4.	**Comparative Analysis**	
	4.1 Comparison of different boards	29
	4.2 Advantages and disadvantages of this system	34
5.	**Proposal Analysis**	35
6..	**Conclusion and future Scope**	41
7.	**Referances**	43

S.No.	Title	Page No.
1.	Arduino Board.	6
2.	Arduino IDE.	12
3.	Screenshot for the LED blink program.	13
4.	Circuit Diagram for LED blink program.	14
5.	Actual circuit for LED blink progam.	15
6.	LDR sensor.	16
7.	Screenshot showing reading of LDR sensor.	19
8.	Circuit Diagram for connecting sensor to Arduino Board.	20
9.	LDR sensor connected with Arduino Board.	20
10.	Circuit showing when there is sufficient light LEDs do not blink.	22
11.	Circuit showing when there is shadow on the sensor LEDs blink.	23
12.	A Relay.	24
13.	Relay Circuit.	25
14.	Circuit Diagram for connecting sensor to Arduino Board.	26
15.	Circuit showing when there is a shadow over the sensor bulb glows.	27
16.	Circuit showing when there is sufficient light the bulb does not glow.	27

Chapter 1

INTRODUCTION

1.1About Microcontrollers

A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP RAM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microproccesors used in personal computers or other general purpose applications. Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption (single-digit milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a Digital Signal Processors (DSP), with higher clock speeds and power consumption.

1.2 Motivation and Background

A lighting control system is an intelligent network based lighting control solution that incorporates communication between various system inputs and outputs related to lighting control with the use of one or more central computing devices. Lighting control systems are widely used on both indoor and outdoor lighting of commercial, industrial, and residential spaces. Lighting control systems serve to provide the right amount of light where and when it is needed.

Light Control System is a simple and powerful concept, which uses microcontroller as a main element to switch ON and OFF the light automatically. By using this system manual works are removed. It automatically switches ON lights when person or the speaker approaches the sensor. It automatically switches OFF lights when speaker star to go away from this system. This is done by a sensor called Light Dependant Resistor (LDR) which senses the light actually like our eyes. This system needs no manual operation of switching ON and OFF. The system itself detects whether there is need for light or not. When darkness rises to a certain value due to presence of someone then automatically light is switched ON and when there is other source of light, the light gets OFF. The extent of darkness at which the light to be switched on can also be tailored using the potentiometer provided in the circuit. Moreover, the circuit is carefully designed to avoid common problems like overload, relay chattering and inductive kick back in relay. The main advantages of this system consist in the reduction of the costs related to energy consumption. A lighting control system is an intelligent network based lighting control solution that incorporates communication between various system inputs and outputs related to lighting control with the use of one or more central computing devices. Lighting control systems are widely used on both indoor and outdoor lighting of commercial, industrial, and residential spaces. Lighting control systems serve to provide the right amount of light where and when it is needed.

Lighting control systems are employed to maximize the energy saving from the lighting system, satisfy building codes for comply with green building and energy conservation programs. Lighting control systems are often referred to under the term smart lighting.

Chapter 2
Literature Review

2.1 Applications of the light control system

The light control system can be used in various application such as applying in the street light control system where with the help of sensor presence of any person can be detected and by detecting the presence the street light can be turned on. The sensor used can be applied at every street light so as if someone passes the light every street light can be turned on and can turn off as the passerby goes away from the street light. This technique can be used to save a lot of energy as the light only gets on only when required and automatically turns off whenever it is not required.

Another application of this system is that it can be used in the bookshelves in the library where a sensor is placed above the book shelves and when any person comes to search any book then the light will turn on by detecting the presence of the person near the sensor and when the person leaves the shell the light will turn off. So again in this case it is used as a energy saver because the lights gets turn on only when it is required and no additional light is required to be installed near the bookshelf in the library.

2.2 Scope of the project

Microcontroller based light control system is a very useful technique for saving the energy in the form of light. This can be used in a lecture theatre where a speaker giving presentation through a projector so lights are needed to be turned off when he is giving presentation to make the visibility on the screen better but if in case he is to explain something by writing on the board he/she will have to turn on the lights to make the visibility on the board better. So this project mainly reduces the effort of the speaker to turn on and off the lights whenever he needed. This project uses a LDR sensor which is somehow fitted over the top of the white board which will detect the presence of the speaker and some coding is done in the arduino software which is given as a input to the Arduino board through which the light gets turn on or off automatically. This can also be used as a energy saver of light as the light gets on only when it is needed.

Requirement of the project

Here is a brief description of the components (hardware or software used in this project)

2.3 Arduino

Arduino is a family of single-board microcontroller intended to make it easier to build interactive objects or environments. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. Current models feature a USB interface together with six analog input pins and 14 digital I/O pins that can accommodate various extension boards.

The first Arduino was introduced in 2005. Its designers sought to provide an inexpensive and easy way for hobbyists, students, and professionals to create devices that interact with their environment using sensors and actuators. Common examples for beginner hobbyists include simple robots, thermostats and motion detectors. Arduino boards come with a simple integrated development environment (IDE) that runs on regular personal computers and allows users to write programs for Arduino using C or C++.

Arduino boards can be purchased assembled or as do-it-yourself kits. Hardware design information is available for those who would like to assemble an Arduino by hand. It was estimated in mid-2011 that over 300,000 official Arduinos had been commercially produced, and in 2013 that 700,000 official boards were in users' hands.

Arduino is open source hardware: the Arduino hardware reference designs are distributed under a Creative Common Attribution Share-Alike 2.5 license and are available on the Arduino Web site. Layout and production files for some versions of the Arduino hardware are also available. The source code for the IDE is available and released under the GNU General Public License, version 2.

Although the hardware and software designs are freely available under copyleft licenses, the developers have requested that the name "Arduino" be exclusive to the official product and not be used for derivative works without permission. The official policy document on the use of the Arduino name emphasizes that the project is open to incorporating work by others into the official product. Several Arduino-compatible products commercially released have avoided the "Arduino" name by using "-duino" name variants.

To start working with an arduino we require

2.3.1 Arduino Board

An Arduino board consists of an Atmel 8-bit AVR microcontroller with complementary components that facilitate programming and incorporation into other circuits. An important aspect of the Arduino is its standard connectors, which lets users connect the CPU board to a variety of interchangeable add-on modules known as *shields*. Some shields communicate with the Arduino board directly over various pins, but many shields are individually addressable via an I^2C serial bus—so many shields can be stacked and used in parallel. Official Arduinos have used the mega AVR series of chips, specifically the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560. A handful of other processors have been used by Arduino compatibles. Most boards include a 5 volt linear regulator and a 16 MHz oscillator (or ceramic resonator in some variants), although some designs such as the LilyPad run at 8 MHz and dispense with the on board voltage regulator due to specific form-factor restrictions. An Arduino microcontroller is also pre-programmed with a boot loader that simplifies uploading of programs to the on-chip flash memory, compared with other devices that typically need an external programmer. This makes using an Arduino more straightforward by allowing the use of an ordinary computer as the programmer.

At a conceptual level, when using the Arduino software stack, all boards are programmed over an RS-232 serial connection, but the way this is implemented varies by hardware version. Serial Arduino boards contain a level shifter circuit to convert between RS-232-level and TTL-level signals. Current Arduino boards are programmed via USB, implemented using USB-to-serial adapter chips such as the FTDIFT232. Some variants, such as the Arduino Mini and the unofficial Boarduino, use a detachable USB-to-serial adapter board or cable, Bluetooth or other methods. (When used with traditional microcontroller tools instead of the Arduino IDE, standard AVR ISP programming is used.)

The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The Diecimila, Duemilanove, and current Uno provide 14 digital I/O pins, six of which can produce pulse-width modulated signals, and six analog inputs, which can also be used as six digital I/O pins. These pins are on the top of the board, via female 0.10-inch (2.5 mm) headers. Several plug-in application shields are also commercially available. The Arduino

Nano, and Arduino-compatible Bare Bones Board and Boarduino boards may provide male header pins on the underside of the board that can plug into solderless breadboard.

There are many Arduino-compatible and Arduino-derived boards. Some are functionally equivalent to an Arduino and can be used interchangeably. Many enhance the basic Arduino by adding output drivers, often for use in school-level education to simplify the construction of buggies and small robots. Others are electrically equivalent but change the form factor—sometimes retaining compatibility with shields, sometimes not. Some variants use completely different processors, with varying levels of compatibility.

Below is an image of an Arduino board

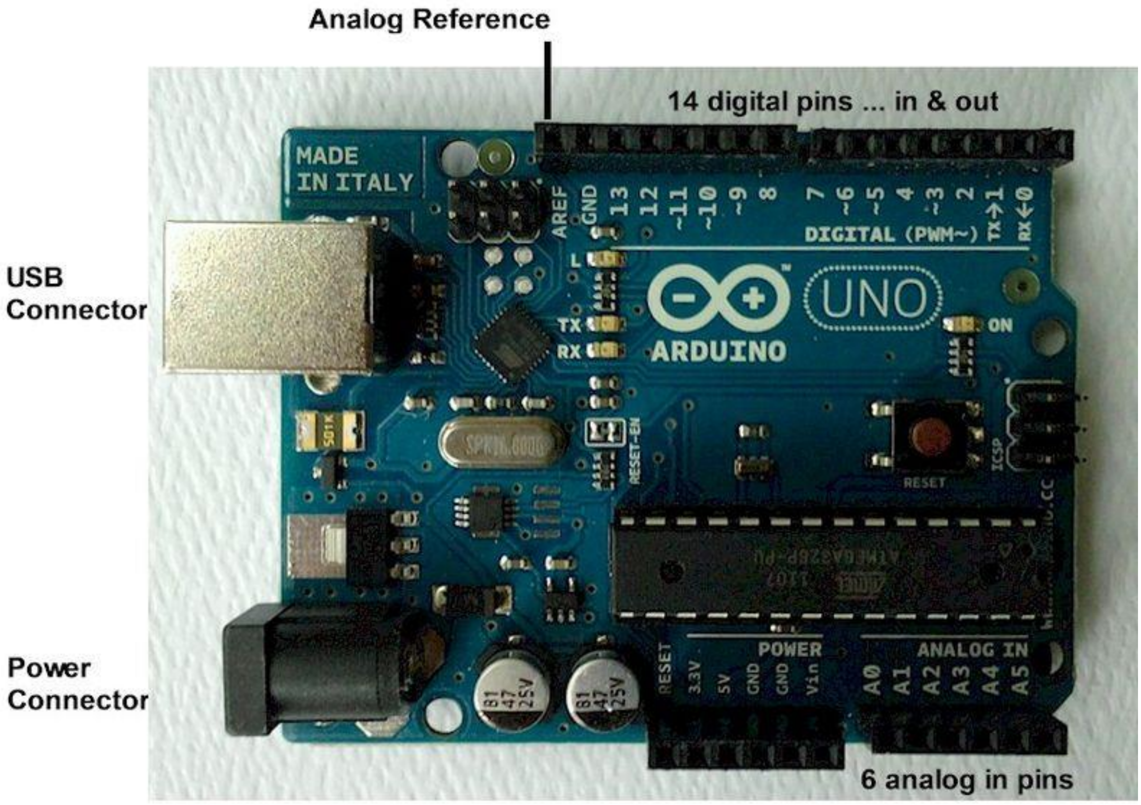


Figure 2.1 Arduino Board.

There are many varieties of Arduino boards that can be used for different purposes. Some boards look a bit different from the one below, but most Arduinos have the majority of these components in common:

Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply (like this) that is terminated in a barrel jack. In the picture above the USB connection is labeled **(1)** and the barrel jack is labeled **(2)**.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in our Installing and Programming Arduino tutorial.

Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjuction with abreadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labeled on the board and used for different functions.

- **GND** : Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
- **5V & 3.3V** : As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
- **Analog** : The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
- **Digital** : Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).

- **PWM** : You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). We have a tutorial on PWM, but for now, think of these pins as being able to simulate analog output (like fading an LED in and out).
- **AREF**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Reset Button

Just like the original Nintendo, the Arduino has a reset button. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times. Unlike the original Nintendo however, blowing on the Arduino doesn't usually fix any problems.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON'. This LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs . These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit . Think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

Voltage Regulator

The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

2.3.2 Arduino Software

The Arduino integrated develop environment (IDE) is a cross-platform application written in Java, and derives from the IDE for the Processing programming language and the wiring projects. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click. A program or code written for Arduino is called a *sketch*

Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. Users only need define two functions to make a runnable cyclic executive program:

- setup():a function run once at the start of a program that can initialize settings
- loop(): a function called repeatedly until the board powers off

Installing the Software

If you have the access to the internet, there are step-by-step directions and the software available at http://arduino.cc/en/Main/Software.Otherwise, the USB stick in your kit has the software under the software directory. There are two directories under that. One is “windows”and the other is “Mac OS X”. If you are installing into Linux, you will need to follow the direction at http://arduino.cc/en/Main /Software.

Windows Installation

1. Plug in your board via USB and wait for Windows to begin its driver installation process. After a few moments, the process will fail.
2. Click on the Start Menu, and open up the Control Panel.
3. While in the control panel ,navigate to System and Security .Next, click on System. Once the system window is up, open the Device Manager.
4. Look under Ports (COM & LPT).You should see an open port named “Arduino UNO(COMxx)”.
5. Right click on the “Arduino UNO (COMxx)”port and choose the “Update Driver Software” option.
6. Next, choose the “Browse my computer for Driver software” option.
7. Finally, navigate to and select the Uno’s driver file, named “ArduinoUNO.inf”, located in the “Driver” folder of the Arduino Software download .
8. Windows will finish up the driver installation from there.
9. Double-click the Arduino application.
10. Open the LED blink example sketch:File>Examples>Basics>Blink.
11. Select Arduino under the Tools > Board menu.
12. Select your serial port (if you do not know which one, disconnect the UNO and the entry that disappears is the right one.)
13. Click the Upload button.
14. After the message “Done uploading ”appears, you should see the “L” LED blinking once a second.(The “L” LED is on the Arduino directly behind the USB port.)

Mac Installation

1. Connect the board via USB.
2. Drag the Arduino application onto your hard drive.
3. When network Preferences comes up, just click “Apply”(remember the /dev/tty/usb.)
4. Start the program.
5. Open the LED blink example:File >Example >1.Basic >Blink
6. Select Arduino Uno under the Tools > Board menu.
7. Select your Serial port(if you do not know which one, disconnect the UNO and the entry that disappears is the right one.)
8. Click the upload button.
9. After the message “done uploading” appears, you should see the “L”LED blinking once a second .(The “L” LED is on the Arduino directly behind the USB connection)

The Integrated Development Environment(IDE)

You use the Arduino IDE on your computer to create, open, and change sketches (Arduino calls programs (sketches). We will use the two words interchangeably in this book.). Sketches define what the board will do. You can either use the buttons along the top of the IDE or the menu items.

The following is an image of the Arduino Integrated Development Environment.

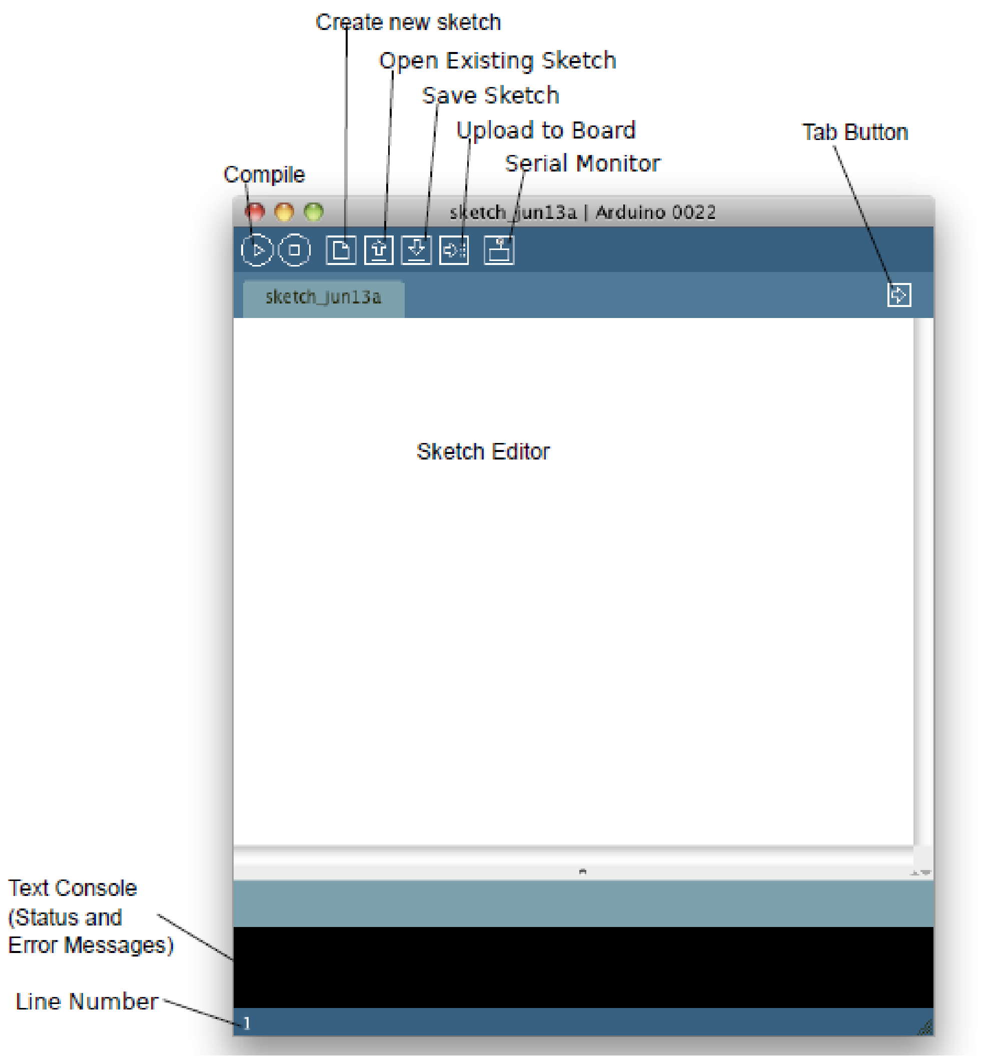


Figure 2.2: Arduino Integrated Development Environment

Parts of the IDE: (from left to right, top to bottom)

1. **Compile** - Before your program "code" can be sent to the board, it needs to be converted into instructions that the board understands. This process is called *compiling.*
2. **Stop** - This stops the compilation process.
3. **Create new Sketch** - This opens a new window to create a new sketch.
4. **Open Existing Sketch** - This loads a sketch from a on your computer.
5. **Save Sketch** - This saves the changes to the sketch you are working on.
6. **Upload to Board** - This compiles and then transmits over the USB cable to your board.

7. **Tab Button** - This lets you create multiple files in your sketch. This is for more advanced programming than we will do in this class.
8. **Sketch Editor** - This is where you write or edit sketches.
9. **Text Console** - This shows you what the IDE is currently doing and is also where error messages display if you make a mistake in typing your program. (often called a syntax error) .
10. **Line Number** - This shows you what line number your cursor is on. It is useful since the compiler gives error messages with a line number.

A typical first program for a microcontroller simply blinks an LED on and off.

Here is the screen shot of the program to blink the LED.

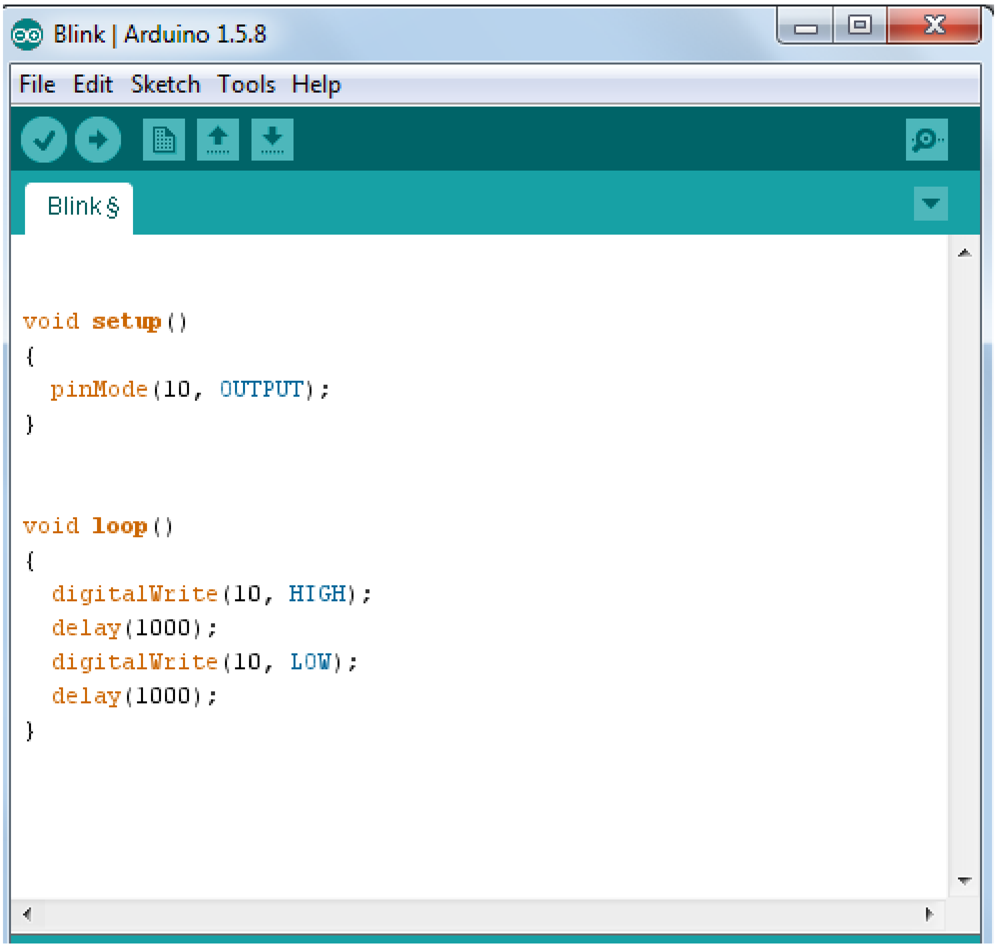


Figure 2.3:Screenshot for the LED blink program

This is a simple and a basic program of Arduino in which a LED is turn ON and OFF after every one second because whatever value is written inside the delay function it is in milliseconds. And the positive end of the LED is connected to the pin 13 of the Arduino board and negative end is connected to the ground pin of the Arduino board.

The circuit diagram for the above code is

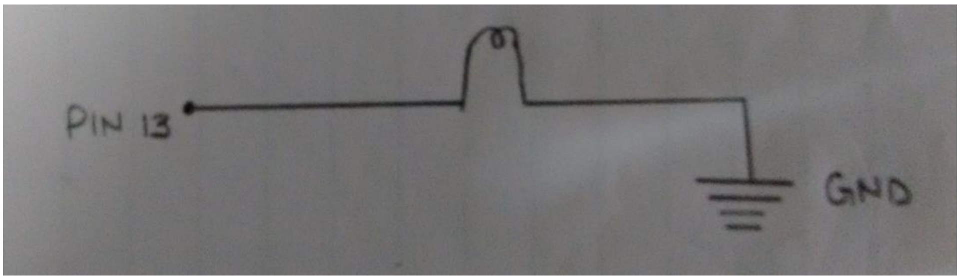


Figure 2.4:Circuit Diagram for LED blink program

The diagram for the above code is

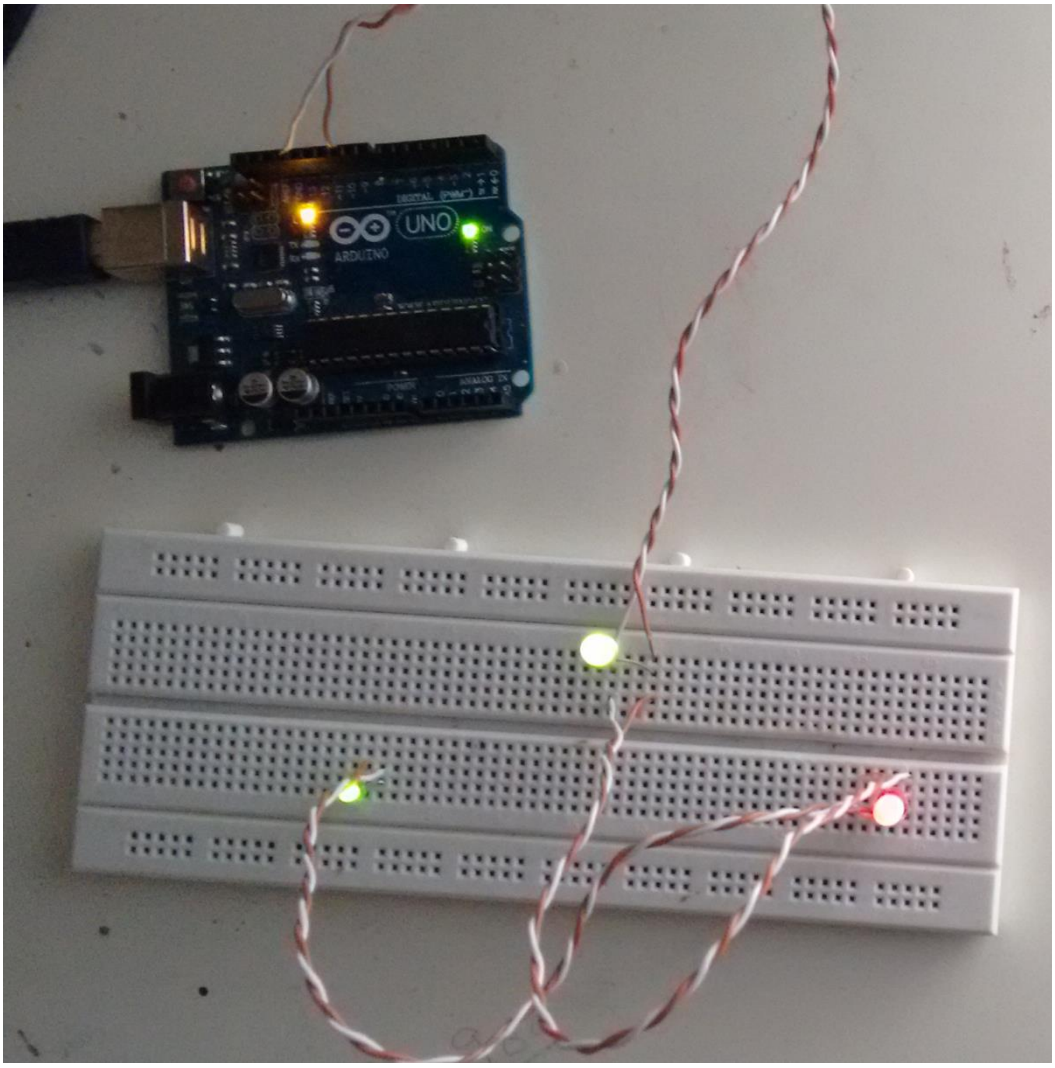


Figure 2.5: Actual circuit for LED blink progam.

2.4 LDR sensor

A LDR or light-dependent resistor is a light-controlled variable resistor. The resistance of a LDR decreases with increasing incident light intensity; in other words, it exhibits photoconductivity. A LDR can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits.

A Light Dependent Resistor (LDR) or a photo resistor is a device whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light sensitive devices. They are also called as photo conductors, photo conductive cells or simply photocells. They are made up of semiconductor materials having high resistance. There are many different symbols used to indicate a LDR, one of the most commonly used symbol is shown in the figure below. The arrow indicates light falling on it.

A LDR is made of a high resistance semiconductor. In the dark, a LDR can have a resistance as high as a few megaohms (MΩ), while in the light, a LDR can have a resistance as low as a few hundred ohms. If incident light on a 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 hole partners) conduct electricity, thereby lowering resistance. The resistance range and sensitivity of a LDR can substantially differ among dissimilar devices. Moreover, unique LDRs may react substantially differently to photons within certain wavelength bands.

A photoelectric device can be either intrinsic or extrinsic. An intrinsic semiconductor has its own charge carriers and is not an efficient semiconductor, for example, silicon. In intrinsic devices the only available electrons are in the valence band, and hence the photon must have enough energy to excite the electron across the entire bandgap. Extrinsic devices have impurities, also called dopants, added whose ground state energy is closer to the conduction band; since the electrons do not have as far to jump, lower energy photons (that is, longer wavelengths and lower frequencies) are sufficient to trigger the device. If a sample of silicon has some of its atoms replaced by phosphorus atoms (impurities), there will be extra electrons available for conduction..

Here is a image for the LDR sensor

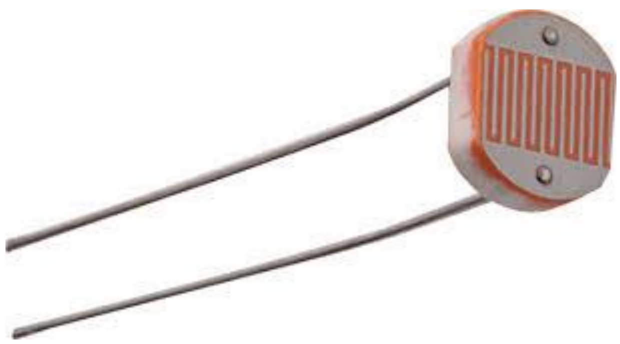


Figure 2.6: LDR sensor.

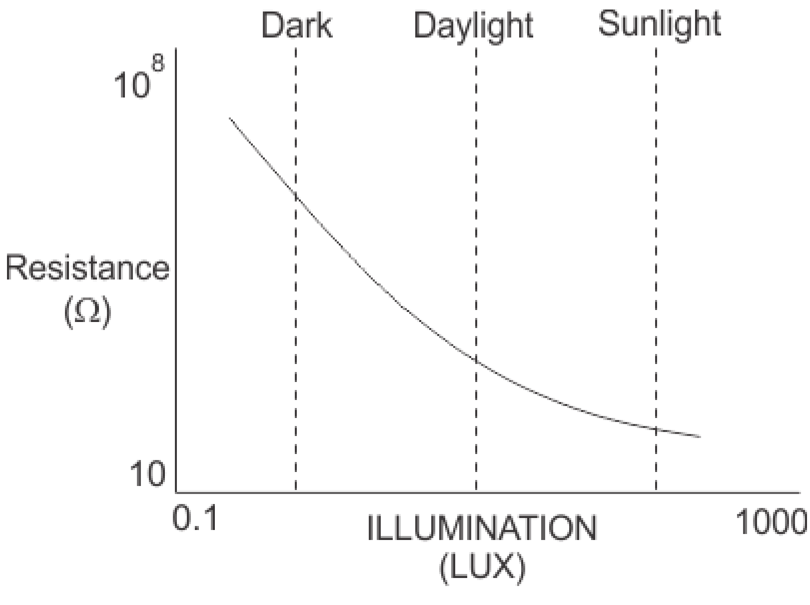
2.4.1 Working Principle of the LDR sensor

A light dependent resistor works on the principle of photo conductivity. Photo conductivity is an optical phenomenon in which the materials conductivity (Hence resistivity) reduces when light is absorbed by the material.

When light falls i.e. when the photons fall on the device, the electrons in the valence band of the semiconductor material are excited to the conduction band. These photons in the incident light should have energy greater than the band gap of the semiconductor material to make the electrons jump from the valence band to the conduction band. Hence when light having enough energy is incident on the device more & more electrons are excited to the conduction band which results in large number of charge carriers. The result of this process is more and more current starts flowing and hence it is said that the resistance of the device has decreased. This is the most common working principle of LDR.

2.4.2 Characteristics of the LDR sensor

LDR's are light dependent devices whose resistance decreases when light falls on them and increases in the dark. When a light dependent resistor is kept in dark, its resistance is very high. This resistance is called as dark resistance. It can be as high as 1012 Ω. And if the device is allowed to absorb light its resistance will decrease drastically. If a constant voltage is applied to it and intensity of light is increased the current starts increasing. Figure below shows resistance vs. illumination curve for a particular LDR.



Photocells or LDR’s are non linear devices. There sensitivity varies with the wavelength of light incident on them. Some photocells might not at all response to a certain range of wavelengths. Based on the material used different cells have different spectral response curves.

When light is incident on a photocell it usually takes about 8 to 12ms for the change in resistance to take place, while it takes seconds for the resistance to rise back again to its initial value after removal of light. This phenomenon is called as resistance recovery rate. This property is used in audio compressors. Also, LDR’s are less sensitive than photo diodes and photo transistor. (A photo diode and a photocell (LDR) are not the same, a photo-diode is a p-n junction semiconductor device that converts light to electricity, whereas a photocell is a passive device, there is no p-n junction in this nor it “converts” light to electricity).

Types of Light Dependent Resistors:

Based on the materials used they are classified as: i) Intrinsic photo resistors (Un doped semiconductor): These are pure semiconductor materials such as silicon or germanium. Electrons get excited from valance band to conduction band when photons of enough energy falls on it and number charge carriers increases.

ii) Extrinsic photo resistors: These are semiconductor materials doped with impurities which are called as dopants. Theses dopants create new energy bands above the valence band which are filled with electrons. Hence this reduces the band gap and less energy is required in exciting them. Extrinsic photo resistors are generally used for long wavelengths.

The value of LDR in the Arduino Integrated development environment can be known by the code given below:

```
int LDR = A0;
int val = 0 ;
void setup ()
{
Serial.begin(9600);
}

void loop()
```

```
{
val = analogRead(LDR);
  Serial.print("value is ");
  Serial.println(val);
 delay(50);
}
```

When we will run the above code in the Arduino Integrated Environment we will get the readings by clicking on the serial monitor screen which is as

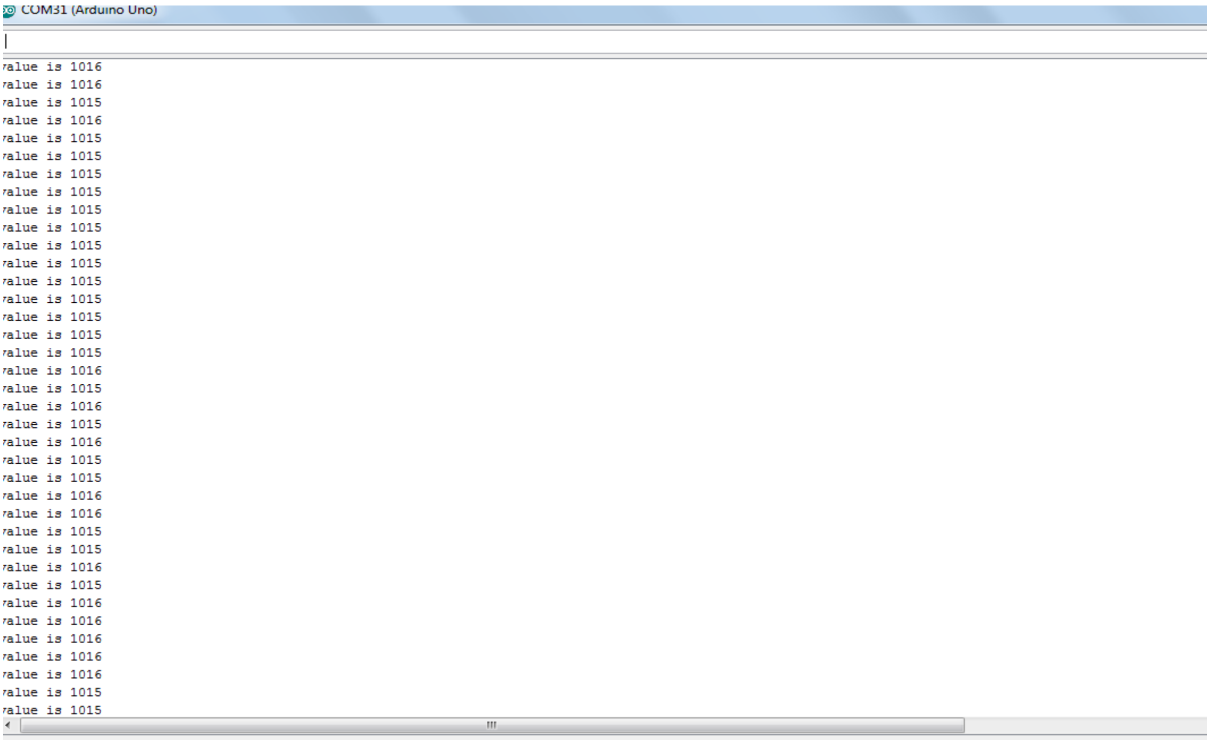


Figure 2.7: Screenshot showing reading of LDR sensor.

As we can see that the LDR value is approximately 1015 in every reading but if there is darkness around the sensor the value will reduce.

The circuit diagram for reading the LDR values is as under.

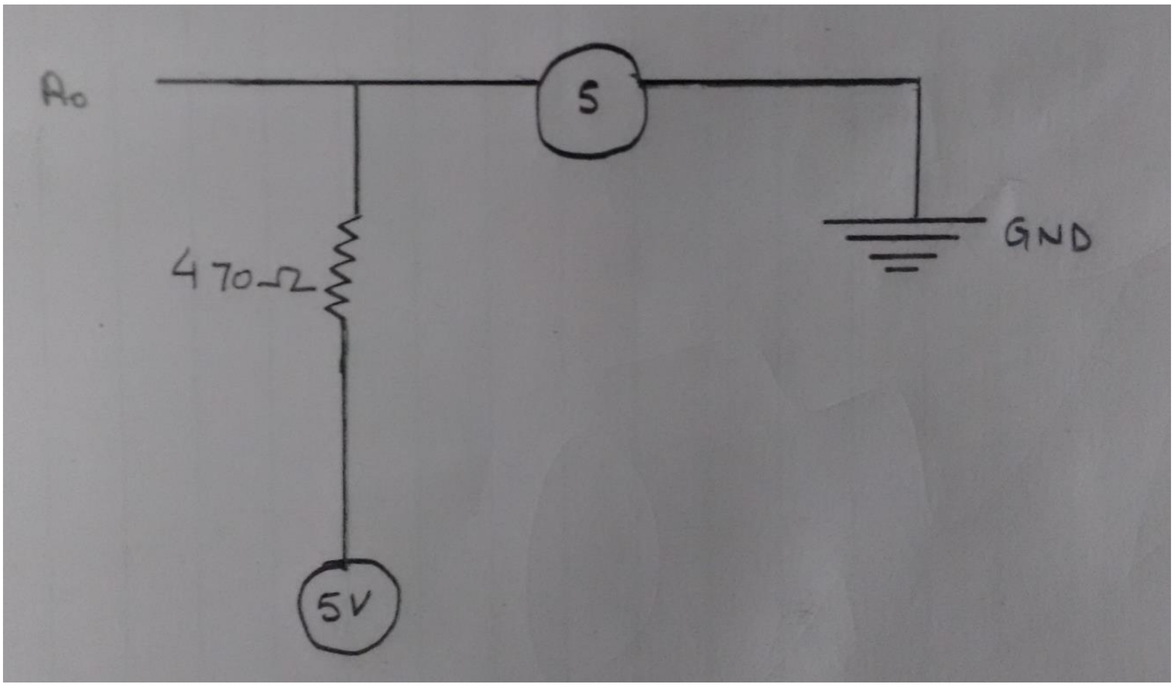


Figure 2.8: Circuit Diagram for connecting sensor to Arduino Board.

The actual implemented cicuit diagram is as under.

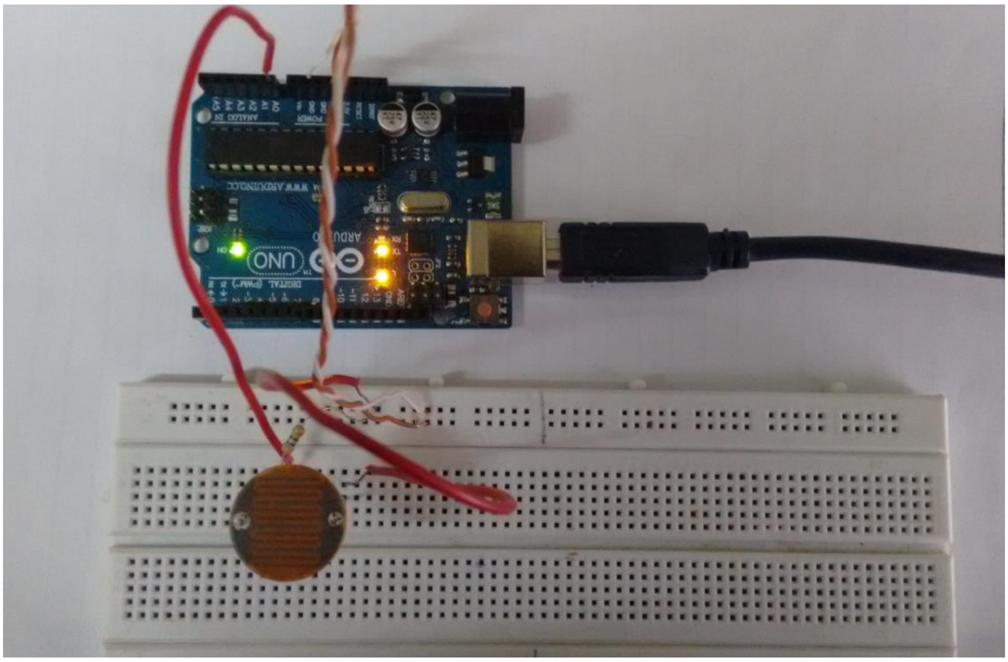


Figure 2.9: LDR sensor connected with Arduino Board.

When we will put darkness around the LDR sensor the LDR value will decrease and when we put the illumination around the sensor the LDR value will decrease.

2.4.3 Implementation of a circuit using LDR sensor

The basic circuit which we implement using the LDR circuit is to turn ON the LEDs when there is darkness around the sensor and turning OFF the LED whenever there is illumination around the LDR sensor.

The code for the above kind of problem is as under

```
int LDR=A0;
int LDRvalue=0;
int light_sensitivity=950;


void setup()
{
Serial.begin(9600);
pinMode(13,OUTPUT);
}
void loop()
{
LDRvalue=analogRead(LDR);
Serial.println(LDRvalue);
delay(50);
if(LDRvalue < light_sensitivity)
{
digitalWrite(13,HIGH);
}
else
```

{

digitalWrite(13,LOW);

}

}

As per this code whenever the LDR value reaches below 950 the LEDs will turn on and whenever the LDR value cross 950 the LEDs will turn OFF.

Here we have two cases. In the first case there is enough light so that LEDs are not turning ON. Whereas in the second case the LEDs are turning ON because there is a shadow of the hand over the LDR sensor.

Here is the case when the enough light is falling on the LDR sensor and LEDs are not turning ON because the LDR value is higher in this case.

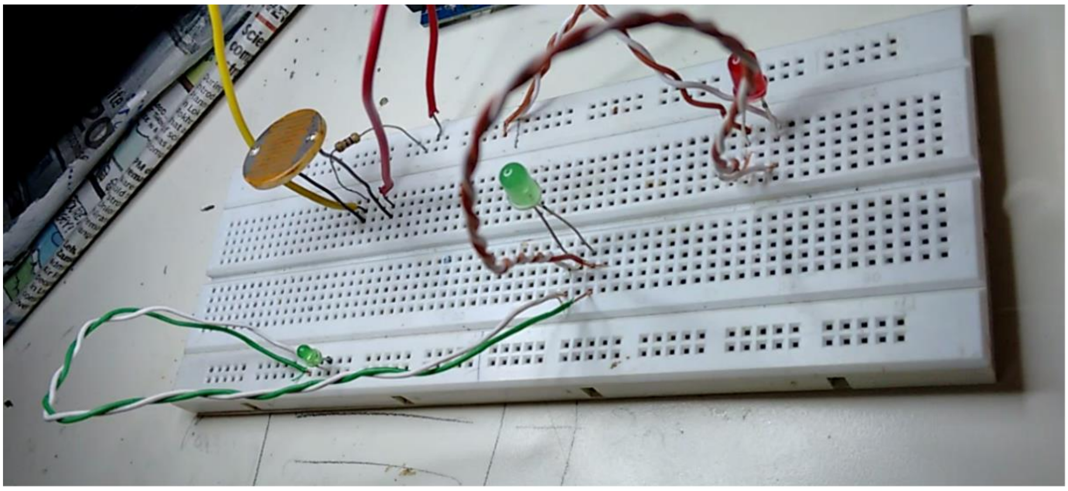


Figure 2.10: Circuit showing when there is sufficient light LEDs do not blink.

As soon as there is a shadow of the hand over the LDR sensor the LDR value will decrease and the LEDs will turn ON.

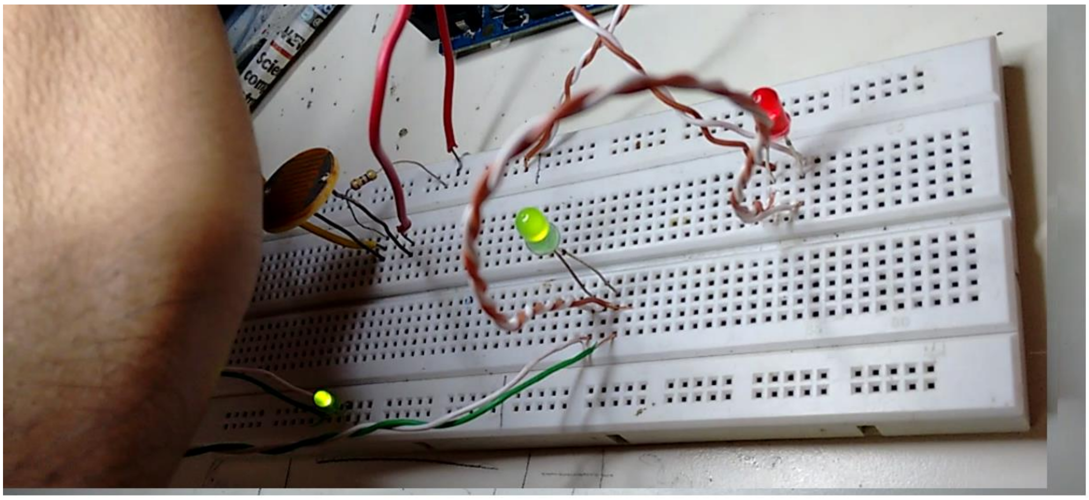


Figure 2.11: Circuit showing when there is shadow on the sensor LEDs blink

2.5 Relay Circuit

A relay is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

Below is an image of a relay

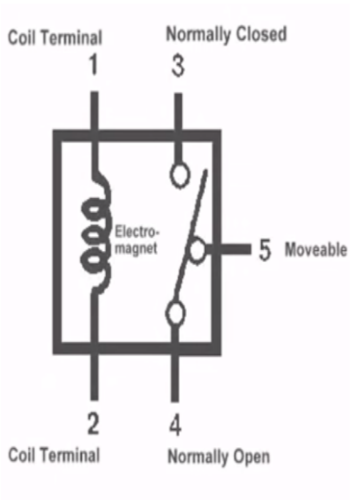


Figure 2.12: A Relay.

Relays are electromechanical devices that use an electromagnet to operate a pair of movable contacts from an open position to a closed position. The advantage of relays is that it takes a relatively small amount of power to operate the relay coil, but the relay itself can be used to control motors, heaters, lamps or AC circuits which themselves can draw a lot more electrical power.

Here is an image of the relay circuit

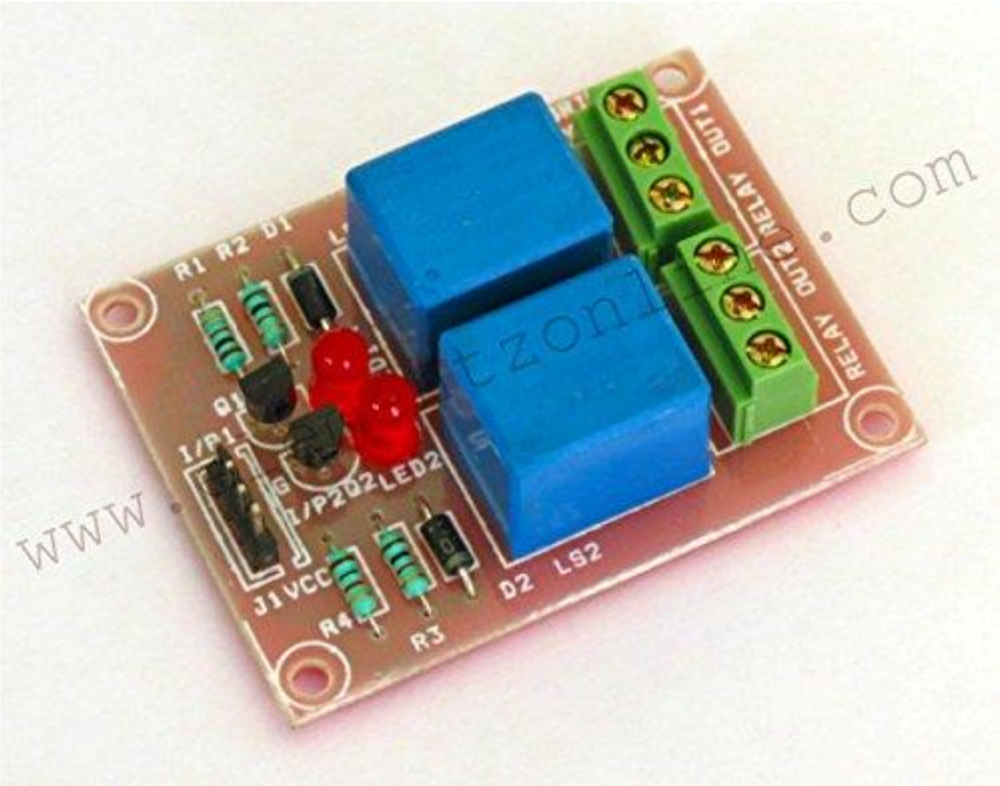


Figure 2.13: Relay Circuit

This is a two channel relay circuit having the specifications as:

- CONTROLLABLE with 5V or 3.3V SIGNAL
- Equipped with high-current relay 7A@250VAC / 10A@24VDC
- It can be used to control both AC and DC appliances such as Solenoids, Motors, lights, fans, etc
- High quality screw terminals (Terminal Block) provided (C, NC , NO) for quick and easy connection
- Free wheeling diode to protect your microcontroller

Chapter 3

Implementation

For implementing the major light control system using the arduino we just connect the LDR sensor to the arduino board and set a LDR value below which the light should turn ON and above which light should turn OFF. The light bulb is connected at pin 13 via a relay circuit and we are setting the LDR value equal to 855.So if the LDR value goes below 855 the lights will turn ON otherwise it will remain OFF.

To connect the LDR sensor we connect one end of the sensor to the Analog pin A3 of the arduino board and the bulb to the pin 12 of the arduino board via relay circuit which will turn OFF and ON accordingly if the LDR value is less than or greater than 855 respectively.

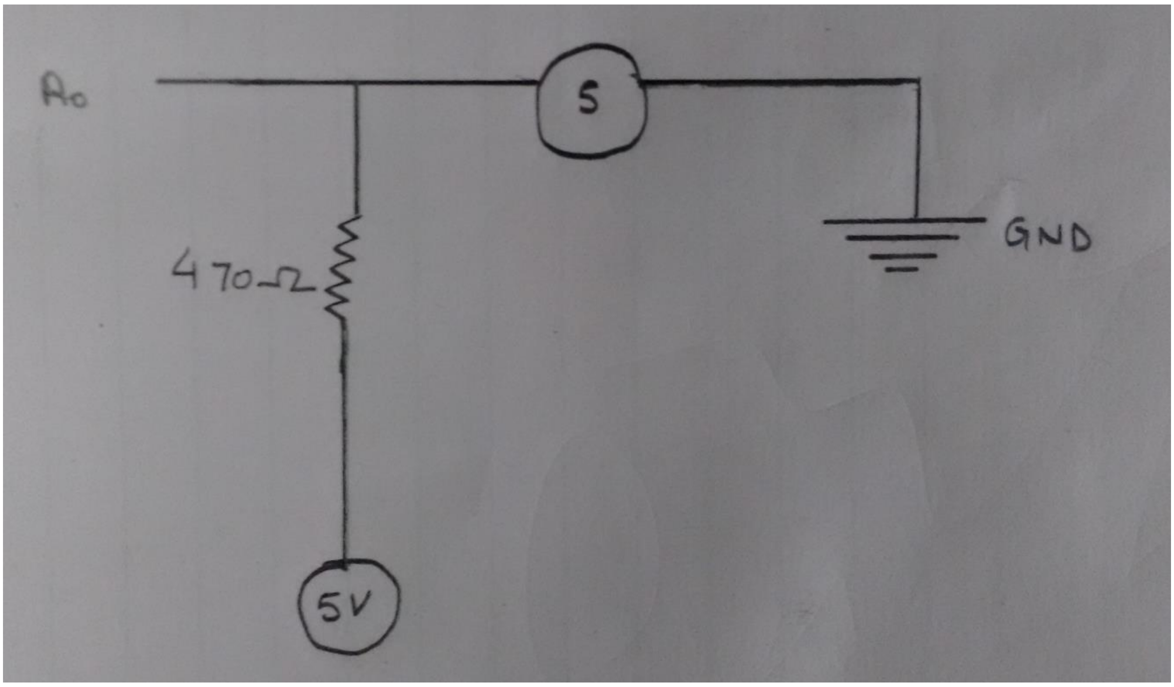


Figure 2.14: Circuit Diagram for connecting sensor to Arduino Board.

Here are the two images which shows the working of whole setup.

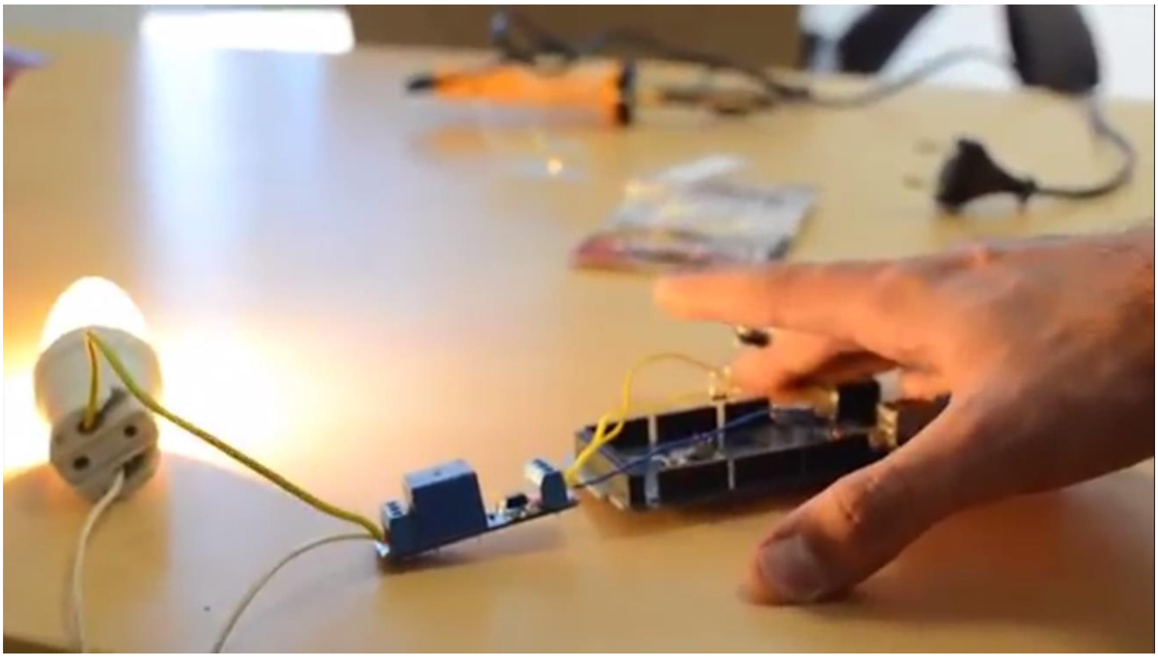


Figure 2.15: Circuit showing when there is a shadow over the sensor bulb glows.

In thee above image the user is covering the LDR sensor by his hand so the LDR value goes below 855 and the bulb gets turned ON .

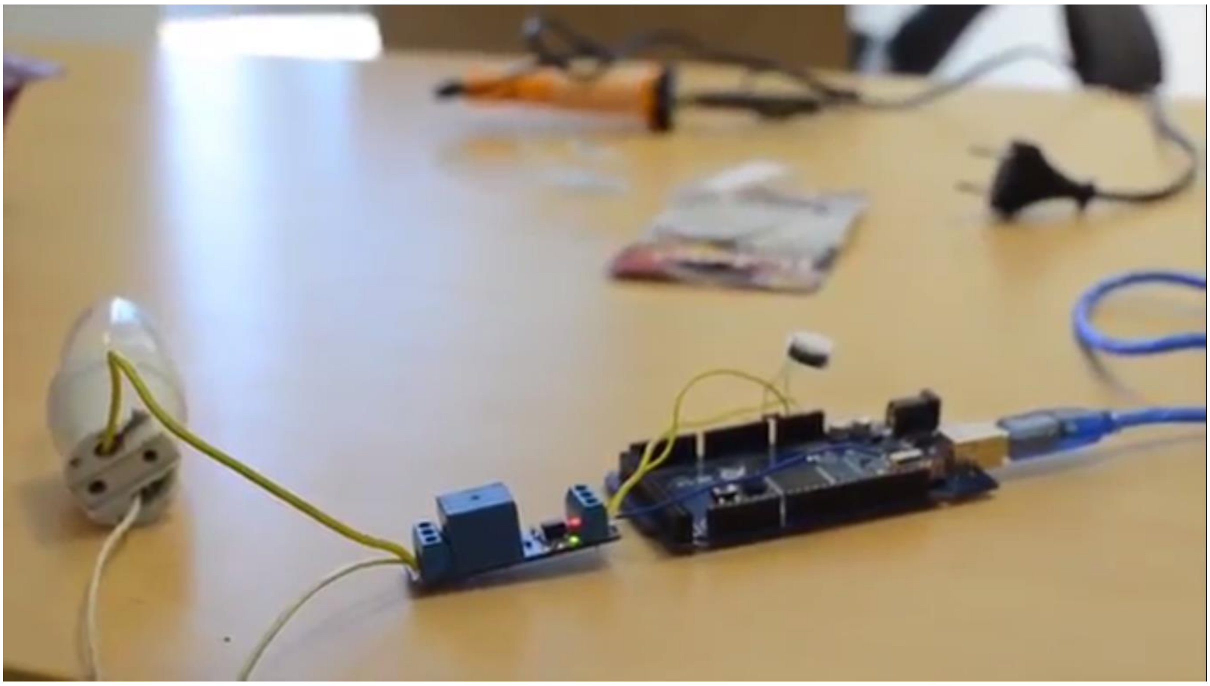


Figure 2.16: Circuit showing when there is sufficient light the bulb does not glow.

In this image as the user removes his hand from the LDR sensor the LDR value goes beyond 855 and the bulb turns Off.

The code for the light control system using arduino is as under:

```
int LDR = A3;
int val = 0 ;
int bulb = 12;

void setup ()
{

pinMode ( bulb , OUTPUT);
Serial.begin(9600);
digitalWrite(bulb , LOW );

}

void loop()
{
  val = analogRead(LDR);
 Serial.print("value is ");
  Serial.println(val);
  delay(50);
 if (val>=855)
{
DigitalWrite(bulb,HIGH);
}
 else
{
 digitalWrite(bulb,LOW);
}
```

CHAPTER 4
Comparative Analysis

Lighting controls present a key opportunity for designers and engineers to tune the lighting system to the needs of the occupants in a dynamic manner while potentially saving significant energy. As the need to reduce lighting energy consumption continues to increase, the ability to dynamically modify the energy use profile within a space is of great value, both to presenter and operators, and to the major utilities whose grid must respond. Innovative and cost effective wireless strategies, when viewed from a life cycle cost standpoint, present an attractive solution for achieving energy savings, flexibility, productivity, reduced maintenance costs and personal control.

With the current availability of energy-efficiency incentives and rebates, utilities are strongly encouraging the use of advanced lighting controls. Combined with the annual energy savings, these incentives and rebates can often reduce the initial cost burden on the presenter, providing further reason to expand on the flexibility of the lighting system.
Under this section we are going to compare the research papers .

The firs paper gives an idea where we can use arduino for various purposes like as a light control system using sesnsors. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.) The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.
The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50
- Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino
- Open source and extensible software- The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

Open source and extensible hardware - The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

A microprocessor is typically found in a desktop PC or laptop and contains a CPU and an external memory interface plus various I/O buses to connect to the outside world such as SPI, I2C, UART, USB, LCD and others. A microprocessors will also have an external crystal to provide a clock.

Microprocessors have very limited read-only memory, generally just enough to contain a boot program that starts the computer up from a hard drive, CD or DVD drive, or in past, a floppy. Unlike microcontrollers, which execute their programs out of read-only memory, microprocessors load their programs into external RAM and execute it from there. (For some microprocessors, even the read-only boot memory is external to the chip.)

A microcontroller on the other hand is a standalone single-chip IC that contains a CPU, read-only memory to store the program, RAM to store variables used in the execution of the program, and various I/O buses to connect to the outside world such as SPI, I2C, UART and others. By itself, it cannot execute any programs without being programmed via an external interface to a PC. A microcontroller may also need an external crystal to provide a clock, however some have an internal clock.

For your purpose, you would want to use a microcontroller, not a microprocessor. To use a microcontroller, you would either have to design your own board, or buy some sort of development board.

An Arduino is such a board, and contains a microcontroller, typical an 8-bit AVR such as the ATmega8, ATmega168, ATmega328, ATmega1280, and ATmega2560, plus power supplies, crystal, and female headers to interface with various peripheral boards.

These peripheral boards are called shields, and are designed to stack on top of each other (there are male pins on the bottom of the boards to connect to the Arduino itself or another shield, and female headers on the top to accept the male pins of a shield stacked on top of it).
Example shields are motor control boards, general I/O boards, relay boards, Ethernet boards, and LCD's, typically with a touch-screen. However I don't know of any resistive touch screens that would be used just for detection (without an LCD).

Arduino libraries are a convenient way to share code such as device drivers or commonly used utility functions. This guide details how to install libraries on your computer. For an excellent introduction to Arduino libraries and what you can do with them, see the Libraries page from Arduino Tips, Tricks and Techniques.

Standard Libraries

The Arduino IDE comes with a set of standard libraries for commonly used functionality. These libraries support all the examples included with the IDE. Standard library functionality includes basic communication functions and support for some of the most common types of hardware like: Servo Motors and Character LCD displays.
Standard Libraries are pre-installed in the "Libraries" folder of the Arduino install. If you have multiple versions of the IDE installed, each version will have its own set of libraries. For the most part, it is not a good idea to change the Standard Libraries or install your libraries in the same folder.
User Installed Libraries

There are many other libraries with useful functionality and device drivers for all sorts of hardware. These drivers are available from places like Arduino Playground, Github and Google Code. Adafruit provides over 100 libraries libraries supporting almost all of our Arduino compatible products. Most of our libraries are hosted on GitHub. Direct links are provided from the product description and/or tutorial pages.

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you're not sure which one is right for your project, check this guide for some helpful hints. Here are a few options that are well-suited to someone new to the world of Arduino:

Arduino Uno (R3)

The Uno is a great choice for your first Arduino. It's got everything you need to get started, and nothing you don't. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

LilyPad Arduino

This is LilyPad Arduino main board! LilyPad is a wearable e-textile technology developed by Leah Buechley and cooperatively designed by Leah and SparkFun. Each LilyPad was creatively designed with large connecting pads and a flat back to allow them to be sewn into clothing with conductive thread. The LilyPad also has its own family of input, output, power, and sensor boards that are also built specifically for e-textiles. They're even washable!

RedBoard

At SparkFun we use many Arduinos and we're always looking for the simplest, most stable one. Each board is a bit different and no one board has everything we want – so we decided to make our own version that combines all our favorite features.

The RedBoard can be programmed over a USB Mini-B cable using the Arduino IDE. It'll work on Windows 8 without having to change your security settings (we used signed drivers, unlike the UNO). It's more stable due to the USB/FTDI chip we used, plus it's completely flat on the back, making it easier to embed in your projects. Just plug in the board, select "Arduino UNO" from the board menu and you're ready to upload code. You can power the RedBoard over USB or through the barrel jack. The on-board power regulator can handle anything from 7 to 15VDC.

Arduino Mega (R3)

The Arduino Mega is like the UNO's big brother. It has lots (*54!*) of digital input/output pins (14 can be used as PWM outputs), 16 analog inputs, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The large number of pins make this board very handy for projects that require a bunch of digital inputs or outputs (like lots of LEDs or buttons).

Arduino Leonardo

The Leonardo is Arduino's first development board to use one microcontroller with built-in USB. This means that it can be cheaper and simpler. Also, because the board is handling USB directly, code libraries are available which allow the board to emulate a computer keyboard, mouse, and more.

ADVANTAGES AND DISADVANTAGES OF THIS SYSTEM

ADVANTAGES

a)Energy Saving: Since it turns on the light only when it is required i.e whenever the person is present near the board and turns off the lights when it is not requied i.e whenever the speaker goes away from the board so it saves a lot of energy in the form of electricity.

b)Reducing Physical Effort: It saves a lot of physical effort of the speaker or the presenter by reducing the manual work of turning on the lights whenever required and turning off whenever it is not required by automatically detecting the prescence of the user near the board.

c)Low Cost: The cost of the over implementation is very low as it uses only arduino, LRD sensors and a relay circuit.

d)Safe and Secure: This system is relatively safe and secure to any other system.

DISADVANTAGE

a)Limitation of Visibility: As discussed earlier that light only turns on when someone is present near the board but if the presenter is not there near the board and something is to be shown written on the board then it is not possible. This is the major disadvantage of this system.

b)Power Limitation: The power is only limited in this setup so it can not be used for a longer period.

CHAPTER 5
PROPOSAL ANALYSIS

This project can be implemented in such educational institutes where there are regular presentations or in the different business meetings. This project is a hardware based project which requires the hardware like Arduino which is a family of single board microcontroller intended to make it easier to build interactive objects or environments and a LDR sensor which can be applied in light-sensitive detector circuits, and light- and dark-activated switching circuits. The presence of a speaker or a presenter is detected using this sensor and a program is written in the Arduino software which is uploaded to the Arduino board. This implementation saves a lot of energy and a lot of effort to again turn on and turn off the light.

Lighting controls have traditionally been used to create moods. Today, they are also used as part of a high quality energy efficient lighting system that integrates daylight and electric light sources to provide a comfortable and visually interesting environment for the occupants of a space. Electric lighting controls are appropriate for a wide variety of spaces, from restrooms to large open offices, from conference rooms to classrooms. They can be incorporated with day lighting to provide flexibility, energy savings, and ecological benefits. Although lighting controls are still most commonly used in commercial buildings, they are also increasingly being used in residential applications.

Electric lighting controls are used in lighting design projects to achieve a high quality energy efficient lighting system. Specifying a layered, daylight-integrated lighting and control system gives the occupants control of the lighting while providing appropriate lighting levels, minimizing glare, balancing surface brightness, and enhancing the surrounding architecture.

When electric lighting controls are used properly, energy will be saved and the life of lamps and ballasts can be extended. Lighting controls will help reduce energy by:

Reducing the amount of power used during the peak demand period by automatically dimming lights or turning them off when they are not needed

Reducing the number of hours per year that the lights are on

Reducing internal heat gains by cutting down lighting use, which allows for reduced HVAC system size and a reduction in the building's cooling needs

Allowing occupants to use controls to lower light levels and save energy

Photosensor

Example of a photosensor

There are other reasons to use electric lighting controls. For example, dimming controls can provide the lighting flexibility which is often required in multi-use rooms or rooms in which projectors are used. Exterior motion detectors and interior occupancy sensors can be used to turn lights on when people (including intruders) are present. Moreover, by tuning an environment for the individual occupant's or group's visibility, comfort and productivity can be improved.

A. Types of Lighting Controls

The most common form of electric lighting control is the on/off "toggle" switch. Other forms of lighting control include occupancy sensors, daylight sensors, clock switches, a variety of manual and automatic dimming devices, and centralized controls. Some controls operate on line-voltage power, while others are low-voltage (DC) powered. Note that fluorescent fixtures that are intended to be dimmed require special compatible dimming ballasts. Also, controls can be linked together which can perform multiple control tasks.

Standard on/off switches and relays can be used to turn groups of lights on and off together. Creative design options can be developed with this simple tool, if the circuiting is properly designed. For example, some of the lamps in each fixture can be switched together, every other fixture can be switched as a group, or lighting near the windows can be turned off when daylight is plentiful.

Occupancy sensors (including passive infrared, ultrasonic, and dual technology sensors) serve three basic functions:

To automatically turn lights on when a room becomes occupied,

To keep the lights on without interruption while the controlled space is occupied, and

To turn the lights off within a preset time period after the space has been vacated.

Some sensors have settings that allow the specifier to select between the functions listed above (manual on instead of automatic on, for example). Note that sensor characteristics may vary considerably from manufacturer to manufacturer, so it is important to carefully evaluate the options for each device.

Passive infrared sensors (PIR) are triggered by the movement of a heat-emitting body through their field of view. Wall-box type PIR occupancy sensors are best suited for small, enclosed spaces such as private offices, where the sensor replaces the light switch on the wall and no extra wiring is required. PIR sensors cannot "see" through opaque walls, partitions, or windows so occupants must be in direct line-of-site of the sensor.

Ultrasonic sensors emit an inaudible sound pattern that is disrupted by any moving object altering the signal returning to the sensor (Doppler shift). They are best suited for spaces where line-of-sight view to the occupant is not always available. This type of sensor detects very minor motion better than most infrared sensors and is often used in restrooms since the hard surfaces will reflect the sound pattern.

Dual-technology occupancy sensors use both passive infrared and ultrasonic technologies for less risk of false triggering (lights coming on when the space is unoccupied). Combining the technologies requires a more reliable, yet slightly larger and more expensive device.

Occupancy sensor

Example of a ceiling-mounted occupancy sensor

Occupancy sensor placement is very important to the successful implementation of the control design intent. Occupancy sensors must be located to ensure that they will not detect movement outside of the desired coverage area, through an open doorway, for example. Ultrasonic devices are sensitive to air movement and should not be placed near an HVAC diffuser, where air movement may cause false tripping.

Occupancy controls can be used in conjunction with dimming or daylight controls to keep the lights from turning completely off when a space is unoccupied, or to keep the lights off when daylight is plentiful and the room is occupied. This control scheme may be appropriate when occupancy sensors control separate groups of luminaires, or "zones", in a large space, such as in a laboratory or an open office area. In these situations, the lights can be dimmed to a predetermined level in one specific area when the space is unoccupied.

There are several different kinds of coverage patterns and mounting configurations for occupancy sensors, such as:

Ceiling-mounted controls with 360° coverage

Ceiling-mounted controls with elongated "corridor" coverage

Wall-mounted controls with a fan-shaped coverage pattern

Ceiling-mounted controls with a rectangular coverage pattern

and more!

Additionally, take note of the difference between each device's sensitivity to minor motion (working at a desk) vs. major motion (walking or half-step activity). The sensor manufacturer should provide coverage diagrams for both levels of activity. HID lamps do not work well with occupancy sensors because most HID lamps take a long time to start each time they are switched off.

Manual dimming gives occupants of a space an added degree of control and satisfaction, as well as an opportunity to save energy. It provides users with the flexibility to instantly change the characteristics of a space to make it a more comfortable and productive environment. There are several families of manual preset dimming control.

Manual hard-wired control

Preset scene control

Remote infrared control

Remote radio frequency control

Manual hard-wired control consists of a dimmer, connected to a single luminaire or zone which is operated by the user at the device. Preset scene dimming controls change the light level settings for multiple zones simultaneously at the press of a button.

Remote control dimming is another form of manual dimming that is well suited for retrofit projects to minimize rewiring. Infrared and radio frequency technologies are most successful in these applications. Remote infrared control operates in a similar fashion to other infrared technologies like television, for example. Radio frequency controls are equipped with a sender that "talks" to other dimmer's receivers. This allows multi-zone control from a single-zone device. Personal control systems are now available that allow users to change levels of lighting, sound, heating/cooling, etc., in their own workspaces. See also WBDG Productive—Design for the Changing Workplace.

Light-level sensors or photosensors can be used to automatically turn lights on or off, or dim them, depending on the available daylight available in the space. Daylight dimming can maintain the desired light level while providing a smooth, barely noticeable transition to or from electric lighting as daylight increases or decreases.

Clock switches turn lights on or off for a specific period of time. They are especially useful for turning off photocell-activated exterior lighting late at night (as long as that lighting is not needed after a certain time).

Centralized controls can be used to automatically turn on, turn off, and/or dim lighting at specific times or under certain load conditions. This type of control can be used in a conference room or on a building-wide scale. Centralized control strategies can also integrate lighting controls with other building systems such as mechanical or security systems.

Distributed controls are based on digital communication protocols. These systems are local, or integral, to the luminaire itself, not housed in a central cabinet or enclosure. They integrate with building automation or energy management system. A Digital Addressable Lighting Interface (DALI) system provides a means of control which can speak to an individual ballast or groups of ballasts. The "control wiring" is independent of the "power wiring" and provides the highest degree of flexibility available at this time. When space configuration or occupant needs change, the system can respond by reassigning the ballasts accordingly.

B. Selecting the Appropriate Lighting Controls

There are many guides and services available for designing lighting control systems. Some controls manufacturers will actually do a controls layout on the building plans or even on electronic drawing files, free of charge or for a nominal fee. If you prefer to do the controls layout yourself, it is advisable that you provide a written "sequence of control" which describes the design intent and a "performance specification" which describes the performance characteristics of individual components. Further, some coordination with the selected controls manufacturer will help to avoid surprises during construction and commissioning.

Chapter 6

CONCLUSION AND FUTURE SCOPE

In this project work, we have studied and implemented a complete working model using a Microcontroller. The programming and interfacing of microcontroller has been mastered during the implementation. This work includes the study of LRD sensors also. So the implementing the project can save a lot of energy and a human effort for turning on and off the light again and again by the presenter or speaker present in that room.

Moreover we can apply two sensors one at one corner of the board and the other at the other corner of the board so that if the speaker is writing from any end the light should get turn on and if he is in middle of the board then also the presence of the speaker can be detected by any of the sensor and the lights will turn on.

In place of the LDR sensor the ultrasonic sensor can also be used which will detect the presence of speaker by some kind of disturbances in the sound. Ultrasonic sensors are based on measuring the properties of sound waves with frequency above the human audible range. They are based on three physical principles: time of flight, the Doppler effect, and the attenuation of sound waves. Ultrasonic sensors are non-intrusive in that they do not require physical contact with their target, and can detect certain clear or shiny targets otherwise obscured to some vision-based sensors. On the other hand, their measurements are very sensitive to temperature and to the angle of the target.

Further improvements can be done in this project by adding features like managing intensity of light as per the intensity of lights already on in the room.

So finally concluding what we have discussed in this report

Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures kits for building digital devices and interactive objects that can sense and control the physical world. Arduino boards may be purchased preassembled, or as do-it-yourself kits; at the same time, the hardware design information is available for those who would like to assemble an Arduino from scratch.

A sensor is a transducer whose purpose is to sense (that is, to detect) some characteristic of its environs. It detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal; for example, a thermocouple converts temperature to an output voltage. But a mercury-in-glass thermometer is also a sensor; it converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube.

Sensors are used in everyday objects such as touch-sensitive elevator buttons (tactile sensor) and lamps which dim or brighten by touching the base, besides innumerable applications of which most people are never aware. With advances in micromachinery and easy-to-use microcontroller platforms, the uses of sensors have expanded beyond the more traditional fields of temperature, pressure or flow measurement, for example into MARG sensors. Moreover, analog sensors such as potentiometers and force-sensing resistors are still widely used. Applications include manufacturing and machinery, airplanes and aerospace, cars, medicine and robotics.

Chapter 7

REFERANCES

[1] Arduino ADK Schematic ,Referance Design and Pin Mapping [online]
Available: http://arduino.cc/en/Main/ArduinoBoardADK.

[2] H.Norton," Transducer Fundamentals", in Handbook of Transducers. Englewood Cliffs ,NJ:Prentice Hall, 1989 ch2.

[3] About LDR sensors[online]Available: https://www.kitronik.co.uk/blog/how-an-ldr-light-dependent-resistor-works/

[4] About relay [online]Available: http://en.wikipedia.org/wiki/Relay

[5] C.E Shannon A symbolic analysis of realy and switching circuits.Trans.AIEE57:713-723(1938)

[6] About relay circuits[online]Available
http://www.electronics-tutorials.ws/blog/relay-switch-circuit.html

[7] light sensors using arduino[online]Available
http://rookieelectronics.com/light-sensor-using-arduino/

[8] motion sensors[online]Available

http://www.instructables.com/id/Arduino-Controlled-Motion-Sensor/