زانکۆی سەلاحەدین - هەولێر
Salahaddin University-Erbil

Arduino-Powered ON/OFF Cycle Calculator For CFL Light Bulbs

Research project

Submitted to the Department of (Electrical Engineering), College of Engineering Salahaddin University in Partial Fulfillment of the Requirements for the degree of (BSc.)

By:

Sara Fuad Abdullah

Avan Tarq Akram

Supervised by:

Dr. Ahmad Khalid

February - 2024

ACKNOWLEDGEMENT

First of all, we give thanks to Allah or God. Then we would like to take this opportunity to express our appreciation and gratitude to our project and thesis supervisor Mr. Ahmad for being dedicated in supporting, motivating and guiding us through this project. This project can't be done without his useful advice and helps. Also thank you very much for giving us opportunity to choose this project. Apart from that, we would like to thank our entire friends for sharing knowledge; information and helping us in making this project a success. Also thanks for lending us some tools and equipment. To our beloved family, we want to give them our deepest love and gratitude for being very supportive and also for their ins Passive Infraredaction and encouragement during our studies in this University.

Table of Contents

Table of figures

Abstract

In this project, we want to know the life cycle of CFL light bulbs by using the Arduino to control the ON/OFF cycles of the light bulb and then recording the number of cycles the Arduino IDE program, a cycle counter can be constructed to calculate the lifespan of a light bulb. It is possible to estimate the total hours of operation of a light bulb by dividing the recorded number of ON/OFF cycles by the average daily usage of the light bulb and then by the average lifespan of the light bulb in hours. In addition, we will discuss how the Arduino microcontroller can be used to conduct research and studies. A microcontroller such as an Arduino can be used to develop small projects involving sensors in a relatively short time. The Arduino microcontroller is an easy-to-learn and easy-to-program device. The Arduino IDE is capable of programming Arduino microcontrollers. A program for Arduino boards can be written using the Arduino IDE. The Arduino IDE is an open-source program that can be downloaded and installed on a computer for free. The Arduino IDE provides several libraries that are ready for use. Arduino developers can save a great deal of time by using these libraries.

Chapter one

1.1 introduction

The project aims to create an automatic ON/OFF cycle calculator for the defined running time of the Compact Fluorescent Lamp (CFL) load using an Arduino development board. In our routine life, we cannot know the running time of any load we are using more specifically when the load consists of a CFL lamp. However, this project aims to calculate the number of times the ON/OFF operation of the load can be used to define ON and OFF time electronically, to operate the load without using the mechanical switch. So, we can define this project as a useful and effective way of using the number of operations of any load by switching it manually ON/OFF. This would consist of a power-saving project. The ON period of the load will be powered through the relay (used as an electronic switch) whereas the Arduino board will control the relay to define the ON and OFF period. We can use a real-time clock (RTC) module to keep track of time. Write a program that calculates the total operating hours based on the ON/OFF cycles and then triggers the CFL to turn OFF after a predetermined number of cycles or hours.

1.2 Feasibility study

A feasibility study for an Arduino-powered on/off cycle calculator for CFL light bulbs involves assessing various aspects to determine the practicality and viability of the project. Here are key considerations:

i. Technical Feasibility:

-**Arduino compatibility**: Verify that Arduino is suitable for interfacing with sensors, relay, and display modules required for cycle calculation.

-**Sensor Accuracy**: refers to how accurately the light sensor can detect the ON/OFF status of the light bulb [13].

-**Data Storage**: Confirm that the chosen method for storing cycle data meets the project requirements.

ii. Economic Feasibility:

-Cost Analysis: Estimate the overall cost of components, including Arduino board, sensor, display, and power supply, to ensure it aligns with the budget.

-Long-Term Costs: consider maintenance and potential future upgrades, if any.

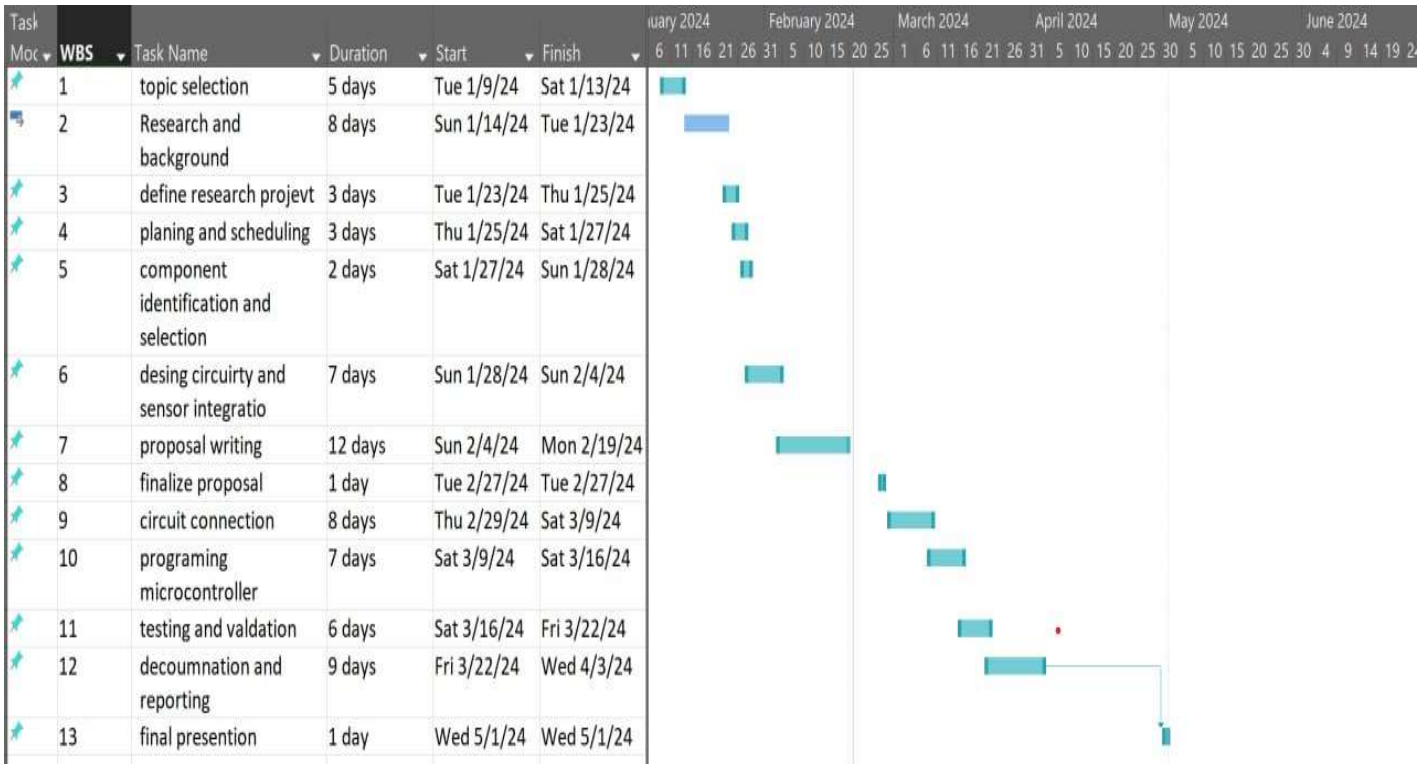
iii. Cultural Feasibility:

A cultural feasibility study for an Arduino-powered on/off cycle calculator for CFL light bulbs would involve assessing whether the target culture is receptive to and compatible with such a technology. Factors to consider include the community's attitudes toward technology, energy conservation practices, and willingness to adopt new devices. Understanding cultural norms and preferences will help in tailoring the product to meet the needs and acceptance levels of the specific audience.

1.3 Task Table

Topic selection	2024-01-09	2024-1-13	5 days
Research and background	2024-01-14	2024-01-23	8 days
Define research project	2024-01-23	2024-01-25	3 days
Planning and scheduling	2024-01-25	2024-01-27	3 days
Component identification and selection	2024-01-27	2024-01-28	2 days
Design circuitry and sensor integration	2024-01-28	2024-02-04	7 days
Proposal writing	2024-02-04	2024-02-19	12days
Finalize proposal	2024-02-27	2024-02-27	1 day
Circuit connection	2024-02-29	2024-03-09	8 days
Programming microcontroller	2024-03-09	2024-03-16	7 days
Testing and validation	2024-03-16	2024-03-22	6 days
Documentation and reporting	2024-03-22	2024-04-03	9 days
Final presentation	2024-05	2024-05	1 day

1.4 Gantt chart



Task Mod	WBS	Task Name	Duration	Start	Finish
	1	topic selection	5 days	Tue 1/9/24	Sat 1/13/24
	2	Research and background	8 days	Sun 1/14/24	Tue 1/23/24
	3	define research projevt	3 days	Tue 1/23/24	Thu 1/25/24
	4	planing and scheduling	3 days	Thu 1/25/24	Sat 1/27/24
	5	component identification and selection	2 days	Sat 1/27/24	Sun 1/28/24
	6	desing circuirty and sensor integratio	7 days	Sun 1/28/24	Sun 2/4/24
	7	proposal writing	12 days	Sun 2/4/24	Mon 2/19/24
	8	finalize proposal	1 day	Tue 2/27/24	Tue 2/27/24
	9	circuit connection	8 days	Thu 2/29/24	Sat 3/9/24
	10	programing microcontroller	7 days	Sat 3/9/24	Sat 3/16/24
	11	testing and valdation	6 days	Sat 3/16/24	Fri 3/22/24
	12	decoumnation and reporting	9 days	Fri 3/22/24	Wed 4/3/24
	13	final presention	1 day	Wed 5/1/24	Wed 5/1/24

Figure 1: Gantt Chart

Chapter two

2.1 Literature review:

1) In this research writer mentions that: Reliability prediction includes the identification of failure modes, mechanisms, and causes. So it is apparent that the reliability of any component, device, or system has two components i.e. probabilistic and deterministic. In probabilistic probability of failure and uncertainty in components are estimated and in deterministic, the various models and causes of failures are revealed. These two components make the reliability prediction process more accurate and simple. The reliability prediction process consists following steps: Identification of the life cycle load Electronic system, device, or component. Identification of the failure mode. Testing the electronic system and monitoring. Prediction based on stress and damage mo. Prediction based on handbooks [14].

2) In this research writer mentions that: Life cycle assessment (LCA) is an approach methodology for assessing environmental aspects and potential environmental impacts throughout a product's life cycle. This paper presents and discusses cases where LCA is used for assessing the environmental impact of electronic products and processes. Four cases in electronics illustrate and critically evaluate LCA including: Consumer electronic products, Interconnect technology in electronics micro-integration, Photovoltaic (PV) solar cells, and Electric vehicles. The environmental and resource impacts commonly included in an LCA are climate change, stratospheric ozone depletion, toxicological stress on human health and ecosystems, and the depletion of energy and material resources [15].

3) In this research writer mentions that: To assess the environmental performance of an electricity system, a holistic system perspective is required. If not, the risk is the only environmental impact arising directly from the technology itself is considered, without taking into account indirect upstream and downstream contributions and savings. The main goal of the project is to develop a systematic framework for the environmental assessment of electricity systems. This aims at (i) providing scientifically sound recommendations for decision-making processes, leading towards more sustainable energy systems, and (ii) providing accurate and transparent electricity supply LCA data, thereby increasing the robustness of LCA results for a multimode of products consuming electricity throughout the life cycle [16].

4) In this research writer mentions that: Concerning the short-life problem of electronic application products. And the steps for life estimation are illustrated. First, allowable cumulative failure probability against operating time that customers expect, i.e. target value of life, is set. Next, the electronic components that underperform the target values are picked up without experiment, among those mounted on the electronic circuit. For the components picked up as having a short life, an accelerated life test is performed to estimate the life [17].

5) In this research writer mentions that: Environmental issues are playing an ever-increasing role in the decision-making process at every level: political, economic, industrial, and individual. More than just a passing trend, the increasing attention given to environmental problems stems from a basic observation: because of this limited capacity to absorb the effects of human activities, the environment sets a limit to society's development. This limit has already been reached in many regions of the planet. Even though LCA has many advantages, it is not devoid of shortcomings. Some applications of LCA have been harshly criticized, suggesting that a given LCA method was selected to obtain the results expected by the sponsor of the study [18].

Chapter three

3.1Objective:

The objective of the project is to create an Arduino-powered ON/OFF cycle counter is to create a system that can accurately count the number of times the light bulb is turned ON and OFF, and a calculator for CFL light bulbs using a photo-resistor and a relay. The goal is to measure the ON/OFF cycle of the CFL (Compact Fluorescent Lamp) light bulb to estimate its lifespan. The project involves using an Arduino, a photo-resistor to detect light, and a relay to control the power to the light bulb. The Arduino will count the number of ON/OFF cycles, and this data can be used to estimate the remaining lifespan of the CFL light bulb, CFL bulbs have a limited number of cycles before they blow up. Compact Fluorescent Lamp (CFL) bulbs generally have a longer lifespan than traditional incandescent bulbs, with ratings typically ranging from 6000 to 15000 hours. If CFL bulbs are turned ON and OFF frequently or used in totally enclosed fixtures, under normal conditions, CFL bulbs can provide 70-80% of their original light output after reaching the end of their rated lifetime. Despite having a shorter lifespan compared to LED bulbs, CFL bulbs continue to be a viable option for energy conservation due to their lower cost and high efficiency. The benefits of using an Arduino-powered on/off cycle calculator for CFL light bulbs include the ability to control the timing and duration of the light cycles, which can help in optimizing energy efficiency and extending the lifespan of the CFL bulbs. By implementing a soft start feature, the calculator can reduce the inrush current that can damage the filament in incandescent lights, thus enhancing the longevity of the bulbs. Additionally, the Arduino allows for precise control over the transition between cool and warm light colors, as well as dimming capabilities, providing flexibility in creating different lighting effects. This level of control and customization can be particularly useful in projects like industrial design prototypes or artistic installations where specific lighting effects are desired.

3.2 Problem statement:

Design an Arduino-based system that monitors and calculates the on/off cycles of CFL (Compact Fluorescent lamp) light bulbs. The system should track the duration of each on/off period and provide a user-friendly interface to display the accumulated cycles. The calculator should be able to detect if a CFL light bulb is broken or burned out and indicate the status through a small LED. Aim to prevent excessive cycling, as frequent switching can reduce the lifespan of CFL bulbs. The solution should be cost-effective, energy-efficient, and capable of accurately recording and displaying the on/off cycle count for each connected CFL light bulb. Finally, the calculator should be able to power and transition the CFL light bulb between different levels of brightness.

CHAPTAR FOUR

METHODOLOGY: -

A) Hardware component:

1. **Microcontroller (Arduino Uno R3):** It is the most used and documented board of the whole Arduino family and is the best board to get started with electronics and coding, The brain of the operation, is responsible for processing and interpreting data [2]. Shown in **Figure 2**.

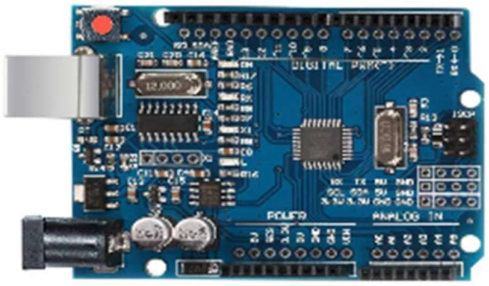


Figure 2: Arduino Uno [10].

2. **Relay (SPDT):** This component acts as a switch, controlling the ON/OFF state of the CFL bulb based on Arduino's instructions, it is a type of relay that is beneficial in modern applications due to its internal configurations [8]. Shown in **Figure 3.**

Figure 3: Relay [11].

3. **Power Supply:** A power supply is a device that provides electrical energy to one or more electric loads, the power supply is responsible for providing the necessary voltage and current to the Arduino board and any connected components [9]. Shown in **Figure 4.**

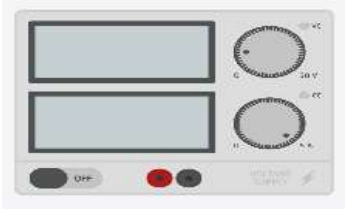


Figure 4: Power supply

4. **Photo-resistor:** A photo-resistor (also known as a Photocell, or Light Dependent Resistor (LDR), Photo Conductive Cell), is a type of resistor whose resistance varies with the intensity of light. It exhibits a decrease in resistance as light levels increase and vice versa. Photo-resistors are commonly used in electronic devices for light detection and automatic control applications [1]. Shown in **Figure 5.**

Figure 5: Photo-resistor [12].

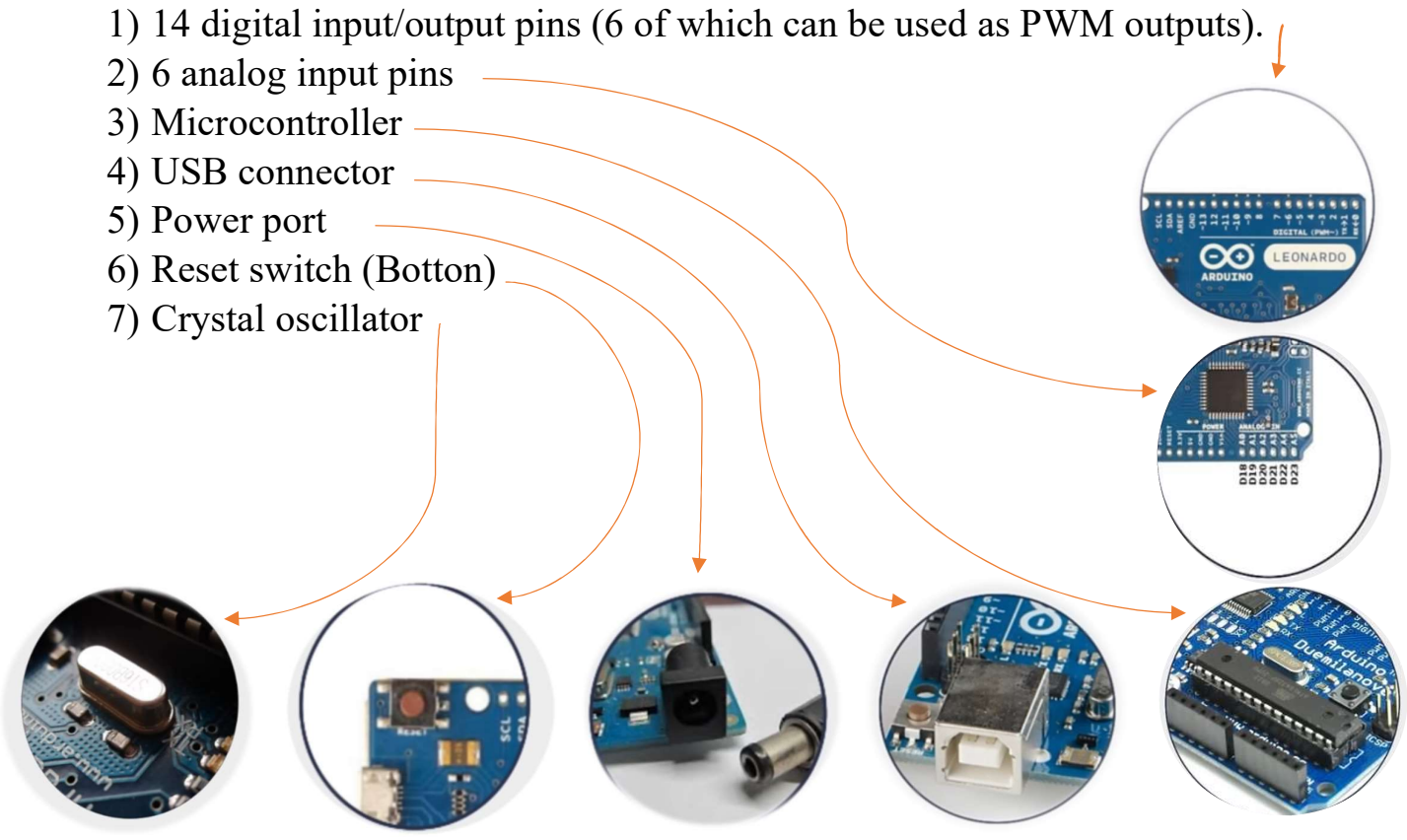
5. **Resistor:** A resistor is an electronic component that restricts the flow of electric current. It is designed to introduce a specific amount of resistance into an electrical circuit, controlling the amount of current that can flow through. Resistors are used for various purposes, such as setting the bias point of transistors, dividing voltages, limiting current, and protecting components in a circuit. They come in different values and types to suit different applications.

6. **Light bulb:** A light bulb is a device that produces light from electricity. It typically consists of a filament, often tungsten, enclosed in a glass bulb filled with an inert gas. When an electric current passes through the filament, it heats up and emits visible light. Light bulbs are widely used for illumination in homes, businesses, and various other applications. Traditional incandescent bulbs have become less common with the rise of more energy-efficient alternatives such as LED (Light Emitting Diode) and CFL (Compact Fluorescent Lamp) bulbs, as shown in **Figure** below.

Figure 6: CFL Bulbs [13].

Arduino Uno:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. The Arduino UNO R3 processing board is illustrated in Figure 2. Working clockwise from the left, the board is equipped with a USB connector to allow programming the processor from a host personal computer (PC) or laptop. The board may also be programmed using In-System Programming (ISP) techniques. A 6-pin ISP programming connector is on the opposite side of the board from the USB connector [3, 4]. Arduino Uno is designed to be easy to use. The Arduino can't emit more than 5V from any of its pins, let alone enough to power a light bulb that needs 30 to 40 times the voltage for an LED.



Relay Module:

An SPDT (Single Pole Double Throw) relay is a type an electromagnetic switch that can control the on/off cycles of the CFL bulbs. It consists of five control terminals, including two electromagnetic relay coils, one common terminal (CM), one normally closed terminal (N/C), and one normally open terminal (N/O). The relay can be used to switch between two circuits, allowing for the control voltage and current flow to the connected CFL bulbs. To system should be designed to ensure accurate timing, consistent illumination levels, and error detection and logging for performance analysis. The SPDT relay is beneficial in modern applications due to its internal configurations, making it suitable for controlling the on/off cycles of the CFL bulbs. It can be connected in a circuit to control the flow of current based on the input power applied to its coil terminals. Additionally, SDPT relays are available in various specifications, such as coil voltage, and contact rating. For example, a high-quality SPDT sealed relay may have a coil rated up to 5V DC, with a minimum switching voltage of 5V. An SPDT relay always has six pins: typically with three male headers for connection to the Arduino and three screw terminal pins to connect to mains electricity [7,8].

Connecting the Relay Module to Arduino

The relay module should have header pins on it with three signs: Signal, 5v, and GND. These are the pins that we're supposed to connect to the Arduino with jumper wires.

The pin works as follows:

1-**Signal**: Turns the relay on or off.

2-**5V**: Accepts 5V to power the relay.

3-**GND**: connects to Arduino's GND pin.

Connecting the Relay module to a Light bulb

On the other side of the relay module, there's a screw terminal with three holes. It should have three signs: N/C, N/O, and common Ground. Each hole connects to a copper wire that carries high-voltage electricity to and from the light bulb.

1- **N/C (Normally Closed):** Connects to a light bulb; closed while relay is off.
2- **N/O (Normally Open):** Connects to a light bulb; open while relay is off.
3- **Common Ground:** Connects to the main source.

We can only have either N/C or N/O connected to the light bulb-never both at the same time, as Explained in **Figure 7.**

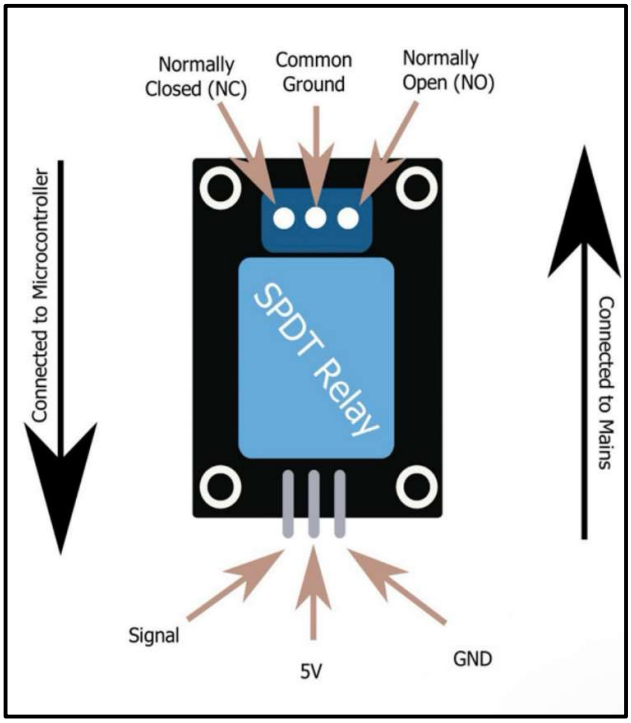


Figure 7 Relay

B) Software Tools:

Arduino IDE for coding:

The Arduino IDE is an open-source software used for writing, compiling, and uploading code to Arduino boards. It is designed by Arduino. cc and supports both C and C++ languages. The IDE consists of two main parts: Editor and Compiler. The Editor is used for writing the required code, while the Compiler is used for compiling and uploading the code. The IDE is available for all operating systems, including MAC, Windows, and Linux, and runs on the Java Platform. It comes with inbuilt functions and commands that play a vital role in debugging, editing, and compiling the code.

The IDE supports the exchange of data with the connected board on the port and allows the installation of new libraries. It also includes a Serial Plotter button used to display the serial data in a plot and check and update the Wi-Fi Firmware of the connected board. The IDE requires the selection of the board from the list of boards, and the selected board must be similar to the board connected to the computer. The IDE also consists of the virtual and real serial devices present on our machine and gives information about the selected board. The IDE allows the customization settings and provides examples of small projects for a better understanding of the IDE and the board.

The IDE supports the creation of new sketches and the opening of existing sketches. The sketch is written in the text editor and is then saved with the file extension .ino. The IDE environment mainly contains two basic parts: Editor and Compiler where the former is used for writing the required code and later is used for compiling and uploading the code. The IDE also supports the burning of the Bootloader onto the microcontroller on an Arduino board. This is useful if you purchase a new ATmega microcontroller, which ensures that you've selected the correct board from the Boards menu before burning the bootloader on the target board.

Figure 8: Arduino IDE

c) The system design and connection:

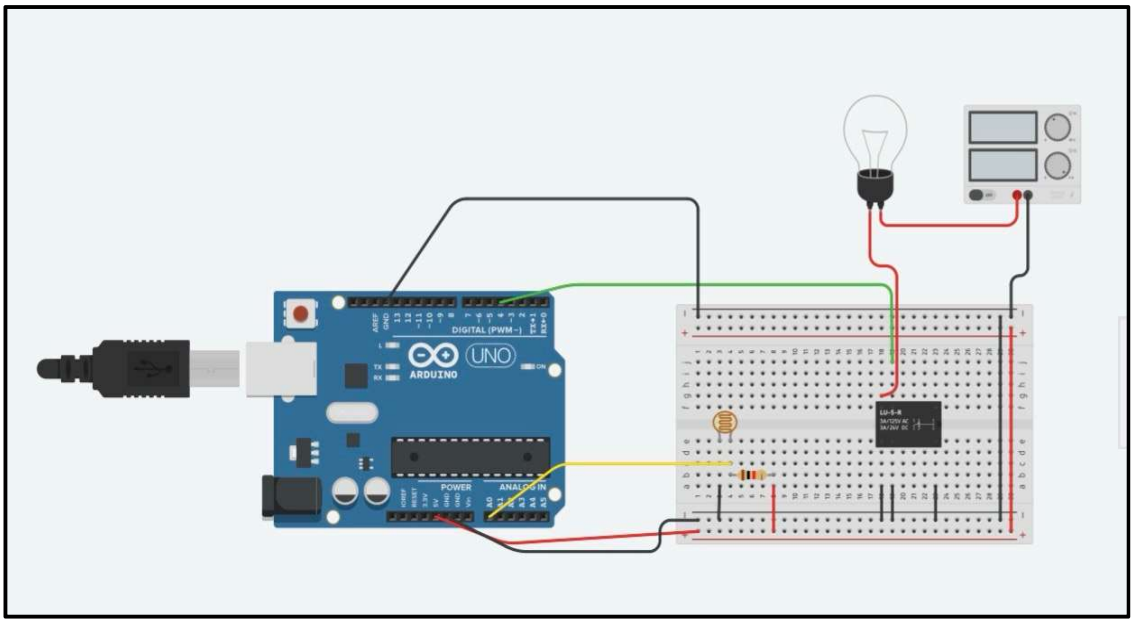


Figure 9: Design Circuit by Tinker-Cad

I. Materials:

1 × Arduino Uno

1 × Photo-resistor (LDR)

1 × Bread Board

1 × Power Supply

12-17 Jumper Wires

1 × Relay (SPDT)

1 × Light bulb

1 × 1kΩ Resistor

II. Connection:

Step 1: Wiring

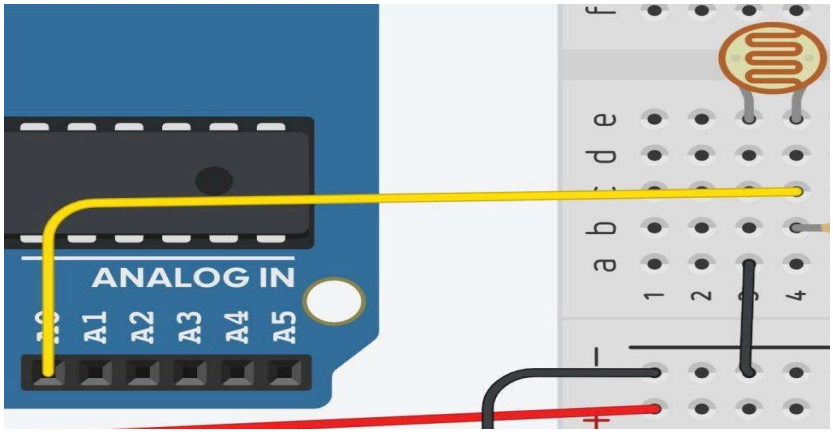


Figure 10: Wiring

Our first step for this project is to connect terminal 2 of the LDR to the A0 pin of the Arduino. A0 pretty much means analog input. Analog input Converts voltage into a digital value which can stored and processed by the software. Shown in **Figure 10**.

Step2: Connecting the LDR (pt.1)

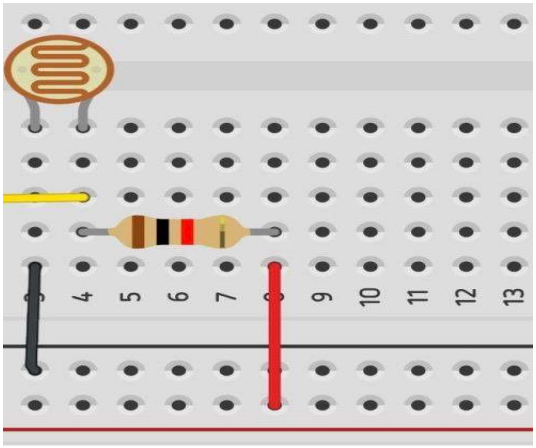


Figure 11: Connecting the LDR (pt.1)

Once this step is over, we must give power to the LDR. To do this we must add a resistor first to terminal 2 and then add power. The purpose of the resistor is to stop the flow of current. Too much current can damage an electronic which is not good that is why we use a resistor. As we can see from the picture the wire connecting to the ground is connected to the resistor which then will be connected to the LDR. Shown in **Figure 11**.

Step3: connecting the LDR (pt.2)

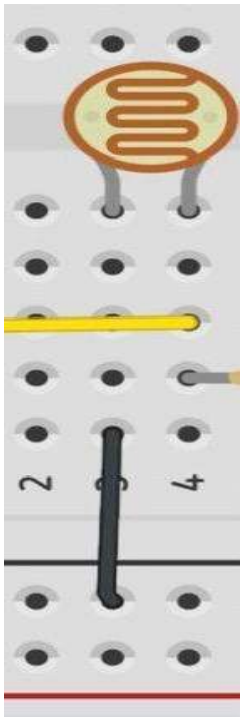


Figure 12: Connecting the LDR (pt.2)

Once this step is over, we must attach terminal 1 of LDR to the ground. This is an important step since we must let the electricity coming in go back to the Arduino. The only way to accomplish this is by connecting one of the railings of the LDR to the ground. Shown in **Figure 12**.

Step4: Relay

We might be wondering what a relay is. The purpose of this component is to control the circuit. There are two main switches on a relay. NC (Normally Closed) and NO (Normally Open). Later in this tutorial, we will know how these two parts of the circuit. Now we must connect the relay to the circuit. Once this part is over, we will start wiring our circuit.

Step 5: Connection Terminal 5

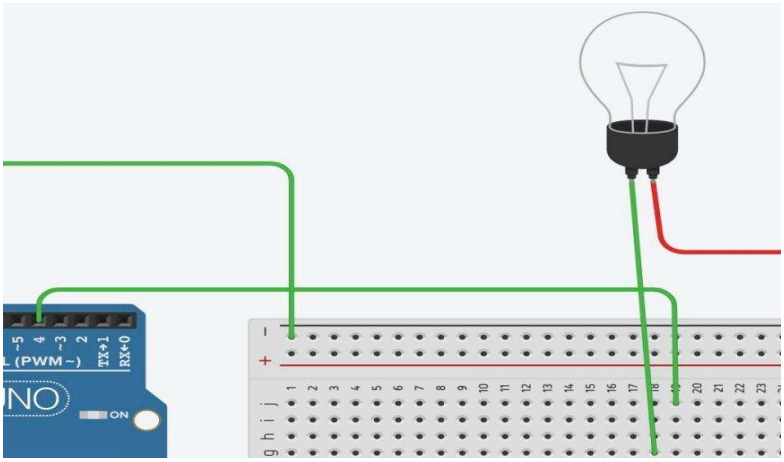


Figure 13: Connection Terminal 5

Now the relay is connected to us breadboard. Once this is done, we will connect terminal 2 of the relay to pin to pin 4. The pin 4 is not any specific pin, we can connect this to any pin we'd like, but this will be very important when it comes to the coding part. Shown in **Figure 13**.

Step 6: Connecting the Light Bulb

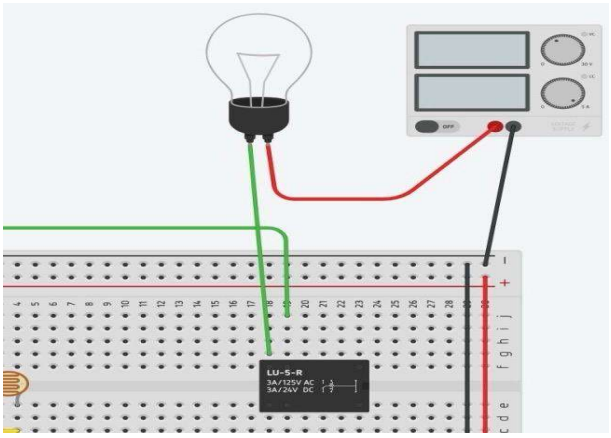


Figure 14: Connecting the Light Bulb

Since the relay has been connected to the breadboard we can connect the light bulb to the relay. We first step is to get terminal 1 of the light bulb and connect it to terminal 1 of the relay. Once this is done we take terminal two of the light bulb and connect it to the positive side of the power supply. Then take the negative side of the power supply and attach it to a negative rail on the breadboard. Shown in **Figure 14**

Step 7: Wiring Power and Ground

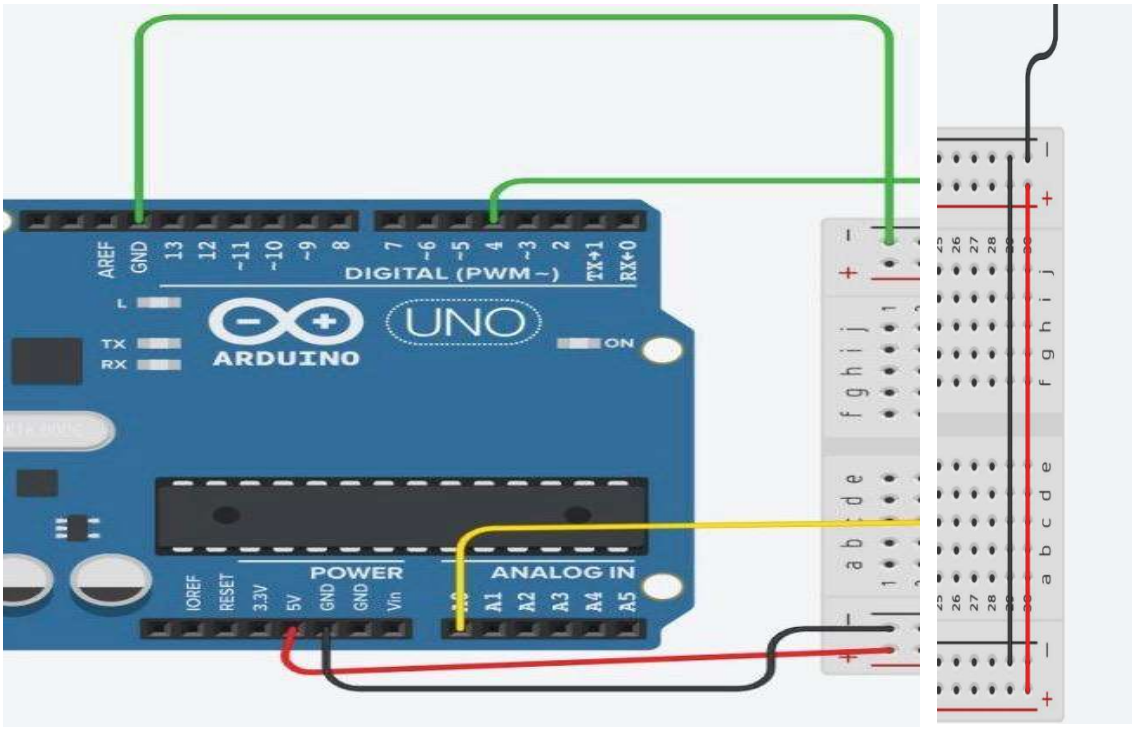


Figure 15: Wiring Power and Ground

Now since the wiring of the electronics is done we must remember that we have to give power and ground to both of the railing across the breadboard.

To achieve this connect one end of the ground railing to the other end of the ground railing. Once this is done, do the same thing for the power railing. This will allow the current to flow out from the Arduino and directly come back.

Before we get all excited don't forget to directly give power and ground to the breadboard. Just because we attached both the power and ground railing doesn't mean the ground and power are going through. To have ground and power on the breadboard we must connect the ground pin of the Arduino to the negative rail on the breadboard and the pin that says 5v on the Arduino to the positive railing of the breadboard. Shown in **Figure 15**.

III. Circuit design:

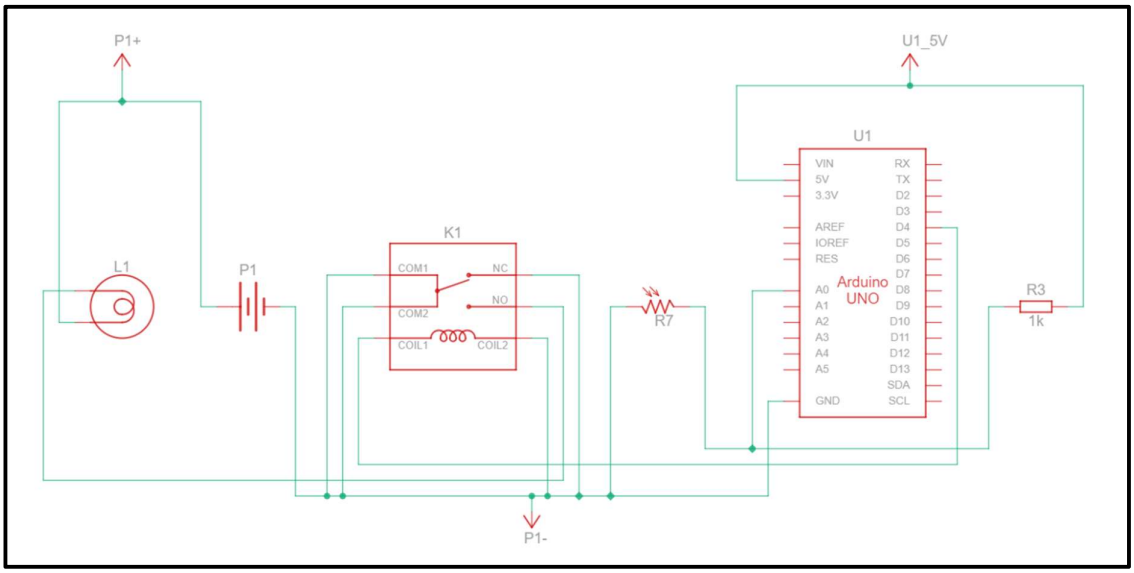


Figure 16: Circuit Design

D) Coding:

```
const int LDR_PIN = A0;  // LDR sensor pin
const int LED_PIN = 4;   // LED pin
const int THRESHOLD = 500; // Threshold value for LDR sensor
const int ON_TIME = 2000; // Time in milliseconds for LED to be on
const int OFF_TIME = 1000; // Time in milliseconds for LED to be off
const int MAX_CYCLES = 10; // Maximum number of LED cycles

int ledState = LOW;      // ledState used to set the LED
unsigned long previousMillis = 0;  // will store last time LED was updated
unsigned long currentMillis; // current time
int cycleCount = 0; // counter for LED cycles
bool ledOn = false; // flag to indicate if LED is currently on

void setup() {
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  currentMillis = millis(); // get the current time

  // Check if it's time to change the LED state
  if (ledOn && currentMillis - previousMillis >= ON_TIME) {
    ledOn = false;
    digitalWrite(LED_PIN, LOW); // turn LED off
    previousMillis = currentMillis; // remember the time
  } else if (!ledOn && currentMillis - previousMillis >= OFF_TIME) {
    ledOn = true;
    digitalWrite(LED_PIN, HIGH); // turn LED on
    previousMillis = currentMillis; // remember the time
    cycleCount++; // increment cycle count when LED turns on
    Serial.print("Cycle Count: ");
    Serial.println(cycleCount);
  }

  // Check if LDR sensor detects light
  int ldrValue = analogRead(LDR_PIN);
  if (ldrValue > THRESHOLD) {
    // If LED is on, reset cycle count
    if (ledOn) {
      cycleCount = 0;
    }
  }

  // Check if maximum cycles reached or LED is continuously on
  if (cycleCount >= MAX_CYCLES || (cycleCount > 0 && !ledOn)) {
    Serial.println("LED is blown. Stopping...");
    // You can add additional actions here, such as turning off the LED
    // or performing some cleanup before stopping the program.
    while (true) {
      // This loop will halt the program and keep it in a stopped state.
    }
  }
}
```

Step 1-6: Variable declarations

• LDR_PIN is set to A0, which is the pin connected to a Light Dependent Resistor (LDR) sensor.

• LED_PIN is set to 4, which is the pin connected to an LED.

• THRESHOLD is set to 500, which is the threshold value for the LDR sensor.

• ON_TIME is set to 2000, which is the time in milliseconds for the LED to be on.

• OFF_TIME is set to 1000, which is the time in milliseconds for the LED to be off.

• MAX_CYCLES is set to 10, which is the maximum number of LED cycles.

Step 7-13: Global variable declarations

• ledState is set to LOW, which is used to set the LED state.

• previousMillis is an unsigned long variable to store the last time the LED was updated.

• currentMillis is an unsigned long variable to store the current time.

• cycleCount is an integer variable to count the number of LED cycles.

• ledon is a boolean variable to indicate if the LED is currently on.

Step 14-17: setup() function

• The setup() function is called once at the beginning of the program.

• It sets the LED_PIN as an output using pinMode (LED_PIN, OUTPUT).

• It initializes the serial communication at a baud rate of 9600 using Serial.begin (9600).

Step 18-53: loop() function

• The loop() function is called repeatedly after the setup() function.

• It gets the current time using currentMillis = millis().

• It checks if it's time to change the LED state:

• If the LED is on (ledon is true) and the time since the last update is greater than or equal to ON_TIME, it turns the LED off, sets ledon to false, and updates previousMillis.

• If the LED is off (ledon is false) and the time since the last update is greater than or equal to OFF_TIME, it turns the LED on, sets ledon to true, updates previousMillis, increments cycleCount, and prints the cycle count to the serial monitor.

• It checks if the LDR sensor detects light:

• If the LDR sensor value is greater than the threshold, it resets the cycle count if the LED is on.

• It checks if the maximum cycles have been reached or if the LED is continuously on:

• If either condition is true, it prints a message to the serial monitor indicating that the LED is blown and stops the program using an infinite loop.

CHAPTAR FIVE

5.1 Working principle:

The circuit is powered via the USB cable connected to the Arduino Uno board. The Arduino Uno can be powered either through the USB connector or through an external power supply connected to the 5v. The power supply provides the necessary voltage and current to operate the Arduino and any connected components. The circuit you provided is an Arduino -based circuit that uses a Light Dependent Resistor (LDR) to control a relay and used as a sensor to detect the presence or absence of light. The LDR is connected to an analog input pin on the Arduino, and its resistance changes based on the amount of the light that falls on it. The relay is connected to a digital output pin on the Arduino and is used to control a 220V power supply (AC). An LDR sensor changes its resistance based on the amount of light that falls on it. When connected to an Arduino, we can read the value of the LDR using an analog input pin. When the Arduino is powered on, it reads the value of the LDR and compares it to a threshold value. If the LDR value is below the threshold and LDR detects light, the Arduino sets the digital output pin to HIGH, which activates the relay or relay to open the circuit and turns on the 220V power supply. If the LDR value is above the threshold, the Arduino sets the digital output pin to LOW, which deactivates the relay or relay to close the circuit and turns off the 220V power supply. And the Arduino code for the circuit would include functions to read the input from the LDR make decisions based on the input, and control the relay accordingly. The code would also include functions to count the number of time the circuit it opened and closed, as well as calculate the lifespan of the load based on the number of cycles.

5.2 Cost Analysis

No.	Name	Quantity	Price(IQD)
1	Arduino uno R3	1	20,000
2	Relay (SPDT)	1	
3	Power Supply	1	
4	Resistor	1	
5	Photoresistor	1	
6	Jumper Wire	1 Set	4,000
7	CFL Light		
8	Bread Bord	1	10,000
	Total		

Chapter five

Result and Discussion:

Reference:

[1]- https://circuitdigest.com/microcontroller-projects/arduino-light-sensor-using-ldr

[2]- https://store.arduino.cc/products/arduino-uno-rev3

[3]- https://eepower.com/resistor-guide/resistor-types/photo-resistor/

[4]- https://www.rs-online.com/designspark/what-is-arduino-uno-a-getting-started-guide

[5]- https://learn.sparkfun.com/tutorials/what-is-an-arduino/all

[6]- https://www.familyhandyman.com/project/cfl-bulbs-heres-what-you-need-to-know/

[7]- https://www.iottechtrends.com/use-relay-module-with-arduino/

[8]- https://www.ourpcb.com/spdt-relay.html

[9]- https://www.monolithicpower.com/en/ac-dc-power-supply-basics

[10]- https://images.app.goo.gl/6j572uyJnPwLopGLA

[11]- https://images.app.goo.gl/7BApJr9hSPLRK1ScA

[12]- https://images.app.goo.gl/6KZeQaFVjJ6BFgv77

[13]- https://images.app.goo.gl/pfRCUrgLVdbti7N56

[14]- [Estimation of Life and Reliability of Electronic Product] BY: Ramesh E Patil, Jerald Leo, Govind R.

Kunkolienkar, Date: April 2017.

[15]- [Life Cycle Assessment of Electronics] BY: Otto Andersen, John Hille, Geoffrey Gilpin, Andres S. G. Andrae,

Date: January 2014.

[16]- [Life Cycle Assessment of Electricity Systems] BY: Roberto Turconi, Date: 2014. Citation: Turconi, R. (2014).

Life Cycle Assessment of Electricity Systems. DTU Environment.

[17]- [Approaches to Life Estimation of Electronic Circuits] BY: Akira MOTOYAMA. Representative, M.A. Reliability

Technology Office.

[18]-[Environmental Life Cycle Assessment] BY: Olivier Jolliet, Myriam Saade-Sbeih, Shanna Shaked, Alexandre

Jolliet, Pierre Crettaz. Version Date:20151012. International Standard Book Number-13:978-1-4398-8766-0

(Hardback).