IoT BASED CONSUMER LINE CONTROLLING SYSTEM

A report submitted to the Department of Electrical and Electronic Engineering, Sonargaon University of Bangladesh in partial fulfillment of the requirements for the Award of Degree of Bachelor of Science in Electrical and Electronic Engineering.

Submitted by

Md. Sohan Mia	EEE2103024103
S.M. Anis Ahmed Jibon	EEE2103024069
Rakibul Ahmed	EEE2103024088
Zonaed Habib	EEE2103024089
Md. Tariqul islam	EEE2103024086



Supervised by

Shithi Dev

Lecturer

Department of Electrical and Electronic Engineering

Sonargaon University (SU)

Dhaka-1215, Bangladesh

April, 2025

DECLARATION

It is declared hereby that this thesis paper or any part of it has not been submitted to anywhere else for the award of any degree

Md. Sohan Mia
EEE2103024103

S.M. Anis Ahmed Jibon
EEE2103024069

Rakibul Ahmed
EEE2103024088

Zonaed Habib
EEE2103024089

Md. Tariqul islam
EEE2103024086

Under Superviosn of

Shithi Dev
Lecturer
Department of EEE
Sonargaon University (SU)

CERTIFICATION

This is to certify that this project entitled **"IoT Based Consumer Line Controlling System"** is done by the following students under my direct supervision. This project work has been carried out by them in the laboratories of the Department of Electrical and Electronic Engineering under the Faculty of Engineering, Sonargaon University (SU) in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering.

Supervisor

_____

Shithi Dev

Lecturer

Department of EEE

Sonargaon University (SU)

ACKNOWLEDGEMENT

The report titled as **"IoT Based Consumer Line Controlling System"** has been prepared to fulfill the requirement of our practicum program. In the process of doing and preparing our practicum report, we would like to pay our gratitude to some persons for their enormous help and vast cooperation.

At first, we would like to show our gratitude to the University authority to permit us to do our practicum. Specially, we would like to thank to our honorable supervisor Shithi Dev, Lecturer, Department of Electrical and Electronics Engineering, SU – Sonargaon University Dhaka, for his valuable and patient advice, sympathetic assistance, cooperation, contribution of new ideas. Deep theoretical and hardware knowledge & keen interest of our supervisor in this field influenced us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

Finally, we would like to thank again the respected Vice-Chancellor of SU, also thanks to the Head of the Department of SU, **Electrical & Electronics Engineering**, because they are designated such an environment for learning through which we got the opportunity to acquire knowledge under **BSc in EEE program**, and that will be very helpful for our prospective career.

We are, indeed, grateful to all those from whom we got sincere cooperation and help for the preparation of this report.

ABSTRACT

IOT Based Consumer Line Controlling System is an innovative energy management device designed to enhance power efficiency and safety in residential and commercial electrical systems. This system automatically disconnects the power supply to individual consumer lines during periods of low usage or when appliances are idle, effectively reducing unnecessary energy consumption. Utilizing microcontroller-based automation, the device monitors real-time load status and makes intelligent decisions to optimize power distribution. It also provides surge protection and prevents electrical hazards by isolating faulty lines instantly. With IoT integration, users can remotely monitor and control the disconnector, making it ideal for smart home and energy-conscious environments. This solution not only contributes to significant power savings but also increases the lifespan of electrical devices by minimizing overuse and overheating. The Power Saver Consumer Line Disconnector represents a practical approach toward sustainable and efficient energy utilization.

TABLE OF CONTENTS

Declaration	ii
Certification	iii
Acknowledgement	iv
Abstract	v
List of Figure	viii

		Page
CHAPTER-1	**INTRODUCTION**	
1.1 Overview		01
1.2 Problem Statement		01
1.3 Motivation		02
1.4 Objective		02
1.5 Methodology		02
1.6 Work Flow Diagram		03
1.7 Research Outline		03
CHAPTER-2	**LITERATURE REVIEW**	04-06
CHAPTER-3	**SYSTEM ARCHITECTURE**	
3.1 Introduction		07
3.2 Block Diagram		07
3.3 Circuit Design		08
3.4 System Design and Architecture		08
3.5 Working Principle		09
3.6 Cost Analysis		09
CHAPTER-4	**HARDWARE AND SOFTWARE ANALYSIS**	
4.1 Introduction		10
4.2 Hardware & Software		10
4.3 Node MCU		10

4.4 SMPS 13
4.5 Relay Module 17
4.6 Arduino IDE 20
4.7 Easy EDA 24

CHAPTER-5 RESULT AND DISCUSSION

5.1 Result Analysis 25
5.2 Final Project 26
5.3 Advantage 26
5.4 Application 27
5.5 Limitation 27
5.6 Discussion 27

CHAPTER-6 CONCLUSION

6.1 Conclusion 28
6.2 Future Scope 28

REFERENCE 29

APPENDIX 31

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	Work Flow Diagram	03
3.1	Block Diagram of our Project	07
3.2	Schematic Design of our Project	08
3.3	Server Interface Block	08
4.1	Node MCU	11
4.2	Node MCU Schematic Diagram	11
4.3	Node MCU pin Out	12
4.4	SMPS	13
4.5	SMPS Circuit	14
4.6	SMPS Circuit Connection	16
4.7	DC Power Supply Way	17
4.8	Relay	17
4.9	Relay Module	18
4.10	Main Voltage Connection	19
4.11	Voltage Connections	19
4.12	Arduino Software Inteface IDE	21
4.13	Easy EDA Software Interface	24
5.1	Complete Project Prototype	26

CHAPTER 1
INTRODUCTION

1.1 Overview

IoT Based Consumer Line Controlling System is a smart energy management system designed to reduce electricity wastage by automatically disconnecting power supply to non-essential or idle appliances during off-peak hours or when not in use. This system integrates a microcontroller or IoT-based control unit and relays to monitor power consumption in real time and trigger disconnection when predefined thresholds are exceeded or idle conditions are detected.

Primarily targeted for domestic and small commercial applications, the device enhances energy efficiency, reduces utility bills, and promotes the sustainable use of electricity. It can be configured to work manually or automatically, and may include user interfaces such as mobile apps monitoring and control. Some advanced versions include overload protection, surge detection, and remote operation capabilities.

The project addresses rising concerns over unnecessary energy consumption and supports the growing global emphasis on smart grid systems and demand-side energy management. By preventing standby power drain (often called “phantom load”), the Power Saver Consumer Line Disconnector ensures optimized usage of electrical power. This solution not only contributes to economic savings for consumers but also aligns with broader goals of energy conservation and environmental responsibility.

1.2 Problem Statement

IoT Based Consumer Line Controlling System aims to address excessive energy consumption and electrical hazards in residential and commercial settings. Many consumers unknowingly waste electricity due to devices left on standby or during peak hours. This system automatically disconnects power to non-essential loads during idle periods or when voltage fluctuations are detected, reducing energy bills and preventing damage to appliances. It ensures safety by isolating faulty lines and optimizing power usage. The project promotes energy conservation, enhances electrical safety, and supports sustainable power management by enabling smart, real-time control over energy distribution at the consumer end.

1.3 Motivation

The motivation behind the IoT Based Consumer Line Controlling System project stems from the growing need for energy efficiency and electrical safety in homes and workplaces. Rising electricity costs, energy wastage, and frequent appliance damage due to voltage fluctuations highlight the importance of smarter power management. This project offers an automated solution to disconnect non-essential loads, reduce standby power consumption, and protect electrical systems. By integrating intelligent control and fault detection, it empowers users to conserve energy effortlessly. The project supports sustainable living, encourages responsible energy use, and contributes to reducing the overall carbon footprint, aligning with global efforts toward energy conservation.

1.4 Objective

The objectives of this project are

- To study,design and construct the IoT Based Consumer Line Controlling System
- To automatically disconnect non-essential electrical loads during idle or standby conditions.
- To reduce energy consumption and lower electricity bills.
- To enhance user safety by isolating faulty lines.
- To promote efficient and sustainable power usage through smart control.
- To put the whole system into action so that we can observe its practical effects and verify our work.
- To study the system performance for future reference and improvement purpose.

1.5 Methodology

Our methodology for the project:

- Creating an idea for design and construction of an IoT Based Consumer Line Controlling System. And designing a block diagram & circuit diagram to know which components we need to construct it.
- Collecting all the components and programming the micro-controller to control the whole system.
- Setting up all the components in a PCB board & then soldering. Lastly, assembling all the blocks in a board and to run the system & for checking purposes.

1.6 Work Flow Diagram

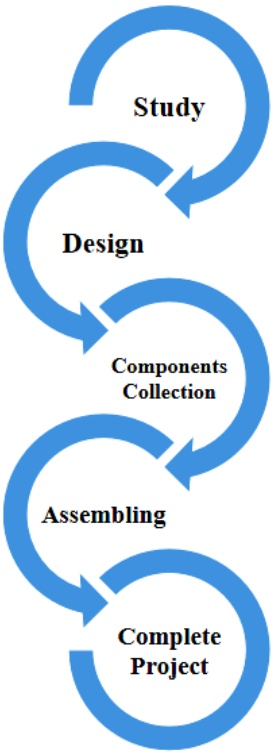


Figure 1.1 : Work Flow Diagram

1.7 Research Outline

This project book consists of Six chapter. The first chapter contains the statement of the Introduction,Overview, Problem Statement, Motivation,Work Flow Diagram, objectives of the study and Methodology. Chapter two contains system Literatur e Review details. Chapter three System Architecture, Discussed in Block Diagram,Circuit Diagram,Working principle and Cost analysis.Chapter Four describes the hardware implementation with component details and the software which we have used for our work. Chapter Five deals with the result Analysis,shows the complete prototype,Advantage, Application and Limitation of the project that we have built. In the final chapter, we have discussed the , Future Scope, conclusion of the project.

CHAPTER 2
LITURATURE REVIEW

In this chapter we discuss our system overview. In this paper we are also work about IoT Based Consumer Line Controlling System. In below section we will describe some previous literature review. From previous literature we gather more knowledge to build this project and we successfully made it .

Kumar and Jain [1] proposed a microcontroller-based smart disconnector for consumer lines that actively monitors electrical parameters and disconnects the power supply in the event of faults or overloads. Their work laid the foundation for automated disconnection using low-cost microcontrollers, emphasizing reliability and consumer safety. The system was designed for quick response and scalability in both urban and rural settings.

Building upon the smart control concept, Sharma and Bhardwaj [2] incorporated IoT technology to develop an advanced energy monitoring and control system capable of disconnecting non-critical appliances during high-demand periods. Their system provides real-time energy analytics and remote control via the internet, introducing a new layer of user interactivity and demand-side load shedding, critical in managing distributed energy resources.

Patel and Chauhan [3] explored a hardware-based solution using Arduino microcontrollers and current sensors to automatically detect abnormal line conditions. Their system automatically isolates the consumer line during fault conditions, improving both system protection and power conservation. This design was notable for its low implementation cost and adaptability for small-scale residential applications.

Another major step in this direction was by Selvakumar et al. [4], who introduced a smart energy meter integrated with an automatic disconnection mechanism. The meter detects not just overloads but also signs of tampering and unauthorized energy usage. Their design emphasizes smart billing, theft detection, and energy auditing, all of which are valuable in today's digitized utility networks.

In contrast, Zhang and Wu [5] took a broader grid-level approach by designing an energy-saving circuit breaker for smart grids. Their research focused on intelligent prioritization, where non-essential or low-priority loads are automatically disconnected during peak

loads. The innovation lies in its seamless integration into existing power infrastructures and its ability to adapt to real-time grid demands, thus promoting load balancing and energy efficiency.

Tandel and Mehta [6] introduced a GSM-based consumer line disconnector that allows users or utilities to control the electrical supply remotely. Their approach is especially beneficial in remote or semi-urban areas where physical access to distribution panels is limited. By integrating GSM communication with real-time load control, their model improves energy reliability and user empowerment.

Fault detection and real-time response were central to the study by Chakraborty and Sinha [7], who developed a line monitoring system with fault isolation. Their system uses real-time current analysis to disconnect faulty sections immediately, preventing accidents and energy waste. The solution enhances safety and is well-suited for layered consumer networks such as apartment complexes or industrial parks.

A user-centered approach to power saving was taken by Ramesh et al. [8], who integrated IoT-based home automation to reduce idle energy consumption. Their system disconnects non-essential loads such as fans or lights in unused rooms, thereby optimizing household energy usage. The idea of integrating occupancy-based load control is a modern energy-saving strategy that aligns with smart home concepts.

The theme of smart home compatibility continues in the work of Rathod and Naik [9], who proposed an automatic disconnection system that relies on relay-based load isolation. Their design includes priority-based operation, where inefficient or low-priority devices are turned off during peak hours, helping consumers stay within energy quotas and utility demand response programs.

Mishra and Raj [10] presented a comprehensive approach combining real-time monitoring, consumer safety, and power saving. Their smart disconnector activates during line faults or hazardous conditions (like overvoltage), ensuring both system

protection and reduction in unnecessary energy consumption. Their system reflects the integration of automation and protection protocols, essential for smart city infrastructure.

From the literature discussed above, we gained a lot of knowledge and we inspired to do this project. We were able to do it with everyone tireless work.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 Introduction

The project objectives, methods, literature evaluation, and other details were all clarified in the preceding chapter. The block diagram, circuit diagram, Working principle,Project final Image and cost analysis will all be covered in this chapter.

3.2 Block Diagram

Here is the block diagram of the **IoT Based Consumer Line Controlling System** with all the essential components. All of the components are shown in below as a block in this diagram.Here we use Battery, Gear Motor, Wireless Coil, Street Light of our project and work together for perform as our desire outcome.

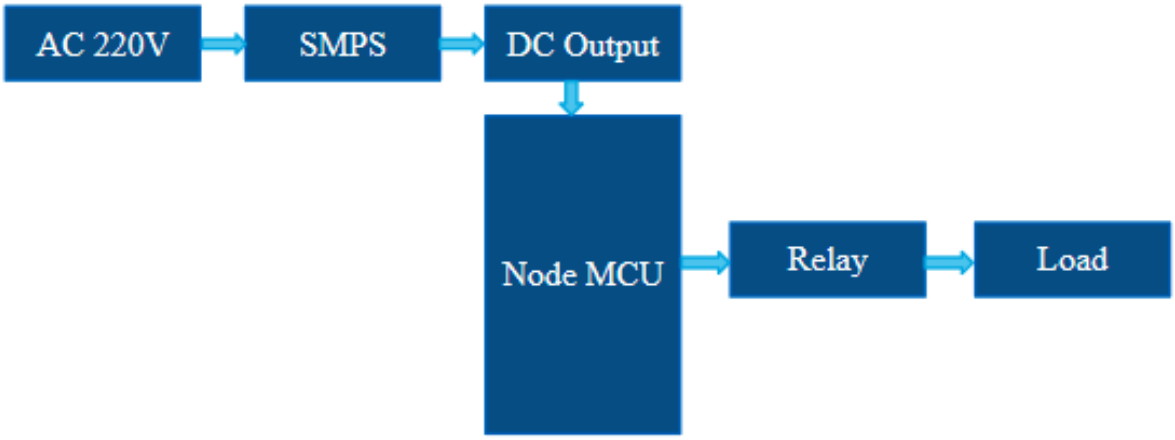


Figure 3.1: Block Diagram of Our Project

3.3 Circuit Design

In this part we show our project circuit design and connect out instrument through standard wire.

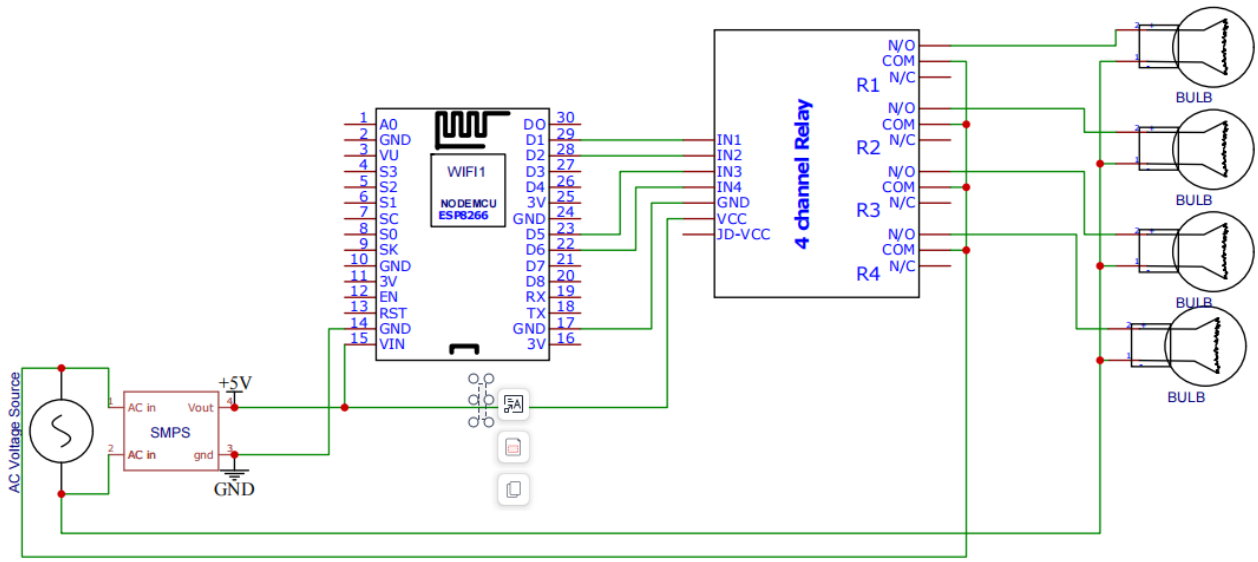


Figure 3.2 : Schematic Design of our Project

3.4 System Design and Architecture

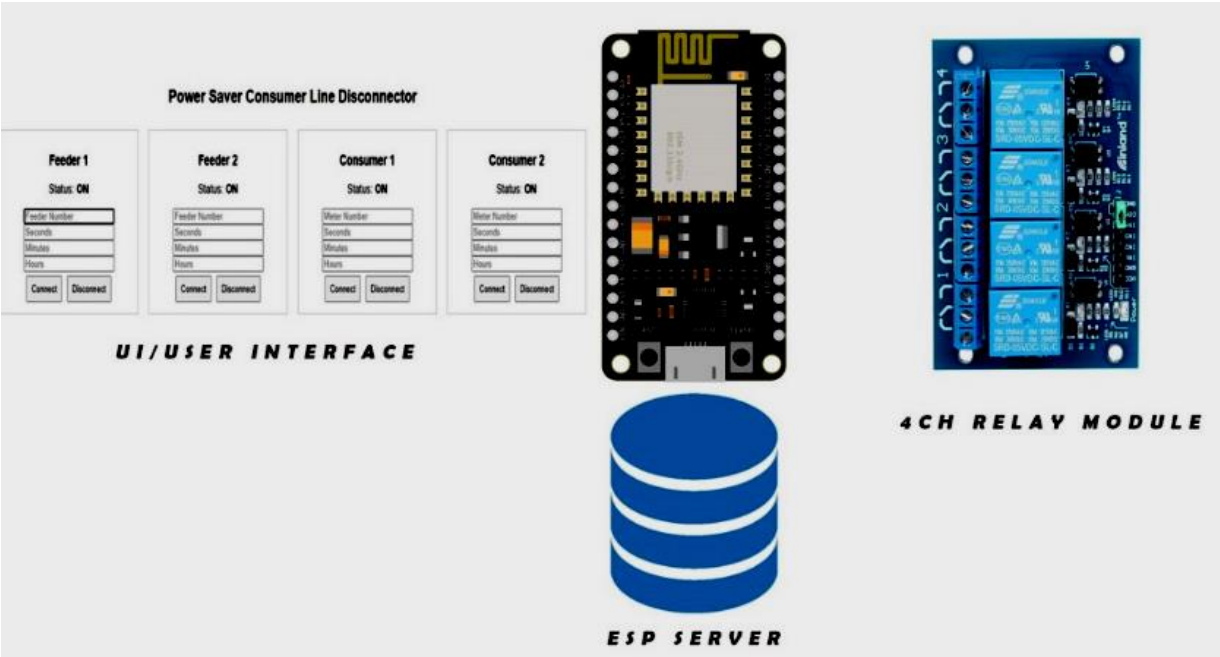


Figure 3.3. : Sever Interface Block

3.5 Working Principle

IoT Based Consumer Line Controlling System operates by monitoring and controlling the power flow to connected loads using a NodeMCU microcontroller. The system is powered by an SMPS (Switched Mode Power Supply), which converts AC to a stable DC voltage suitable for electronic components. The NodeMCU, programmed with logic to detect power thresholds or remotely receive control commands, processes input data and activates a relays accordingly. The relays serves as a switch, disconnecting or connecting the electrical load based on predefined conditions such as overload, remote command, or scheduled power saving. This ensures energy conservation and safety by preventing continuous power supply to idle or faulty appliances, enhancing both efficiency and system protection in residential or commercial settings.

3.6 Cost Analysis

Table 1:List of Component with Price

Sl.no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)
1	SMPS	5V 7A	1	800	800
2	Node MCU	ESP8266(italy version)	1	1500	1500
3	Relay Module	4 Channel	1	700	700
4	Wire & Holder & Bulb		4	500	500
5	PVC Board			500	500
6	Software			2000	2000
7	Others				1000
				Total	7000/=

CHAPTER 4

HARDWARE AND SOFTWARE ANALYSIS

4.1 Introduction

In this section, we will discuss elaborately about "IoT Based Consumer Line Controlling System" and the component description, features, working procedure of our all equipment. The system hardware fabricates composed of micro-controller unit, power unit, source unit, power store unite, sensor unit and many more related components.

4.2 Hardware & Software

Hardware

- SMPS
- Node MCU
- Relay Module

Software

- Arduino IDE
- Easy EDA

4.3 Node MCU

Node MCU is an open source firmware for which open source prototyping board designs are available. The name "Node MCU" combines "node" and "MCU" (micro-controller unit). sand built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson and SPIFFS. Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit ESP32 has also been implemented.

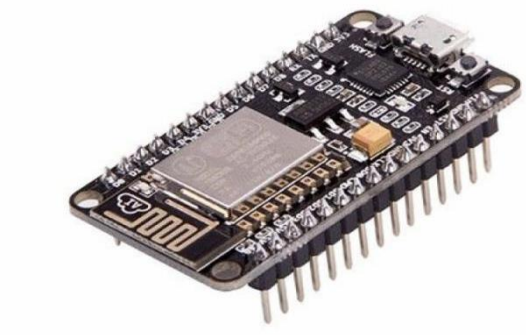


Figure 4.1: Node MCU

The prototyping hardware typically used is a circuit board functioning as a dual in-line package (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on breadboards. The design was initially was based on the ESP-12 module of the ESP8266, which is a Wi-Fi SoC integrated with a Tensilica Xtensa LX106 core, widely used in IoT applications. This an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Express if Systems, and hardware which is based on the ESP-12 module. The term "Node MCU" by default refers to the firmware rather than the development kits. The firmware uses the Luascripting language. It is based on the eLua project, and built on the Es press if Non-OS SDK for ESP8266. Node MCU was created shortly after the ESP8266 came out.

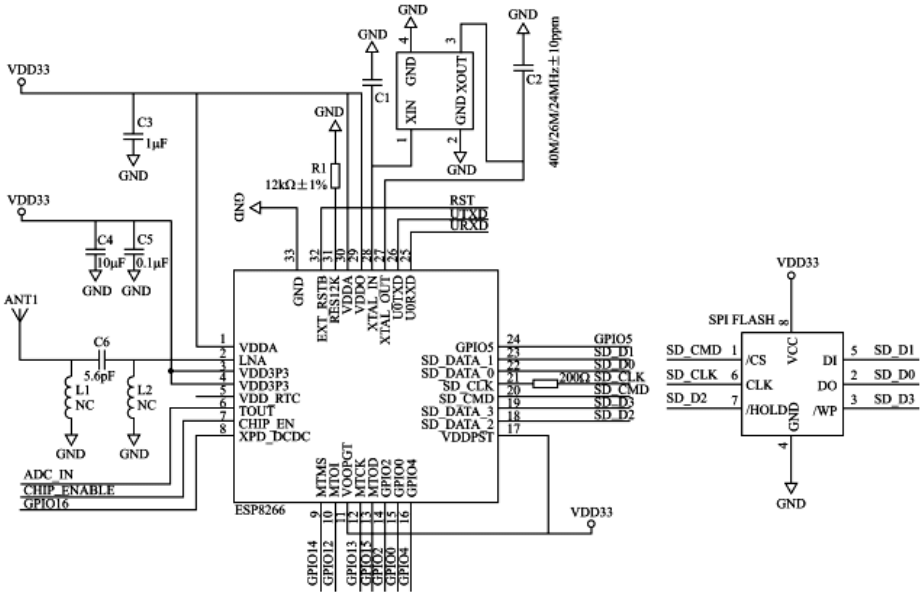


Figure 4.2: Node MCU Schematic Diagram

Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glibto Node MCU project, enabling Node MCU to easily drive LCD, Screen, OLED, even VGA displays. In summer 2015 the creators abandoned the firmware project and a group of independent contributors took over. By summer 2016 the Node MCU included more than 40 different modules. Due to resource constraints users need to select the modules relevant for their project and build a firmware tailored to their needs.

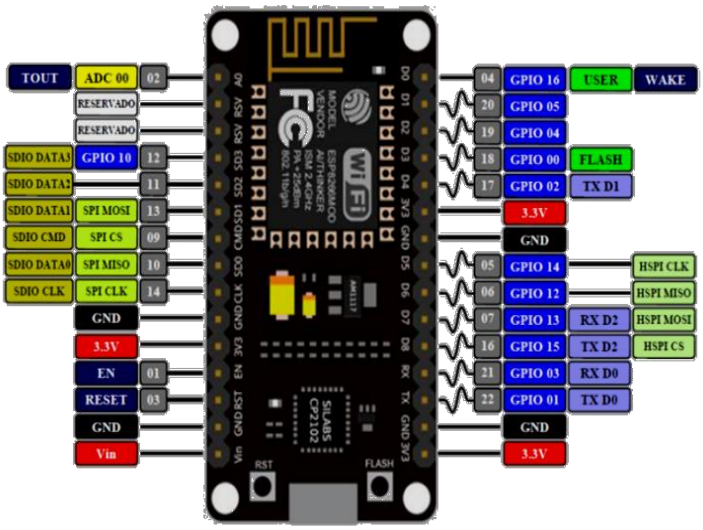


Figure 4.3: Node MCU Pin Out

Node MCU V3 ESP8266 ESP-12E is Wi-Fi development board that helps you to prototype your IoT product with few Lua script lines, or through Arduino IDE. The board is based on ESP8266 ESP-12E variant, unlike other ESP-12E, you won't need to buy a separate breakout board, usb to serial adapter, or even solder it to a PCB to get started, you will only need a usb cable (Micro USB).

Features

1. Communication interface voltage: 3.3V.
2. Antenna type: Built-in PCB antenna is available.
3. Wireless 802.11 b/g/n standard
4. WiFi at 2.4GHz, support WPA / WPA2 security mode
5. Support STA/AP/STA + AP three operating modes
6. Built-in TCP/IP protocol stack to support multiple TCP Client connections (5 MAX)
7. D0 ~ D8, SD1 ~ SD3: used as GPIO, PWM, IIC, etc., port driver capability 15mA
8. AD0: 1 channel ADC
9. Power input: 4.5V ~ 9V (10VMAX), USB-powered
10. Current: continuous transmission: ≈70mA (200mA MAX), Standby: <200uA
11. Transfer rate: 110-460800bps
12. Support UART / GPIO data communication interface
13. Remote firmware upgrade (OTA)

4.4 SMPS

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power) to DC loads, such as a personal computer, while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. A hypothetical ideal switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time (also known as duty cycles). In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply. Switched-mode power supplies may also be substantially smaller and lighter than a linear supply due to the smaller transformer size and weight.



Figure 4.4: SMPS

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size or lighter weight are required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor. Switched-mode power supplies are classified according to the type of input and output voltages. The four major categories are:

- AC to DC
- DC to DC
- DC to AC
- AC to AC

A basic isolated AC to DC switched-mode power supply consists of:

- Input rectifier and filter
- Inverter consisting of switching devices such as MOSFETs
- Transformer
- Output rectifier and filter
- Feedback and control circuit

The input DC supply from a rectifier or battery is fed to the inverter where it is turned on and off at high frequencies of between 20 KHz and 200 KHz by the switching MOSFET or power transistors. The high-frequency voltage pulses from the inverter are fed to the transformer primary winding, and the secondary AC output is rectified and smoothed to produce the required DC voltages. A feedback circuit monitors the output voltage and instructs the control circuit to adjust the duty cycle to maintain the output at the desired level.

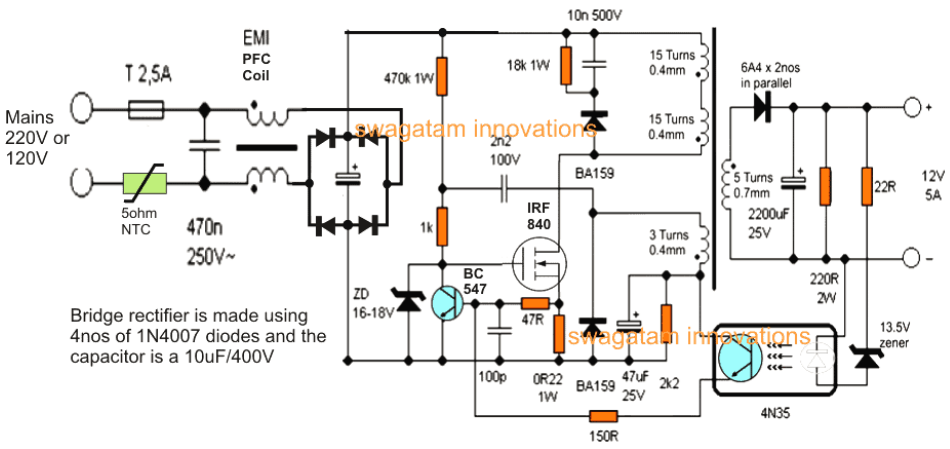


Figure 4.5: SMPS Circuit

Basic working concept of an SMPS

A switching regulator does the regulation in the SMPS. A series switching element turns the current supply to a smoothing capacitor on and off. The voltage on the capacitor controls the time the series element is turned. The continuous switching of the capacitor maintains the voltage at the required level.

Design basics

AC power first passes through fuses and a line filter. Then it is rectified by a full-wave bridge rectifier. The rectified voltage is next applied to the power factor correction (PFC) pre-regulator followed by the downstream DC-DC converter(s). Most computers and small appliances use the International Electrotechnical Commission (IEC) style input connector. As for output connectors and pin outs, except for some industries, such as PC and compact PCI, in general, they are not standardized and are left up to the manufacturer.

There are different circuit configurations known as topologies, each having unique characteristics, advantages and modes of operation, which determines how the input power is transferred to the output. Most of the commonly used topologies such as flyback, push-pull, half bridge and full bridge, consist of a transformer to provide isolation, voltage scaling, and multiple output voltages. The non-isolated configurations do not have a transformer and the power conversion is provided by the inductive energy transfer.

Advantages of switched-mode power supplies:

- Higher efficiency of 68% to 90%
- Regulated and reliable outputs regardless of variations in input supply voltage
- Small size and lighter
- Flexible technology
- High power density
- Disadvantages:
- Generates electromagnetic interference
- Complex circuit design

Switched-mode power supplies are used to power a wide variety of equipment such as computers, sensitive electronics, battery-operated devices and other equipment requiring high efficiency. Linear voltage IC regulators have been the basis of power supply designs for many years as they are very good at supplying a continuous fixed voltage output. Linear voltage regulators are generally much more efficient and easier to use than equivalent voltage regulator circuits made from discrete components such a zener diode and a resistor, or transistors and even op-amps.The most popular linear and fixed output voltage regulator types are by far the 78… positive output voltage series, and the 79… negative output voltage series.

These two types of complementary voltage regulators produce a precise and stable voltage output ranging from about 5 volts up to about 24 volts for use in many electronic circuits.There is a wide range of these three-terminal fixed voltage regulators available each with its own built-in voltage regulation and current limiting circuits. This allows us to create a whole host of different power supply rails and outputs, either single or dual supply, suitable for most electronic circuits and applications. There are even variable voltage linear regulators available as well providing an output voltage which is continually variable from just above zero to a few volts below its maximum voltage output.

Switch Mode Power Supply

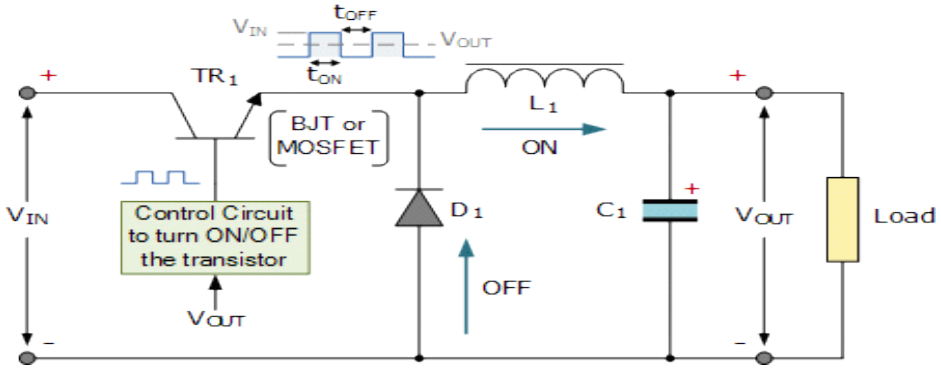


Figure 4.6: SMPS Circuit Connection

Most d.c. power supplies comprise of a large and heavy step-down mains transformer, diode rectification, either full-wave or half-wave, a filter circuit to remove any ripple

content from the rectified d.c. producing a suitably smooth d.c. voltage, and some form of voltage regulator or stabiliser circuit, either linear or switching to ensure the correct regulation of the power supplies output voltage under varying load conditions. Then a typical d.c. power supply would look something like this:

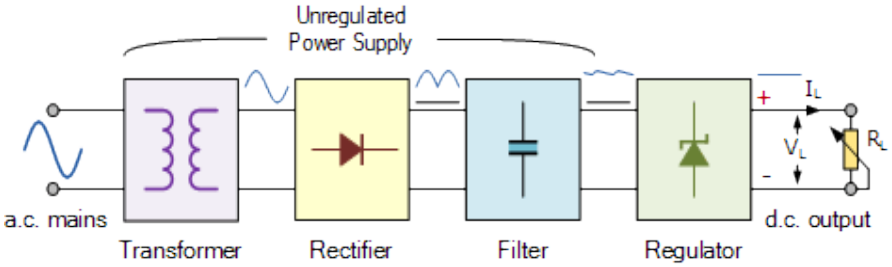


Figure 4.7: DC Power supply way

These typical power supply designs contain a large mains transformer (which also provides isolation between the input and output) and a dissipative series regulator circuit. The regulator circuit could consist of a single zener diode or a three-terminal linear series regulator to produce the required output voltage. The advantage of a linear regulator is that the power supply circuit only needs an input capacitor, output capacitor and some feedback resistors to set the output voltage.

4.5 Relay Module

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 separate low-power signal, 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.

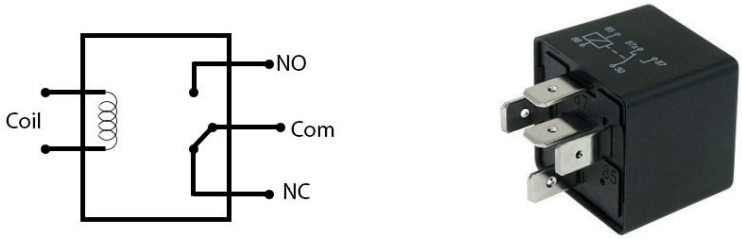


Figure 4.8: Relay

A type of relay that can handle the high power required to directly control an electric motor or other loads is called a contactor. Solid-state relays control power circuits with no moving parts, instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "protective relays".

Magnetic latching relays require one pulse of coil power to move their contacts in one direction, and another, redirected pulse to move them back. Repeated pulses from the same input have no effect. Magnetic latching relays are useful in applications where interrupted power should not be able to transition the contacts. Magnetic latching relays can have either single or dual coils. On a single coil device, the relay will operate in one direction when power is applied with one polarity, and will reset when the polarity is reversed. On a dual coil device, when polarized voltage is applied to the reset coil the contacts will transition. AC controlled magnetic latch relays have single coils that employ steering diodes to differentiate between operate and reset commands. The relay module is the one in the figure below.

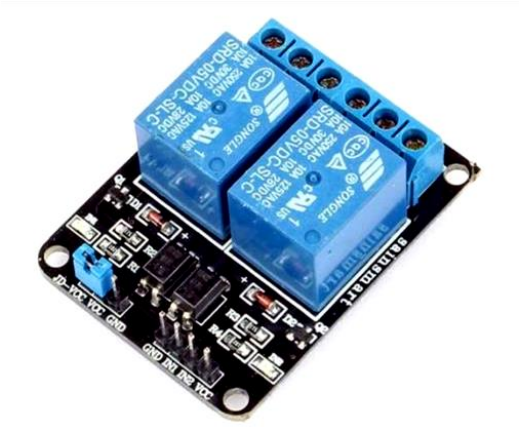


Figure 4.9: Relay Module.

This module has two channels (those blue cubes). There are other varieties with one, four and eight channels.

Main's voltage connections.

In relation to mains voltage, relays have 3 possible connections:

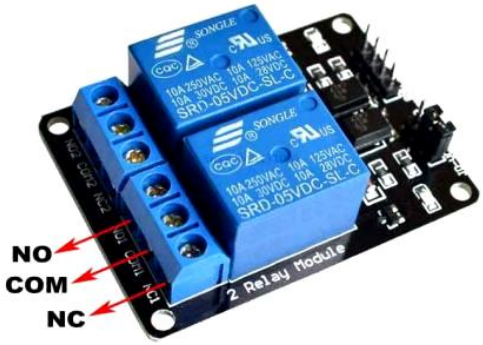


Figure 4.10: Main's voltage connections.

COM: Common pin

NO (Normally Open): There is no contact between the common pin and the normally open pin. So, when you trigger the relay, it connects to the COM pin and supply is provided to a load

NC (Normally Closed): There is contact between the common pin and the normally closed pin. There is always connection between the COM and NC pins, even when the relay is turned off. When you trigger the relay, the circuit is opened and there is no supply provided to a load.



Figure 4.11: Voltage Connections

Pin wiring

The connections between the relay module and the Arduino are really simple:

GND: Goes to ground

IN: Controls the first relay (it will be connected to an Arduino digital pin)

IN2: Controls the second relay (it should be connected to an Arduino digital pin if you are using this second relay. Otherwise, you don't need to connect it)

VCC: Goes to 5V

SOFTWARE

4.5 Arduino IDE

The digital microcontroller unit named as Arduino UNO can be programmed with the Arduino software IDE. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino UNO from the Tools, Board menu (according to the microcontroller on our board). The IC used named as ATmega328 on the Arduino UNO comes pre burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.The Arduino UNO is one of the latest digital microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .in file is

required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Uno digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses argued. As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

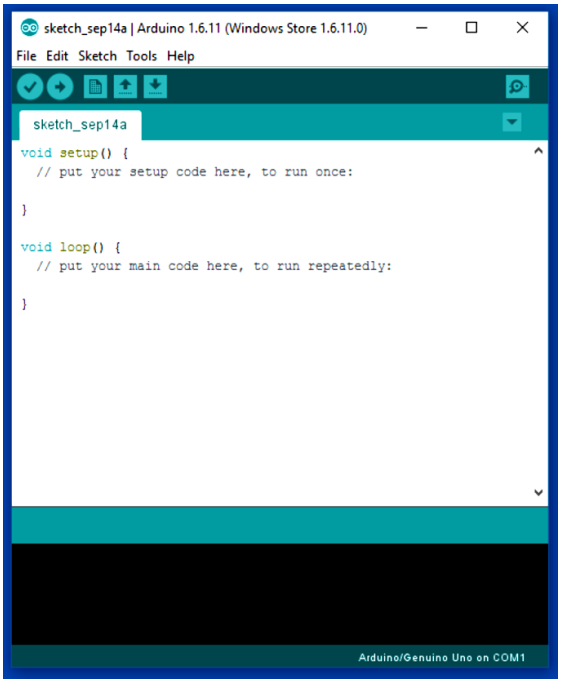


Figure 4.12: Arduino Software Interface IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the .pde extension. You may still open .pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be

/dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino boot loader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The boot loader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

Serial Monitor

This displays serial sent from the Arduino or Genuino board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial.begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

4.7 Easy EDA

Easy EDA is a web-based EDA tool suite that enables hardware engineers to design, simulate, share-publicly and privately-and discuss schematics, simulations and printed circuit boards. Other features include the creation of a bill of materials, Gerber files and pick and place files and documentary outputs in PDF, PNG and SVG formats. Easy EDA allows the creation and editing of schematic diagrams, SPICE simulation of mixed analogue and digital circuits and the creation and editing of printed circuit board layouts and, optionally, the manufacture of printed circuit boards.

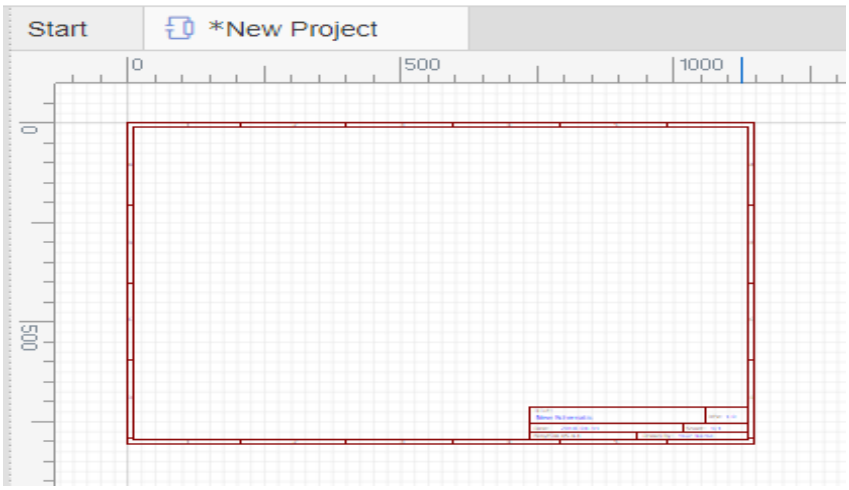


Figure 4.13: Easy EDA Software Interface

Subscription-free membership is offered for public plus a limited number of private projects. The number of private projects can be increased by contributing high quality

public projects, schematic symbols, and PCB footprints and/or by paying a monthly subscription.Registered users can download Gerber files from the tool free of charge; but for a fee, Easy EDA offers a PCB fabrication service. This service is also able to accept Gerber file inputs from third party tools.

CHAPTER 5
RESULT & DISCUSSION

5.1 Result Analysis

After finally completing this project, we ran it & we observed the output of this project. We can see that it is working well as expected. After making our project we observe it very careful. It works as we desire. Our project give output perfectly and all equipment are work perfectly. We check how much it works and we get perfect output from this project.

- Finally, we have completed our project successfully & check our project its run accurately according to our objective.
- At first, we start our system.
- Following are The outputs of our project.
- SMPS provided consistent DC output for reliable system performance.
- NodeMCU accurately monitored and processed load conditions.
- Relay operated smoothly to connect/disconnect the load as per logic.
- Power consumption reduced by automatically cutting off idle or excess load.
- NodeMCU enabled wireless control, enhancing system flexibility.
- Demonstrated suitability for homes, small offices, and smart energy systems.

5.2 Final Project

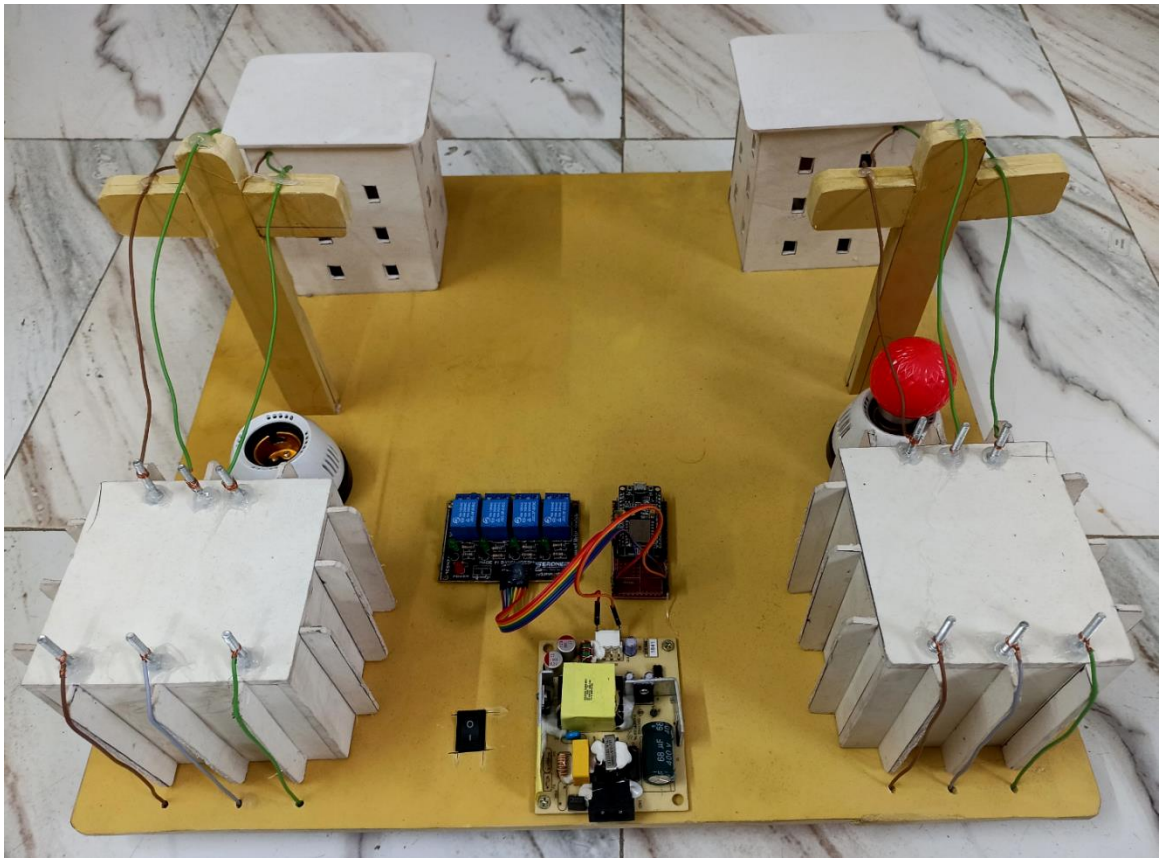


Figure 5.1: Complete Project Prototype

5.3 Advantages

There are certainly many advantages of our project and some of the major ones have been given below

- Energy Efficiency
- Automatic Load Disconnection
- Overload Protection
- Smart Monitoring
- Remote Control (Server-Based)
- System Reliability
- Cost-Effective Design
- Compact Hardware
- User Safety
- Low Power Consumption
- Real-Time Response

5.4 Application

The application areas for this project in this modern and practical world are huge and some of these are given below:

- Smart Homes
- Energy Management Systems
- Residential Load Control
- Commercial Buildings
- Industrial Automation
- Remote Load Switching
- Power Distribution Panels

5.5 Limitation

It is a demo project so we found some limitation. In future we will work for reduce this kind of limitation. These limitations are –

- NodeMCU requires a stable internet connection for remote operation.
- This project can now be only used for small scale purposes
- No energy backup.
- Used Most of Chinese products for the prototype so there's some processing delay present in the circuit.

5.6 Discussion

While working on our project, we did face some difficulties as it is a very complex system but the end results, we came up with were quite satisfactory. We have put the whole system through several tasks to validate our work and also have taken necessary notes for future improvements. Some future recommendations that we have involves improvement in system design and wiring, adding features for more efficient.

IoT Based Consumer Line Controlling System project effectively demonstrates energy conservation by automatically disconnecting loads during faults or idle conditions. Using NodeMCU, relay, and SMPS, the system ensures reliable operation and remote control. It offers a practical solution for smart energy management in homes and offices, though it has limitations in scalability, load capacity, and dependency on internet connectivity.

CHAPTER 6
CONCLUSION

6.1 Conclusion

In conclusion, the "Energy Generation Using Speed Breakers" project successfully demonstrated the potential of harnessing mechanical energy from vehicle motion to generate electricity. The system utilized a gear motor to capture energy during vehicle compression on speed breakers, which was then stored in a battery for later use. Wireless energy transfer between transmitter and receiver coils effectively allowed the stored energy to be distributed for practical applications, such as powering streetlights and charging small vehicles. The project highlighted the feasibility of utilizing renewable energy sources in high-traffic areas, offering an eco-friendly solution to meet local energy demands. By reducing reliance on conventional energy sources and minimizing environmental impact, this system represents an innovative step toward sustainable energy solutions in urban settings. Overall, the project has successfully met its objectives, contributing to the broader goal of energy conservation and promoting the adoption of green technologies for future infrastructure development.

5.2 Future Scope

We are thinking about adding many features to our project in the future to get more desirable out comes. Some of the steps that we are thinking about taking are given below:

- In future development this project can be develop by Microcontroller Unit .
- In future, we will try it to apply in real field.
- In Future we will add extra power system.

REFERENCE

[1] A. Kumar and R. Jain, "Smart Disconnector for Consumer Line Protection," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE)*, vol. 7, no. 3, 2018. [Online]. Available: https://www.ijareeie.com

[2] N. Sharma and M. Bhardwaj, "IoT-Based Energy Monitoring and Control System," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 4, pp. 1120–1124, 2019. [Online]. Available: https://www.irjet.net

[3] R. Patel and D. Chauhan, "Design and Implementation of Automatic Power Line Disconnector using Arduino," in *Proc. IEEE Conf. Smart Energy Grid Engineering*, 2020. [Online]. Available: https://ieeexplore.ieee.org

[4] R. Selvakumar, M. Kumar, and S. R. Menon, "Smart Energy Meter with Auto Disconnection Feature," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 6, pp. 78–82, 2021. [Online]. Available: https://www.ijeat.org

[5] Y. Zhang and J. Wu, "A Novel Energy-Saving Circuit Breaker for Smart Grids," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2354–2362, 2017. [Online]. Available: https://ieeexplore.ieee.org

[6] D. Tandel and K. Mehta, "GSM-Based Consumer Line Disconnection System," *International Journal of Scientific Research in Engineering and Management (IJSREM)*, vol. 4, no. 9, pp. 90–94, 2022. [Online]. Available: https://ijsrem.com

[7] A. Chakraborty and P. Sinha, "Energy Efficient Power Line Monitoring with Fault Isolation," in *Lecture Notes in Electrical Engineering*, vol. 658, Springer, 2020, pp. 113–122. [Online]. Available: https://link.springer.com

[8] T. Ramesh, A. R. Shankar, and M. B. Raj, "Home Automation with Energy Optimization using IoT," *International Journal of Computer Applications*, vol. 182, no. 24, pp. 5–9, 2018. [Online]. Available: https://www.ijcaonline.org

[9] S. Rathod and V. Naik, "Automatic Line Disconnection System for Smart Homes," *International Journal of Engineering Research & Technology (IJERT)*, vol. 12, no. 1, pp. 15–18, 2023. [Online]. Available: https://www.ijert.org

[10] P. Mishra and V. Raj, "Consumer Protection Using Smart Disconnectors in Urban Power Systems," in *IEEE Power & Energy Society General Meeting*, 2019. [Online]. Available: https://ieeexplore.ieee.org

APPENDIX

Program Code:

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <EEPROM.h>

ESP8266WebServer server(80);

const int relayPins[4] = {5, 4, 14, 12};  // D1, D2, D5, D6
const char* meterNumbers[4] = {"1", "2", "3", "4"};
bool relayStatus[4];
unsigned long relayTimers[4] = {0, 0, 0, 0};

void saveRelayState() {
  for (int i = 0; i < 4; i++) {
    EEPROM.write(i, relayStatus[i] ? 1 : 0);
  }
  EEPROM.commit();
}

void loadRelayState() {
  for (int i = 0; i < 4; i++) {
    relayStatus[i] = EEPROM.read(i) == 1;
    digitalWrite(relayPins[i], relayStatus[i] ? LOW : HIGH);
  }
}

void setup() {
  Serial.begin(115200);
  delay(5000);

  EEPROM.begin(16);
  WiFi.softAP("MeterControlSystem", "12345678");

  for (int i = 0; i < 4; i++) {
    pinMode(relayPins[i], OUTPUT);
  }
  loadRelayState();

  server.on("/", handleRoot);
  server.on("/control", handleControl);
  server.on("/status", handleStatus);
  server.begin();
}
```

```
void loop() {
  server.handleClient();
  for (int i = 0; i < 4; i++) {
    if (relayTimers[i] > 0 && millis() >= relayTimers[i]) {
      relayStatus[i] = !relayStatus[i];
      digitalWrite(relayPins[i], relayStatus[i] ? LOW : HIGH);
      relayTimers[i] = 0;
      saveRelayState();
    }
  }
}

void handleRoot() {
  String html = "<html><head><title>Meter Control System</title>";
  html += "<style>body{font-family:sans-serif;text-align:center;}";
  html += ".box{border:1px solid #ccc;padding:10px;margin:10px;display:inline-
block;width:250px;}";
  html += ".btn{padding:8px;margin:5px;cursor:pointer;}</style></head><body>";
  html += "<h2>Power Saver Consumer Line Disconnector</h2>";

  for (int i = 0; i < 4; i++) {
    String title;
    String placeholderText;

    if (i == 0) {
      title = "Feeder 1";
      placeholderText = "Feeder Number";
    } else if (i == 1) {
      title = "Feeder 2";
      placeholderText = "Feeder Number";
    } else if (i == 2) {
      title = "Consumer 1";
      placeholderText = "Meter Number";
    } else {
      title = "Consumer 2";
      placeholderText = "Meter Number";
    }

    html += "<div class='box'><h3>" + title + "</h3>";
    html += "<p>Status: <b id='status" + String(i) + "'></b></p>";
    html += "<input type='text' id='meter" + String(i) + "' placeholder='" +
placeholderText + "'><br>";
    html += "<input type='number' id='sec" + String(i) + "' placeholder='Seconds'
min='0'><br>";
    html += "<input type='number' id='min" + String(i) + "' placeholder='Minutes'
min='0'><br>";
    html += "<input type='number' id='hour" + String(i) + "' placeholder='Hours'
min='0'><br>";
    html += "<button class='btn' onclick='control(" + String(i) +
",1)'>Connect</button>";
```

```
        html += "<button class='btn' onclick='control(" + String(i) +
",0)'>Disconnect</button>";
        html += "</div>";
    }

    html += "<script>function control(r, action) {";
    html += "var meter = document.getElementById('meter'+r).value;";
    html += "var sec = document.getElementById('sec'+r).value || 0;";
    html += "var min = document.getElementById('min'+r).value || 0;";
    html += "var hour = document.getElementById('hour'+r).value || 0;";
    html += "var time = (hour*3600 + min*60 + parseInt(sec)) * 1000;";
    html += "fetch('/control?r=' + r + '&a=' + action + '&m=' + meter + '&t=' + time)";
    html += ".then(res => res.text()).then(res => {alert(res); updateStatus();});";
    html += "}";
    html += "function updateStatus() { fetch('/status').then(res => res.json()).then(data =>
{";
    html += "for (let i = 0; i < 4; i++) { document.getElementById('status' + i).innerText =
data[i] ? 'OFF' : 'ON'; }";
    html += "});}";
    html += "updateStatus(); setInterval(updateStatus, 2000);";
    html += "</script></body></html>";

    server.send(200, "text/html", html);
}



void handleControl() {
    int relay = server.arg("r").toInt();
    int action = server.arg("a").toInt();
    String meter = server.arg("m");
    unsigned long delayTime = server.arg("t").toInt();

    if (relay < 0 || relay >= 4) {
        server.send(400, "text/plain", "Invalid Relay");
        return;
    }

    if (meter != meterNumbers[relay]) {
        server.send(403, "text/plain", "Incorrect Meter Number!");
        return;
    }

    if (delayTime > 0) {
        relayTimers[relay] = millis() + delayTime;
    } else {
        relayStatus[relay] = (action == 0);
        digitalWrite(relayPins[relay], relayStatus[relay] ? LOW : HIGH);
        saveRelayState();
    }
```

```
    server.send(200, "text/plain", "Relay " + String(relay+1) + " will " + (action ? "Connect" : "Disconnect") + (delayTime > 0 ? " in " + String(delayTime/1000) + " seconds." : " Now."));
}

void handleStatus() {
    String json = "[";
    for (int i = 0; i < 4; i++) {
        json += relayStatus[i] ? "1" : "0";
        if (i < 3) json += ",";
    }
    json += "]";
    server.send(200, "application/json", json);
}
```