IoT-BASED SMART HOME SECURITY AND HOME AUTOMATION SYSTEM

A main project report submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted By

SK. SHAHABAJ	**20471A0249**
SK. MD. MOHIDDIN BASHA	**20471A0247**
S. MAHESHWARI	**20471A0243**
B. VENKATESWARA RAO	**20471A0203**

Under the Esteemed Guidance of

Mr. SK. ABDUL KALAM M.Tech,(Ph.D)

Asst. Professor, Dept. of EEE

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING
NARASARAOPETA ENGINEERING COLLEGE(AUTONOMOUS)
(Accredited ‘A^{+}’ Grade by NAAC, NBA & ISO 9001:2015 Certified institution)
(Approved by AICTE, New Delhi &Permanent Affiliated to JNTUK, Kakinada)
NARASARAOPET-522601(A.P)
2023-2024

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

(Accredited ‘A^{+}’ Grade by NAAC, NBA & ISO 9001:2015 Certified institution)

(Approved by AICTE, New Delhi &Permanent Affiliated to JNTUK Kakinada)

NARASARAOPET-522601(A.P)

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

CERTIFICATE

This is to certify that the Main Project entitled **“**IoT-Based Smart Home Security and Home Automation System**”** is a bonafide work carried out by **SK. SHAHABAJ (20471A0249), S. MAHESHWARI (20471A0243), SK. MD. MOHIDDIN BASHA (20471A0247), B. VENKATESWARA RAO (20471A0203)** under our supervision and guidance in partial fulfillment of requirements for the award of Degree of BACHELOR OF TECHNOLOGY in ELECTRICAL & ELECTRONICS ENGINEERING, from JNTUK, Kakinada, Andhra Pradesh, during the year 2023-32024.

PROJECT GUIDE

Mr. SK. ABDUL KALAM M. Tech (Ph.D)

Asst. Professor, Dept. of EEE

Narasaraopeta Engineering College

Narasaraopet.

HEAD OF DEPARTMENT

Dr. P.LAKSHMANAN M.E,Ph.D

Professor & HOD, Dept. of EEE

Narasaraopeta Engineering College

Narasaraopet.

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

It gives us immense pleasure to express our gratitude to everyone who helped us in the successful completion of our project titled **"IoT-Based Smart Home Security and Home Automation System"**

It is with great pleasure that we acknowledge our sincere thanks and deep sense of gratitude to our guide **Mr. SK. ABDUL KALAM M. Tech (Ph.D),** Asst.Professor, Dept. of EEE, for his valuable glance through the course of this work. His unflinching help, suggestions, direction and guidance have helped in the project work.

We express our sincere thanks to **Dr.P.LAKSHMANAN M.E., Ph.D,** Professor & Head of the Dept, of Electrical and Electronics Engineering for his indispensable encouragement to complete the project work.

We convey our special thanks to our beloved and honorable Principal **Dr.M.SREENIVASA KUMAR, M.TECH.,Ph.D.(UK),MISTE,FIE(I)** and to our college management for providing excellent lab Facilities for the completion of project with in our campus. Finally, we express indebtedness to everyone remotely involved with this project.

We are very much gratitude to all faculty members of the Department for their kind cooperation without which this work could not have been to this shape.

Project Associates

SK. SHAHABAJ	**20471A0249**
SK. MD. MOHIDDIN BASHA	**20471A0247**
S. MAHESHWARI	**20471A0243**
B. VENKATESWARA RAO	**20471A0203**

INDEX

ABSTRACT

Home security based on the Internet of Things (IoT) has been getting huge attention of mass people in recent years. Smart home eases and secures the management of home appliances. This project's main aim is a low-cost and reliable smart home system that assists users in managing home appliances without the need for their physical presence. It can store and display information on the temperature & humidity of a home and notifies the users of switching on/off time of light, fan, and other home appliances using the IoT platform.

The system includes gas leakage & fire alarms. It has a leakage gas removal & fire extinguishing facility and notification system using an IT platform. The system uses real IP and RESTful API for controlling, monitoring, and accessing home appliances remotely from anywhere in the world using an Android-based smartphone app or web app. This system is user-friendly and energy-efficient.

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1

1.1 Introduction

Just imagine, how beneficial it will be to be able to switch on our air condition for half an hour before we reach our home in the summertime. When we leave our home for some work without realizing that some appliances such as fans, air conditioners, and lights are on; then by using our mobile phone or internet, we are able to turn off power to those devices. It will be even more useful if the system detects unauthorized movement in the house and alerts us or sends messages on our mobile phones or we can know the status of our house anytime. Such systems provide security from natural, incidental, intended, unintended, accidental and human made problems by continuously monitoring homes with different sensory systems like motion, smoke, gas, temperature, glass break or door break detectors and fire alarm systems. Security is a big challenge everywhere because thefts are increasing day by day owing to the unsafe and insecure security systems in homes, commercial complexes and industries. Several conventional technologies are available to keep home properties safe from intruders, but most common smart home security systems work on wireless GSM communication.

1.2 Literature review

Vinay Sagar K N (2016) et al present a low-cost cost flexible and reliable home automation system with additional security using an Arduino microcontroller, with IP connectivity through local Wi-Fi for accessing and controlling devices by authorized users remotely using a Smart smartphone application. The proposed system is server-independent and uses the Internet of Things to control human-desired appliances starting from industrial machines to consumer goods. The user can also use different devices for controlling with the help of a web browser, smartphone, or IR remote module. To demonstrate the effectiveness and feasibility of this system, in this paper, we present a home automation system using Arduino UNO microcontroller and esp8266-01 as a connectivity module. It helps the user to control various appliances such as lights, fans, and TVs and can make decisions based on the feedback of sensors remotely.

Pooja N. Pawar (2018) et al designed a people prefer more automatic systems rather than manual systems. With the influence of the Internet in people's life lots of new technologies are coming up. One of the latest, emerging, and trending technologies is the ‘Internet of Things’. This technology is expected to rule the world within a few years. Home Automation System uses the technology of the Internet of Things for monitoring and controlling the electrical and electronic appliances at home from any remote location by simply using a Smartphone. The implementation of a low-cost, flexible home automation system is presented. It enhances the use of wireless communication which provides the user with remote control of various electronic and electrical appliances.

Shweta Singh (2017) et al proposed home automation with the proliferation of IoT is becoming a reality now, and a variety of players like, Apple, Amazon, Google, and Samsung, are all converging into this space to provide the platform and solutions for smart homes. In Light of this, the present study addresses IoT concepts through a systematic review of scholarly research papers, corporate white papers, professional discussions with experts, and online databases. The main objective of this paper is to provide an overview of the Internet of Things, architectures, and vital technologies and their usages in our daily life

K. Saiteja (2017) et al develop a system that will provide remote control of home appliances and also provide security against mishaps when the host is not at home. This paper is mainly concerned with the automatic control of light or any other home appliances using the internet. It is meant to save the electric power and human energy. This application is made with the help of the Internet of Things and Raspberry Pi. The various appliances connected to the Raspberry Pi is using the wireless network.

Priyanka Zambare (2018) et al designed an IoT that has nowadays become an emerging and trending technology. It is a system of physical things embedded with sensors, software, electronics, and connectivity to allow it to perform better by exchanging information with other connected devices, the operator, or the manufacturer. Home automation based on IoT allows users to access and control various home applications remotely using smartphones. It is mainly useful for physically disabled people and also to provide security to our house. It improves the standard and quality of people’s lives. And also make our home and life safer.

1.3 Existing System

Wireless devices monitoring and controlling systems is a means that allows users to control electric appliances of varying kinds. Bluetooth and RF Communications have drawbacks in that they can control from 10 meters and 30 meters long range, whereas coming to GSM technology we can only control the home appliances whenever the Network coverage is good and also it should contain the SMS balance to control. Where coming to IoT is a system that uses computers or mobile devices to control basic home functions and features automatically through the Internet from anywhere around the world using mobile Internet or WIFI

CHAPTER-2

PROPOSED SYSTEM

The Internet of thing is a growing network of everyday objects, from industrial machines to consumer goods that can share information and complete tasks while you are busy with other activities Because of the advanced development in computer technology, microprocessors are not only on the desktop but also exist everywhere wireless devices monitoring and controlling Swallows us to control household appliances like light, door, fan, AC, etc. It also provides wireless devices monitoring and controlling an emergency system to be activated. wireless device monitoring and controlling not only refers to reducing human efforts but also energy efficiency and time-saving. Microprocessors are embedded in electronic appliances in our homes today. In the past, the appliances are working on standalone and cannot cooperate. But in recent years, these appliances can be monitored and controlled by embedded microprocessors and displayed on terminals.

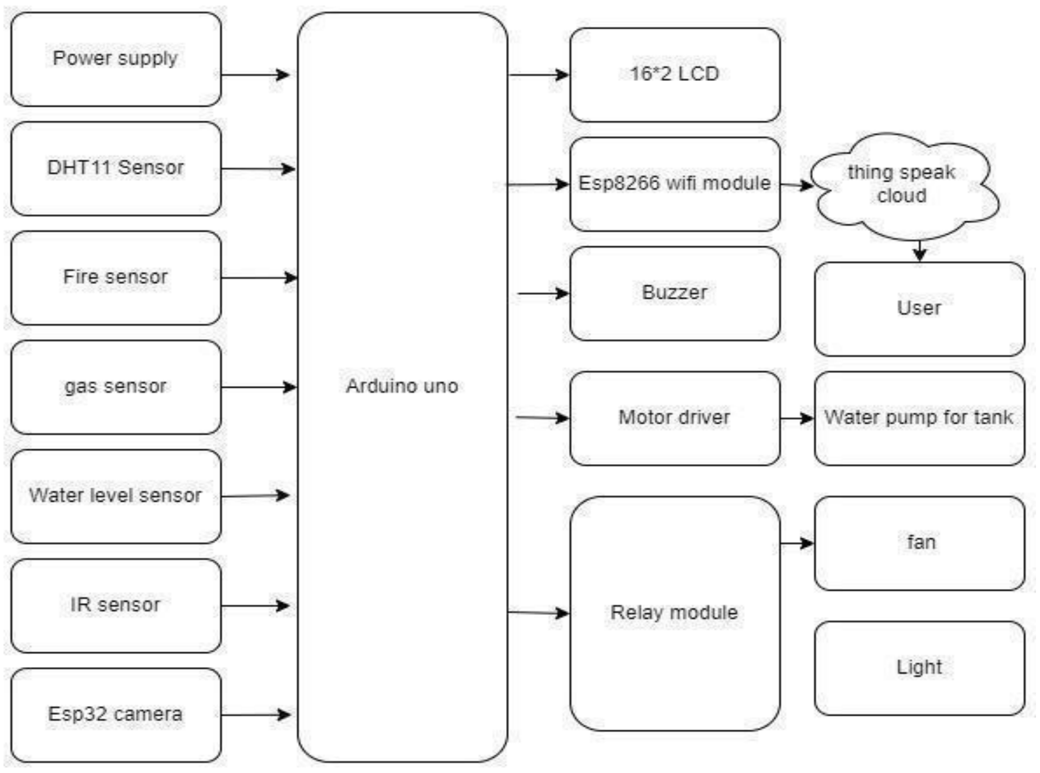


Fig.2.1: Block Diagram of Proposed System

2.1 Description

Monitoring and controlling wireless devices with IoT and Arduino involves using an Arduino board connected to sensors and actuators. The Arduino collects data from sensors and communicates with the IoT platform over the internet using Wi-Fi or other wireless modules. The IoT platform processes and stores the data, allowing remote monitoring through a web or mobile app. Control signals can also be sent to the Arduino to actuate devices. Security measures are implemented to protect data and ensure the system's reliability.

2.2 ADVANTAGES AND APPLICATIONS

- Energy Savings. Home automation systems have proven themselves in the arena of energy efficiency. ...
- Convenience. In today's fast-paced society, the less you have to worry about, the better. ...
- Security. ...
- Installation. ...
- Complex Technology. ...
- System Compatibility. ...
- Cost.

Applications of the wireless devices monitoring and controlling system:

1. **Smart Home Automation:** IoT and Arduino enable remote control of lights, thermostats, and appliances for energy savings and convenience.
2. **Industrial IoT:** Monitoring and controlling machinery and equipment in factories for predictive maintenance and increased productivity.
3. **Agricultural Automation:** Monitoring soil conditions and automating irrigation for efficient farming practices.

4. **Healthcare:** Tracking vital signs and medication adherence for remote patient monitoring and telemedicine.

5. **Environmental Monitoring:** Measuring air quality, water quality, and weather conditions for better environmental management.

6. **Traffic Management:** Optimizing traffic signals and monitoring vehicle flow for smoother urban transportation.

7. **Energy Management:** Monitoring energy consumption in buildings and optimizing usage for cost savings.

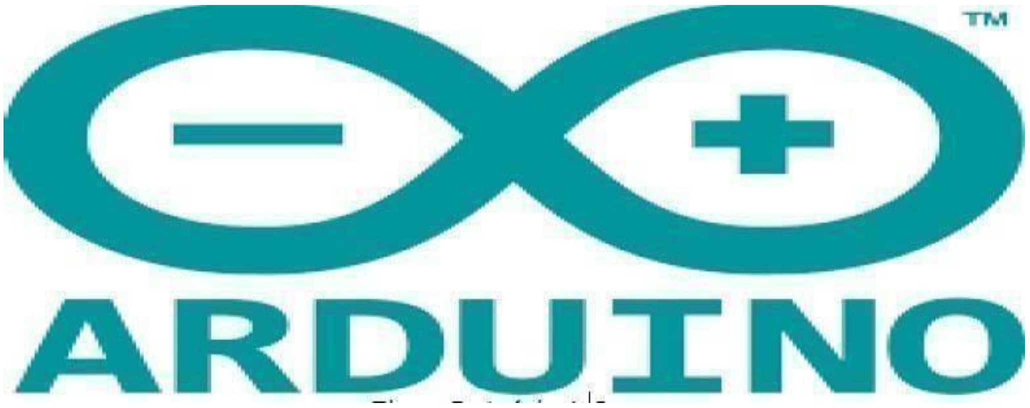
8. **Wearable Devices:** Tracking fitness metrics and health data for personal well-being.

9. **Security Systems:** Integrating surveillance cameras and access control for enhanced security in homes and businesses.

CHAPTER-3

HARDWARE DESCRIPTION

Arduino is open source physical processing which is based on a microcontroller board and an incorporated development environment for the board to be programmed. Arduino gains a few inputs, for example, switches or sensors, and controls a few multiple outputs, for example, lights, engines, and others. Arduino programs can run on Windows, Macintosh, and Linux operating systems (OS) opposite to most microcontrollers' frameworks which run only on Windows. Arduino programming is easy to learn and apply to beginners and amateurs. Arduino is an instrument used to build a better version of a computer that can control, interact, and sense more than a normal desktop computer. It's an open-source physical processing stage focused around a straightforward microcontroller board, and an environment for composing programs for the board. Arduino can be utilized to create interactive items, taking inputs from a diverse collection of switches or sensors, and controlling an assortment of lights, engines, and other physical outputs. Arduino activities can remain solitary, or they can be associated with programs running on your machine (e.g. Flash, Processing, and Maxmsp.) The board can be amassed by hand or bought preassembled; the open-source IDE can be downloaded free of charge. Focused on the Processing media programming environment, the Arduino programming language is an execution of Wiring, a comparative physical computing platform. Figure 7- Arduino's



Why choosing Arduino

There are numerous different microcontrollers and microcontroller platforms accessible for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handy board, and numerous others offer comparative usefulness. These apparatuses take the chaotic subtle elements of microcontroller programming and wrap them up in a simple to-utilize bundle. Arduino additionally rearranges the methodology of working with microcontrollers; moreover, it offers some advantages for instructors, students, and intrigued individuals:

• Inexpensive - Arduino boards are moderately cheap compared with other microcontroller boards. The cheapest version of the Arduino module can be amassed by hand, and even the preassembled Arduino modules cost short of $50.

• Cross-platform - The Arduino programming runs multiple operating systems Windows, Macintosh OSX, and Linux working frameworks. So, we conclude that Arduino has an advantage as most microcontroller frameworks are constrained to Windows.

• Straightforward, clear programming method - The Arduino programming environment is easy to use for novices, yet sufficiently versatile for cutting-edge customers to adventure as well. For educators, it favourably engaged around the Processing programming environment, so
understudies finding ways to understand how to program in that environment will be familiar
with the nature of Arduino.

Open source and extensible programming. The Arduino program language is available as open source, and available for development by experienced engineers. The lingo can be reached through C++ libraries, and people expecting to understand the specific purposes of different interests can make the leap from Arduino to the AVR C programming language on which it is based. You can incorporate AVR-C code clearly into your Arduino programs if you have to.

Open source and extensible hardware - The Arduino is concentrated around Atmel's Atmega8 and Atmega168 microcontrollers. The plans for the modules are circulated under a Creative Commons license, so experienced circuit designers can make their own particular interpretation of the module, extending it and improving it. slightly inexperienced customers can build the breadboard variation of the module remembering the finished objective to perceive how it capacities and save money.

3.1 Arduino UNO:

The Arduino UNO is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 Analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. "Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform;

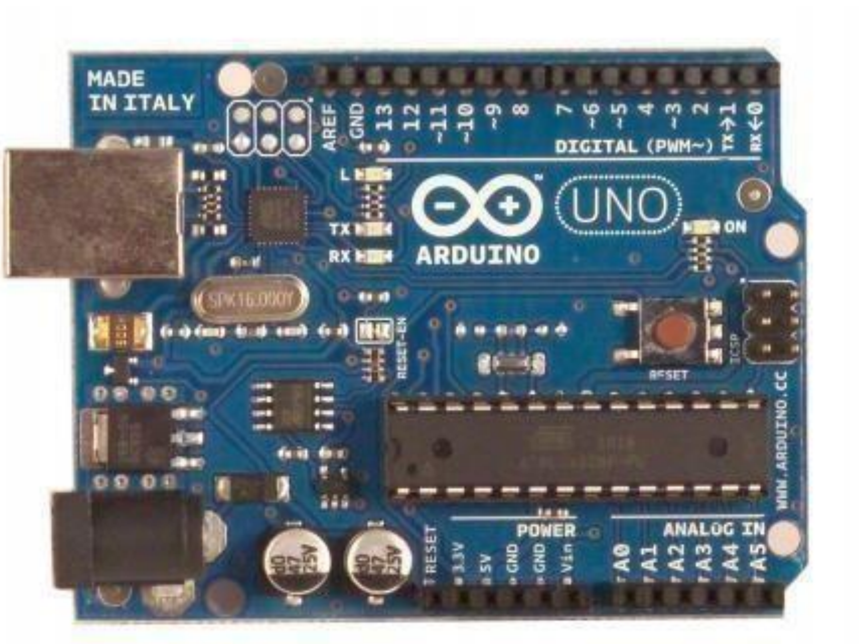


Fig.3.1(a): Arduino Uno

3.1.1 Technical specifications of Arduino:

Microcontroller: ATmega328

Operating Voltage: 5V

Input Voltage (recommended): 7-12V

Input Voltage (limits): 6-20V

Digital I/O Pins 14 (of which 6 provide PWM output)

Analog Input Pins 6

DC Current per I/O Pin 40 mA

DC Current for 3.3V Pin 50 mA

Flash Memory

32 KB of which 0.5 KB was used by

bootloader

SRAM 2 KB

EEPROM 1 KB

Clock Speed 16 MHz

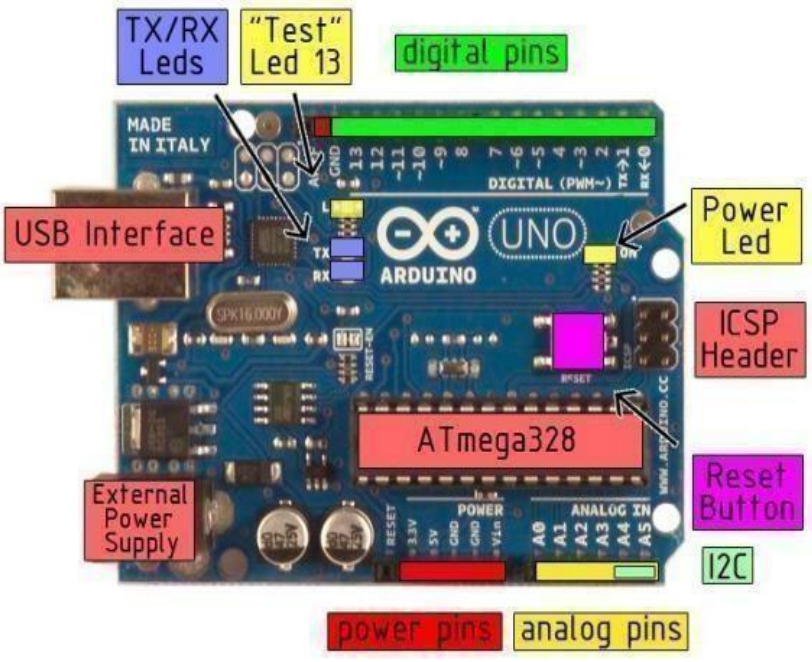


Fig.3.1(b): Arduino UNO

POWER

The Arduino Uno can be powered via a USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or a battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the GND and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

• **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

• **5V**. The regulated power supply is used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

• **3V3**. A 3.3-volt supply is generated by the on-board regulator. Maximum current draw is 50 mA.

• **GND**. Ground pins.

MEMORY:

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

INPUT/OUTPUT

Each of the 14 digital pins on the Uno can be used as an input or output, using PinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 KOhms. In addition, some pins have specialized functions:

• Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

• External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

• PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

• SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

• LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 Analog inputs, each of which provides 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function? Additionally, some pins have specialized functionality:

• I 2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library. There are a couple of other pins on the board:

• AREF. Reference voltage for the Analog inputs. Used with analogReference().

• Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields that block the one on the board.

Communication:

The Arduino Uno has several facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer. The '8U2 firmware uses the standard

USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required.

The Arduino software includes a serial monitor that 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 UNO's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify the use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software. The ATmega328 on the Arduino Uno comes pre-burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C headerfiles). You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, 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. 10 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. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nano farad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data. The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110-ohm resistor from 5V to the reset line. 11

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100-mil spacing of the other pins.

3.2 LCD:

3.2.1 LCD 16×2 Pin Configuration and Its Working

Nowadays, we always use devices that are made up of LCDs such as CD players, DVD players, digital watches, computers, etc. These are commonly used in the screen industries to replace the utilization of CRTs. Cathode Ray Tubes use huge power when compared with LCDs, and CRTs are heavier as well as bigger. These devices are thinner as well power consumption is extremely low. The LCD 16×2 working principle is, it blocks the light rather than dissipates it. This article discusses an overview of LCD 16X2, pin configuration, and it's working.

What is the LCD 16×2?

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simple programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

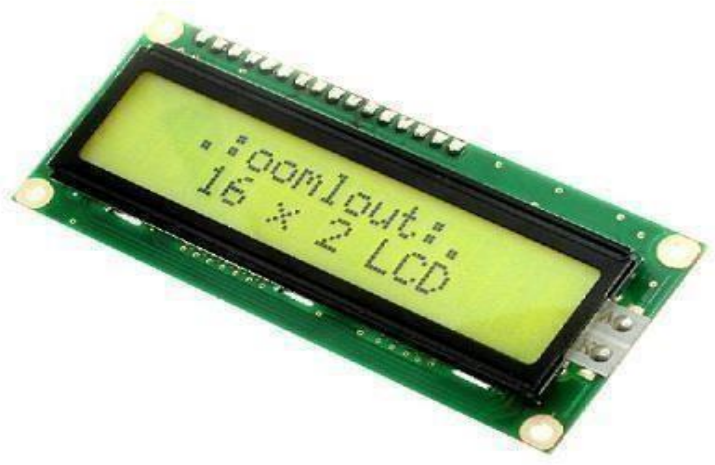


Fig.3.2(a): LCD 16X2

3.2.2 LCD 16×2 Pin Diagram

The 16×2 LCD pinout is shown below.

- Pin1 (Ground/Source Pin): This is a GND pin of display, used to connect the GND terminal of the microcontroller unit or power source.
- Pin2 (VCC/Source Pin): This is the voltage supply pin of the display, used to connect the supply pin of the power source.
- Pin3 (V0/VEE/Control Pin): This pin regulates the difference of the display, used to connect a changeable POT that can supply 0 to 5V.
- Pin4 (Register Select/Control Pin): This pin toggles among command or data register, is used to connect a microcontroller unit pin, and obtains either 0 or 1(0 = data mode, and 1 = command mode).
- Pin5 (Read/Write/Control Pin): This pin toggles the display among the read or write operation, and it is connected to a microcontroller unit pin to get either 0 or 1 (0 = Write Operation and 1 = Read Operation).
- Pin 6 (Enable/Control Pin): This pin should be held high to execute the Read/Write process, and it is connected to the microcontroller unit & constantly held high.
- Pins 7-14 (Data Pins): These pins are used to send data to the display. These pins are connected in two-wire modes like 4-wire mode and 8-wire mode. In the 4-wire mode, only four pins are connected to the microcontroller unit like 0 to 3, whereas in the 8-wire mode, 8-pins are connected to the microcontroller unit like 0 to 7.
- Pin15 (+ve pin of the LED): This pin is connected to +5V)
- Pin 16 (-ve pin of the LED): This pin is connected to GND).

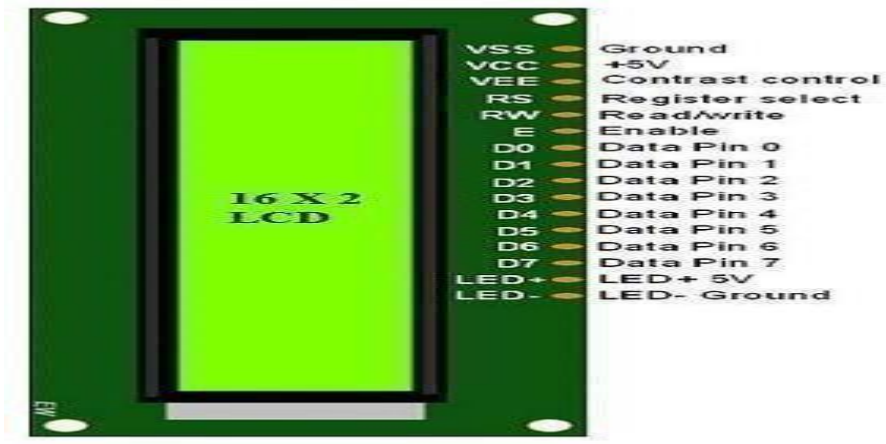


Fig.3.2(b): LCD-16×2-pin-diagram

3.2.3 Features of LCD16x2

The features of this LCD mainly include the following.

- The operating voltage of this LCD is 4.7V-5.3V
- It includes two rows where each row can produce 16 characters.
- The utilization of current is 1mA with no backlight
- Every character can be built with a 5×8 Pixel box
- The alphanumeric LCDs alphabets & numbers
- Is display can work on two modes like 4-bit & 8-bit
- These are obtainable in Blue & Green Backlight
- It displays a few custom-generated characters

3.2.4 Registers of LCD

A 16×2 LCD has two registers like data register and a command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as a command register. Similarly, when the register set is ‘1’, then it is known as a data register.

Command Register

The main function of the command register is to store the instructions of commands which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, setting the cursor place, and display control. Here command processing can occur within the register.

Data Register

The main function of the data register is to store the information that is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information that is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process starts there. When register set =1, then the data register will be selected.

3.2.5 16×2 LCD Commands

The commands of LCD 16X2 include the following.

- For Hex Code-01, the LCD command will be the clear LCD screen
- For Hex Code-02, the LCD command will be returning home
- For Hex Code-04, the LCD command will be decrement cursor
- For Hex Code-06, the LCD command will be Increment cursor
- For Hex Code-05, the LCD command will be Shift display right
- For Hex Code-07, the LCD command will be Shift display left
- For Hex Code-08, the LCD command will be Display off, cursor off
- For Hex Code-0A, the LCD command will be cursor on and display off
- For Hex Code-0C, the LCD command will be cursor off, display on
- For Hex Code-0E, the LCD command will be cursor blinking, Display on
- For Hex Code-0F, the LCD command will be cursor blinking, Display on
- For Hex Code-10, the LCD command will be to Shift the cursor position to the left
- For Hex Code-14, the LCD command will be to Shift the cursor position to the right
- For Hex Code-18, the LCD command will be Shift the entire display to the left
- For Hex Code-1C, the LCD command will be Shift the entire display to the right
- For Hex Code-80, the LCD command will Force the cursor to the beginning (1st line)
- For Hex Code-C0, the LCD command will Force the cursor to the beginning (2nd line)
- For Hex Code-38, the LCD command will be 2 lines and a 5×7 matrix

3.3 Relay

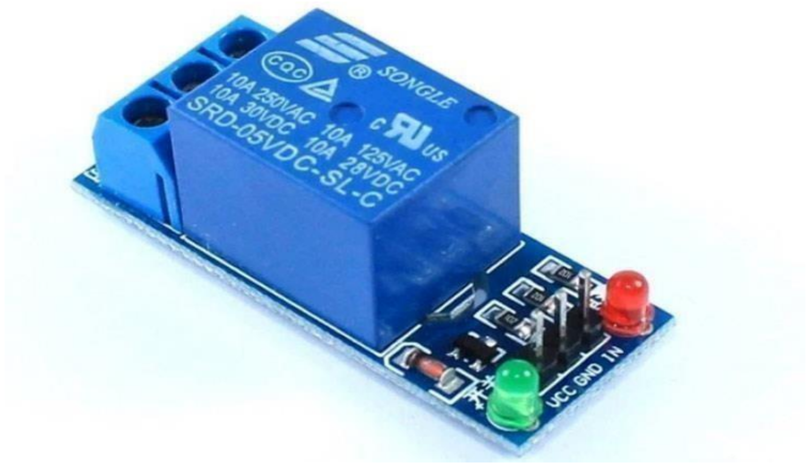


Fig.3.3(a): Relay

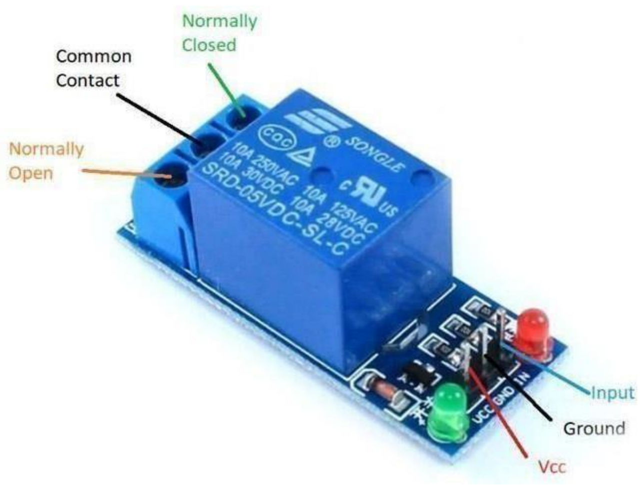


Fig.3.3(b): 5V Single-Channel Relay Module Pinout

5V Single-Channel Relay Module

Single-Channel Relay Module Pinout

A relay is an electromechanical device that uses an electric current to open or close the contacts of a switch. The single-channel relay module is much more than just a plain relay, it comprises components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not

3.3.1 Single-Channel Relay Module Pin Description

Pin Number	**Pin Name**	**Description**
1	Relay Trigger	Input to activate the relay
2	Ground	0V reference
3	VCC	Supply input for powering the relay coil
4	Normally Open	Normally open terminal of the relay
5	Common	The common terminal of the relay
6	Normally Closed	Normally closed contact of the relay

Table 3.1: Single-Channel Relay Module Pin Description

3.3.2 Single-Channel Relay Module Specifications

- Supply voltage – 3.75V to 6V
- Quiescent current: 2mA
- Current when the relay is active: ~70mA
- Relay maximum contact voltage – 250VAC or 30VDC
- Relay maximum current – 10A

3.3.3 Components Present on a 5V Single Channel Relay Module

The following are the major components present in a relay module; we will get into the details later in this article.

5V Relay, Transistor, Diode, LEDs, Resistors, Male Header pins, 3-pin screw-type terminal connector, etc.

Understanding 5V Single-Channel Relay Module

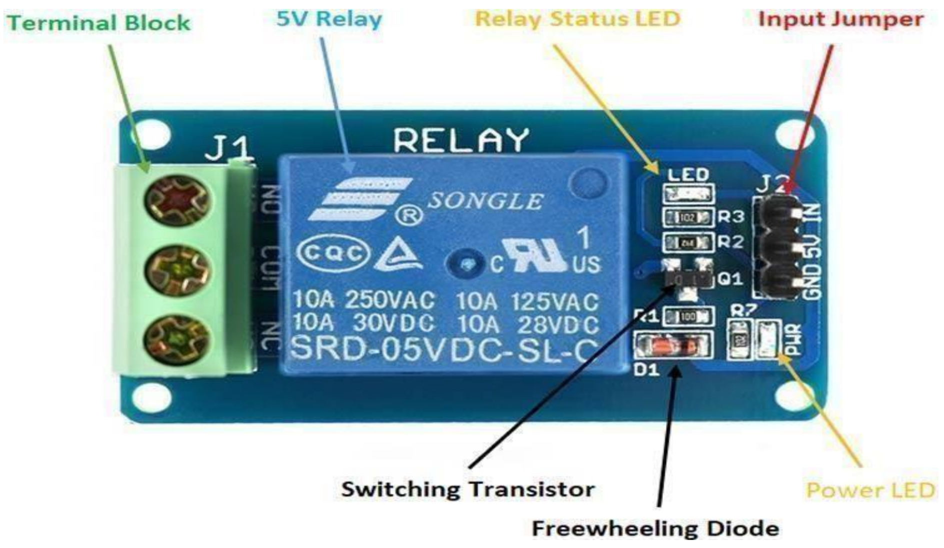


Fig.3.3(c): Understanding Single-Channel Relay

The single-channel relay module is much more than just a plain relay, it contains components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active.

First is the screw terminal block. This is the part of the module that is in contact with mains so a reliable connection is needed. Adding screw terminals makes it easier to connect thick main cables, which might be difficult to solder directly. The three connections on the terminal block are connected to the normally open, normally closed, and common terminals of the relay.

The second is the relay itself, which, in this case, is a blue plastic case. Lots of information can be gleaned from the markings on the relay itself. The part number of the relay on the bottom says “05VDC”, which means that the relay coil is activated at 5V minimum – any voltage lower than this will not be able to reliably close the contacts of the relay. There are also voltage and current markings, which represent the maximum voltage and current, the relay can switch. For example, the top left marking says “10A 250VAC”, which means the relay can switch a maximum load of 10A when connected to a 250V mains circuit. The bottom left rating says “10A 30VDC”, meaning the relay can switch a maximum current of 10A DC before the contacts get damaged.

The 'relay status LED' turns on whenever the relay is active and provides an indication of current flowing through the relay coil.

The input jumper is used to supply power to the relay coil and LEDs. The jumper also has the input pin, which when pulled high activates the relay.

The switching transistor takes an input that cannot supply enough current to directly drive the relay coil and amplifies it using the supply voltage to drive the relay coil. This way, the input can be driven from a microcontroller or sensor output. The freewheeling diode prevents voltage spikes when the relay is switched off.

The power LED is connected to V_{CC} and turns on whenever the module is powered.

3.3.4 How Does A Relay Work?

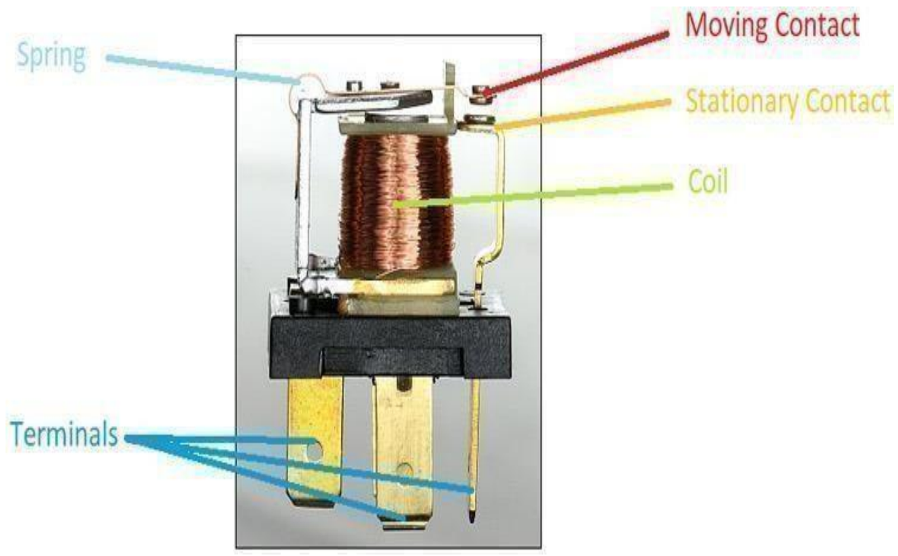


Fig.3.3(d): How Relay Works

The relay uses an electric current to open or close the contacts of a switch. This is usually done using the help of a coil that attracts the contacts of a switch and pulls them together when activated, and a spring pushes them apart when the coil is not energized.

There are two advantages of this system – First, the current required to activate the relay is much smaller than the current that relay contacts are capable of switching, and second, the coil and the contacts are galvanically isolated, meaning there is no electrical connection between them.

This means that the relay can be used to switch mains current through an isolated low-voltage digital system like a microcontroller.

3.3.5 Internal Circuit Diagram for Single Channel Relay Module

The circuit on the PCB is quite simple.

LEDs can be added to this basic circuit to act as indicators, sometimes even optical isolation is added to input to further improve isolation

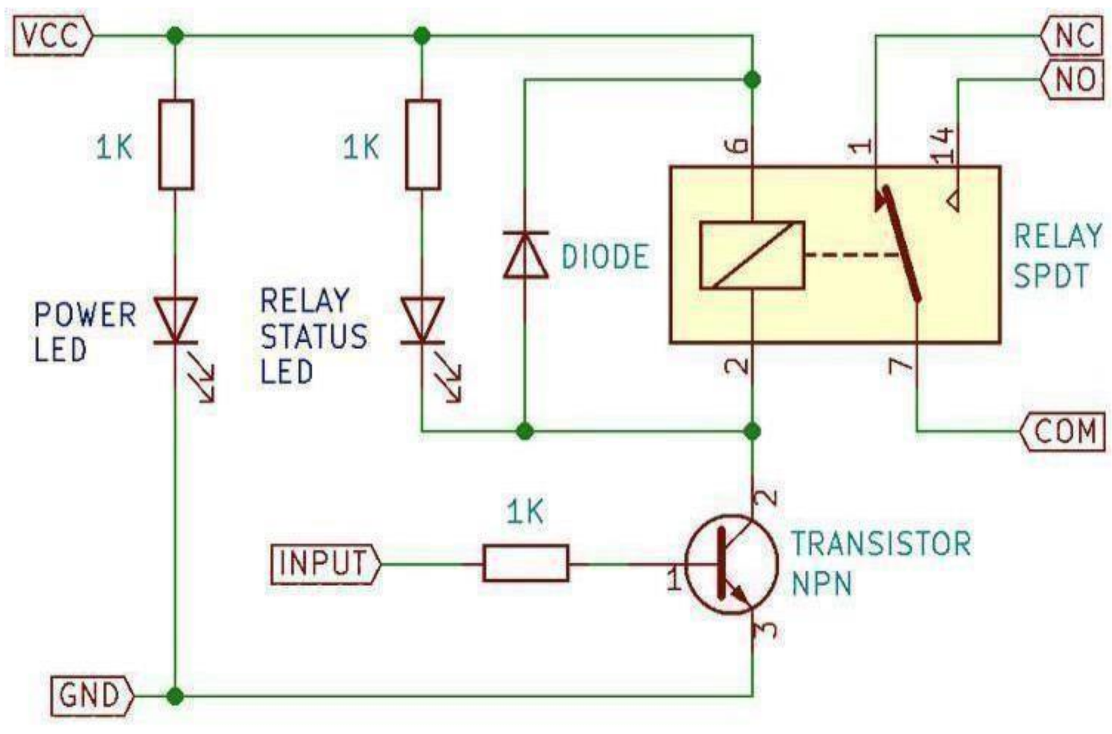


Fig.3.3(e): Internal Circuit of Single-Channel Relay

The extra components apart from the relay are there since it would not be possible to drive the relay directly from the pins of a microcontroller. digital logic or a sensor. This is because although the coil consumes much less current than the currents it can switch, it still needs relatively significant current – low-power relays consume around 50mA while higher-power relays consume around 500mA. The coil is also an inductive load, so when the coil is switched off, a large flyback voltage is developed which can damage the device by turning it on and off. For this reason, a flyback diode is added anti-parallel to the relay coil to clamp the flyback voltage.

LEDs can be added to this basic circuit to act as indicators, and sometimes even optical isolation is added to the input to further improve the isolation.

How to use Single-Channel Relay Module

Relay modules like this one are commonly used to drive mains loads from a microcontroller like the Arduino or a sensor. In cases like this, the common circuit diagram would be as follows.

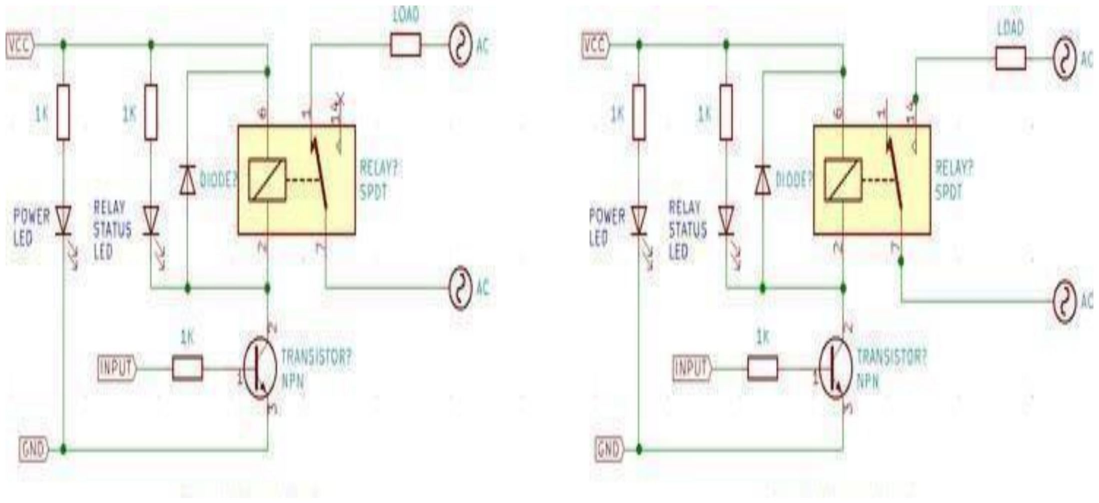


Fig.3.3(f): How to Use Single-Channel Relay

For simple on/off applications, the relay can be connected as shown above. One terminal of mains is connected to common, and the other is connected to NO or NC depending on whether the load should be connected/disconnected when the relay is active.

Check out the image below to see how the relay module is connected to a microcontroller and mains source and load.

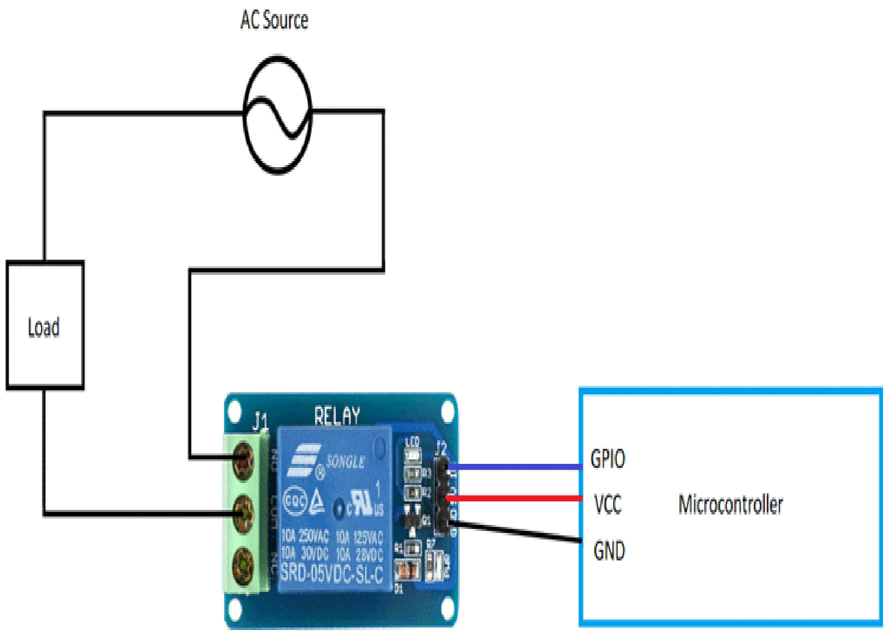


Fig.3.3(g): Relay Connected to Microcontroller

The mains wiring is screwed to the terminal block, and the microcontroller can be connected using jumper cables.

3.3.6 Single-Channel Relay Module Basic Trouble Shooting

If the relay does not switch on, i.e. no audible clicking sound is heard:

- The contacts might be stuck - Check by physically shaking the relay, if a light clicking sound is not heard, then tap the relay hard, in most cases, this should ‘unstick’ both the contacts.
- If the contacts do click when the relay is shaken, then the transistor or the flyback diode might be damaged and must be replaced.

Single-Channel Relay Module Applications

- Mains switching
- High current switching
- Isolated power delivery
- Home automation

3.4 BUZZER

Fig.3.4(a): Active Passive Buzzer

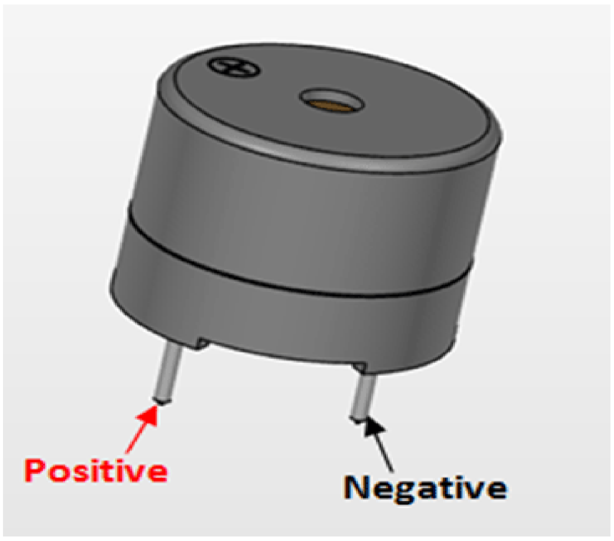


Fig.3.4(b): Active Passive Buzzer Pinout

3.4.1 Buzzer Pin Configuration

Pin Number	**Pin Name**	**Description**
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

Table 3.2: Buzzer Pin Configuration

3.4.2 Buzzer Features and Specifications

- Rated Voltage: 6V DC
- Operating Voltage: 4-8V DC
- Rated current: <30mA
- Sound Type: Continuous Beep
- Resonant Frequency: ~2300 Hz
- Small and neat sealed package
- Breadboard and Perf board friendly

3.4.3 How to use a Buzzer

A buzzer is a small yet efficient component to add sound features to our project/system. It is a very small and compact 2-pin structure hence can be easily used on breadboards, Perf Board, and even on PCBs which makes this a widely used component in most electronic applications.

There are two types are buzzers that are commonly available. The one shown here is a simple buzzer which when powered will make a Continuous Beep sound, the other type is called a readymade buzzer which will look bulkier than this and will produce a Beep. Beep. Beep. Sound due to the internal oscillating circuit present inside it. But, the one shown here is most widely used because it can be customized with the help of other circuits to fit easily in our application.

This buzzer can be used by simply powering it using a DC power supply ranging from 4V to 9V. A simple 9V battery can also be used, but it is recommended to use a regulated +5V or +6V DC supply. The buzzer is normally associated with a switching circuit to turn ON or turn OFF the buzzer at the required time and required interval.

Applications of Buzzer

- Alarming Circuits, where the user has to be alarmed about something
- Communication types of equipment
- Automobile electronics
- Portable types of equipment, due to its compact size

3.5 DHT11 Temperature and Humidity sensor:

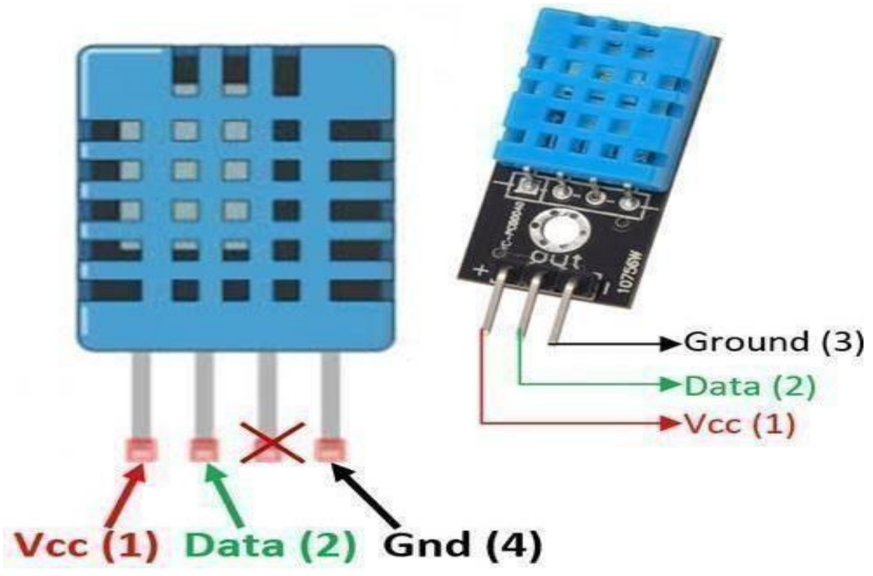


Fig.3.5 DHT11 Temperature and Humidity Sensor

- The **DHT11** is a commonly used **Temperature and humidity sensor that** comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

3.5.1 DHT11 Pinout Configuration

No:	Pin Name	Description
For DHT11 Sensor		
1	VCC	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	NC	No Connection and hence not used
4	Ground	Connected to the ground of the circuit

For the DHT11 Sensor module		
1	VCC	Power supply 3.5V to 5.5V
2	Data	Outputs both Temperature and Humidity through serial Data
3	Ground	Connected to the ground of the circuit

Table 3.3: DHT11 Pin-out Configuration

DHT11 Specifications

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: ±1°C and ±1%

3.6 Ultrasonic sensor:

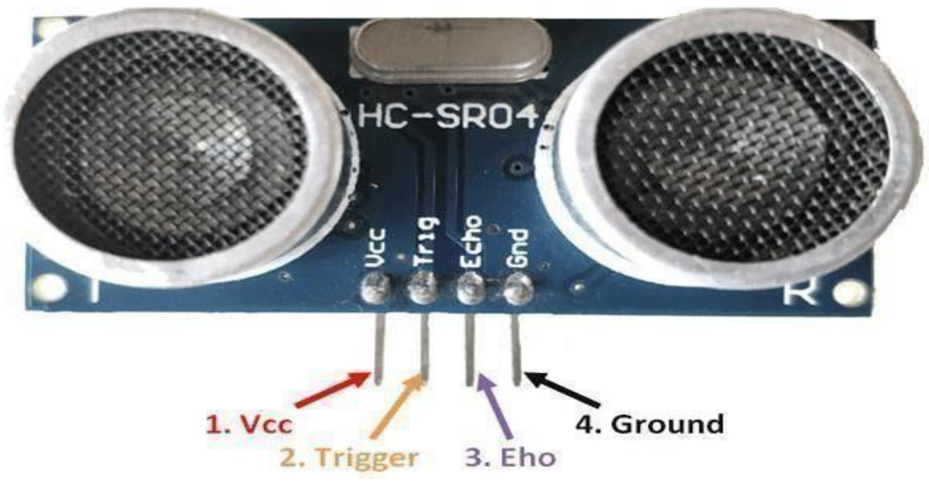


Fig.3.6(a): Ultrasonic Sensor Pinout Configuration

3.6.1 Pin Out Configuration Of Ultrasonic Sensor

Pin Number	**Pin Name**	**Description**
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	A trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending a US wave.
3	Echo	The echo pin is an Output pin. This pin goes high for a period that will be equal to the time taken for the US wave to return to the sensor.
4	Ground	This pin is connected to the Ground of the system.

Table 3.4: HC-SRO4 Pinout Configuration

3.6.2 HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

More details can be found in the **HC-SR04 ultrasonic sensor datasheet** attached at the bottom of this article.

Equivalent distance-measuring Sensors

US transmitter Receiver pair, IR sensor module, IR sensor pair, IR Analog distance sensor.

3.6.3 HC-SR04 Ultrasonic Sensor – Working and Applications

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4-pin module, whose pin names are Vcc, Trigger, Echo, and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes-like projects in the front which form the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in the air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

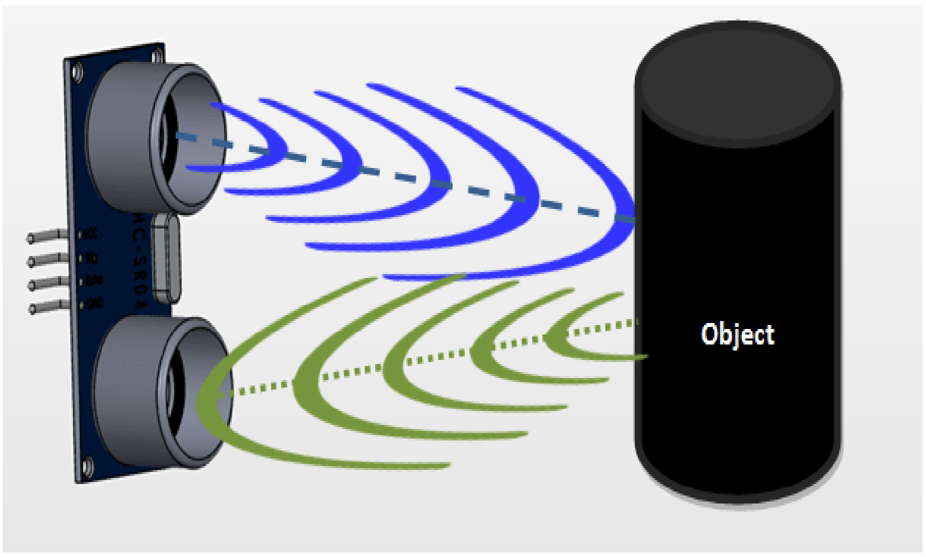


Fig.3.6(b): HC-SRO4 Transmitter Transmitting Waves

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic Wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry built on the module will calculate the time taken for the US wave to come back and turn on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

How to use the HC-SR04 Ultrasonic Sensor

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, ARM, PIC, Raspberry Pie, etc. The following guide is universal since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc and Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the onboard 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to the I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at the frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it gets reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return to the sensor.

The amount of time during which the Echo pin stays high is measured by the MCU/MPU as it gives the information about the time taken for the wave to return to the Sensor. Using this information the distance is measured as explained in the above heading.

Applications

- Used to avoid and detect obstacles with robots like biped robots, obstacle avoider robots, path-finding robots, etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits, etc can be measured since the waves can penetrate through water

3.7 Power Supply

The power supply is a reference to a source of electrical power. A device or system that supplies electrical or other types of energy to an output load or group of loads is called a power supply unit or PSU. The term is most commonly applied to electrical energy supplies, less often to mechanical ones, and rarely to others

This power supply section is required to convert the AC signal to a DC signal and also to reduce the amplitude of the signal. The available voltage signal from the mains is 230V/50Hz which is an AC voltage, but the required is DC Voltage(no frequency) with the amplitude of +5V and +12V for various applications.

In this section we have the Transformer and bridge rectifier, which are connected serially, and voltage regulators for +5V and +12V (7805 and 7812) via a capacitor (1000µF) in parallel are connected parallel as shown in the circuit diagram below. Each voltage regulator output is again connected to the capacitors of values (100, 10µF, 1 µF, 0.1 µF) are connected parallel through which the corresponding output(+5V or +12V) is taken into consideration.

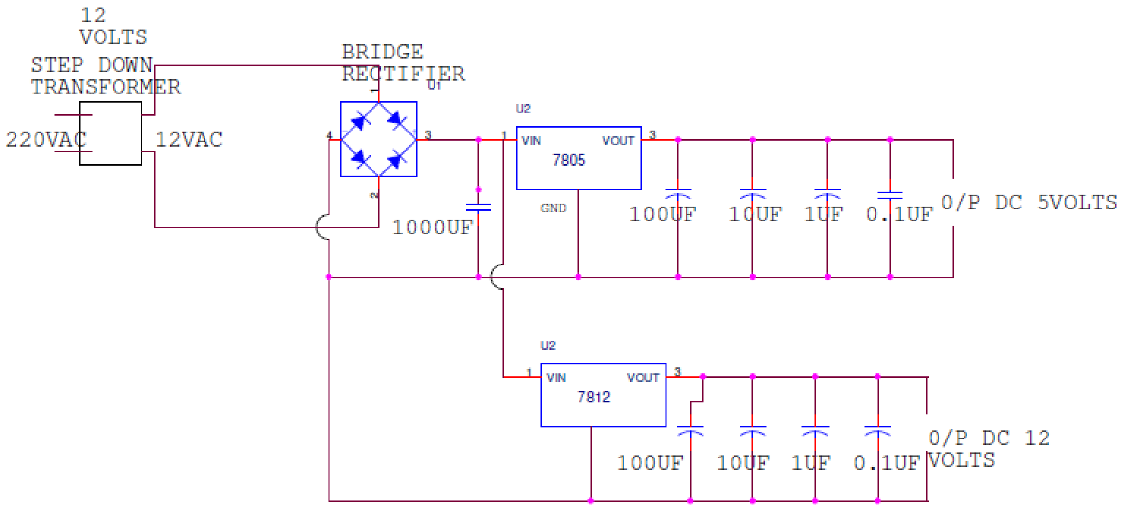


Fig.3.7: Transformer Bridge Rectifier

3.7.1 Circuit Explanation

1) Transformer

A transformer is a device that transfers electrical energy from one circuit to another through inductively coupled electrical conductors. A changing current in the first circuit (the primary) creates a changing magnetic field; in turn, this magnetic field induces a changing voltage in the second circuit (the secondary). By adding a load to the secondary circuit, one can make current flow in the transformer, thus transferring energy from one circuit to the other.

The secondary induced voltage V_S, of an ideal transformer, is scaled from the primary V_P by a factor equal to the ratio of the number of turns of wire in their respective windings:

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

Basic principle

The transformer is based on two principles: firstly, that an electric current can produce a magnetic field (electromagnetism) and secondly that a changing magnetic field within a coil of wire induces a voltage across the ends of the coil (electromagnetic induction). By changing the current in the primary coil, it changes the strength of its magnetic field; since the changing magnetic field extends into the secondary coil, a voltage is induced across the secondary.

A simplified transformer design is shown below. A current passing through the primary coil creates a magnetic field. The primary and secondary coils are wrapped around a core of very high magnetic permeability, such as iron; this ensures that most of the magnetic field lines produced by the primary current are within the iron and pass through the secondary coil as well as the primary coil.

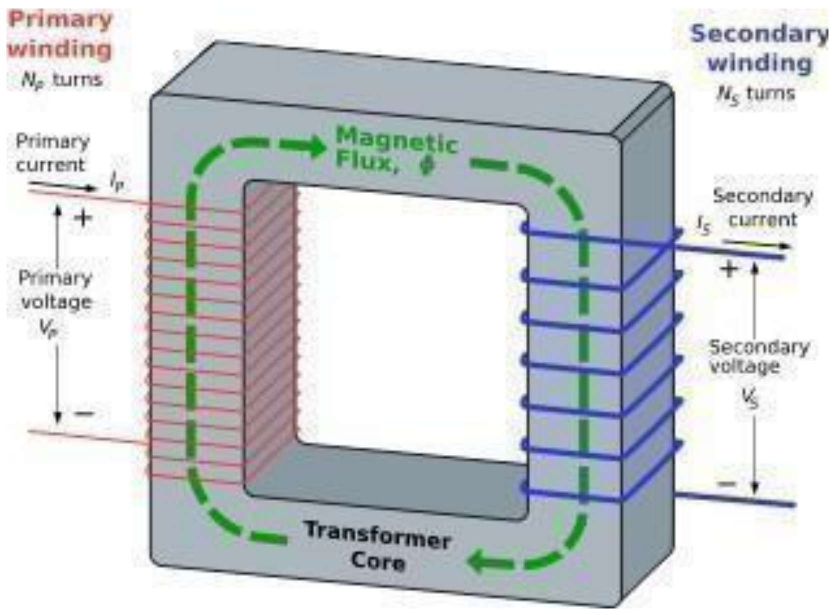


Fig.3.8(a): An Ideal Step-Down Transformer Showing Magnetic Flux in Core

Induction law

The voltage induced across the secondary coil may be calculated from Faraday's law of induction, which states that:

$$V_S = N_S \frac{\mathrm{d}\Phi}{\mathrm{d}t}$$

Where V_S is the instantaneous voltage, N_S is the number of turns in the secondary coil and Φ equals the magnetic flux through one turn of the coil. If the turns of the coil are oriented perpendicular to the magnetic field lines, the flux is the product of the magnetic field strength B and the area A through which it cuts. The area is constant, being equal to the cross-sectional area of the transformer core, whereas the magnetic field varies with time according to the excitation of the primary. Since the same magnetic flux passes through both the primary and secondary coils in an ideal transformer, the instantaneous voltage across the primary winding equals

$$V_P = N_P \frac{\mathrm{d}\Phi}{\mathrm{d}t}$$

Taking the ratio of the two equations for V_S and V_P gives the basic equation for stepping up or stepping down the voltage

$$\frac{V_S}{V_P} = \frac{N_S}{N_P}$$

Ideal power equation

If the secondary coil is attached to a load that allows current to flow, electrical power is transmitted from the primary circuit to the secondary circuit. Ideally, the transformer is perfectly efficient; all the incoming energy is transformed from the primary circuit to the magnetic field and into the secondary circuit. If this condition is met, the incoming electric power must equal the outgoing power.

$$P_{incoming} = I_P V_P = P_{outgoing} = I_S V_S$$

giving the ideal transformer equation

$$\frac{V_S}{V_P} = \frac{N_S}{N_P} = \frac{I_P}{I_S}$$

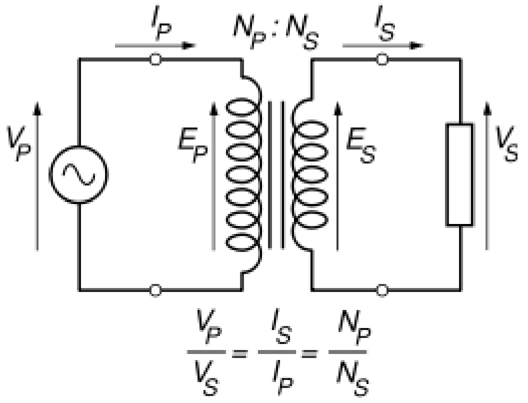


Fig.3.8(b): Circuit Diagram of Ideal Transformer

$$P_{in\text{-}coming} = I_P V_P = P_{out\text{-}going} = I_S V_S$$

giving the ideal transformer equation

$$\frac{V_S}{V_P} = \frac{N_S}{N_P} = \frac{I_P}{I_S}$$

If the voltage is increased (stepped up) ($V_S > V_P$), then the current is decreased (stepped down) ($I_S < I_P$) by the same factor. Transformers are efficient so this formula is a reasonable approximation.

If the voltage is increased (stepped up) ($V_S > V_P$), then the current is decreased (stepped down) ($I_S < I_P$) by the same factor. Transformers are efficient so this formula is a reasonable approximation.

The impedance in one circuit is transformed by the *square* of the turns ratio. For example, if an impedance Z_S is attached across the terminals of the secondary coil, it appears to the primary circuit to have an impedance of

$$Z_S\left(\frac{N_P}{N_S}\right)^2$$

This relationship is reciprocal, so that the impedance Z_P of the primary circuit appears to the secondary to be

$$Z_P\left(\frac{N_S}{N_P}\right)^2$$

Detailed operation

The simplified description above neglects several practical factors, in particular the primary current required to establish a magnetic field in the core, and the contribution to the field due to current in the secondary circuit.

Models of an ideal transformer typically assume a core of negligible reluctance with two windings of zero resistance. When a voltage is applied to the primary winding, a small current flows, driving flux around the magnetic circuit of the core. The current required to create the flux is termed the magnetizing current; since the ideal core has been assumed to have near-zero reluctance, the magnetizing current is negligible, although still required to create the magnetic field.

The changing magnetic field induces an electromotive force (EMF) across each winding. Since the ideal windings have no impedance, they have no associated voltage drop, and so the voltages V_P and V_S measured at the terminals of the transformer, are equal to the corresponding EMFs. The primary EMF, acting as it does in opposition to the primary voltage, is sometimes termed the "back EMF". This is due to Lenz's law which states that the induction of EMF would always be such that it will oppose development of any such change in magnetic field.

2) Bridge Rectifier

A diode bridge or bridge rectifier is an arrangement of four diodes in a bridge configuration that provides the same polarity of output voltage for any polarity of input voltage. When used in its most common application, for conversion of alternating current (AC) input into direct current (DC) output, it is known as a bridge rectifier. A bridge rectifier provides full-wave rectification from a two-wire AC input, resulting in lower cost and weight as compared to a center-tapped transformer design, but has two diode drops rather than one, thus exhibiting reduced efficiency over a center-tapped design for the same output voltage.

Basic Operation

When the input connected at the left corner of the diamond is positive with respect to the one connected at the right-hand corner, current flows to the right along the upper colored path to the output, and returns to the input supply via the lower one.

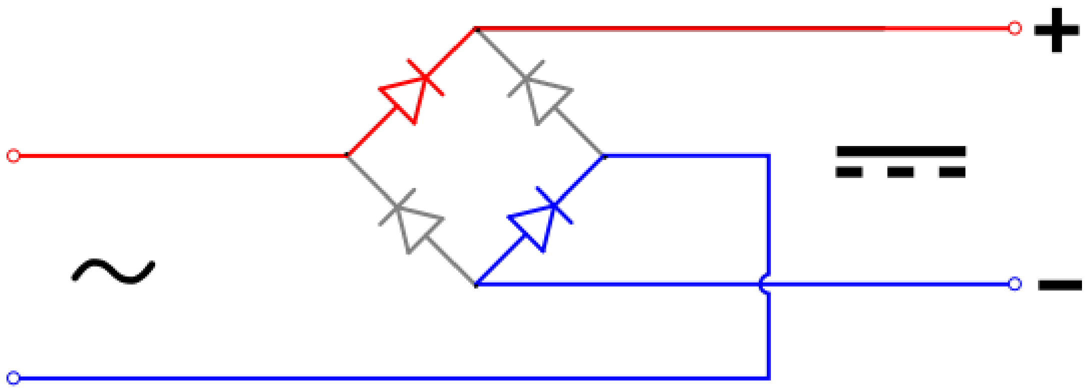


Fig.3.9(a): Current Flow in Bridge Rectifier

When the right-hand corner is positive relative to the left hand corner, current flows along the upper colored path and returns to the supply via the lower colored path

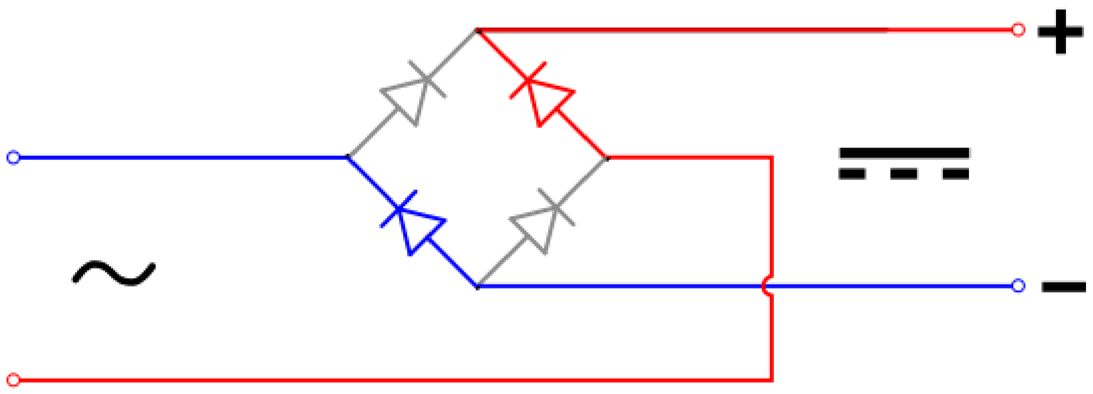


Fig.3.9(b): Current Flow in Bridge Rectifier

In each case, the upper right output remains positive with respect to the lower right one. Since this is true whether the input is AC or DC, this circuit not only produces DC power when supplied with AC power: it also can provide what is sometimes called "reverse polarity protection". That is, it permits normal functioning when batteries are installed backwards or DC input-power supply wiring "has its wires crossed" (and protects the circuitry it powers against damage that might occur without this circuit in place).

Prior to availability of integrated electronics, such a bridge rectifier was always constructed from discrete components. Since about 1950, a single four-terminal component containing the four diodes connected in the bridge configuration became a standard commercial component and is now available with various voltage and current ratings.

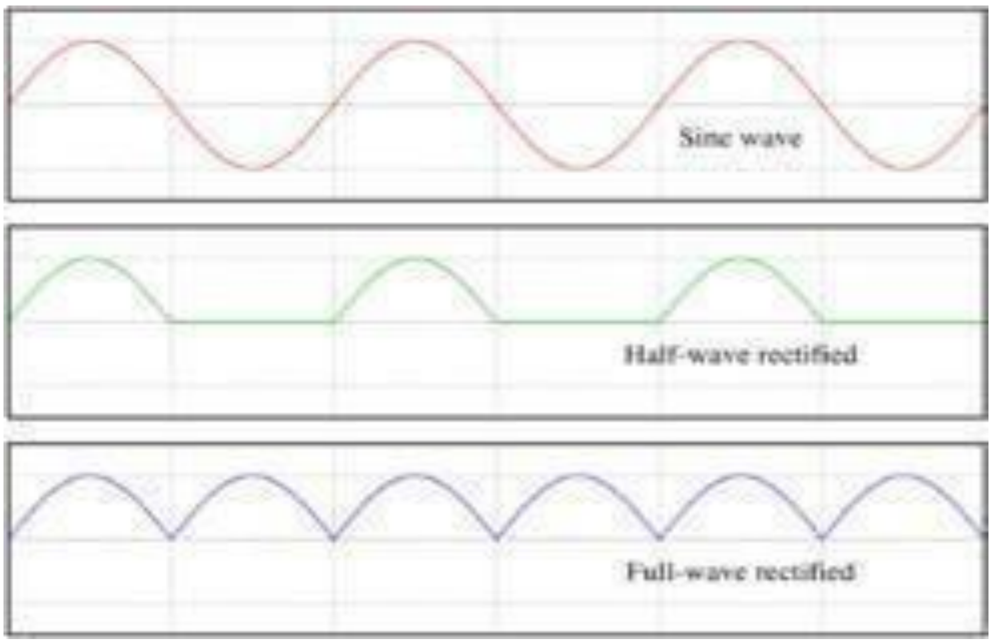


Fig.3.10: Wave Forms of Half Wave and Full Wave Rectifier

Output smoothing (Using Capacitor)

For many applications, especially with single phase AC where the full-wave bridge serves to convert an AC input into a DC output, the addition of a capacitor may be important because the bridge alone supplies an output voltage of fixed polarity but pulsating magnitude (see diagram above).

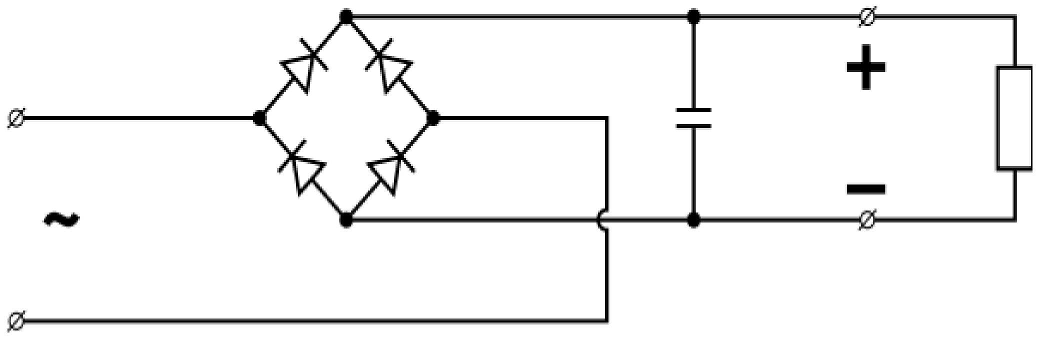


Fig.3.11: Smoothing of Bridge Rectifier

The function of this capacitor, known as a reservoir capacitor (aka smoothing capacitor) is to lessen the variation in (or 'smooth') the rectified AC output voltage waveform from the bridge. One explanation of 'smoothing' is that the capacitor provides a low impedance path to the AC component of the output, reducing the AC voltage across, and AC current through, the resistive load. In less technical terms, any drop in the output voltage and current of the bridge tends to be cancelled by loss of charge in the capacitor.

This charge flows out as additional current through the load. Thus, the change of load current and voltage is reduced relative to what would occur without the capacitor. Increases of voltage correspondingly store excess charge in the capacitor, thus moderating the change in output voltage / current. Also see rectifier output smoothing.

The simplified circuit shown has a well-deserved reputation for being dangerous, because, in some applications, the capacitor can retain a lethal charge after the AC power source is removed. If supplying a dangerous voltage, a practical circuit should include a reliable way to safely discharge the capacitor. If the normal load can not be guaranteed to perform this function, perhaps because it can be disconnected, the circuit should include a bleeder resistor connected as close as practical across the capacitor. This resistor should consume a current large enough to discharge the capacitor in a reasonable time, but small enough to avoid unnecessary power waste.

Because a bleeder sets a minimum current drain, the regulation of the circuit, defined as percentage voltage change from minimum to maximum load, is improved. However, in many cases the improvement is of insignificant magnitude.

The capacitor and the load resistance have a typical time constant $\tau = RC$ where C and R are the capacitance and load resistance respectively. As long as the load resistor is large enough so that this time constant is much longer than the time of one ripple cycle, the above configuration will produce a smoothed DC voltage across the load.

In some designs, a series resistor at the load side of the capacitor is added. The smoothing can then be improved by adding additional stages of capacitor–resistor pairs, often done only for sub-supplies to critical high-gain circuits that tend to be sensitive to supply voltage noise.

The idealized waveforms shown above are seen for both voltage and current when the load on the bridge is resistive. When the load includes a smoothing capacitor, both the voltage and the current waveforms will be greatly changed. While the voltage is smoothed, as described above, current will flow through the bridge only during the time when the input voltage is greater than the capacitor voltage. For example, if the load draws an average current of n Amps, and the diodes conduct for 10% of the time, the average diode current during conduction must be 10n Amps. This non-sinusoidal current leads to harmonic distortion and a poor power factor in the AC supply.

In a practical circuit, when a capacitor is directly connected to the output of a bridge, the bridge diodes must be sized to withstand the current surge that occurs when the power is turned on at the peak of the AC voltage and the capacitor is fully discharged. Sometimes a small series resistor is included before the capacitor to limit this current, though in most applications the power supply transformer's resistance is already sufficient.

Output can also be smoothed using a choke and second capacitor. The choke tends to keep the current (rather than the voltage) more constant. Due to the relatively high cost of an effective choke compared to a resistor and capacitor this is not employed in modern equipment.

Some early console radios created the speaker's constant field with the current from the high voltage ("B +") power supply, which was then routed to the consuming circuits, (permanent magnets were considered too weak for good performance) to create the speaker's constant magnetic field. The speaker field coil thus performed 2 jobs in one: it acted as a choke, filtering the power supply, and it produced the magnetic field to operate the speaker.

3) Voltage Regulator

A voltage regulator is an electrical regulator designed to automatically maintain a constant voltage level.

The 78xx (also sometimes known as LM78xx series of devices is a family of self-contained fixed linear voltage regulator integrated circuits. The 78xx family is a very popular choice for many electronic circuits which require a regulated power supply, due to their ease of use and relative cheapness. When specifying individual ICs within this family, the xx is replaced with a two-digit number, which indicates the output voltage the particular device is designed to provide (for example, the 7805 has a 5 Volt output, while the 7812 produces 12 volts). The 78xx line is positive voltage regulators, meaning that they are designed to produce a voltage that is positive relative to a common ground. There is a related line of 79xx devices which are complementary negative voltage regulators. 78xx and 79xx ICs can be used in combination to provide both positive and negative supply voltages in the same circuit, if necessary.

78xx ICs have three terminals and are most commonly found in the TO220 form factor, although smaller surface-mount and larger TrO3 packages are also available from some manufacturers. These devices typically support an input voltage which can be anywhere from a couple of volts over the intended output voltage, up to a maximum of 35 or 40 volts, and can typically provide up to around 1 or 1.5 amps of current (though smaller or larger packages may have a lower or higher current rating).

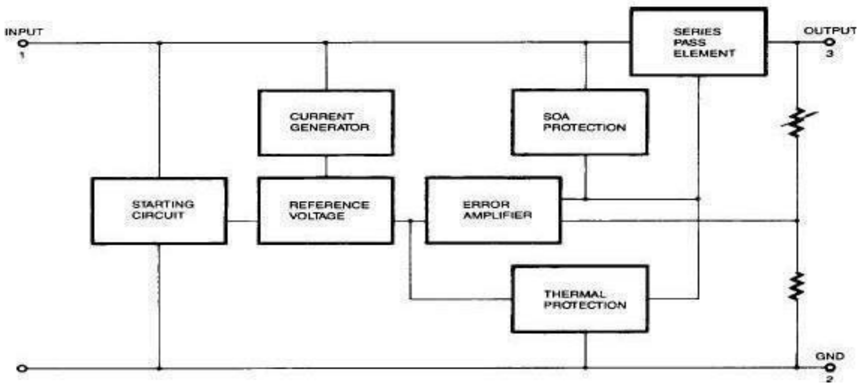


Fig.3.12: Internal Block Diagram

3.8 Introduction to Gas Sensor

A gas sensor is a device which detects the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. Based on this voltage value the type and concentration of the gas can be estimated.

The type of gas the sensor could detect depends on the **sensing material** present inside the sensor. Normally these sensors are available as modules with comparators as shown above. These comparators can be set for a particular threshold value of gas concentration. When the concentration of the gas exceeds this threshold the digital pin goes high. The Analog pin can be used to measure the concentration of the gas.

3.8.1 Different Types of Gas sensors

Gas sensors are typically classified into various types based on the type of the sensing element it is built with. Below is the classification of the various types of gas sensors based on the sensing element that are generally used in various applications:

- Metal Oxide based gas Sensor.
- Optical gas Sensor.
- Electrochemical gas Sensor.
- Capacitance-based gas Sensor.
- Calorimetric gas Sensor.
- Acoustic based gas Sensor.

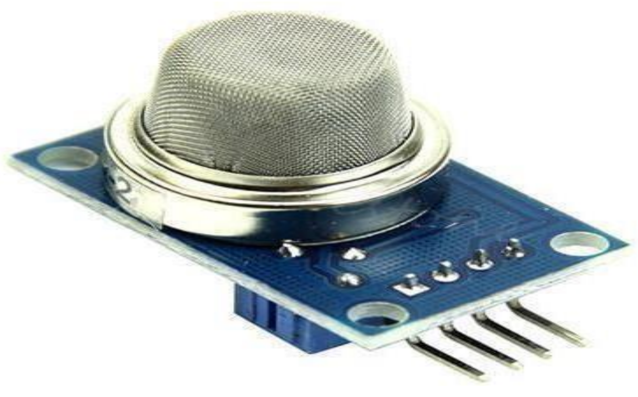


Fig.3.13: Gas Sensor

3.9 WIFI MODULE (ESP8266)

ESP8266 was designed by the Chinese company Espressif Systems for uses in Internet of Things (IoT) systems. ESP8266 is a complete WIFI system on chip that incorporates a 32-bit processor, some RAM and depending on the vendor between 512KB and 4MB of flash memory. This allows the chip to either function as a wireless adapter that can extend other systems with WIFI functionality, or as a standalone unit that can by itself execute simple applications. Depending on the specific module variant (ESP-1 to ESP-12 at the time of this thesis) between 0 and 7 General Purpose Input/Output (GPIO) pins are available, in addition to Rx and Tx pins of the UART, making the module very suitable for IoT applications. The Software Development Kit (SDK) provided by Espressif contains a lightweight implementation of a TCP/IP control stack (IWIP) for WIFI communication. The modules house libraries for optional services such as Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), JavaScript Object Notation (JSON) and Secure Socket Layer (SSL) libraries for Application Level programming. It incorporates 802.11 MAC extensions such as 802.11b/g/n/d/e/h/i/k/r that manage signal transmission, encapsulation, encryption, collision management and roaming functionality. The chip generally comes as part of a module, soldered to a Printed Circuit Board (PCB), however it is possible to purchase only the chip itself in order to create a truly custom module. The module variants currently available on the market may include an antenna (PCB or ceramic) or a U-FL connector, a hardware component for serial communication and a myriad of other auxiliary components such as resistors, capacitors and LEDs.

3.9.1 Overview and Specification

The module comes in many different variations (ESP-01 to ESP-12), along with non-Espressif vendors such as Olimex and NODEMCU. The main differences between these modules are size and additional components of the PCB, with some having an inbuilt PCB antenna and up to 7 GPIO pins, while others provide no easy access to GPIO and no antenna, but at a lower cost and module dimensions.

The processor inside the module is a low power, 80MHz, 32bit Tensilica Xtensa LX. It is classified as a DPU, which is Tensilicas own type of CPU combining the strengths of a traditional CPU and a DSP to achieve better performance for data-intensive tasks. Multiple compilation tools exist for this processor with the ESP community even attempting to design their own version of gcc compiler to achieve more efficient code density and better performance.

The amount of programmable memory varies depending on the module manufacturer, but generally ESPs come with either 512KB, 1MB, 2MB or 4MB of flash memory. ESP8266 is interrupt driven, with a relatively simple OS and three levels of task priority, meaning that only three user tasks that can respond to interrupts can be defined. A function user_init() configures the module once its provided with power, and can be used to schedule the next task, or define a fully event-driven configuration.

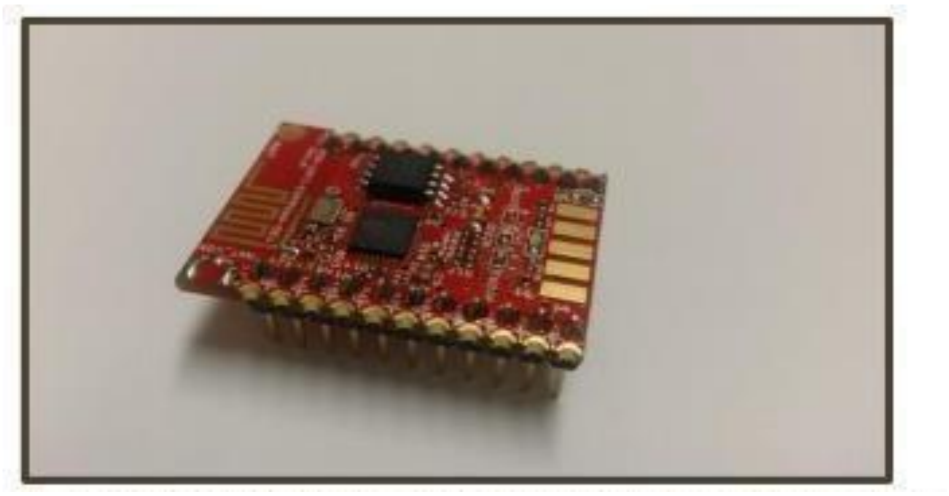
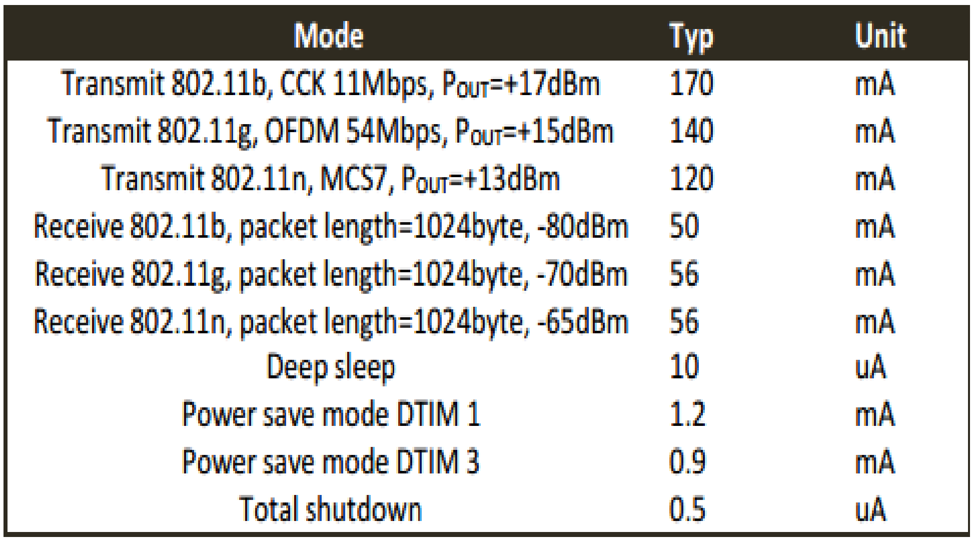


Fig.3.14: WI-FI Module(Esp8266)

The module used in this thesis is an Olimex MOD-WIFI-ESP8266-DEV with all of the basic components of ESP8266, a PCB antenna, crystal and an easily accessed UART with support for SPI and I2C, 2Mbytes of flash, but more importantly for this thesis it has all the available chip pins mapped out for easier access.

3.9.2 Power Supply and Consumption

Being a WIFI SoC, this chip requires a fair amount of power to operate its transceiver. It has incorporated some impressive power management features, including highly integrated components that allow for greater optimization and increased efficiency. All this makes ESP8266 one of the least power-hungry chips in the WIFI IC industry! Unfortunately, its levels of demand are still higher than of those based on wireless technologies such as Bluetooth, or ZigBee. The official ESP8266 datasheet states this regarding current draw:



Mode	Typ	Unit
Transmit 802.11b, CCK 11Mbps, P_{OUT}=+17dBm	170	mA
Transmit 802.11g, OFDM 54Mbps, P_{OUT}=+15dBm	140	mA
Transmit 802.11n, MCS7, P_{OUT}=+13dBm	120	mA
Receive 802.11b, packet length=1024byte, -80dBm	50	mA
Receive 802.11g, packet length=1024byte, -70dBm	56	mA
Receive 802.11n, packet length=1024byte, -65dBm	56	mA
Deep sleep	10	uA
Power save mode DTIM 1	1.2	mA
Power save mode DTIM 3	0.9	mA
Total shutdown	0.5	uA

Table 3.5: ESP8266EX Current Draw at 3.3v as Listed in the Official Documentation from Espressif

However, this is just the power consumption of the ESP8266EX chip, the entire module, featuring additional hardware such as LEDs, crystals, capacitors and registers revealed that actual consumption of the MOD-WiFi-ESP8266-DEV varied greatly from this table. The approximate idle (while ready to receive packets) current of the module was measured to be 70mA, with somewhat higher when receiving packets in 802.11n mode. Transmission drew 80mA current.

The module was also prone to high current spikes in the range of 300mA at unpredictable points in time, often causing a full module restart. This problem does seem to have been dealt with in most recent SDK, however most manufacturers still choose to add additional capacitances parallel to power supply in order to prevent such instances from occurring. The ESP8266 can operate in a total of 3 power saving modes all of which sacrifice a portion of functionality to achieve lower power consumption. These modes are: Light Sleep• Modem Sleep• Deep Sleep• The Light and Modem modes are so called WIFI sleep modes. They are designed to be used when the module is in STA mode, meaning it does not have to actively send beacons to announce its presence and verify clients' statuses. The Light Sleep WIFI mode is to be used when the module needs to maintain WLAN connection without actively transmitting or receiving data, allowing the CPU to operate at a lower voltage (or be suspended altogether) and turning off the WIFI modem between AP status beacons. This would allow the module to save power while still answer to beacons, effectively maintaining the wireless connection. In this mode a DTIM3 setup at the AP, with 300ms sleep and 3ms wake cycle can, according to the datasheet, lower power consumption to 0.9mA [15]. Modem Sleep is used when the CPU needs to be active. In this mode the WIFI modem is turned off between the AP status beacons, maintaining the connection at minimal cost while allowing the CPU to perform without interference. A similar DTIM3 setup as in previous example leaves the current consumption at 15mA [15]. The Deep Sleep turns of all functionality (CPU and WIFI modem), while maintaining only the RTC clock, allowing the module to be woken up by a timed interrupt. When waking up, the module performs a complete reset, meaning all RAM data is erased (although there is limited space in RTC memory block that does not get erased) and the WIFI connection needs to be re-established. Espressif claims a 300s sleep and 1s wake cycle (claimed as enough to connect to AP) results in average consumption of less than 1mA [15]. Important to note is that WIFI sleep-modes described above may not always be true. In test conducted in this thesis they only seemed to lower the power consumption to 20-50mA in Light Sleep and 40mA in Modem Sleep. Although conventional Deep Sleep resulted in an average of 10uA, there were instances when the deep-sleep sequence seemed to execute in an incorrect manner, leaving the power consumption at standby levels (~80mA). This issue has been known to Espressif that has promised to address it in future releases of the SDK. It should also be noted that the module is specified for voltages between 1.7V and 3.6V, meaning it can be powered by two AA alkaline batteries placed in series (achieving 3V). However, the ESP will not work with a Lithium Ion (or LiPo) based batteries without additional power regulating circuits.

3.9.3 Serial Communication

ESP8266 has multiple peripherals through which it can interface with other modules in a classic embedded fashion. In this section only the setup of the communication link will be presented, since the exact flow of bits to achieve such communication was handled automatically by the module and is therefore deemed of no immediate interest for this thesis. In this case classical UART was used to decode output and encoding data to be sent to the sensor. Similarly, EM50 data logger has an

UART of its own and can do the same thing on its end. Serial asynchronous communication does not require a common clock, however in order for the data to be processed correctly and at right intervals a common baud rate (can be viewed as symbols per second) needs to be set for both devices. The baud-rates supported by ESPs UART component range from 9600 to 921600bps, while the EM50 is configured for 9600bps as default.

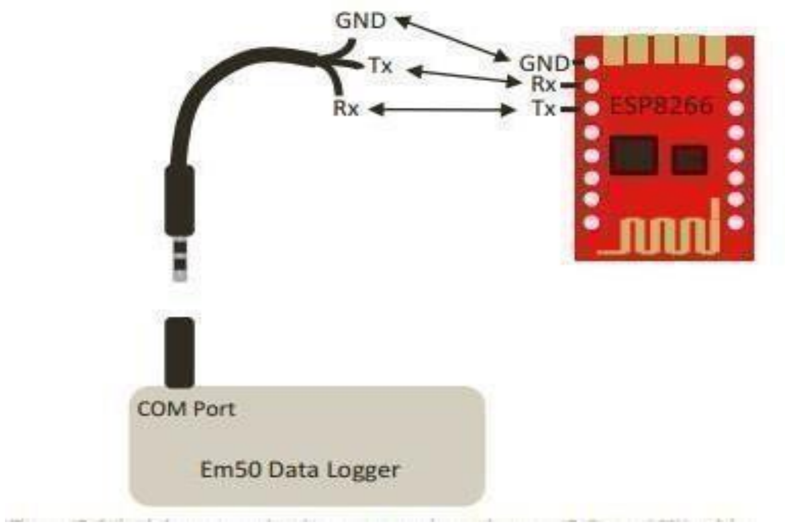


Fig.3.15: Serial Communication to ESP8266 Through AUX Cable

CHAPTER-4
SOFTWARE DESCRIPTION

4.1 Arduino 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.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

✓

Verify
Checks your code for errors compiling it.

➔

Upload
Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New
Creates a new sketch.

Open
Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the **File | Sketchbook**menu instead.

Save
Saves your sketch.

Serial Monitor
Opens the serial monitor.

Additional commands are found within the five menus: **File**, **Edit**, **Sketch**, **Tools**, **Help**. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

File

- *New*

  Creates a new instance of the editor, with the bare minimum structure of a sketch already in place.
- *Open*

  Allows to load a sketch file browsing through the computer drives and folders.
- *Open Recent*

  Provides a short list of the most recent sketches, ready to be opened.

- *Sketchbook*

  Shows the current sketches within the sketchbook folder structure; clicking on any name opens the corresponding sketch in a new editor instance.
- *Examples*

  Any example provided by the Arduino Software (IDE) or library shows up in this menu item. All the examples are structured in a tree that allows easy access by topic or library.
- *Close*

  Closes the instance of the Arduino Software from which it is clicked.
- *Save*

  Saves the sketch with the current name. If the file hasn't been named before, a name will be provided in a "Save as.." window.
- *Save as...*

  Allows to save the current sketch with a different name.
- *Page Setup*

  It shows the Page Setup window for printing.
- *Print*

  Sends the current sketch to the printer according to the settings defined in Page Setup.
- *Preferences*

  Opens the Preferences window where some settings of the IDE may be customized, as the language of the IDE interface.
- *Quit*

  Closes all IDE windows. The same sketches open when Quit was chosen will be automatically reopened the next time you start the IDE.

Edit

- *Undo/Redo*

  Goes back of one or more steps you did while editing; when you go back, you may go forward with Redo.
- *Cut*

  Removes the selected text from the editor and places it into the clipboard.
- *Copy*

  Duplicates the selected text in the editor and places it into the clipboard.

- *Copy for Forum*
  Copies the code of your sketch to the clipboard in a form suitable for posting to the forum, complete with syntax coloring.
- *Copy as HTML*
  Copies the code of your sketch to the clipboard as HTML, suitable for embedding in web pages.
- *Paste*
  Puts the contents of the clipboard at the cursor position, in the editor.
- *Select All*
  Selects and highlights the whole content of the editor.
- *Comment/Uncomment*
  Puts or removes the // comment marker at the beginning of each selected line.
- *Increase/Decrease Indent*
  Adds or subtracts a space at the beginning of each selected line, moving the text one space on the right or eliminating a space at the beginning.
- *Find*
  Opens the Find and Replace window where you can specify text to search inside the current sketch according to several options.
- *Find Next*
  Highlights the next occurrence - if any - of the string specified as the search item in the Find window, relative to the cursor position.
- *Find Previous*
  Highlights the previous occurrence - if any - of the string specified as the search item in the Find window relative to the cursor position.

Sketch

- *Verify/Compile*
  Checks your sketch for errors compiling it; it will report memory usage for code and variables in the console area.
- *Upload*
  Compiles and loads the binary file onto the configured board through the configured Port.

- *Upload Using Programmer*
  This will overwrite the bootloader on the board; you will need to use Tools > Burn Bootloader to restore it and be able to Upload to USB serial port again. However, it allows you to use the full capacity of the Flash memory for your sketch. Please note that this command will NOT burn the fuses. To do so a *Tools -> Burn Bootloader* command must be executed.
- *Export Compiled Binary*
  Saves a .hex file that may be kept as archive or sent to the board using other tools.
- *Show Sketch Folder*
  Opens the current sketch folder.
- *Include Library*
  Adds a library to your s etch by inserting #include statementsat the start of your code. For more details, see li aries below. Additionally, from this m nu item you can access the Library Manager an import new libraries from .zip files.
- *Add File...*
  Adds a source file to the sketch (it will be copied from its current location). The new file appears in a new tab in the sketch window. Files can be removed from the sketch using the tab menu accessible clicking on the small triangle icon below the serial monitor one on the right side o the toolbar.

Tools

- *Auto Format*
  This formats your code nicely: i.e. indents it so that opening and closing curly braces line up, and that the statements inside curly braces are indented more.
- *Archive Sketch*
  Archives a copy of the current sketch in .zip format. The archive is placed in the same directory as the sketch.
- *Fix Encoding & Reload*
  Fixes possible discrepancies between the editor char map encoding and other operating systems char maps.
- *Serial Monitor*
  Opens the serial monitor window and initiates the exchange of data with any connected

board on the currently selected Port. This usually resets the board, if the board supports Reset over serial port opening.

- *Board*
  Select the board that you're using. See below for descriptions of the various boards.
- *Port*
  This menu contains all the serial devices (real or virtual) on your machine. It should automatically refresh every time you open the top-level tools menu.
- *Programmer*
  For selecting a hardware programmer when programming a board or chip and not using the onboard USB-serial connection. Normally you won't need this, but if you're burning a bootloader to a new microcontroller, you will use this.
- *Burn Bootloader*
  The items in this menu allow you to burn a bootloader onto the microcontroller on an Arduino board. This is not required for normal use of an Arduino or Genuino board but is useful if you purchase a new ATmega microcontroller (which normally come without a bootloader). Ensure that you've selected the correct board from the Boards menu before burning the bootloader on the target board. This command also set the right fuses.

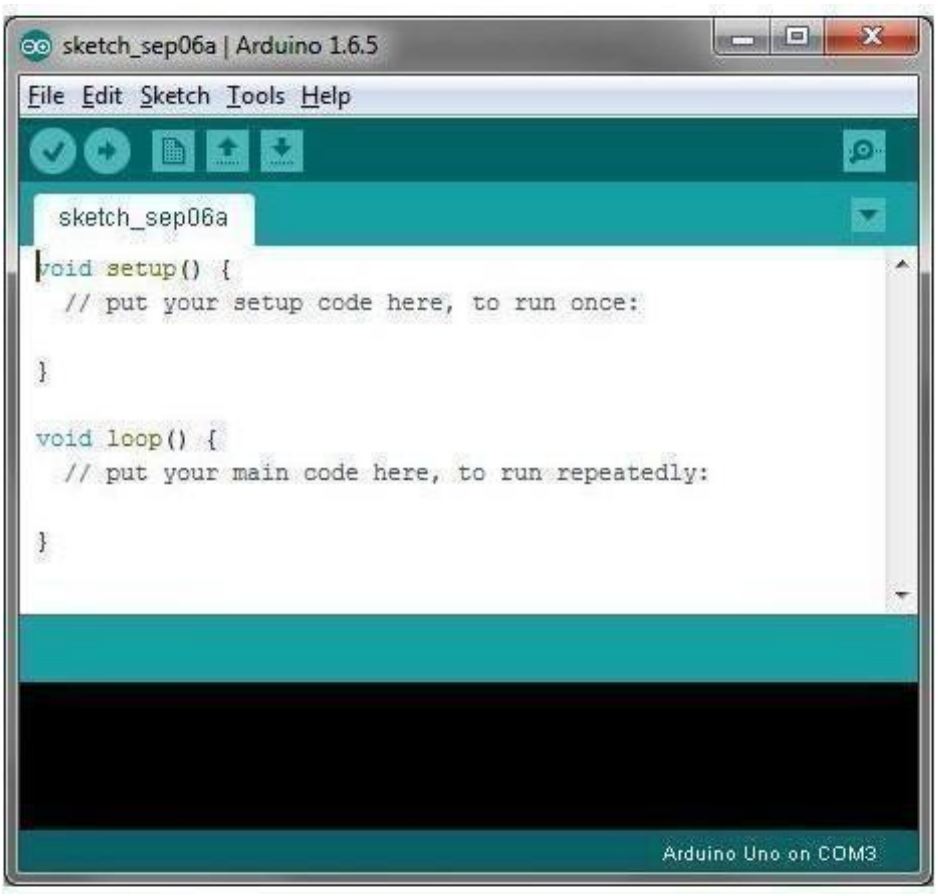
Help

Here you find easy access to a number of documents that come with the Arduino Software (IDE). You have access to Getting Started, Reference, this guide to the IDE and other documents locally, without an internet connection. The documents are a local copy of the online ones and may link back to our online website.

- *Find in Reference*
  This is the only interactive function of the Help menu: it directly selects the relevant page in the local copy of the Reference for the function or command under the cursor.

4.1.1 Arduino IDE: Initial Setup

This is the Arduino IDE once it's been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.

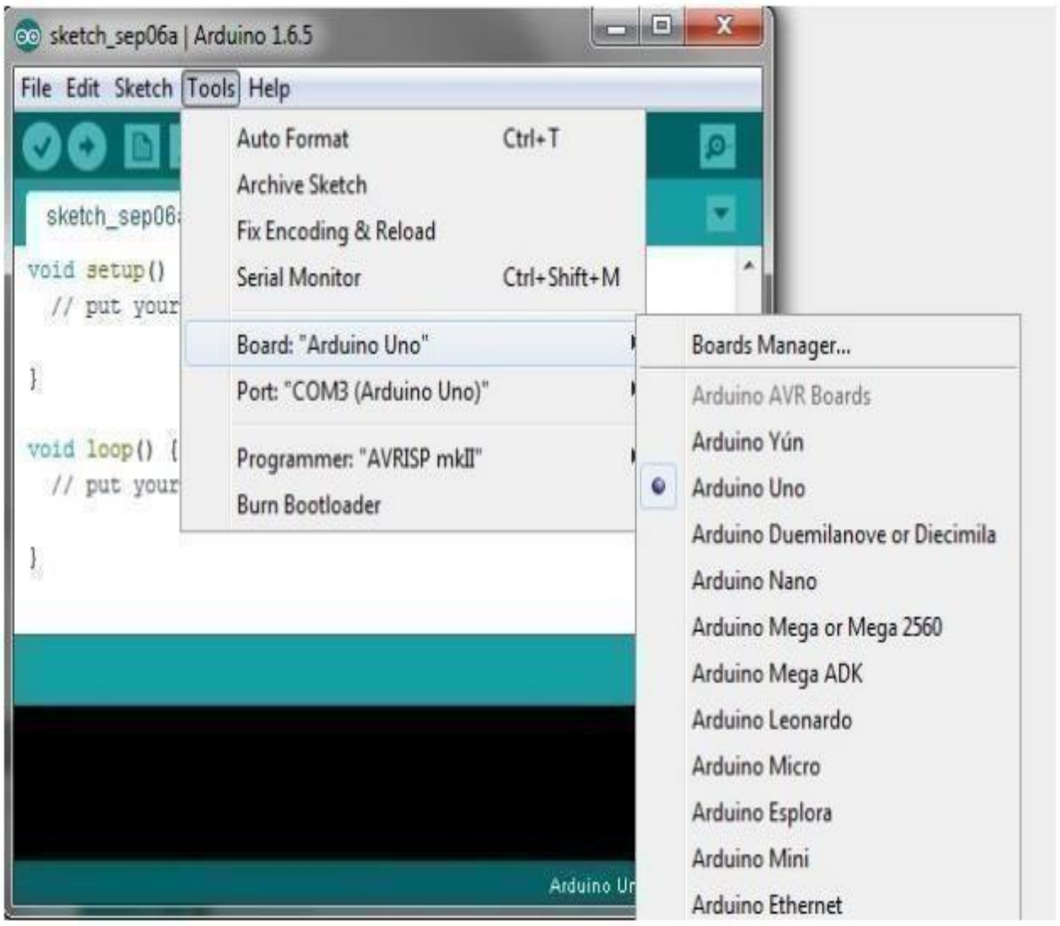


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Fig.4.1: Arduino IDE Default Window

4.1.2 IDE: Board Setup

You have to tell the Arduino IDE what board you are uploading to. Select the Toolspulldown menu and go to Board.This list is populated by default with the currently available Arduino Boards that are developed by Arduino. If you are using an Uno or an Uno-Compatible Clone (ex. Funduino, SainSmart, IEIK, etc.), select Arduino Uno. If you are using another board/clone, select that board.

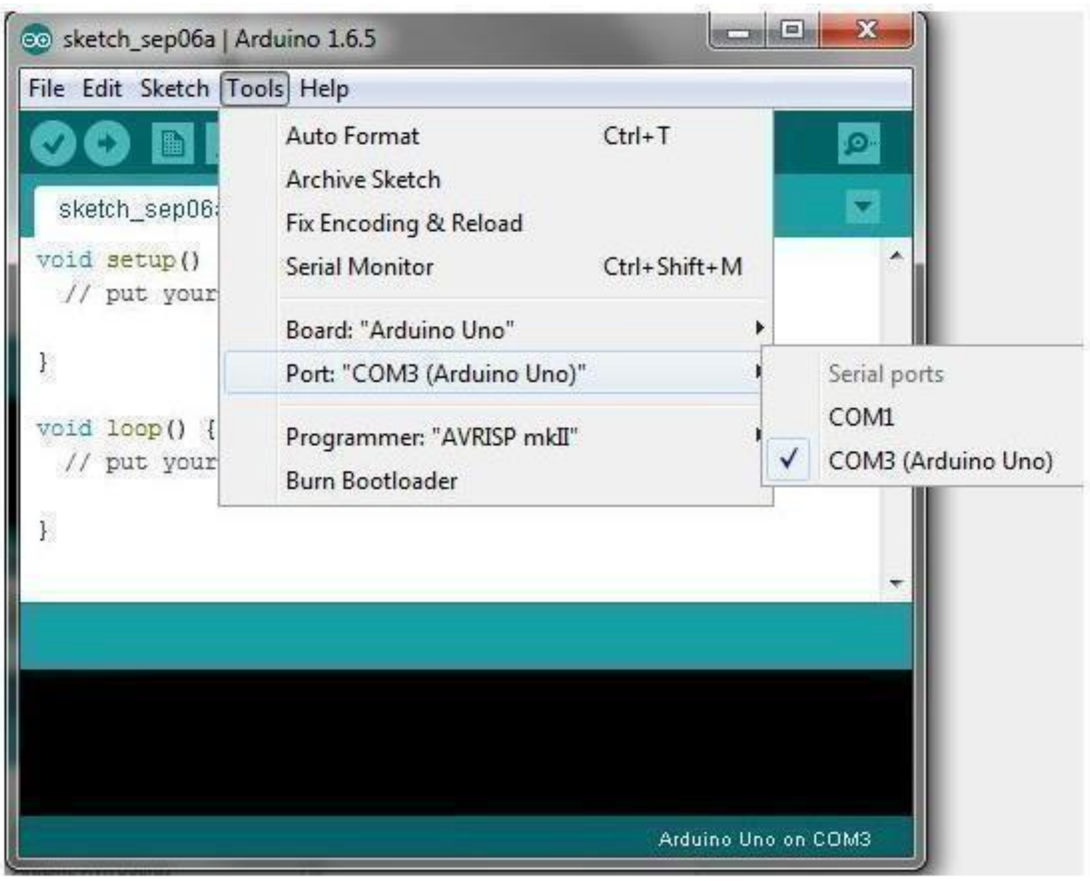


```
void setup()
  // put your
}

void loop() {
  // put your
}
```

Fig.4.2: Arduino IDE: Board Setup Procedure

4.1.3 IDE: COM Port Setup

If you downloaded the Arduino IDE before plugging in your Arduino board, when you plugged in the board, the USB drivers should have installed automatically. The most recent Arduino IDE should recognize connected boards and label them with which COM port they are using. Select the Tools pulldown menu and then Port.Here it should list all open COM ports, and if there is a recognized Arduino Board, it will also give it’s name. Select the Arduino board that you have connected to the PC. If the setup was successful, in the bottom right of the Arduino IDE, you should see the board type and COM number of the board you plan to program. Note: the Arduino Uno occupies the next available COM port; it will not always be COM3.



```
void setup()
  // put your
}

void loop() {
  // put your
}
```

Fig.4.3 Arduino IDE: COM Port Setup

4.2 Testing Your Settings:

Uploading Blink One common procedure to test whether the board you are using is properly set up is to upload the "Blink" sketch. This sketch is included with all Arduino IDE releases and can be accessed by the Filepull-down menu and going to Examples, 01. Basics, and then select Blink. Standard Arduino Boards include a surface-mounted LED labeled "L" or "LED" next to the "RX" and "TX" LEDs, that is connected to digital pin 13. This sketch will blink the LED at a regular interval, and is an easy way to confirm if your board is set up properly and you were successful in uploading code. Open the "Blink" sketch and press the "Upload" button in the upper- left corner to upload "Blink" to the board.

Upload Button:

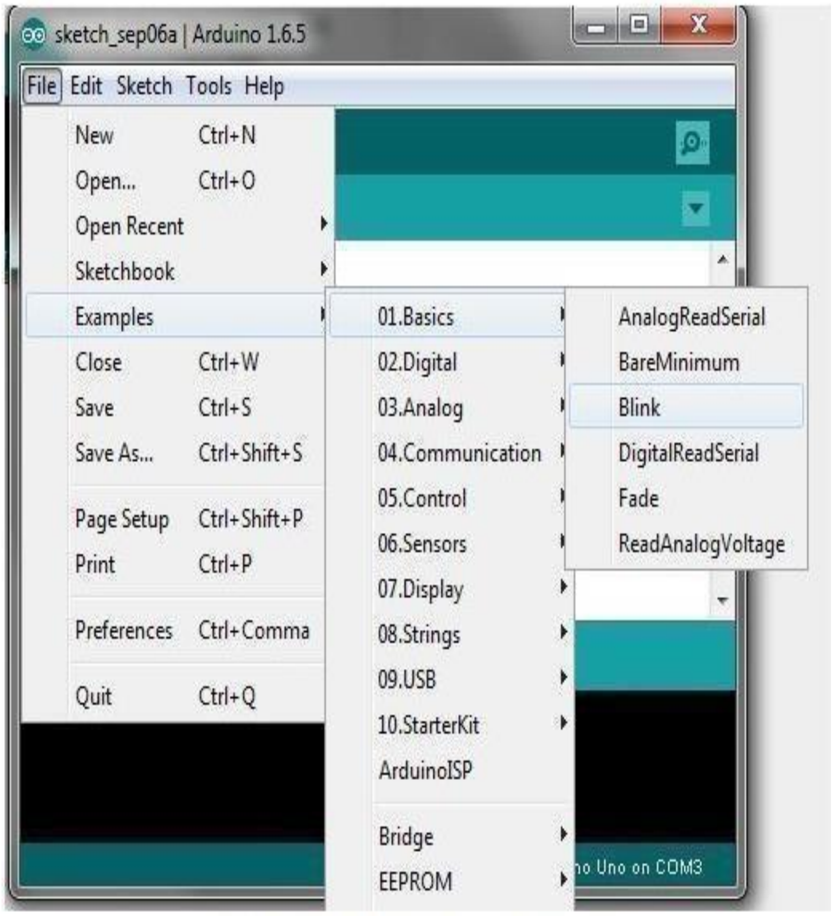
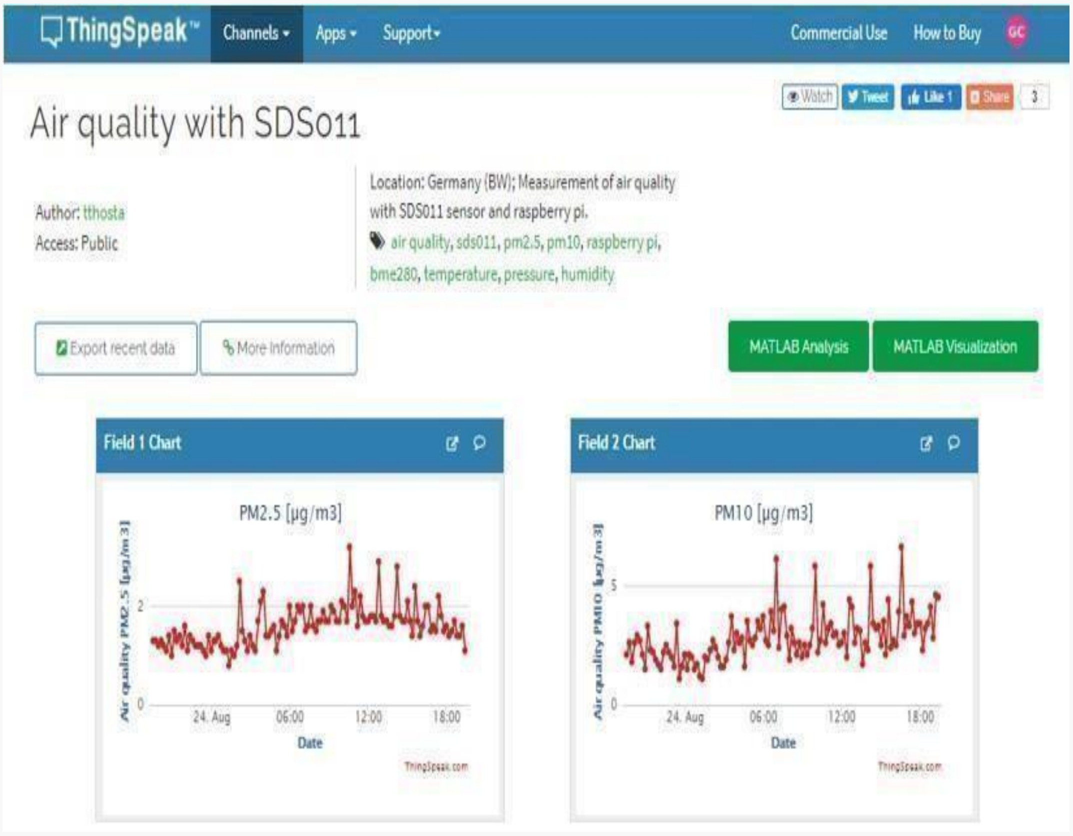


Fig.4.4 Arduino IDE: Loading Blink Sketch

4.3 ThingSpeak IoT Platform

ThingSpeak is IoT platform for user to gather real-time data; for instance, climate information, location data and other device data. In different channels in ThingSpeak, you can summarize information and visualize data online in charts and analyze useful information. ThingSpeak can integrate IOT: bit (micro: bit) and other software/ hardware platforms. Through IOT:bit, you can upload sensors data to ThingSpeak (e.g. temperature, humidity, light intensity, noise, motion, raindrop, distance and other device information).


Air quality with SDS011

Author: tthosta
Access: Public

Location: Germany (BW); Measurement of air quality with SDS011 sensor and raspberry pi.
air quality, sds011, pm2.5, pm10, raspberry pi, bme280, temperature, pressure, humidity

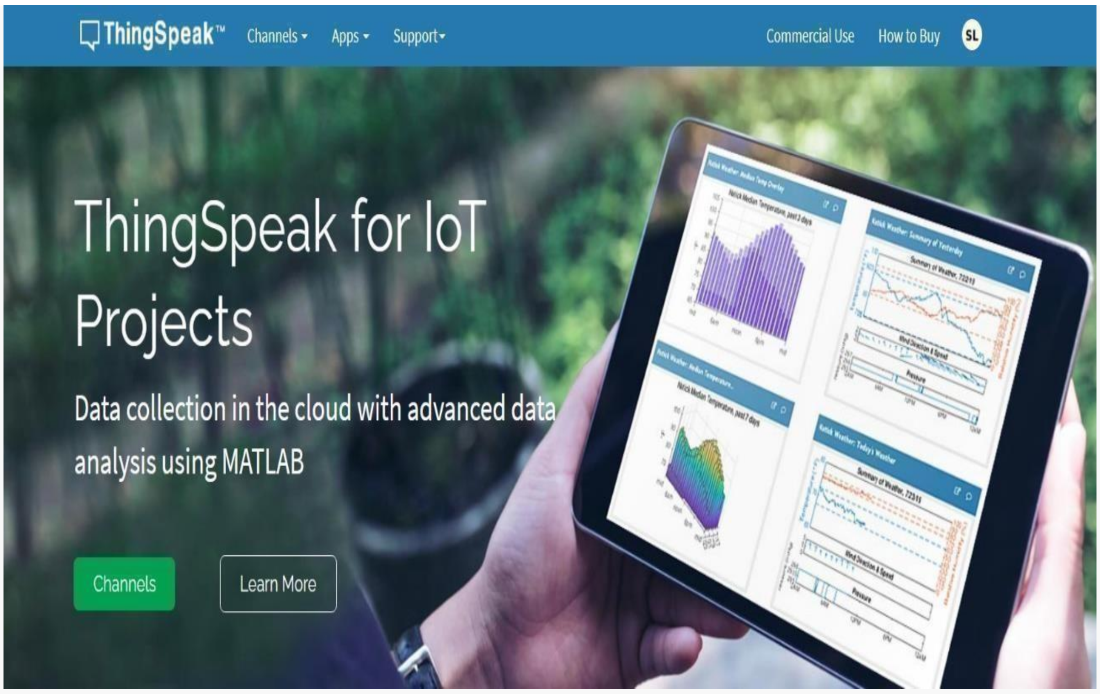
5.1. Thingspeak Configuration

Goal:

we need to create the thingspeak channel and get the key

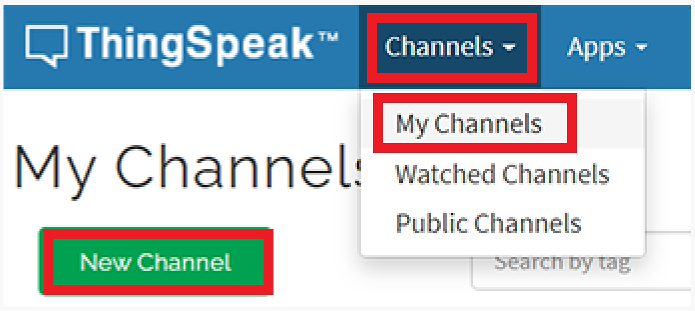
Step 1

Go to https://thingspeak.com/, register an account and login to the platform


ThingSpeak™
Channels
Apps
Support
Commercial Use
How to Buy
ThingSpeak for IoT Projects
Data collection in the cloud with advanced data analysis using MATLAB
Channels
Learn More

Step 2

Choose Channels -> My Channels -> New Channel

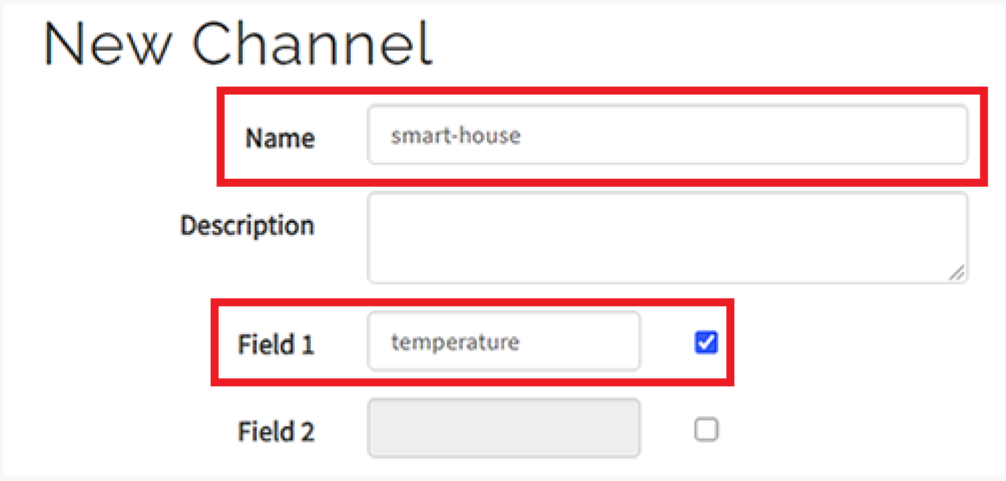


Step 3

Input Channel name, Field1 , then click "Save Channel"

- Channel name: smart-house

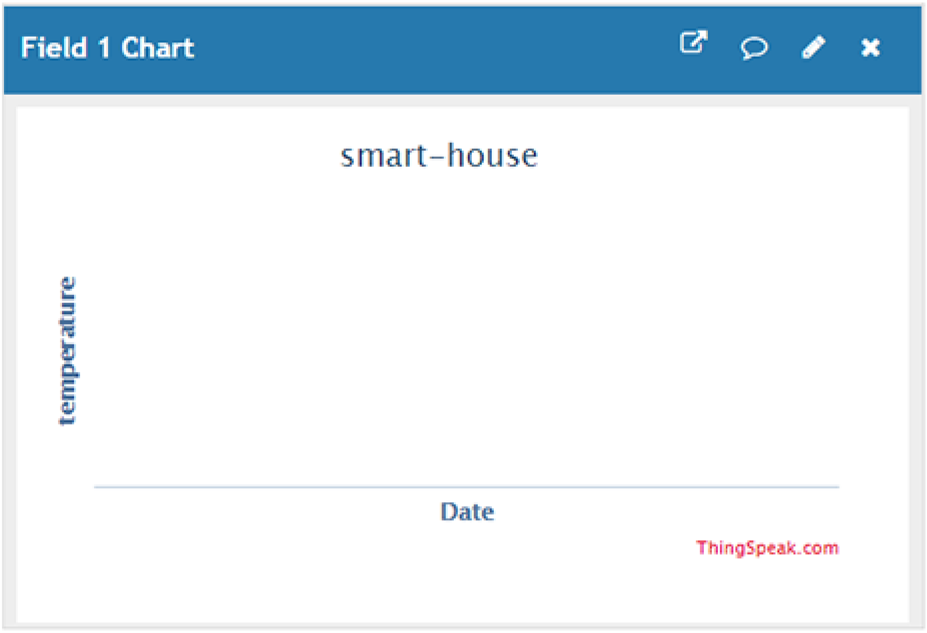
- Field 1: Temperature


New Channel

Name	smart-house	
Description		
Field 1	temperature	✓
Field 2		☐

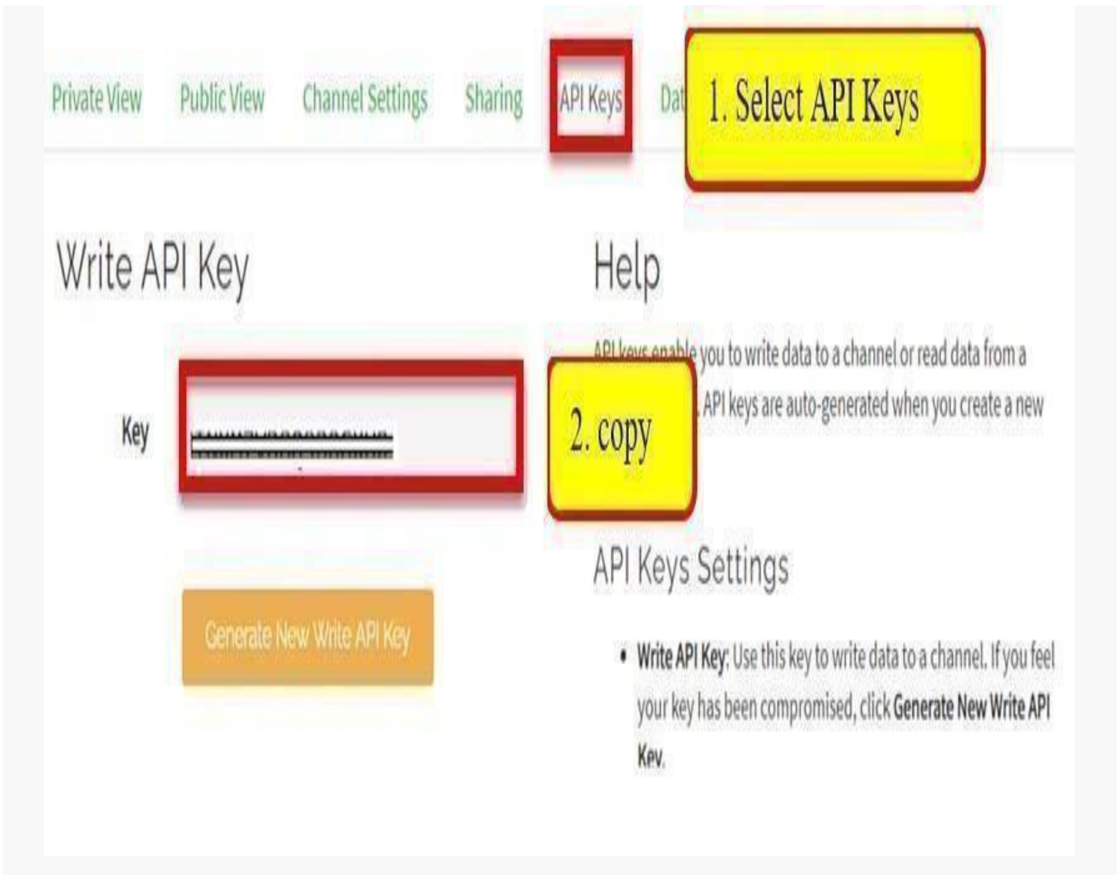
Step 4

You will see a chat for data field1



Step 5

Open your web browser, go to https://thingspeak.com , select your channel > “API Keys” , copy the API key as follows:



Write API Key

Key [redacted]

Help

API keys enable you to write data to a channel or read data from a . API keys are auto-generated when you create a new

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**

CHAPTER-5

RESULT

5.1 Hardware Kit

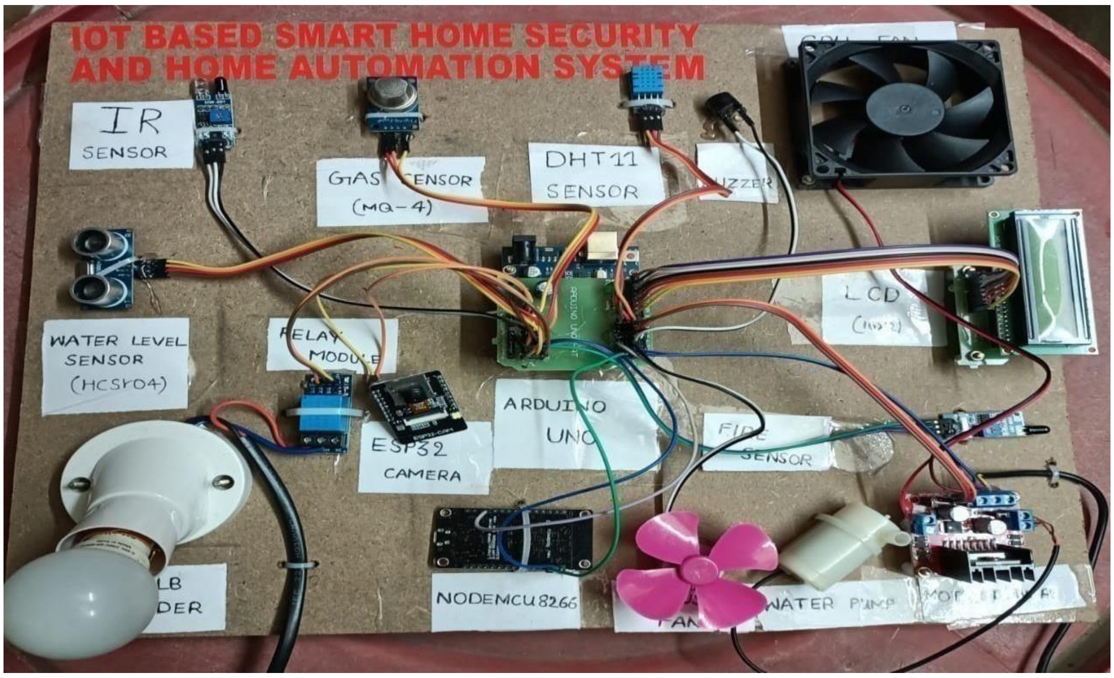


Fig.5.1 Hardware Kit of IoT Based Smart Home Security and Home Automation System

CASE 1: Day time	**CASE 2: Night time**
Temperature: 40^0c	Temperature: 33^0 c
Humidity: 40 g.m^{-3}	Humidity: 66 g.m^{-3}
Water level: 53cm	Water level: 45cm
Gas value: 109ppm	Gas value: 106ppm
Fire: 1(no fire)	Fire: 1(no fire)
Object/human detection: 1	Object/human detection: 0

CHAPTER-6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion:

The wireless devices monitoring and controlling using Internet of Things has been experimentally proven to work satisfactorily by connecting simple appliances to it and the appliances were successfully controlled remotely through internet.

6.3 The future scope of the power theft detection system includes the following:

The future scope of wireless device monitoring and control using IoT and Arduino includes the potential for expanded automation, improved energy efficiency, and enhanced connectivity for a wide range of applications, from smart homes to industrial processes.

The journey of home automation began with basic remote-controlled devices, such as garage door openers and programmable thermostats. However, with the proliferation of smart technology, homes are now equipped with interconnected devices that can be controlled remotely via smartphones or voice commands. From smart lighting and climate control systems to intelligent security cameras and door locks, the possibilities for automation are virtually endless.

He future of home automation holds immense promise, with several trends poised to shape its trajectory. Enhanced connectivity will enable even greater integration between devices, leading to a more cohesive and synchronized smart home experience. Predictive analytics powered by AI will anticipate user needs and automate routine tasks, further enhancing convenience and efficiency. Moreover, the proliferation of smart appliances and energy management systems will promote sustainability by optimizing energy usage and reducing waste. Biometric authentication, blockchain technology, and decentralized architectures may also be employed to enhance security and privacy in smart homes.

REFERENCES

[1] Vinay sagar K N, Kusuma S M, Jan-2016, “Home Automation Using Internet of Things”, International Research Journal of Engineering and Technology, vol. 02, no. 03, pp. 01-10..

[2] Pooja N.Pawar, Shruti Ramachandran, Nisha P.Singh, Varsha V.Wagh, April 2018, “A Home Automation System using Internet of Things”, International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 4, pp. 54-63.

[3] Shweta Singh, Kishore Kumar Ray, 2017, “home automation system using internet of things”, International Journal of Computer Engineering and Applications, pp. 45-49.

[4] K.Saiteja, S.Aruna deepthi, G.Raghu, B.Ravali, April 2017, “Home Automation Using IOT”, International Journal of Engineering Trends and Technology, pp. 229-235. 66

[5] A. Amudha, 2017, “Home Automation using IoT”, International Journal of Electronics Engineering Research,vol. 9, no. 6, pp. 939-944.

[6] Priyanka Zambare , Pooja Madake , Aparna Pottabathini , Prof. Jayant Sawarkar, February 2018, “The Survey on IoT Based Home Automation”, International Journal of Innovative Research in Science Engineering and Technology, vol. 7, no. 2, pp. 223-31.

[7] Abdul Aziz Md, K Harshasri, K Shanmukharao, March 2017, “Cost Effective Voice Controlled Home Automation Using IoT”, International Journal of Engineering Research in Computer Science and Engineering, vol. 4, no. 3, pp. 01-10.

[8] Mamata Khatu, Neethu Kaimal, Pratik Jadhav, Syedali Adnan Rizvi, February 2015, “Implementation of Internet of Things for Home Automation”, International Journal of Emerging Engineering Research and Technology, vol. 3, no. 2, pp. 7-11. [9] B.P Kulkarni, Aniket V Joshi, Vaibhav V Jadhav, Akshaykumar, April 2017, v“IoT Based Home Automation Using Raspberry PI”, International Journal of Innovative Studies in Sciences and Engineering Technology, vol. 3, no. 4, pp. 13- 19. [10] Dhakad Kunal, Dhake Tushar, Undegaonkar Pooja, Zope Vaibhav, February 2016, “Smart Home Automation using IoT”, International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 2, pp. 56-61.