

Effective Energy Consumption

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

Bachelor of Technology.

in

Information Technology

Under the Supervision of

Mr. Punit Gupta

By

Gavish Singla (111441)

to



Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled “Effective Energy Consumption”, submitted by Gavish Singla in partial fulfillment for the award of degree of Bachelor of Technology in Information Technology 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.

Date:

Mr. Punit Gupta

Associate Professor

Acknowledgement

I gratefully acknowledge the Management and Administration of Jaypee University of Information Technology for providing me the opportunity and hence the environment to initiate and complete my project.

I would like to express my gratitude to Prof. Dr. RMK Sinha, Dean and Head of Department, Information Technology JUIT and Brig. (Retd) S.P.Ghrera Head of Department, Computer Science, JUIT Waknaghat for guiding, encouraging and inspiring by giving me valuable thoughts to carry out the project and also like to thank my project guide, Mr. Punit Gupta, Assistant Professor, Dept. of Information Technology for his expert guidance, encouragement and valuable suggestions at every step.

Date:

Gavish Singla

Table of Content

S. No.	Topic	Page No.
1.	Introduction	2
1.1	Introduction: Embedded System	2
1.1.1	General Characteristics	3
1.2	2 nd Generation Intel Galileo Board	3
1.2.1	General Characteristics	6
1.2.1.1	Specifications	6
1.2.1.2	Processor Features	7
1.2.1.3	Storage Options	7
1.2.2	Enhancements	8
1.2.3	Description of Key Components	8
2.	Introduction to SDK	11
2.1	Arduino SDK	11
2.1.1	Modes of Communication	12
2.1.2	Language Reference of Arduino	13
3.	Literature Review	14
4.	Stimulation and work done	19

5.	Project Design	24
5.1	Project Introduction	24
5.2	Hardware Requirements	24
5.3	Software Requirements	31
5.4	Block Diagram	32
5.5	Circuit Diagram	33
5.6	Implementation	35
6.	Conclusion and Future Work	41
	References	42
	Useful Links	43

List of Figures

S.No.	Title	Page No.
1.	Upper view of 2 nd Gen Intel Galileo Board	4
2.	Pin configuration of 2 nd Gen Intel Galileo Board	5
3.	Detailed View of Board	8
4.	Output when pin is HIGH	20
5.	Output when pin is LOW	20
6.	Output on Serial Monitor	22
7.	When value of sens >400	23
8.	When value of sens < 400	23
9.	Colour Coding	26
10.	ORP12 Cadmium Sulphide photoconductive cell	28
11.	Photo Sensor	29
12.	Block Diagram	32
13.	Schematic Circuit Diagram	33
14.	Breadboard Circuit Diagram	34
15.	Connection of board with prototype	35
16.	Scenario 1	36
17.	Scenario 2	37
18.	Scenario 3	38
19.	Scenario 4	39
20.	Scenario 5	40

List of Tables

S.No.	Title	Page No.
1	Embedded System Characteristics	3
2	Specifications of board	6
3	Processor Features	7
4	Storage Options	7
5	Component Description	8
6	Language Reference	13

Abstract

Electricity is a resource which is becoming hard to generate and thereby leading to higher cost with each coming day. The idea of this project came so as to save the power consumption and to minimize the never ending bad habit of wasting. Even if it is for small amount of time, saving electricity power can reduce a significant amount of expenditure. With the help of my project I am hoping to reduce unnecessary use of electricity as much as possible.

In this generation where fully automatic systems are used almost in every life purpose, we have designed this project to serve the purpose of human need. This project is an application of the microcontroller that we use in our daily life. Normally we do not care to switch off the electrical devices used at our homes or offices i.e. tubes, bulbs, etc. even if there is no need for it to be switched on. Every functionality of these devices has a particular power ratings according to which it consumes electricity, now the working of these devices without any use will lead to wastage of power and ultimately to higher electricity bills.

The project can be used to avoid the above problem. In this project every room (in my project only one room) will be monitored by a single microcontroller and the microcontroller will record the intensity of sunlight and the electrical light present in that particular room. If the intensity of light is sufficient for the user to work then it will automatically turn off the electrical light source or else if needed would adjust the brightness of the light source depending on the light intensity in the room being monitored.

In the project I will be using Intel Galileo Gen 2 board as the main hardware along with various sensors for the implementation of the project. Arduino SDK (version 1.5.3) will be used as the SDK for programming of the board.

CHAPTER 1

INTRODUCTION

1.1 Introduction : Embedded System

An embedded system can be broadly defined as a device that contain tightly coupled hardware and software components to perform a single function, forms part of a larger system, is not intended to be independently programmable by the user, and is expected to work with minimal or no human interaction.

Embedded system often require real time operations and multitasking capabilities. Most embedded systems interact directly with processes or the environment, making decisions based on the inputs. This means it is necessary for the system to be reactive, responding in real time to process inputs to ensure proper operation. These systems operate in constrained environments where memory, computing power, and power supply are limited. Multitasking is the capability to perform many functions in a simultaneous or quasi-simultaneous manner. The embedded controller is often responsible of monitoring several aspects of a system and responding according when need arises.

The concept of embedded system is as old as the concept of an electronic computer, and in a certain way, it can be said to precede the concept of a general purpose computer. If we look at the earliest forms of computing devices, they adhere better to the definition of an embedded system than to that of a general purpose computer. Embedded systems are found in a variety of common electronic devices, such as consumer electronics (cell phones, digital cameras and calculators), home appliances (microwaves ovens, answering machines, and thermostats), office automation (fax machines, printers, scanners) and automobiles etc. Since the embedded system is dedicated to specific tasks, design engineers can optimize it, reducing the size and cost of the product, or increasing the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

1.1.1 General Characteristics

- Embedded systems are application specific and single functioned, the programs are executed repeatedly.
- Efficiency are optimized for energy, code size, execution time, weight & dimensions, and cost.
- Embedded systems are typically designed to meet real time constraints, a real time system reacts to stimuli from the controlled object/ operator within the time interval dictated by the environment. For real time systems, right answers arriving too late (or even too early) are wrong.
- Embedded systems often interact (sense, manipulate & communicate) with external world through sensors and actuators and hence are typically reactive systems, a reactive system is in continual interaction with the environment and executes at a pace determined by that environment.
- They generally have minimal or no user interface.

Characteristic	Description
Single Functioned	Executes a specific program repeatedly.
Tightly Constrained	Must cost few dollars, must be fast and power efficient
Dependable	Depends on output produced by connected system
Reliable	Increased by introducing redundancy
Real Time	Response should occur in a specific time
Efficiency	Consumes less power, small space, less cost more speed

Table 1: Embedded System Characteristics

1.2 Introduction : 2nd generation Intel Galileo Board

The 2nd generation Intel Galileo board provides a single board controller for maker community, students and professional developers. Based on the Intel Quark SoCX1000, a 32-bit Intel Pentium processor- class system on a chip (SoC), the genuine board provide a full featured offering for a wide range of applications. Arduino-certificated and designed to be hardware, software, and pin compatible with large range

of Arduino Uno R3 shields, the board also provides a simple and more cost effective development environment compared to the Intel Atom processor and Intel Core processor based designs. Based on a new micro-architecture, the processor is designed for a two-chip platform consisting of a processor and Platform Controller Hub (PCH). The platform enables higher performance, lower cost, easier validation, and improved x-y footprint. The processor includes Integrated Display Engine, Processor Graphics, PCI Express ports, and Integrated Memory Controller. The processor is designed for desktop platforms. It supports up to 12 Processor Graphics execution units (EUs).

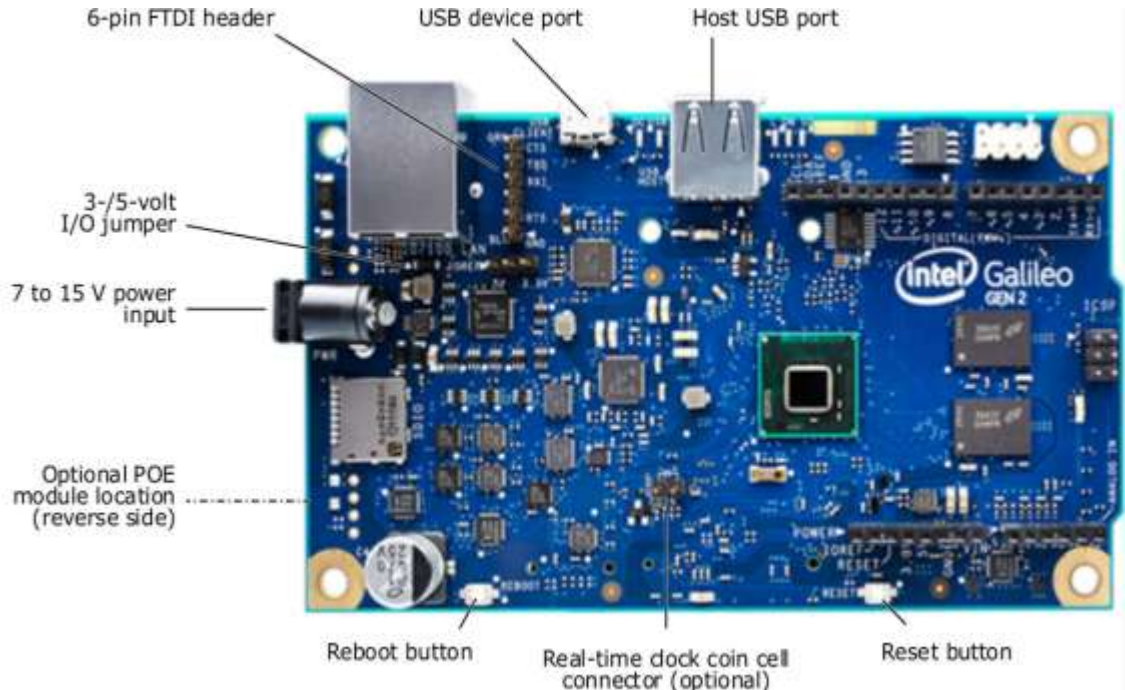


Figure 1: Upper view of 2nd Gen Intel Galileo Board

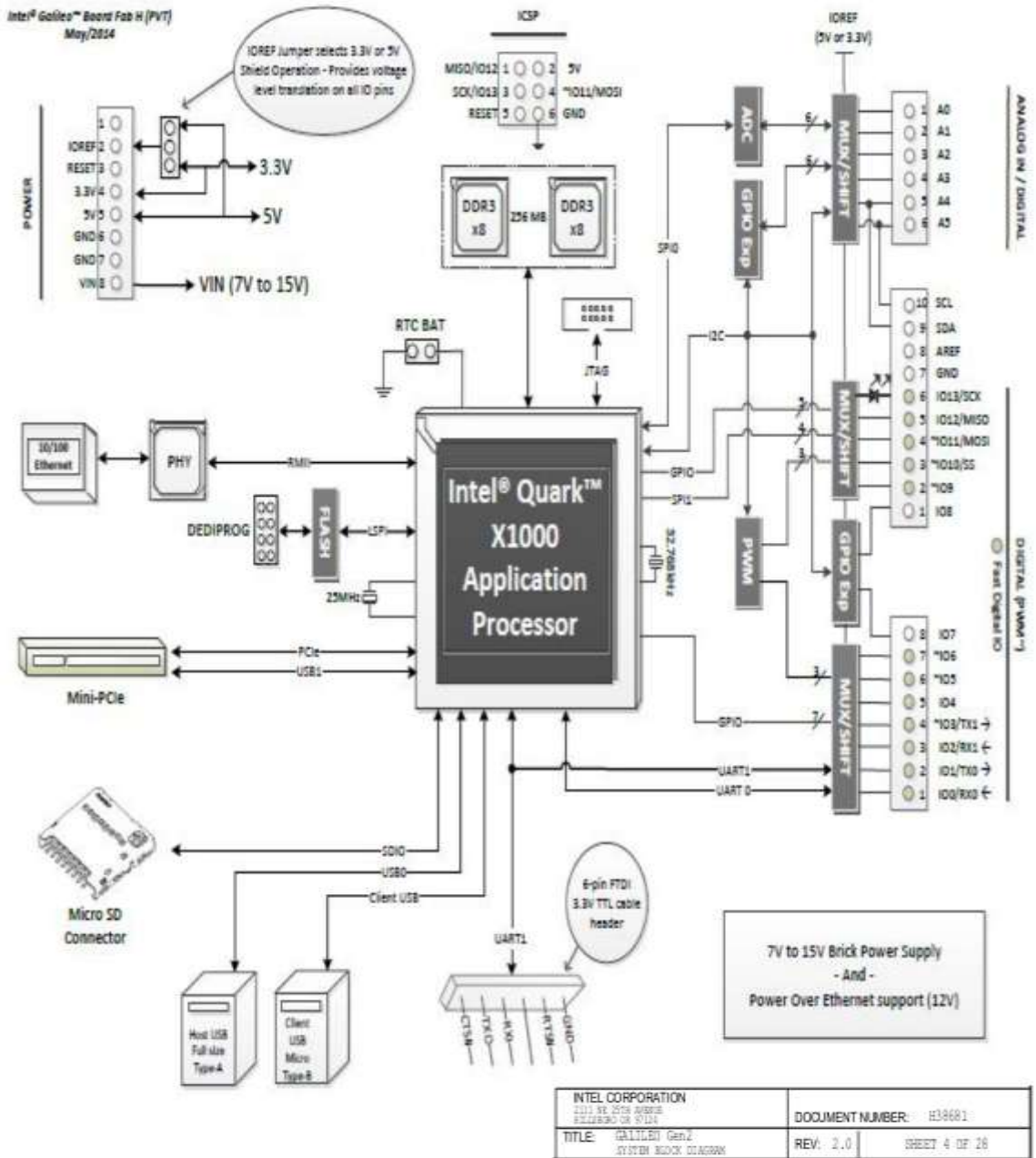


Figure 2: Pin configuration of 2nd Gen Intel Galileo Board

1.2 .1 General Characteristics

1.2.1.1 Specifications

DIMENSIONS	123.8 mm(L) x 72.0 mm(W)
ATTACHMENTS	Four screw holes 4mm (diameter) Arduino compatible headers containing: <ul style="list-style-type: none">• 20 digital I/O• 6 analog inputs• 6 PWMs with 12-bit resolution• 1 SPI master• 2 URATs (1 shared with console URAT)• 1 I²C master
CONNECTORS	6-pin console URAT (compatible with FTDI USB converters) 6-pin ICSP 10-pin JTAG for debugging RJ45 Ethernet, Power over Ethernet capable USB 2.0 Host(Standard Type A) USB 2.0 Client(micro- USB Type A) Mini-PCI Express 1x slot
POWER	Jack with increased range (7 to 15 V) Supports Power-over-Ethernet (requires PoE module installation) Header for RTC power
BUTTONS	Reset for sketch and attached shield resets 10/100 Mbps Ethernet Reboot to reset the Intel Quark SoC X1000

Table 2: Specifications of board

1.2.1.2 Processor Features

MODEL	Intel Quark SoC X1000
SPEED	400 MHz
CORES/ THREADS	1/1
INSTRUCTION SET ARCHITECTURE (ISA)	32-bit Intel Pentium processor compatible ISA
L1 CACHE	16 Kb
SRAM	512 Kb on- die, embedded
PACKAGING	15 mm x 15 mm BGA ACPI- compatible with CPU sleep states
TECHNOLOGIES SUPPORTED	Integrated real time clock (RTC) Optional 3 V coin cell battery for operation between turn on cycles

Table 3: Processor Features

1.2.1.3 Storage Options

FIRMWARE/ BOOT LOADER	8 MB NOR Flash
DRAM	256 MB DDR3, 800 MT/s
SD CARD (OPTIONAL)	Up to 32 GB
USB	Compatible with any USB 2.0 storage device (USB drive/ stick)
EEPROM	8 KB (programmed via the EEPROM library)

Table 4: Storage Options

1.2.2 Enhancements

The Intel Galileo board (Gen 2) delivers improved features and functionality in following areas:

- 12 GPIOs fully native for greater speed and improved drive strength.
- 12-bit PWM for more precise control of servos and smoother response.
- 12 V Power-over-Ethernet capable.
- Power supplies from 7 V to 15 V are supported.
- Serial console URAT header is compatible with FTDI USB converters.
- Console URAT1 can be redirected to Arduino headers in sketches, which can eliminate the need for soft-serial.

1.2.3 Description of Key Components

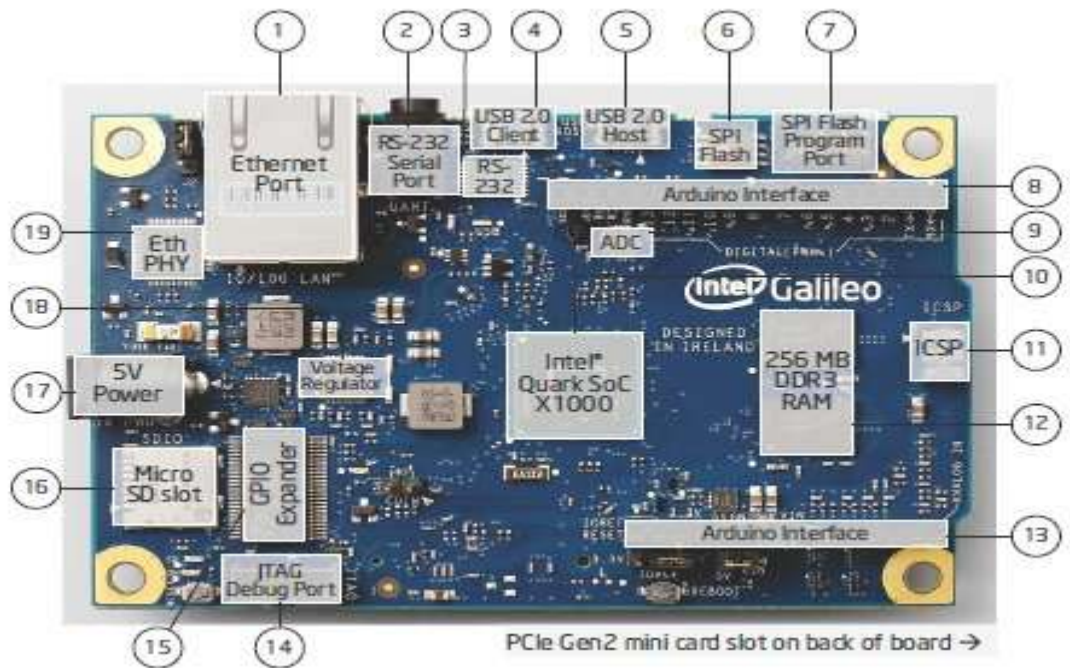


Figure 3: Detailed View of Board

Number	Component	Description
1	Ethernet Port	10/100 Ethernet connector
2	RS-232 Serial Port	3-pin 3.5 mm jack
3	RS-232	RS-232 transceiver

4	USB 2.0 Client	A fully compliant USB 2.0 Device controller, for programming
5	USB 2.0 Host	Supports up to 128 USB end point devices
6	SPI Flash	To store the bootloader and the latest sketch
7	SPI Flash Program Port	7-pin header for Serial Peripheral Interface (SPI) programming Defaults to 4 MHz to support Arduino Uno shields. Programmable up to 25 MHz.
8	Shield Interface	Complies with Arduino Uno Revision 3 shield pin out.
9	ADC	Analog to Digital Converter
10	Intel Quark SoC X1000	400 MHz 32-bit Intel Pentium instruction set architecture (ISA)-compatible processor
11	ICSP	6-pin in-circuit serial programming (ICSP) header. These pins support SPI communication using the SPI library.
12	256 MB DDR3 RAM	Enabled by the firmware by default.
13	Arduino Interface	Complies with Arduino Uno Revision 3 pinout.
14	JTAG Debug Port	10-pin standard JTAG header for debugging
15	GPIO Expander	GPIO pulse width modulation provided by a single I ² C I/O expander
16	Micro SD slot	Supports micro SD card up to 32 GB
17	5V Power	The board is powered via an AC-to-DC adapter, connected by plugging a 2.1 mm center-positive plug into the board's power jack, output rating of the power adapter is 5V at up to 3A.

18	Voltage Regulator	Generates 3.3 volt supply. Maximum current draw to the shield is 800 mA.
19	Eth PHY	Ethernet Physical layer transceiver

Table 5: Component Description

CHAPTER 2

INTRODUCTION TO SDK

A software development kit (SDK or "devkit") is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

It may be something as simple as the implementation of one or more application programming interfaces (APIs) in the form of some libraries to interface to a particular programming language or to include sophisticated hardware that can communicate with a particular embedded system. Common tools include debugging facilities and other utilities, often presented in an integrated development environment (IDE). SDKs also frequently include sample code and supporting technical notes or other supporting documentation to help clarify points made by the primary reference material.

SDKs may have attached licenses that make them unsuitable for building software intended to be developed under an incompatible license. Providers of SDKs for specific systems or subsystems may sometimes substitute a more specific term instead of software. For instance, both Microsoft and Apple provide driver development kits (DDK) for developing device drivers.

2.1 Introduction: Arduino SDK

The Arduino integrated development 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

The Arduino IDE uses the GNU tool chain and AVR Libc to compile programs, and uses avrdude to upload programs to the board. Arduino SDK is a software application that provides comprehensive facilities to computer programmers for software development.

We will be using Arduino version 1.5.3 which will support Intel Galileo Gen 2 board.

2.1.1 Modes of Communication

There are two modes of communication:

- I. *Serial*: Used for communication between the Arduino board and a computer or other devices. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. Thus, if we use these functions, we cannot also use pins 0 and 1 for digital input or output. We can use the Arduino environment's built-in serial monitor to communicate with an Arduino board.
- II. *Stream*: Stream is the base class for character and binary based streams. It is not called directly, but invoked whenever you use a function that relies on it. Stream defines the reading functions in Arduino. When using any core functionality that uses a `read()` or similar method, you can safely assume it calls on the Stream class.

2.1.2 Language Reference of Arduino

The Arduino programs can be divided in three main parts:

- I. Structure
- II. Values (variables and constants)
- III. Functions

Structure	Variables	Functions
Control Structures	Constants	Digital I/O
Syntax	Data Types	Analog I/O
Arithmetic Operators	Conversion	Math
Comparison Operators	Variables Scope and Qualifiers	Trigonometry
Boolean Operators	Utilities	Random Numbers
Pointer Access Operators		Bits and Bytes
Compound Operators		Interrupts (External and Internal)
Bitwise Operators		USB

Table 6: Language Reference

CHAPTER 3

Literature Review

3.1 Method and apparatus for controlling minimum brightness of a fluorescent lamp [Publication number US6946806 B1]

This provided us efficient power conversion circuit for driving a fluorescent lamp that uses a minimum pulse generator circuit to control the minimum on-time of a time modulated signal to increase the dimming range of the fluorescent lamp operating over a wide range of temperature and supply voltage. It further tells us that a minimum number of lamp current cycles with respective amplitudes above a preset threshold is typically required to avoid flickering or shimmering during minimum brightness. The minimum pulse generator circuit counts the lamp current cycles and adjusts the on-time accordingly to guarantee the minimum number of cycles with respective amplitudes above a preset threshold under all operating conditions.

Fluorescent lamps are used in a number of applications where light is required but the power required to generate light is limited. One particular type of fluorescent lamp is a cold cathode fluorescent lamp (CCFL). CCFLs are used for back or edge lighting of liquid crystal displays (LCDs) which are typically used in notebook computers, web browsers, automotive and industrial instrumentation, and entertainment systems. A power conversion circuit is used for driving the CCFL. The power conversion circuit accepts a direct current (DC) supply voltage and provides a substantially sinusoidal output voltage to the CCFL. The brightness of the CCFL is controlled by controlling the current (i.e., lamp current) through the CCFL. The lamp current is time modulated by selectively turning off the sinusoidal output voltage provided to the CCFL for varying time durations. The dimming range of the CCFL is generally limited by the minimum brightness that can be achieved without flickering or shimmering. To achieve minimum brightness without flickering or shimmering, the on-time of the sinusoidal output voltage needs to be the minimum time possible to produce a lamp current with a minimum number of cycles with respective amplitudes above a preset threshold.

The minimum pulse generator circuit is part of a controller in a power conversion circuit for driving the CCFL. The controller generates signals with active states and inactive states corresponding respectively to the on-times and the off-times of the CCFL. The durations of the respective active states are equal to or greater than a minimum duration determined by the minimum pulse generator circuit which counts cycles of current flowing through the CCFL with respective amplitudes above a preset threshold. One or more control signals are provided to the controller indicating a control value for comparison with a value representing the cycles counted by the minimum pulse generator circuit.

The pulse generator circuit is configured to determine an initial duration for the active states (i.e., on-times of the CCFL). The minimum pulse generator circuit is configured to determine the minimum duration for the active states. A logic gate is configured to output a signal to the pulse width modulation circuit with a duty cycle corresponding to a greater of the initial duration duty cycle and the minimum duration duty cycle. the control value is communicated via control signals and is stored in a memory element of the minimum pulse generator circuit. The differential amplifier includes internal hysteresis.

3.2 Inverter controller with automatic brightness adjustment circuitry [Publication Number US 7095392 B2]

It provides a brightness controller for an LCD panel display, comprising an optical sensor generating a signal indicative of ambient light around the LCD panel, a look-up table receiving the signal indicative of ambient light around the LCD panel and adapted to generate a selected one of a plurality of target panel brightness level signals based on the signal indicative of ambient light around the LCD panel, and an inverter circuit adapted to receive one of the plurality of target panel brightness level signals to control power delivered to the LCD panel.

It also provides an LCD panel, comprising a plurality of lamps, and a brightness controller controlling the brightness of said lamps, said brightness controller comprising an optical

sensor generating a signal indicative of ambient light around said LCD panel, a look-up table receiving said signal indicative of ambient light around said

LCD panel and adapted to generate a selected one of a plurality of target panel brightness level signals based on said signal indicative of ambient light around said LCD panel, and an inverter circuit adapted to receive one of said plurality of target panel brightness level signals to control power delivered to said lamps.

The present invention also provides a computer system that includes the LCD panel, lamps and brightness controller. In another exemplary embodiment, the present invention provides a master/slave brightness controller for an LCD panel display, comprising:

- I. An optical sensor generating a signal indicative of ambient light around said LCD panel
- II. A micro controller controlling said optical sensor to generate said signal indicative of ambient light around said LCD panel
- III. A look-up table receiving said signal indicative of ambient light around said LCD panel and adapted to generate a selected one of a plurality of target panel brightness level signals based on said signal indicative of ambient light around said LCD panel
- IV. An inverter circuit adapted to receive one of said plurality of target panel brightness level signals to control power delivered to LCD panel.

3.3 Lighting control [Publication Number EP1657614 A1]

Method for controlling a lighting system, which at a given illumination intensity (E_0) controls in a region to be illuminated, in particular at a workplace, with changing reflection conditions:

At least one from the illuminated region remotely located light sensor, the entire illumination intensity (E) measures and on exceeding an adjustable threshold value, the lighting system off, when it falls below turn on.

An evaluation and control unit from the comparison of the transmitted measurement signals and the reflected measuring signals determines a correction factor, this rated a predetermined calibrated curve and the measured illumination intensity is corrected, and

With a deviation of the corrected illumination intensity (E) from the prespecified illumination intensity (E_0), the light intensity (I) of the illumination system is automatically increased or decreased. The illumination system is controlled from at least one light sensor or of at least one light sensor and at least one presence detector or controlled. A transmitter emits not perceptible by human senses measurement signals, preferably IR signals.

The transmitter operated with a peak current, and from the change in amplitude of the reflected measurement signal, the correction factor is determined. It is performed with a substantially straight, on a white base is calibrated curve which defines a color-dependent correction factor. It is a distance from the area to be illuminated to the transmitter is carried out / receiver is dependent, calibrated curve which extends substantially straight.

Control ment an illumination system and evaluation by at least one light sensor, at least one transmitter for modulated or continuous measurement signals, at least one receiver for the reflected measurement signals and a control unit for performing the method accordingly. The photodiode or photoresistor of the light sensor is arranged in a direction of a to the illuminating zone open sleeve which is preferably rectangular illuminated to the center of region is aligned.

3.4 Blind and Lighting Control to Maintain Comfort Light Intensity of The Classroom Utilizing Microcontroller ATmega 8535

A stable light intensity level of a classroom can actually be obtained by integrating automatic lamp brightness controller with automatic blind system. A brightness controller of the lamp, especially incandescent type can be developed adopting the principle of phase firing angle. It should be implemented in such a way so that the firing angle is set precisely to keep the classroom's light intensity stays in a steady level. Next, the opening and closing process of the blind can also be done using stepper motor which will be controlled

automatically by a microcontroller through a hardware driver. This is intended to take the advantage of light contribution from the outside instead of directly switching on or brightening the lamp.

Phase firing angle is a time proportioning method to control the amount of power that goes into a load (either resistive load or inductive load) by cycle basis. By varying the portion of firing delay time in each cycle of the alternating current wave will determine the total voltage being delivered into the load. This means the shorter the firing delay, the more the power is delivered into the load and vice versa. The device has successfully operated to keep the light intensity in a constant level. In the normal mode, if the light intensity drops below the predetermined set point then the stepper motor opens the blind while the lamp stays OFF. If the blind has already opened and the light intensity still below the set point, the lamp will ON and get brighter until the set point is reached. On the other hand, if light intensity is greater than the predetermined set point, the lamp will be dimmed until the set point is reached (in this case the lamp might be completely OFF). If the light intensity is still greater than the set point and the lamp is OFF, then the stepper motor will close the blind.

CHAPTER 4

Stimulation and work done

Program 1: Blink

```
// Pin 13 has an LED connected on most Arduino boards.

// give it a name:

int led = 13;

// the setup routine runs once when you press reset:

void setup() {

  // initialize the digital pin as an output.

  pinMode(led, OUTPUT);

}

// the loop routine runs over and over again forever:

void loop() {

  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

  delay(1000); // wait for a second

  digitalWrite(led, LOW); // turn the LED off by making the voltage LOW

  delay(1000); // wait for a second

}
```

In this example a LED is repeatedly turned ON and OFF for one second each. Here a simple circuit is implemented on a bread board and is connected with the Intel Galileo Gen 2 board which is further connected to the power supply by the adapter and also with a micro USB cable that helps us to transfer the code serially to the board.

Output:

Scenario 1:

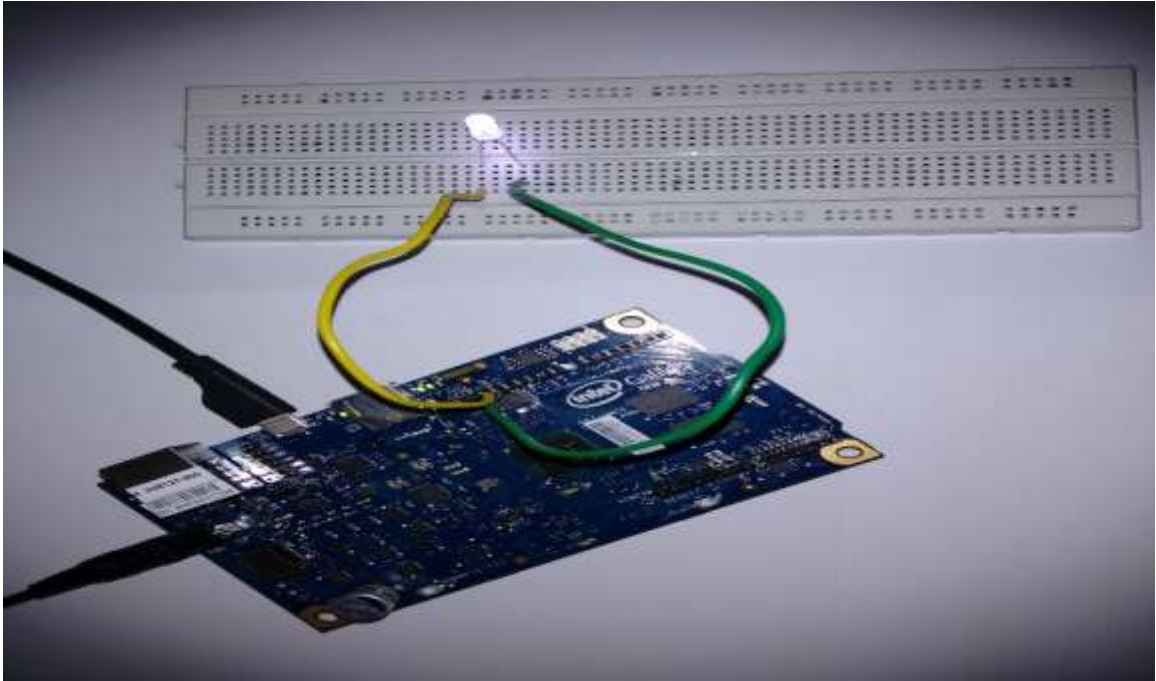


Figure 4: Output when pin is HIGH

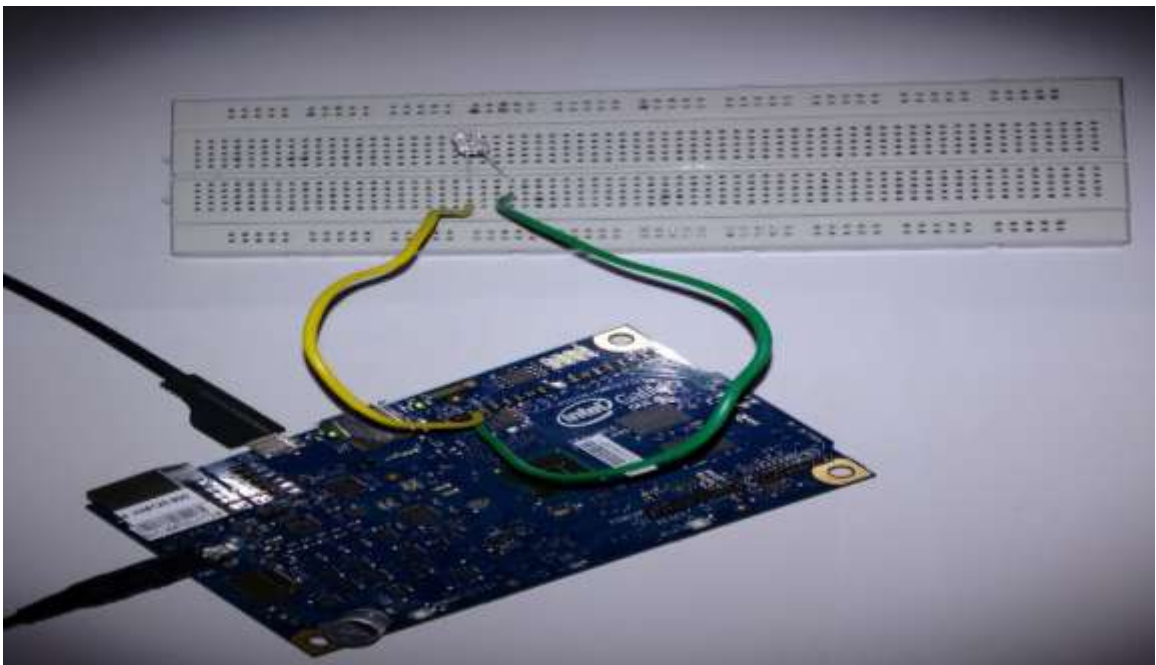


Figure 5: Output when pin is LOW

Program 2: Photo sensing circuit

```
int ldr=0;

int value=0;

int sens=400;

void setup()

{ Serial.begin(9600);

  pinMode(13,OUTPUT);

}

void loop()

{

  value= analogRead(ldr);

  Serial.println(value);

  delay(500);

  if(value<sens)

  {

    digitalWrite(13,HIGH);

  }

  else if(value>sens)

  {

    digitalWrite(13,LOW);

  }

}
```

In this program we have included a photo sensor with a LED and resistor and has mounted it on the bread board. Here the sensor will measure the light intensity and display it on the serial monitor which has been initiated by the command

```
Serial.begin(9600);
```

The reading being measured by the console is being recorded into a variable called value also a variable called sens will have the threshold value stored in it. This threshold value will redirect a signal to the LED that whether it is to be turned ON or OFF, to turn ON

```
digitalWrite(13,HIGH);
```

Is used and to turn OFF the LED we use LOW in place of HIGH.

Output:

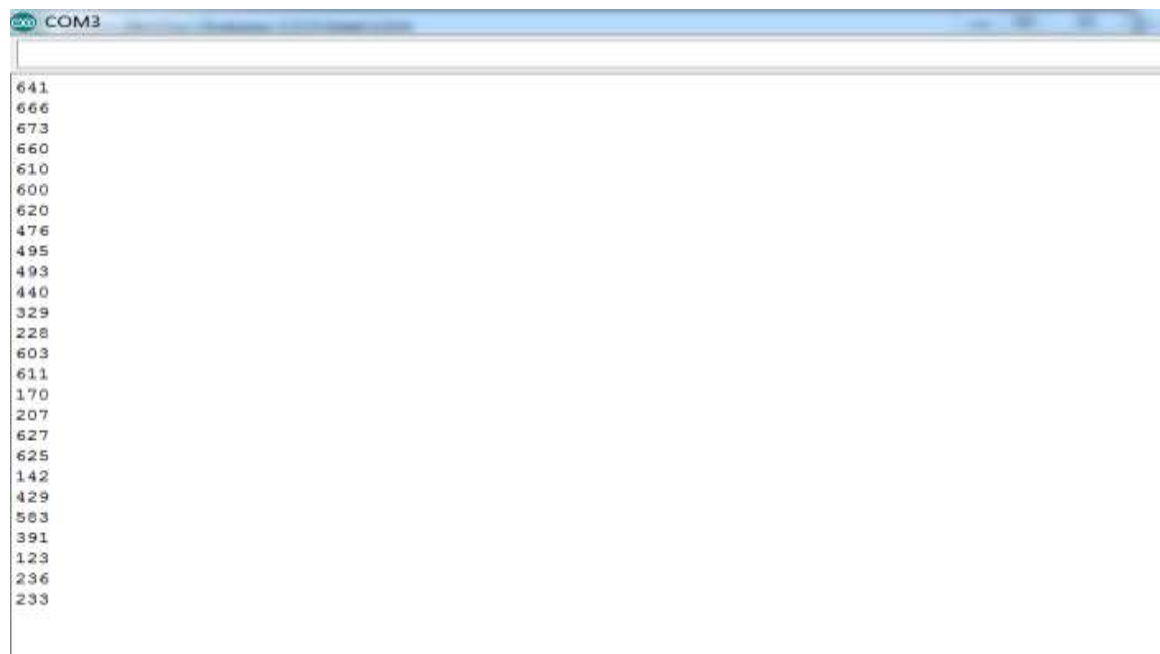


Figure 6: Output on Serial Monitor

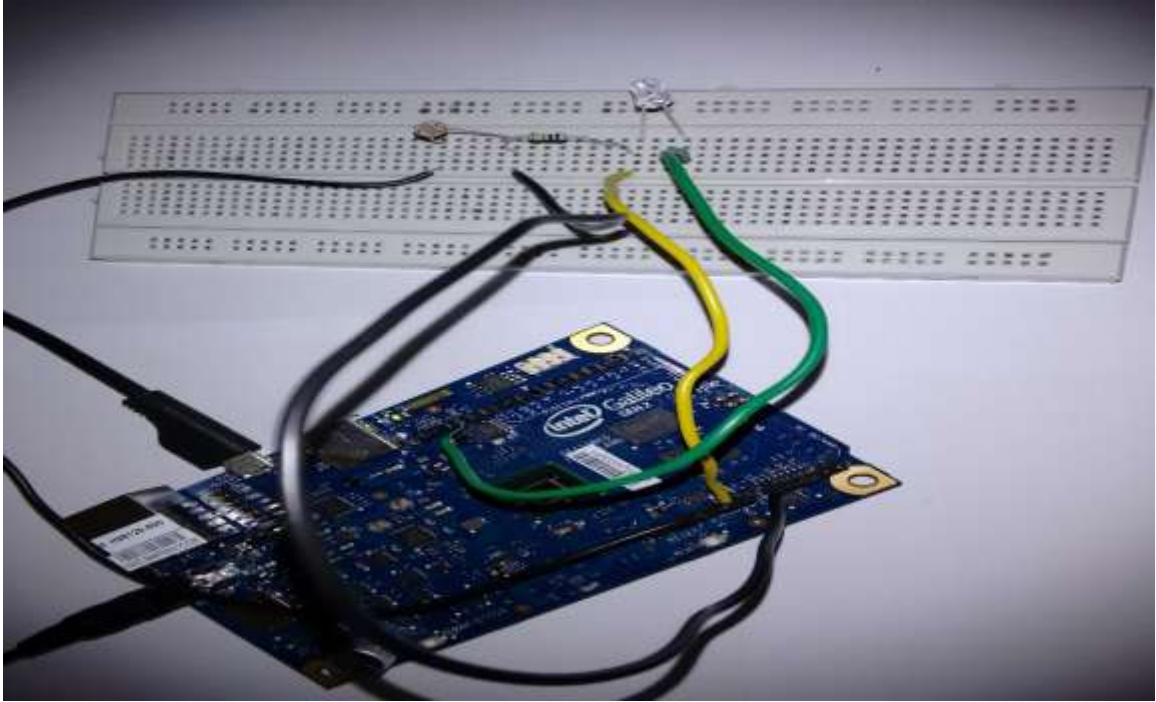


Figure 7: When value of sens > 400

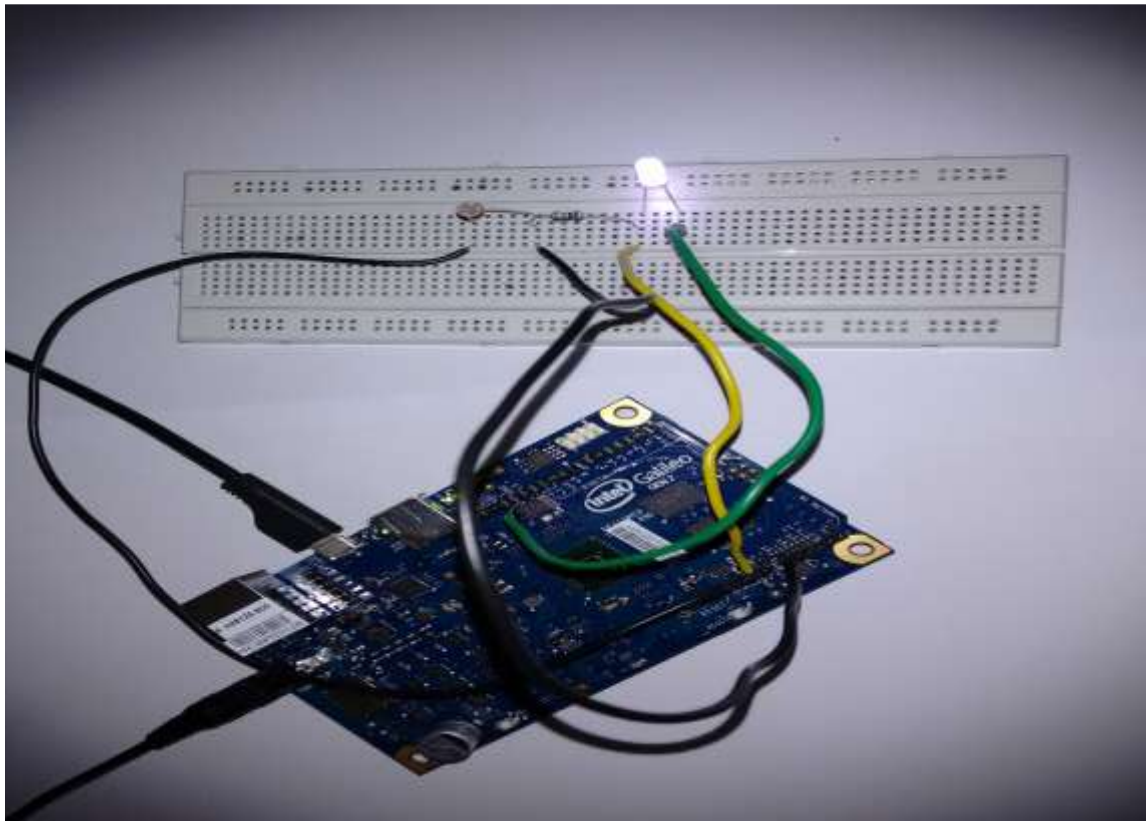


Figure 8: When value of sens < 400

CHAPTER 5

PROJECT DESIGN

5.1 Project Introduction

In our day to day activities we are introduced to many intelligent to many intelligent systems and one of them would be this i.e. a system capable enough to monitor the light intensity at that place and there after controlling the brightness of the lamps installed there in order to attain low power consumption and efficient energy. It will help the user to get low energy bills as will limit the never ending bad habit of wasting resources.

5.2 Hardware Requirements

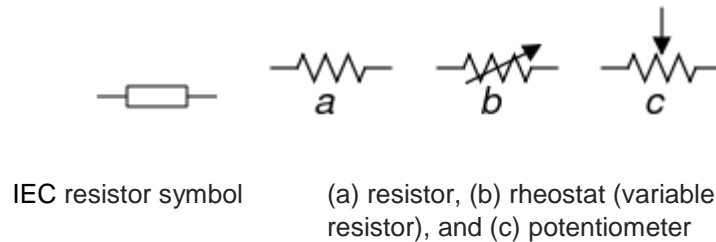
The major hardware requirements required for the project are:

- I. 2nd Gen Intel Galileo board
- II. Resistors
- III. Photo Sensors
- IV. Infrared sensors
- V. Light Emitting Diode (LED)

2nd Gen Intel Galileo board: This board is a processor board provides a single board controller for maker community, students and professional developers. Based on the Intel Quark SoC X1000, a 32-bit Intel Pentium processor- class system on a chip (SoC). All the programs designed and implemented for Arduino UNO board with the microcontroller ATMEGA328P can be used for the implementation on this board. [For further specifications of the board please refer to the chapter 1.2 Introduction to 2nd Gen Intel Galileo board]

Resistors: A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. Resistors act to reduce current flow, and, at the same time, act to lower voltage levels within circuits. In electronic circuits resistors are used to limit current flow, to adjust signal levels, bias active elements, terminate transmission lines among other uses. High-power resistors that can dissipate

many watts of electrical power as heat may be used as part of motor controls, in power distribution systems, or as test loads for generators. Resistors can have fixed resistances that only change slightly with temperature, time or operating voltage. The electronic symbol and notation for resistor is



The behavior of an ideal resistor is dictated by the relationship specified by Ohm's law:

$$V = I.R$$

The unit for resistance is ohm. At any instant of time, the power P consumed by a resistor of resistance R (ohms) is calculated as:

$$P = I^2 R = IV = (V^2/R)$$

Where V (volts) is the voltage across the resistor and I (amps) is the current flowing through it. Using Ohm's law, the two other forms can be derived. This power is converted into heat which must be dissipated by the resistor's package before its temperature rises excessively. Practical resistors have a series inductance and a small parallel capacitance, these specifications can be important in high-frequency applications. In a low-noise amplifier or pre-amp, the noise characteristics of a resistor may be an issue.

We will be using carbon composition resistors which come under the fixed resistors. These resistors consist of a solid cylindrical resistive element with embedded wire leads or metal end caps to which the lead wires are attached. The body of the resistor is protected with paint or plastic. The resistive element is made from a mixture of finely ground (powdered) carbon and an insulating material (usually ceramic). A resin holds the mixture together. The resistance is determined by the ratio of the fill material (the powdered ceramic) to the carbon.



The value of the resistance can be determined by using the following figure:

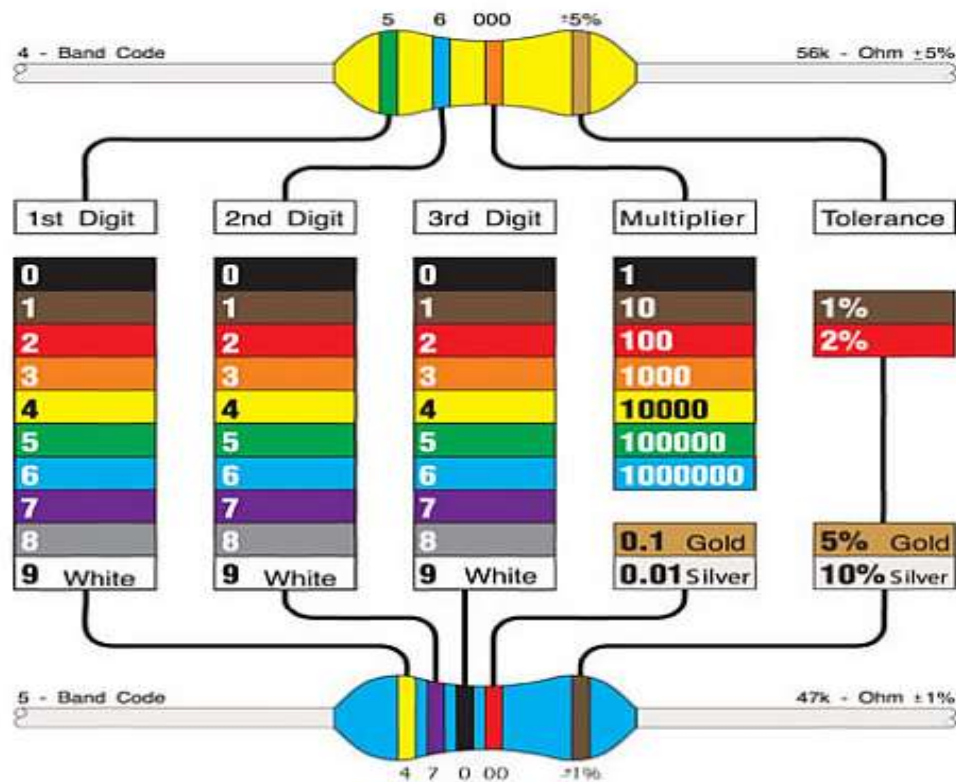


Figure 9: Colour Coding

Photo Sensors: The Light Sensor is a passive devices that convert this “light energy” whether visible or in the infra-red parts of the spectrum into an electrical signal output. Light sensors are more commonly known as “Photoelectric Devices” or “Photo Sensors” because the convert light energy (photons) into electricity (electrons).

Photoelectric devices can be grouped into two main categories, those which generate electricity when illuminated, such as Photo-voltaic or Photo-emissive, etc., and those which change their electrical properties in some way such as Photo-resistors or Photo-conductors. This leads to the following classification of devices.

- I. *Photo-emissive Cells* – These are photo devices which release free electrons from a light sensitive material such as caesium when struck by a photon of sufficient energy. The amount of energy the photons have depends on the frequency of the light and the higher the frequency, the more energy the photons have converting light energy into electrical energy.
- II. *Photo-conductive Cells* – These photo devices vary their electrical resistance when subjected to light. Photoconductivity results from light hitting a semiconductor material which controls the current flow through it. Thus, more light increase the current for a given applied voltage. The most common photoconductive material is Cadmium Sulphide used in LDR photocells.
- III. *Photo-voltaic Cells* – These photo devices generate an emf in proportion to the radiant light energy received and is similar in effect to photoconductivity. Light energy falls on to two semiconductor materials sandwiched together creating a voltage of approximately 0.5V. The most common photovoltaic material is Selenium used in solar cells.
- IV. *Photo-junction Devices* – These photodevices are mainly true semiconductor devices such as the photodiode or phototransistor which use light to control the flow of electrons and holes across their PN-junction. Photojunction devices are specifically designed for detector application and light penetration with their spectral response tuned to the wavelength of incident light.

Photoconductive light sensor does not produce electricity but simply changes its physical properties when subjected to light energy. The most common type of photoconductive device is the Photo resistor which changes its electrical resistance in response to changes in the light intensity. resistors are Semiconductor devices that use light energy to control

the flow of electrons, and hence the current flowing through them. The commonly used Photoconductive Cell is called the Light Dependent Resistor or LDR.

Light Dependent Resistor (LDR) is made from a piece of exposed semiconductor material such as cadmium sulphide that changes its electrical resistance from several thousand Ohms in the dark to only a few hundred Ohms when light falls upon it by creating hole-electron pairs in the material. The net effect is an improvement in its conductivity with a decrease in resistance for an increase in illumination. Also, photo resistive cells have a long response time requiring many seconds to respond to a change in the light intensity.



Figure 10: ORP12 Cadmium Sulphide photoconductive cell

Materials used as the semiconductor substrate include, lead sulphide (PbS), lead selenide (PbSe), indium antimonide (InSb) which detect light in the infra-red range with the most commonly used of all photo resistive light sensors being Cadmium Sulphide (CdS). Cadmium sulphide is used in the manufacture of photoconductive cells because its spectral response curve closely matches that of the human eye and can even be controlled using a simple torch as a light source. Typically then, it has a peak sensitivity wavelength (λ_p) of about 560nm to 600nm in the visible spectral range.

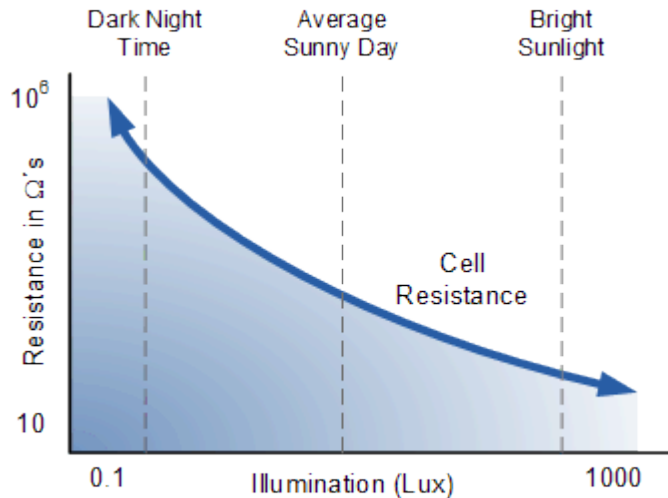
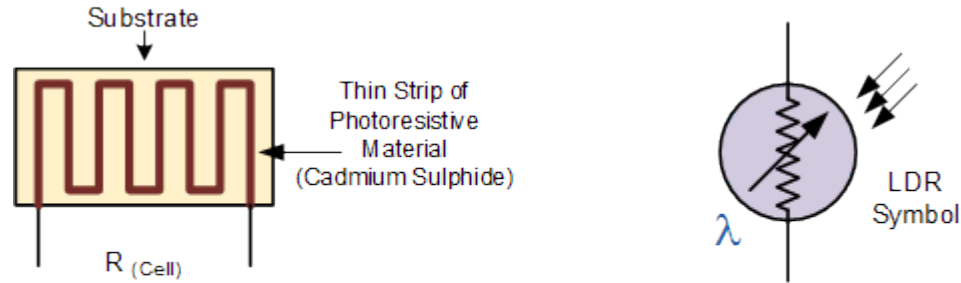


Figure 11: Photo Sensor

The most commonly used photoresistive light sensor is the **ORP12** Cadmium Sulphide photoconductive cell. This light dependent resistor has a spectral response of about 610nm in the yellow to orange region of light. The resistance of the cell when unilluminated (dark resistance) is very high at about 10MΩ's which falls to about 100Ω's when fully illuminated (lit resistance).

To increase the dark resistance and therefore reduce the dark current, the resistive path forms a zigzag pattern across the ceramic substrate. The CdS photocell is a very low cost device often used in auto dimming, darkness or twilight detection for turning the street lights “ON” and “OFF”, and for photographic exposure meter type applications.

Infrared sensors: A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. All objects with a temperature above absolute zero emit heat energy in the form of radiation. Usually this radiation is invisible to

the human eye because it radiates at infrared wavelengths, but it can be detected by electronic devices designed for such a purpose.

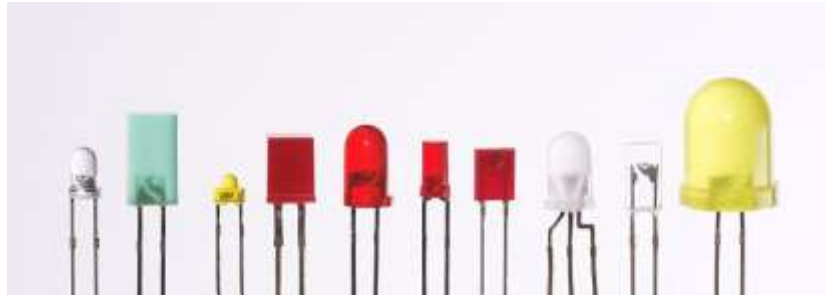
The term passive in this instance refers to the fact that PIR devices do not generate or radiate any energy for detection purposes. They work entirely by detecting the energy given off by other objects. PIR sensors don't detect or measure "heat"; instead they detect the infrared radiation emitted or reflected from an object. Infrared radiation enters through the front of the sensor, known as the 'sensor face'. At the core of a PIR sensor is a solid state sensor or set of sensors, made from pyro electric materials—materials which generate energy when exposed to heat.

LED: A light-emitting diode (LED) is a two-lead semiconductor light source. It is a basic pn-junction diode, which emits light when activated. When a suitable voltage is applied to the leads, electrons are able to recombine with electron holes within the device, releasing energy in the form of photons. This effect is called electroluminescence, and the color of the light (corresponding to the energy of the photon) is determined by the energy band gap of the semiconductor. Infrared LEDs are still frequently used as transmitting elements in remote-control circuits, such as those in remote controls for a wide variety of consumer electronics. The first visible-light LEDs were also of low intensity, and limited to red. Modern LEDs are available across the visible, ultraviolet, and infrared wavelengths, with very high brightness.

LEDs have many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Light-emitting diodes are now used in applications as diverse as aviation lighting, automotive headlamps, advertising, general lighting, traffic signals, and camera flashes. However, LEDs powerful enough for room lighting are still relatively expensive, and require more precise current and heat management than compact fluorescent lamp sources of comparable output.

The first commercial LEDs were commonly used as replacements for incandescent and neon indicator lamps, and in seven-segment displays, first in expensive equipment such as laboratory and electronics test equipment, then later in such appliances as TVs, radios, telephones, calculators, as well as watches As LED materials

technology grew more advanced, light output rose, while maintaining efficiency and reliability at acceptable levels. One of the key advantages of LED-based lighting sources is high luminous efficacy. White LEDs quickly matched and overtook the efficacy of standard incandescent lighting systems.



Red LEDs were notable for their short service life. With the development of high-power LEDs the devices are subjected to higher junction temperatures and higher current densities than traditional devices. LED technology may be a good replacement in uses such as supermarket freezer lighting and will last longer than other technologies it is because LEDs emit less heat than incandescent bulbs, they are an energy-efficient technology for uses such as in freezers and refrigerators. However, because they emit little heat, ice and snow may build up on the LED light fixture in colder climates.

5.3 Software Requirements

This project needs an Arduino SDK (version 1.5.3) as the SDK for the programming of the 2nd Gen Intel Galileo board. This version of Arduino is specially introduced for the board. 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*. [For further specifications of the software please refer to the chapter 2.1 Introduction to Arduino SDK]

5.4 Block Diagram

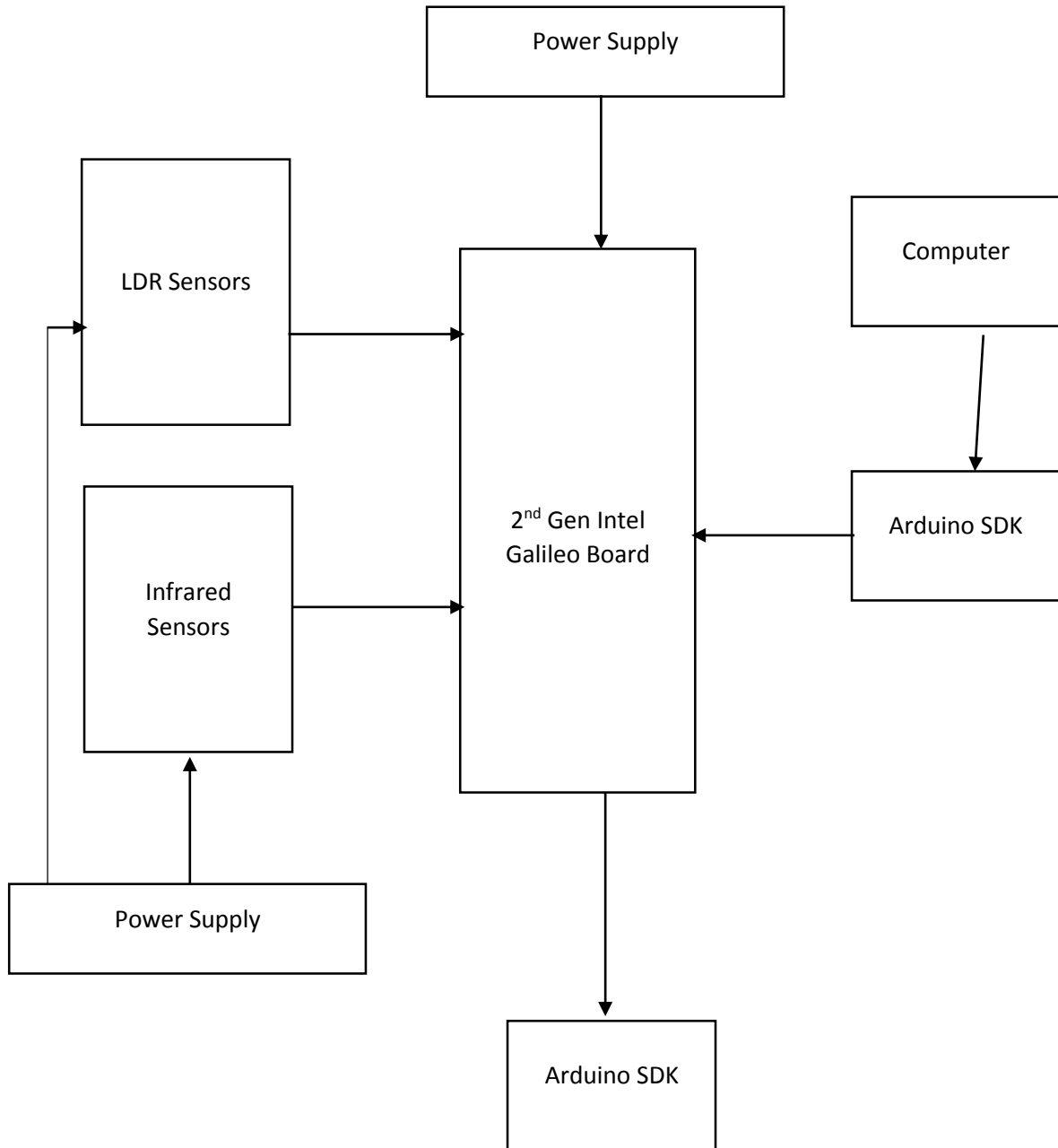


Figure 12: Block Diagram

5.5 Circuit Diagram

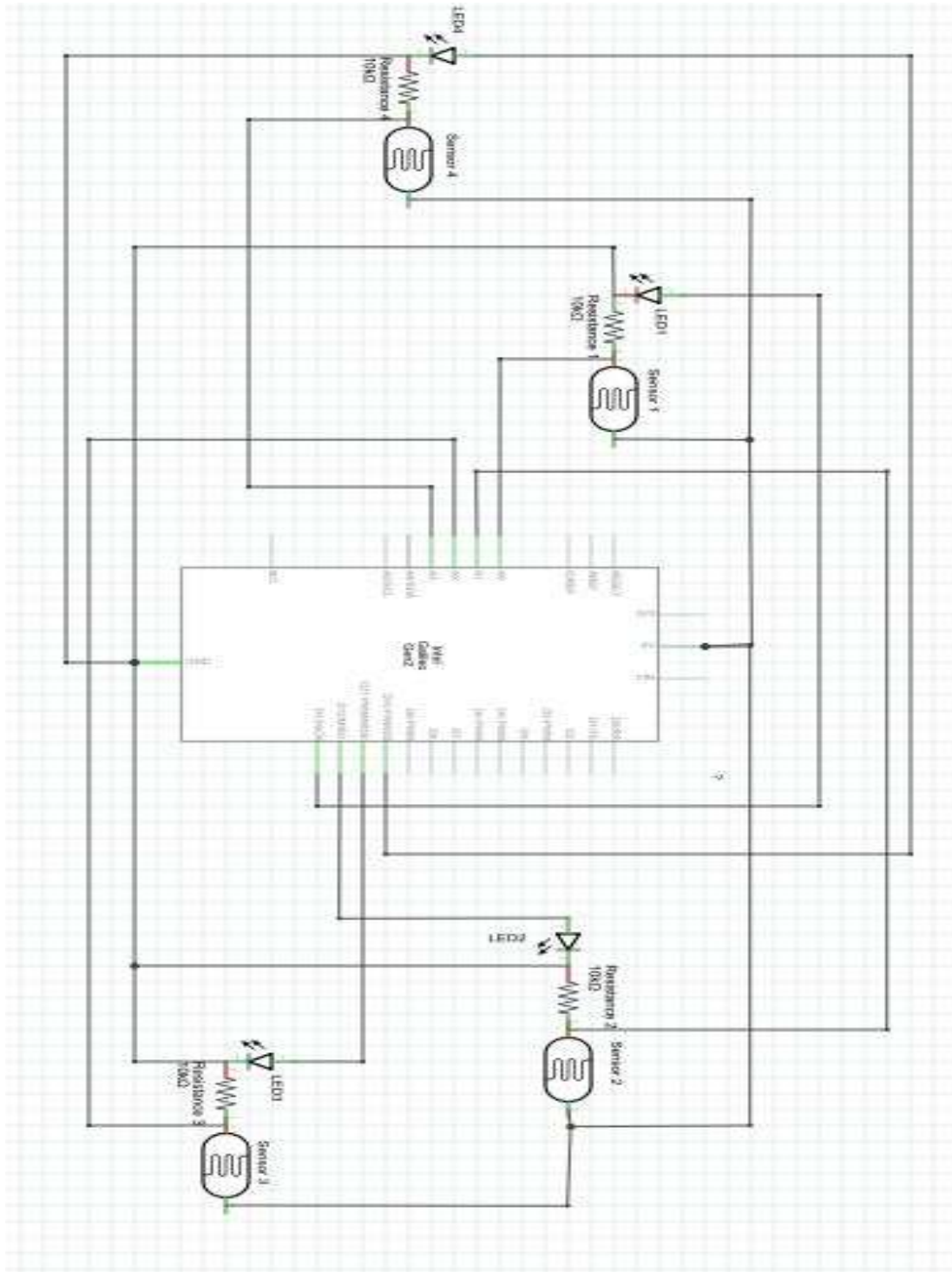


Figure 13: Schematic Circuit Diagram

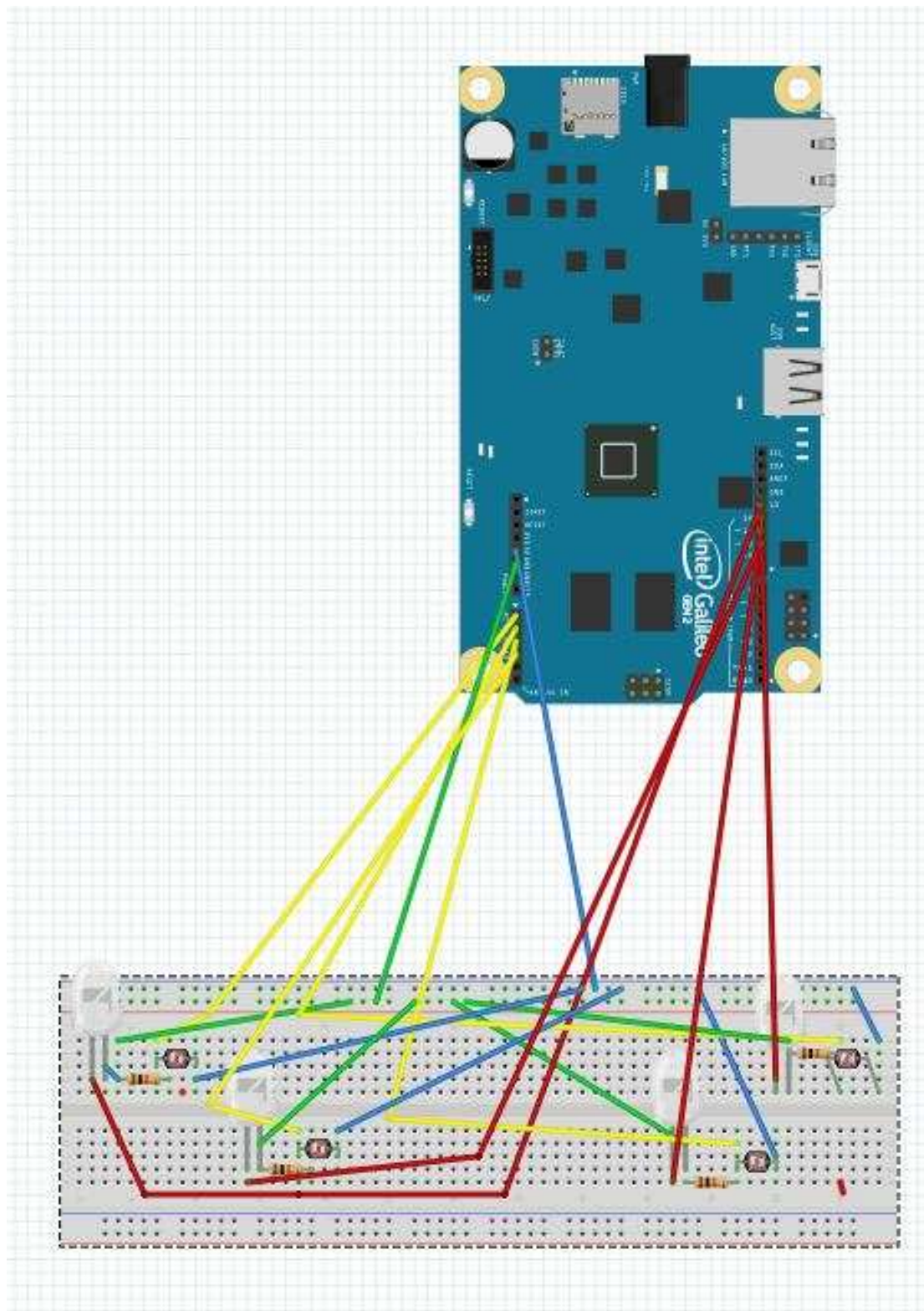


Figure 14: Breadboard Circuit Diagram

5.6 Implementation

While implementing this project we have considered a scenario of a work place say a office where the model will have 4 pillars where the sensors have been fixed at the four corners of the room. Here considering a general scenario that light can enter from any direction I have kept the sensors in open so that all the possible cases can be checked. In the following figure prototype is connected to the 2nd Gen Intel Galileo board and the red circles signify the photo sensors.

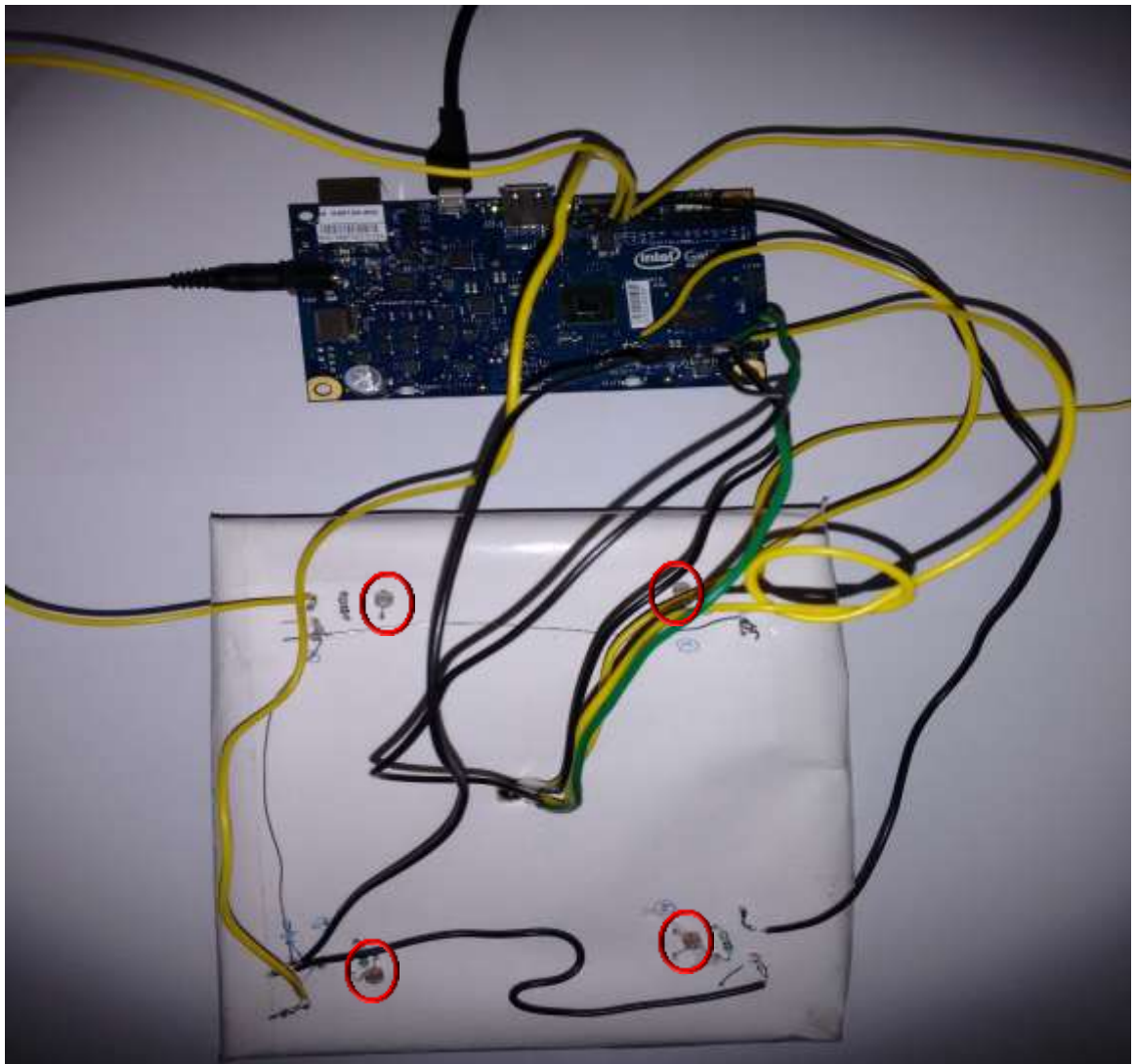


Figure 15: Connection of board with the prototype

Scenario 1: All the LED's are OFF

In this scenario as we can see in the figure that none of the LED's is working. It is because all the four sensors are receiving enough light intensity that can cross the threshold value been initialized in the program. Hence it depicts that the processor senses that there is no need for it to turn ON any of the LED's.

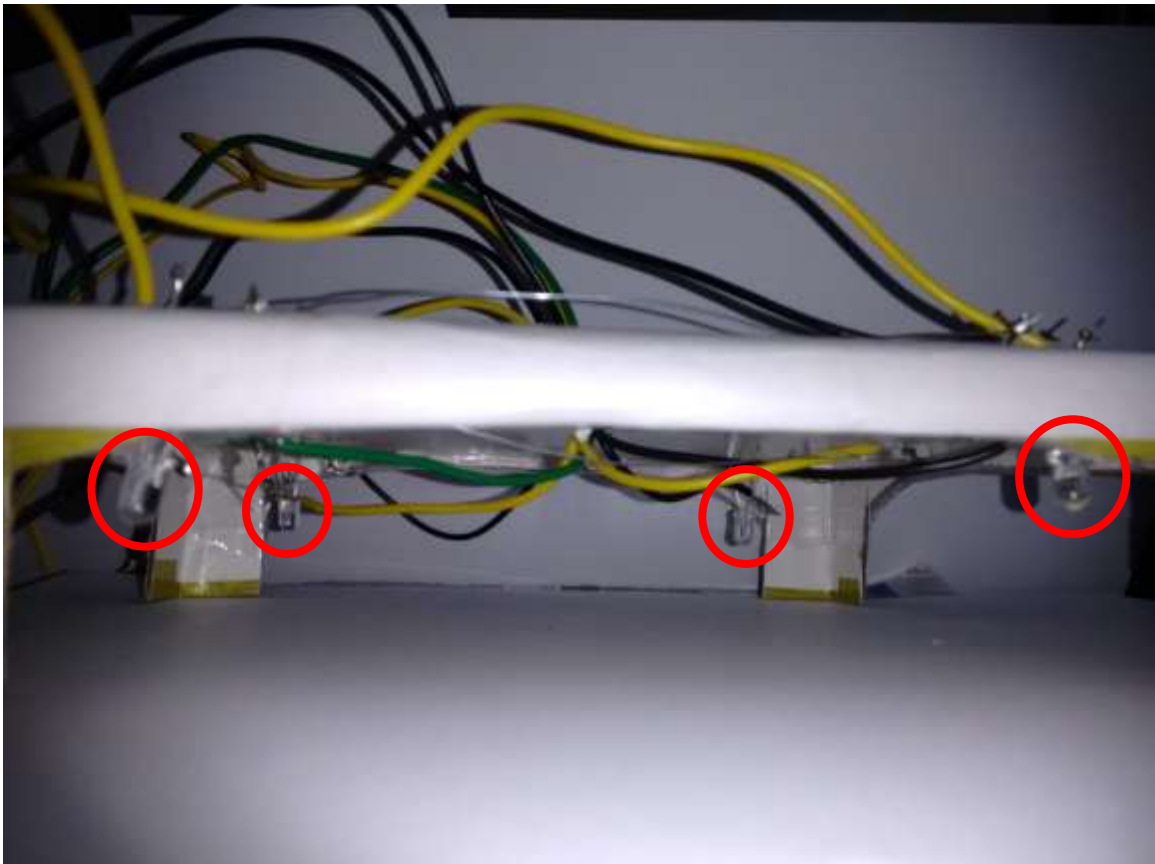


Figure 16: Scenario 1

Here red shows the LED's turned OFF.

Scenario 2: Three of the four LED's are OFF

In this scenario as we can see in the figure that only one of the four LED's is working. It is because the other three sensors are receiving enough light intensity that can cross the threshold value been initialized in the program. Hence it depicts that the processor senses that there is no need for it to turn ON those three LED's. Whereas the fourth sensor does not receive enough light intensity that it can cross the threshold value been set in the

program, hence the controller sends a signal to turn ON the LED connected to that particular sensor.

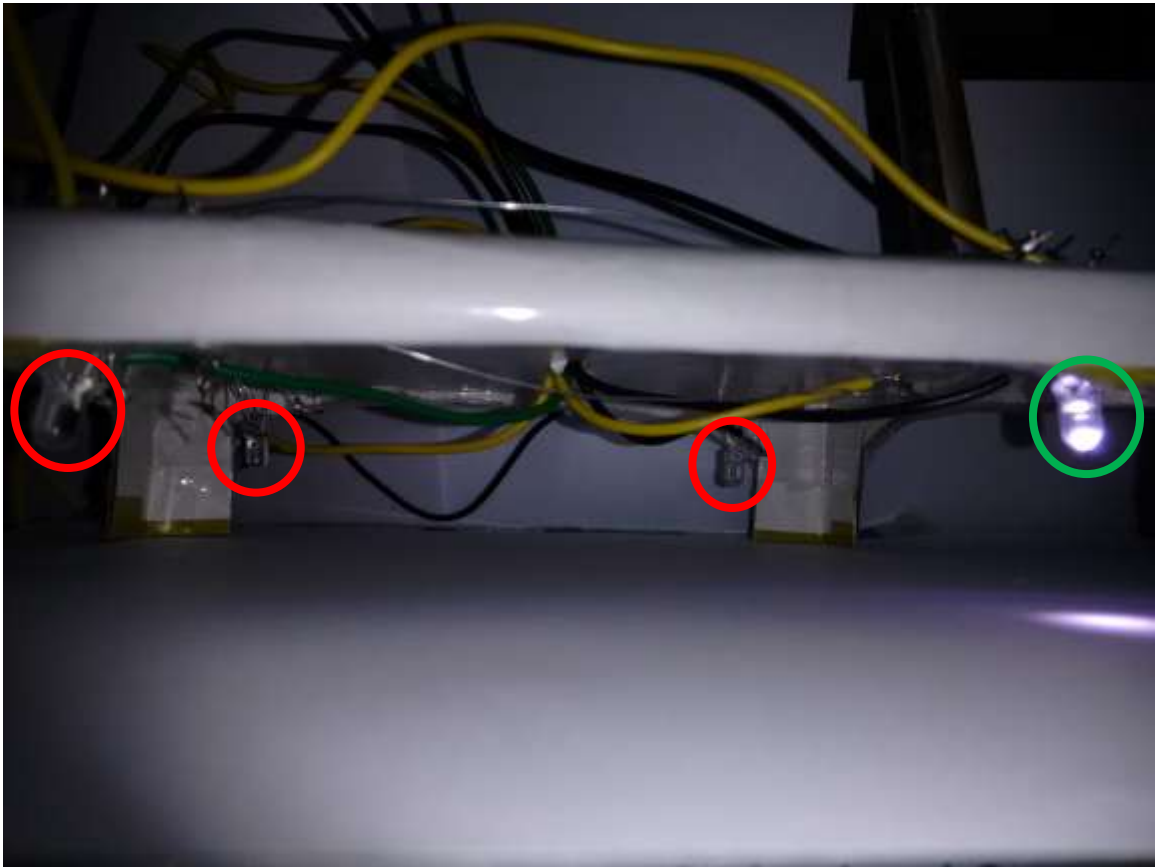


Figure 17: Scenario 2

In this the green one shows the LED that has been turned ON whereas the red one's show the LED which are still OFF.

Scenario 3: Two of the four LED's are OFF

In this scenario as we can see in the figure that only two of the four LED's are working. It is because two sensors are receiving enough light intensity that can cross the threshold value been initialized in the program. Hence it depicts that the processor senses that there is no need for it to turn ON those two LED's. Whereas the other two sensors does not receive enough light intensity that it can cross the threshold value been set in the

program, hence the controller sends a signal to turn ON the LED connected to those particular sensors.

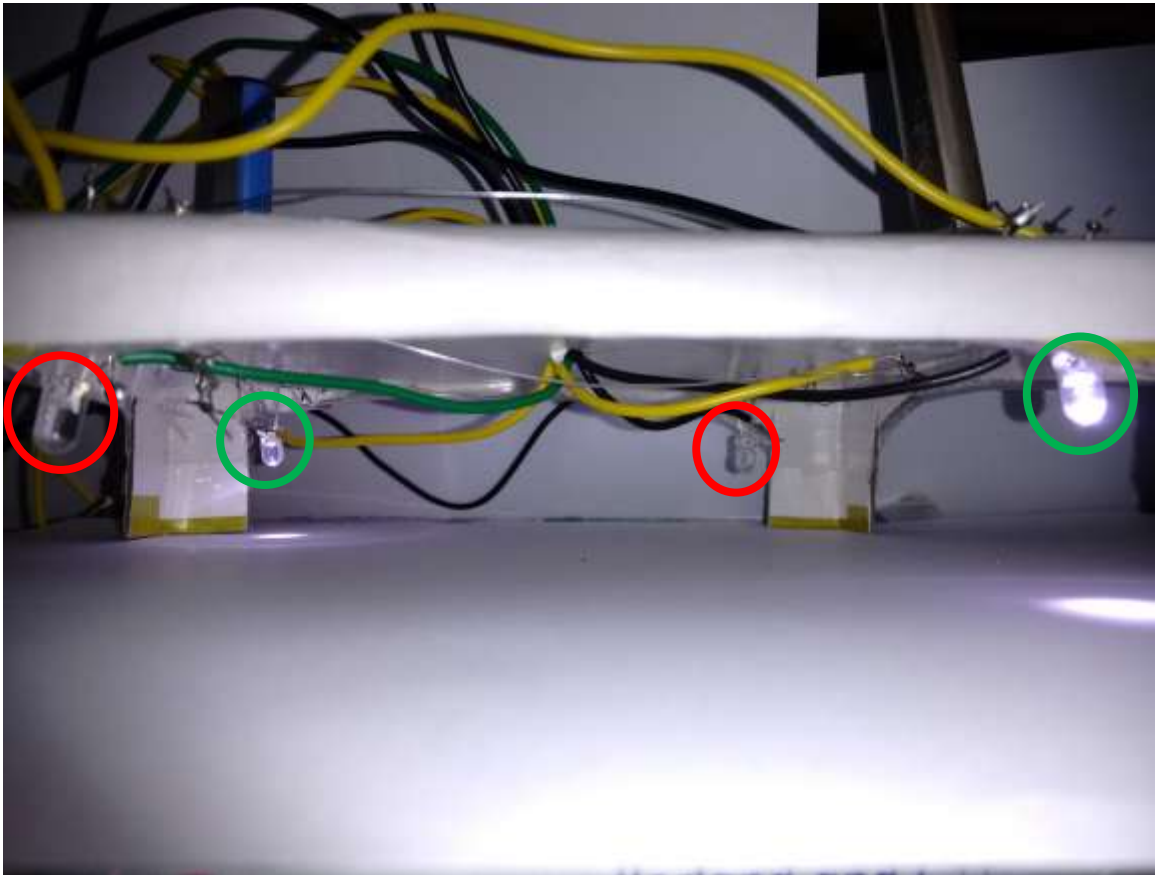


Figure 18: Scenario 3

In this the green one shows the LED's that has been turned ON whereas the red one's show the LED which are still OFF.

Scenario 4: Three of the four LED's are ON

In this scenario as we can see in the figure that only one of the four LED's are not working. It is because the three sensors are not receiving enough light intensity that can cross the threshold value been initialized in the program. Hence it depicts that the processor senses that there is a need for it to turn ON those two LED's. Whereas the other the forth sensors does receive enough light intensity that it can cross the threshold value been set in the program, hence the controller sends a signal to turn OFF the LED connected to those particular sensors.

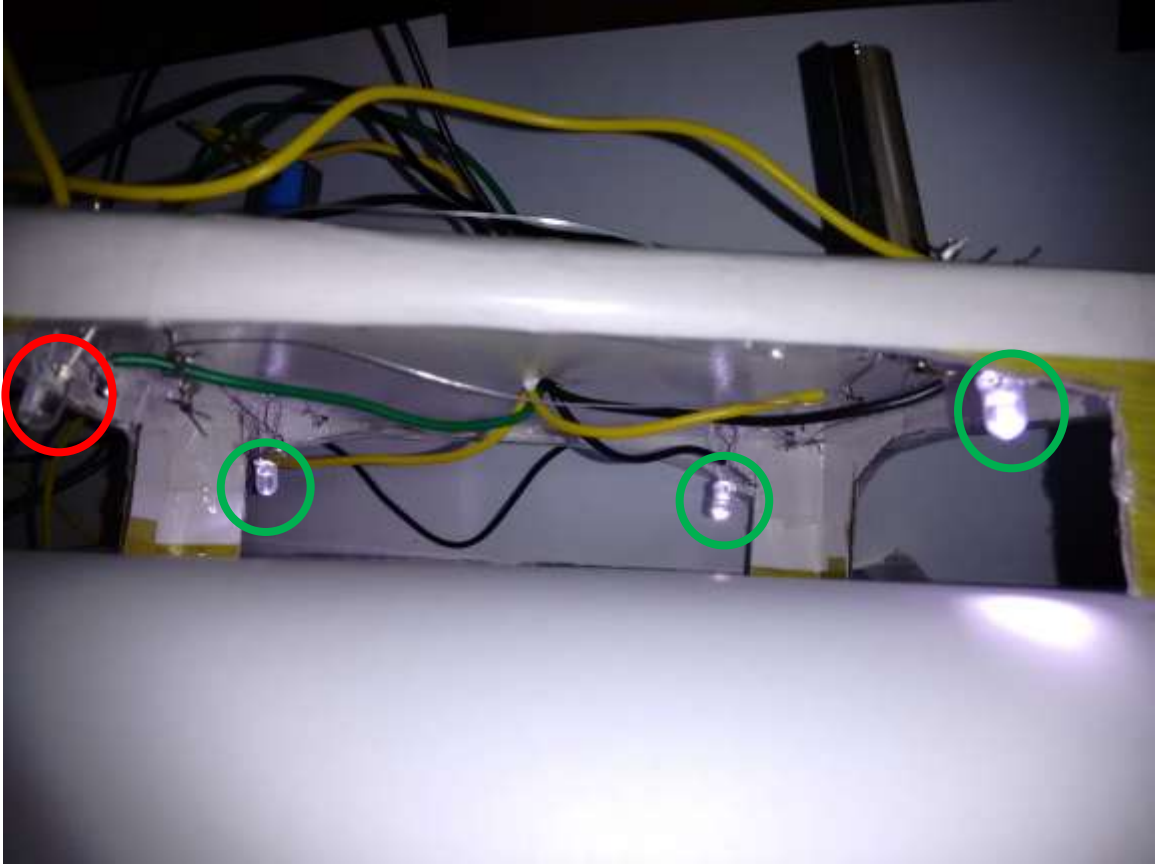


Figure 19: Scenario 4

In this the green one shows the LED's that has been turned ON whereas the red one how the LED which are still OFF.

Scenario 5: All the LED's are ON

In this scenario as we can see in the figure that all of the LED's are working. It is because all the four sensors are not receiving enough light intensity that can cross the threshold value been initialized in the program. Hence it depicts that the processor senses that there is a need for it to turn ON all the LED's.



Figure 20: Scenario 5

In this the green one shows the LED's that has been turned ON.

CHAPTER 6

Conclusion and Future Work

The prototype building part of the overall problem has come to a mid-point, where I have included the major part which is desired by the audience round the globe i.e. saving the energy by turning ON and OFF of the lights. Now I have to concentrate on coming up with an algorithm and to develop a program to be burned on the processor in order to calculate power efficiency and power consumption being provided by the help of the processor, also to send the sensed data to the server and fetch the schedule of the workplace in order to work more efficiently.

The final project will be well equipped with the features like monitoring light at a workplace and then controlling the brightness of the lights installed according to the preference of the user i.e. the installed light will turn off when the light intensity in the room is sufficient enough to carry out work, installed light will turn on with full brightness when the light intensity in the room is not sufficient enough to carry out work else will be on but with variable brightness depending on the intensity sensed by the sensor. The project will also be capable enough to calculate the power consumption and efficiency which will be done by fetching from and sending the sensed data to the server.

References

- [1] M. H. Rashid, Power Electronics Handbook (3rd edition). Oxford. UK: Elsevier. 2011
- [2] Thyristor Theory and Design Considerations. On Semiconductor. 2006.
- [3] T. Igoe, and D. O'Sullivan, Physical Computing: Sensing and Controlling the Physical World with Computers. Boston, MA: Thomson Course Technology PTR, 2004
- [4] A. A. Birjandi, and Z. Ameli, "Three phase controlled rectifier study in terms of firing angle variations", ACEEE International Journal on Electrical and Power Engineering, Vol. 02, No. 02, May 2012, pp. 25-28.
- [5] F Swainston. A Systems Approach to Programmable Controllers. Albany, New York: Delmar Publishers Inc., 1992, pp. 23
- [6] US4139801: On Jan 26, 1977, Linares Raul F, Automatic automobile light control system, Feb 13, 1979.
- [7] US4461977: On Dec 1, 1982, The United States of America As Represented by the Secretary Of The Navy, Single Zone Lightning Controller, Jul 24, 1984.

Useful Links

1. <http://home.howstuffworks.com/smart-home3.htm>
2. http://www.electronics-tutorials.ws/io/io_4.html
3. <http://www.fritzing.org>
4. <http://arduino.cc/>
5. <http://www.intel.com/content/www/us/en/do-it-yourself/galileo-maker-quark-board.html>
6. <http://www.atmel.com/devices/atmega328p.aspx>