IOT BASED ROBOTIC ARM FOR ASSISTING OLDER PEOPLE AND SMART HOME AUTOMATION SYSTEM

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

S.Hassireen (318126512048)   P.Vaishnavi(318126512037)

B.Sai Siva Harsha (318126512012)   D.Soma Sekhar (318126512052)

Under the guidance of

Mr.Nitin Kumar

Asst.Professor

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)
Sangivalasa, bheemili mandal, visakhapatnam dist.(A.P)
2021-2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"IOT BASED ROBOTIC ARM FOR ASSISTING OLDER PEOPLE AND SMART HOME AUTOMATION SYSTEM"** submitted **Sk.Hassireen (318126512048) ,P.Vaishnavi (318126512037), B.Sai Siva Harsha (318126512012) , D.Soma Sekhar (318126512052)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Nitin Kumar
31/5/22

Project Guide

Mr.Nitin Kumar

Asst. Professor

Department of E.C.E

ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of the Department

Dr. V.Rajyalakshmi

Professor&HOD

Department of E.C.E

ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa-531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Nitin Kumar** Designation, Department of Electronics and Communication Engineering, ANITS, for his/her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

S.Hassireen(318126512048),
P.Vaishnavi(313126512037),
B.Sai Siva Harsha(313126512012),
D.Soma Sekhar(313126512052),

ABSTRACT

Robotic arm is a reprogrammable and multifunctional manipulator design to assist human in various surroundings. It is able to overcome human inefficiency in performing repetitive task such as pick and place operation. The Robotic Arm is controlled by Raspberry pi 4 with the help of Internet of Things(IOT). The programming used in Raspberry pi 4 is java to control the Robotic Arm. Internet of things (IOT) allow data to be exchange between devices through the connection of many devices. The integration of internet of things with robotic arm allows smart industry to be realized. The Robotic Arm helps older people in giving medicines etc. The Robotic Arm consists of servo motors and assigned with Pulse Width Modulation Signals. We can control the movement of Robotic Arm by using Blynk App with the help IOT.

In Addition to this we are controlling the bulb and dc fan with the help of IOT by using Node MCU with the help of IOT. Node MCU is an open source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. To see the visuals of the old age people we have used camera which is interfaced with raspberry pi 4 board so, that we can see the visuals by using Blynk app. With the help of this prototype we can control the old age persons from anyplace with the help of IOT

CONTENTS

ACKNOWLEDGEMENT
ABSTRACT
LIST OF FIGURES

CHAPTER 1

CHAPTER 2

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In today's world 15% of the population or some 1 billion individuals live with one or more disabling conditions. More than 46 percent of older persons , those aged 60 years and over have disabilities and more than 250 million older people experience moderate to severe disability. Looking ahead, the global trends in ageing populations and the higher risk of disability in older people are likely to lead to further increases in the population affected by disability. Older people or physically handicapped people usually cannot walk and do things as efficiently as young people . Such people need a care taker to carry out their regular activities. Instead of such care takers, one can make use of current technologies and develop an embedded system that can help them in carrying out their regular activities without the need to move from one place to other. One such application is robotic arm that can help elderly people or physically handicapped people in bringing the things they require with the help of a controlling device .

1.2 Embedded Systems

An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. At the core is an integrated circuit designed to carry out computation for real-time operations.Complexities range from a single microcontroller to a suite of processors with connected peripherals and networks; from no user interface to complex graphical user interfaces. The complexity of an embedded system varies significantly depending on the task for which it is designed.

Embedded system applications range from digital watches and microwaves to hybrid vehicles and avionics. As much as 98 percent of all microprocessors manufactured are used in embedded systems.

Embedded systems are managed by microcontrollers or digital signal processors (DSP), application-specific integrated circuits (ASIC), field-programmable gate arrays (FPGA), GPU technology, and gate arrays. These processing systems are integrated with components dedicated to handling electric and/or mechanical interfacing.

Embedded systems programming instructions, referred to as firmware, are stored in read-only memory or flash memory chips, running with limited computer hardware resources. Embedded systems connect with the outside world through peripherals, linking input and output devices.

1.2.1 Basic structure of embedded systems

The basic structure of an embedded system includes the following components:

Sensor: The sensor measures and converts the physical quantity to an electrical signal, which can then be read by an embedded systems engineer or any electronic instrument. A sensor stores the measured quantity to the memory.

A-D Converter: An analog-to-digital converter converts the analog signal sent by the sensor into a digital signal.

Processor & ASICs: Processors assess the data to measure the output and store it to the memory.

D-A Converter: A digital-to-analog converter changes the digital data fed by the processor to analog data

Actuator: An actuator compares the output given by the D-A Converter to the actual output stored and stores the approved output.

The current Internet revolution along with the increasing robotics in many day-to-day operations of the Internet of Things has taken over all
the heavy loads from the person to itself.

1.3 Internet of things

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

A thing in the internet of things can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low or any other natural or man-made object that can be assigned an Internet Protocol (IP) address and is able to transfer data over a network.Increasingly, organizations in a variety of industries are using IoT to operate more efficiently, better understand customers to deliver enhanced customer service, improve decision-making and increase the value of the business.

An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices for instance, to set them up, give them instructions or access the data.

1.4 Robotic arm

Robot are used in various environment ranging from homes to industries to provide assistant to human for a range of purpose such as operation in hazardous situations or operation in manufacturing sector . The design of a robot can be varied depending on its function, application and the environment. Robotic arm is most commonly used in industry such as manufacturing and assembling. The use of robotic arm is to overcome human inefficiency in performing repetitive task such as pick and place operation. Robotic arm is a reprogrammable and multifunctional manipulator design to assist human in various surroundings.

Robotic arm is able to overcome human inefficiency in performing repetitive task such as pick and place. Robotic is a fusion of not only electronic, mechanical and electrical engineering but it also includes other engineering fields such as computer science, mathematics, technology and science . In general, robotic arm is a reprogrammable and multifunctional manipulator that resemble a human arm. The movement of robotic arm is driven by actuator which is located at the joint of the robotic arm. Motion of human arm differs from the motion of articulated robot arms. Although the robot joints have fewer degrees of freedom, they are able to move in larger angles. For example, the elbow of an articulated robot can bend up or down whereas a person can only bend their elbow in one direction with respect to the straight arm position . Actuator is a device or component that converts energy into motion. Example of actuator are AC motor, DC motor, stepper motor, hydraulic piston and pneumatic piston. End effector is the tool that perform work or operation and is usually located at the end of the kinematic chain . If we combine the robots and the Internet of things t hen we can do more than we would imagine.

CHAPTER 2

HARDWARE REQUIRED

2.1 COMPONENTS REQUIRED

- Raspberry pi 4 Model b
- Node MCU esp8266
- Relay module
- Robotic Arm
- Servo Motor
- L298 Motor Driver IC
- Raspberry pi 4 camera
- Lead Acid Battery
- DC fan
- Electric bulb
- Connecting Jumper wires

2.2 Raspberry pi 4

The Raspberry pi is a single computer board with credit card size, that can be used for many tasks that your computer does, like games, word processing, spreadsheets and also to play HD video. It was established by the Raspberry pi foundation from the UK. It has been ready for public consumption since 2012 with the idea of making a low-cost educational microcomputer for students and children. The main purpose of designing the raspberry pi board is, to encourage learning, experimentation and innovation for school level students. The raspberry pi board is a portable and low cost. Maximum of the raspberry pi computers is used in mobile phones. In the 20th century, the growth of mobile computing technologies is very high, a huge segment of this being driven by the mobile industries. The 98% of the mobile phones were using ARM technology.

Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems. This product’s key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 8GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on). The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.

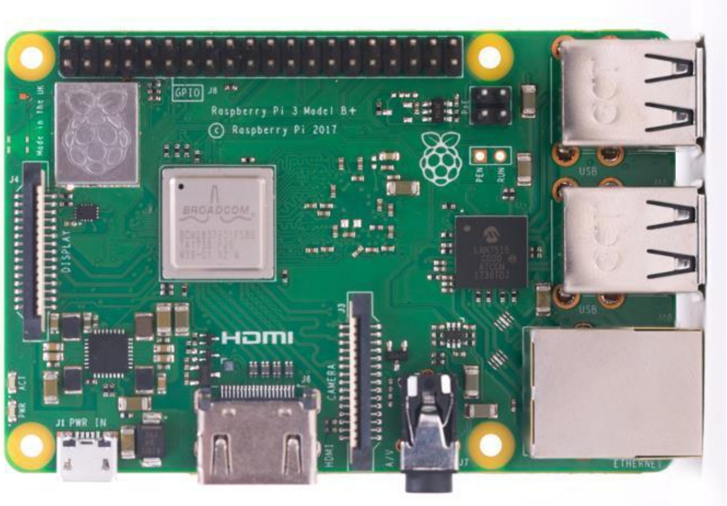


Fig 2.1 Raspberry pi 4

2.2.1 Pin Configuation of Raspberry pi 4

A Raspberry Pi 4 board has 40 pins on it. Among these pins, we have four power pins on the Raspberry Pi, two of which are 5v pins and another two are 3.3v pins. The 5v power pins are connected directly to the Raspberry Pi's power input and we can use these pins to run low power applications.

Then there are the ground pins. There are eight ground pins and all of these are connected to each other; you can use any of these ground pins for your projects.

That leaves us with 28 GPIO pins, labeled starting from GPIO 0 and going up to GPIO 27.

The GPIO pins, as indicated by their full form, can be programmed to be output pins or input pins. So we can set values of output pins and we can even read values of input pins. The GPIO pins can be digitally programmed so that they can be turned ON or OFF. The output of any GPIO pin is 3.3v and can be used to control output components like an LED or a motor. These ON/OFF conditions can also be interpreted as a Boolean True/False, 1/0 or HIGH/LOW.

2.2.2 Ports of Raspberry pi 4

USB port: It consists of two USB 3.0 and two USB 2.0 ports to connect it to an external keyboard, mouse, or other peripheral devices.

USB power port: It consists of a 5.1V, 3A USB type-C power port.

HDMI port: Two micro HDMI ports capable of supporting up to 4k@60HZ resolution.

Ethernet Port: It comes with true Gigabit Ethernet capable of sending *Ethernet* frames at a rate of one *gigabit* per second (1 billion bits per second).

Composite Video Output: Both the audio output socket and the video composite socket reside in a single 4-pole 3.5mm socket.

SD card Slot: A micro-SD card slot is used for booting up the operating system and storage purposes.

2.2.3 Power Pins

The raspberry pi 4 model B board consists of two 5V pins, two 3V3 pins, and 7 ground pins (0V).

5V: The 5v pin outputs the 5 volts coming from the USB Type-C port.

3.3V: The 3v pin is used to provide a stable 3.3v supply to external components.

GND: The ground pin is commonly referred to as GND.

2.2.4 GPIO PINS

A pin that can be set as a input or output and is controlled in run time is called GPIO pin.

PWM (pulse-width modulation) pins:

PWM stands for "Pulse Width Modulation". It means that an analog value is being modulated on a digital signal.

Software PWM is available on all pins.

Hardware PWM is available on these pins only: **GPIO12, GPIO13, GPIO18, GPIO19**

SPI PINS:

SPI (Serial Peripheral Interface) is a type of serial communication protocol. It is used by the Raspberry Pi for master-slave communication to quickly communicate between one or more peripheral devices.

The data is synchronized using a clock (**SCLK** at GPIO11) from the master (RPi).
The Pi sends this data to an SPI device using **MOSI** (Master Out Slave In) pin. And when the SPI device needs to communicate back to the Raspberry Pi, it sends the data back through the **MISO** (Master In Slave Out) pin.

5 pins are required for SPI communication:

- **GND**: Connect the GND pin from all the slave components and the Raspberry Pi 4 board together.
- **SCLK**: Clock for SPI communication.
- **MOSI**: It stands for Master Out Slave In. This pin is used to send data from the master to a slave.
- **MISO**: It stands for Master In Slave Out. This pin is used to receive data from a slave to the master.
- **CE**: It stands for Chip Enable. We need to connect one CE pin per slave (or peripheral devices) in our circuit. By default, we have two CE pins but we can configure more CE pins from the other available GPIO pins.

SPI0: GPIO9 (MISO), GPIO10 (MOSI), GPIO11 (SCLK), GPIO8 (CE0), GPIO7 (CE1)

SPI1: GPIO19 (MISO), GPIO20 (MOSI), GPIO21 (SCLK), GPIO18 (CE0), GPIO17 (CE1), GPIO16 (CE2)

I2C PINS:

I2C pins on the Raspberry Pi board are used to communicate with peripheral devices that are compatible with **Inter-Integrated Circuit** (a low-speed two-wire serial communication protocol).

This serial communication protocol requires master-slave roles between both, the board and the peripheral devices.

I2C protocol requires two connections: **SDA (Serial Data)** and **SCL (Serial Clock)**. They work by transmitting data using the SDA connection, and the speed of data transfer is controlled via the SCLK pin.

Data: (GPIO2), Clock (GPIO3)

EEPROM Data: (GPIO0), EEPROM Clock (GPIO1)

UART PINS:

The **UART** (Universal Asynchronous Receiver / Transmitter) is an asynchronous protocol that provides a way to communicate between two microcontrollers or devices.

TX pin transmits the serial data to the RX pin of another device and RX pin receives the serial data coming from TX pin of the other device.

TX : GPIO14

RX : GPIO15

2.2.5 Specifications of Raspberry pi 4

- Broadcom BCM2711 chip consist of Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 2GB, 4GB, and 8GB of LPDDR4 SDRAM (depending on the version of the board)
- Dual-channel 2.4/5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- Two USB 3.0 ports and two USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header
- Two micro-HDMI ports (support up to 4kp60 resolution)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- 265 (4k@60 decode), H264 (1080@60 decode and 1080@30 encode)
- OpenGL ES 3.0 graphics
- Micro-SD card slot for loading operating system and data storage
- 1V/3A DC via USB-C connector
- Power over Ethernet (PoE) enabled (requires PoE HAT board)
- Operating temperature: 0 – 50^{o}C

2.3 NODE_MCU:

2.3.1 About Node_MCU:

Node MCU is an open-source Lua based firmware and **development board** specially targeted for IoT based Applications. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module.

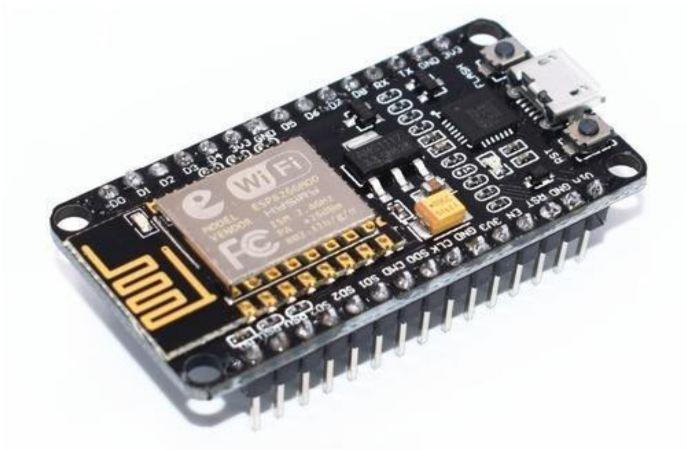


Fig 2.2 Node_MCU ESP8266

The **Node MCU ESP8266 development board** comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency.

NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects. Node MCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

In fig.4 The ESP8266 is the low cost WiFi microchip this small module allows a microcontroller to connect to the WiFi network. Node_MCU is an open source firmware for which we can program it using Arduino IDE so we have uploaded a program for motion control of our robot. Node_MCU can be powered using Micro USB jack and VIN pin External Supply Pin. It supports UART, SPI, and I2C interface.

Node_MCU Development Board can programmed with an Arduino IDE since it is easy to use. . Programming Node MCU with the Arduino IDE is much easier. All you need is the Arduino IDE,a USB cable and the Node MCU board.

2.3.2 Pin Configuation of Node MCU

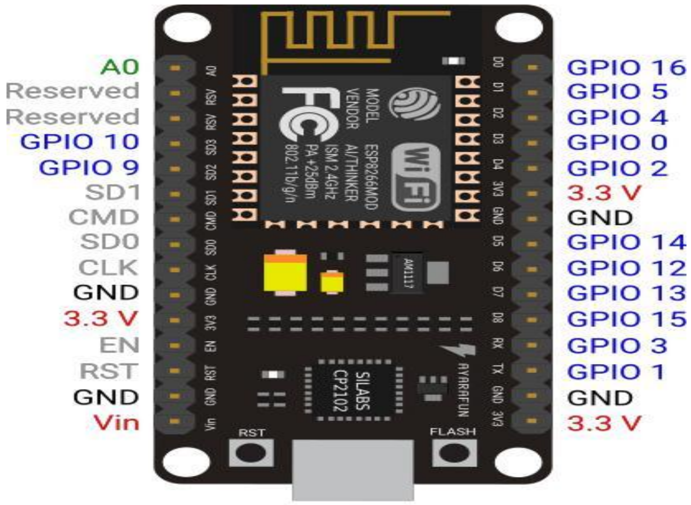


Fig 2.3 Node_MCU Pin diagram

Power Pins

There are four power pins viz. one VIN pin & three 3.3V pins. The VIN pin can be used to directly supply the ESP8266 and its peripherals, if you have a regulated 5V voltage source. The 3.3V pins are the output of an on-board voltage regulator. These pins can be used to supply power to external components.

Ground

It is a ground pin of ESP8266 NodeMCU development board.

I2C pins

These pins are used to hook up all sorts of I2C sensors and peripherals in your project. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins

ESP8266 NodeMCU has 17 GPIO pins which can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel

The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC viz. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins

ESP8266 NodeMCU has 2 UART interfaces, i.e. UART0 and UART1, which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. It supports fluid control. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins

ESP8266 features two SPIs (SPI and HSPI) in slave and master modes.

These SPIs also support the following general-purpose SPI features:

1. 4 timing modes of the SPI format transfer
2. Up to 80 MHz and the divided clocks of 80 MHz
3. Up to 64-Byte FIFO

SDIO Pins

ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins

The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 μs to 10000 μs, i.e. between 100 Hz and 1 kHz.

Control Pins

These pins are used to control ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

EN pin

The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.

RST pin

RST pin is used to reset the ESP8266 chip.

WAKE pin

Wake pin is used to wake the chip from deep-sleep.

2.3.3 Power Requirement

As the operating voltage range of ESP8266 is **3V to 3.6V**, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as **80mA during RF transmissions**. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

2.3.4 NodeMCU ESP8266 Specifications & Features:

1. Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
2. Operating Voltage: 3.3V
3. Input Voltage: 7-12V
4. Digital I/O Pins (DIO): 16
5. Analog Input Pins (ADC): 1
6. UARTs: 1 PIN
7. SPIs: 1 PIN
8. I2Cs: 1 PIN

9. Flash Memory: 4 MB
10. SRAM: 64 KB
11. Clock Speed: 80 MHz
12. USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
13. PCB Antenna
14. Small Sized module to fit smartly inside your IoT projects.

2.3.5 Applications of Node MCU

- Prototyping of IoT devices
- Low power battery operated applications
- Network projects
- Projects requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities
- Smallest IoT Home Automation using ESP8266 – 01 with Vide
- Geolocation using ESP8266.
- ESP8266 Projects: Wireless Web Server. ...
- World's smallest IoT project. ...
- Air Pollution Meter. ...
- Humidity and temperature monitoring.

2.4 RELAY MODULE

2.4.1 About Relay Module

The relay module is an electrically operated switch that can be turned on or off deciding to let current flow through or not. They are designed to be controlled with low voltages like 3.3V like the ESP32, ESP8266, etc, or 5V like your Arduino.

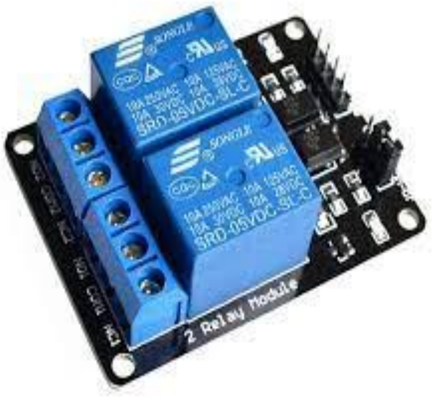


Fig 2.4 Relay Module

Relay is one kind of electro-mechanical component that functions as a switch. The relay coil is energized by DC so that contact switches can be opened or closed. A single channel 5V relay module generally includes a coil, and two contacts like normally open (NO) and normally closed (NC)

2.4.2 Pin Configuation

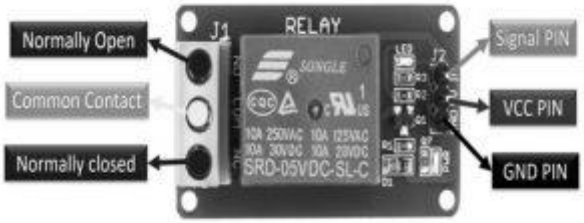


Fig 2.5 Pin Configuation of Relay Module

Normally Open (NO): This pin is normally open unless we provide a signal to the relay modules signal pin. So, the common contact pin smashes its link through the NC pin to make a connection through the NO pin

Common Contact: This pin is used to connect through the load that we desire to switch by using the module.

Normally Closed (NC): This NC pin is connected through the COM pin to form a closed circuit. However, this NC connection will break once the relay is switched through providing an active high/low signal toward the signal pin from a microcontroller.

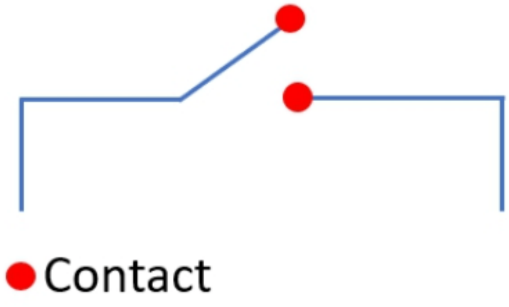
Signal Pin: The signal pin is mainly used for controlling the relay. This pin works in two cases like active low otherwise active high. So, in active low case, the relay activates once we provide an active low signal toward the signal pin, whereas, in an active high case, the relay will trigger once we provide a high signal toward the signal pin.
However, these modules generally work on an active high signal which will strengthen the relay coil to make contact with the common terminal with the normally open terminal.

5V VCC: This pin needs 5V DC to work. So 5V DC power supply is provided to this pin.

Ground: This pin connects the GND terminal of the power supply

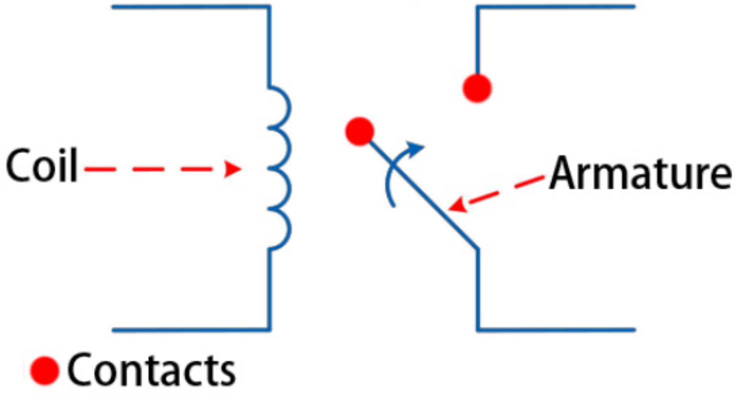
2.4.3 Working of Relay Module

The relay is an electrically operated switch where the relay opens when the two contacts are disconnected, while the relay is closed when the two contacts touch. When set to high, the relay will close allowing current to flow.



Relay Contact

Even though there are many types of relays, electromechanical relays are the most commonly used which we are going to talk about them and how they work. They consist of coils, armatures, and contacts:



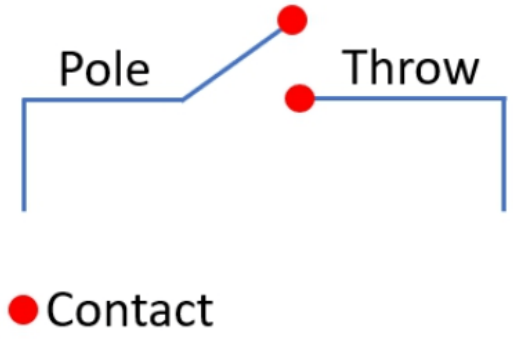
When the coil is energized, the induced magnetic field moves the armature, which opens or closes the contact.

Each contact connects to an input or output terminal. The input terminal is called **Pole**, and the output terminal is called **Throw**. According to the number of terminals, the relay is divided into several types. where the commonly used are the **SPST and SPDT.** Let’s look at how the SPST and the SPDT works:

SPST (Single Pole Single Throw)

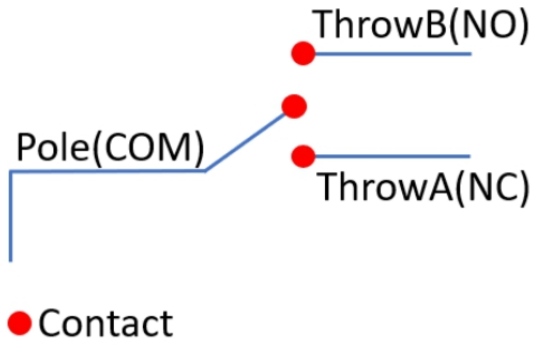
The SPST is the simplest relay, you can consider it as a button. They have 2 terminals that can be connected or disconnected. Including 2 for the coil, an SPST relay has 4 terminals in total

This switch is normally open and when the trigger signal comes, the pole contact will connect to the throw contact which causes the switch to be closed. It is great for applications that need only an on or off state. If this the relay you are looking for, the Grove-Relay is perfect for you.



SPDT(Single Pole Double Throw):

The SPDT relay is also known as the A/B switch, as you can see below, there are two throws, this kind of relay is great for selecting between two options.



- **COM –** Common terminal
- **NC –** Normally closed 120 – 240V terminal
- **NO –** Normally open 120 – 240V terminal

2.4.4 Specfications of Relay Module

Operating Voltage (VDC)	2.5 ~ 5.5
Trigger Voltage (VDC)	5
Trigger Current (mA)	20
Switching Voltage (VAC)	250@10A
Switching Voltage (VDC)	30@10A
Operating Temperature (°C)	-40 to 85
Opt. Relative Humidity(RH)	20% ～ 85%
Storage condition (℃)	– 65 to 125

Length (mm)	43
Width (mm)	43
Height (mm)	17
Weight (gm)	25
Shipment Weight	0.03 kg
Shipment Dimensions	7 × 10 × 2 cm

2.4.5 Features of Relay Module

- The module using Songle relay control.
- AC voltage 250V, AC current 10A, maximum DC voltage 30V DC current maximum 10A.
- Power indicator (green), Two ways of relay status indicator light (red).
- Using 8550 transistor-driven, strong driving ability.
- Has the fixed bolt hole and easy installation.

2.5 ROBOTIC ARM

2.5.1 About Robotic Arm

A robotic arm is a robot manipulator and a mechanical arm, usuallyprogrammable, with similar functions to a human arm. The links of such a manipulator are connected by joints allowing either rotational motion (such as inan articulated robot) or translational (linear) displacement. The links of themanipulator can be considered to form a kinematic chain. The business end of the kinematic chain of the manipulator is called the end effectors and it is analogousto the human hand. The end effectors can be designed to perform any desiredtask such as welding, gripping, spinning etc., depending on the application. Therobot arms can be autonomous or controlled manually and can be used to perform a variety of tasks with great accuracy. The robotic arm can be fixed ormobile (i.e. wheeled) and can be designed for industrial or home applications.

2.5.2 Principle of robotic arm

The computer controls the robot by rotating individual stepper motors connected to each joint (some larger arms use hydraulics or pneumatics). Unlike ordinary motors, step motors move in exact increments. This allows the computer to move the arm very precisely, performing the same movement over and over.

Fig 2.6 Robotic Arm

2.5.3 Uses of Robotic Arm

Robotic arms can be used for all manner of industrial production, processing and manufacturing roles - any task in which extremely precise, fast and repeatable movements are required, in fact.

Robotic arms of all kinds are used today at every scale of manufacturing, from minutely detailed circuit board assembly to large-volume heavy industries such as automotive production lines, as well as in a huge range of 'pick and place' (conveyor belt) applications. This means that it's important to know which types of programmable robotic arms are better suited to which sorts of environments and tasks before you begin planning a purchase.

In every case, selecting the right type of programmable robot arm for a given role or task should involve consideration of the intended application's precise nature and requirements. These will typically include:

- **Load**
  - All types of robotic arms have a given load capacity, and this manufacturer-specified number always needs to exceed the total weight of the payload involved in any job you expect the arm to perform (including tools and attachments).
  - Different sorts of robot arms are supported by differently designed frameworks, which can increase or decrease overall load capacity - this must be balanced with consideration of physical placement and footprint.

- **Orientation**
    - This criterion is generally defined by the footprint and mounting position of the robotic arm, and how well it fits alongside the other equipment in your production line for the range of movements and manipulations it's expected to perform. This will in turn influence where the arm can physically be positioned relative to the objects it will be moving.
    - Certain types of robotic arms require bulkier pedestals or more physical clearance space to perform their programmed range of movements, and these factors must be considered in terms of other equipment or workers in the vicinity.
- **Speed**
    - Particularly when choosing robotic arms for picking and placement applications, it's important to pay attention to manufacturer ratings for speed, and especially in terms of acceleration over longer distances.
    - Changes and upgrades to speed ratings can be achieved in some types of robotic arm through changes made to the choice of belts, motors or actuators used.
- **Travel**
    - Tolerances and accuracy over wider spans can be reduced in certain types of robot arms, due to arm deflection and differences in support framework design.
    - If the application requires longer travel distances between payloads or work areas, this may dictate which sorts of robotic arms would be suitable or unsuitable for performing the task, depending on the tightness of tolerances required.
- **Precision**
    - Certain types of programmable robotic arms are inherently designed to be more precise in their range of movements and articulations than others. This may come at higher cost for a more complex machine, and involve a compromise against other factors such as footprint, speed, potential travel distance and orientation.
    - For many industrial applications such as picking and placement, robotic arms capable of extremely precise repeatable movement may be an unnecessary expense. However, for tooling applications, precision will be a key consideration before most other factors. Again, changes and upgrades can be made to improve precision for certain types of robotic arm, but not all.
- **Environment**
    - Consideration of atmospheric conditions and potential hazards (including dust, dirt and moisture levels) in the immediate working environment will be important when choosing an appropriate type of robotic arm for a specific location.
    - Physical footprint, orientation and range of movement will also influence how suitable a particular model or arm type is for use in a particular environment, with other equipment and workers taken into account.

2.5.4 SPECIFICATIONS OF ROBOTIC ARM

1. The degree of Freedom (DOF): 2
2. Paw material: aluminum alloy

3. Paw weight: about 68g (without motor)
4. Claws overall length: 108mm (overall maximum length when closed paw)
5. Claws overall width: 98mm (maximum overall width of the paw when open)
6. Paw maximum opening angle: 55mm

2.5.5 Applications of Robotic Arm

Palletizing

Robotic arms can be used to automate the process of placing goods or products onto pallets. By automating the process, palletizing becomes more accurate, cost-effective, and predictable. The use of robotic arms also frees human workers from performing tasks that present a risk of bodily injury.

Material Handling

Material-handling robotic arms can help create a safe and efficient warehouse by ensuring goods and materials are properly stored, easy to find, or transported correctly. Automating these processes can help accelerate the delivery of goods to customers, prevent workplace accidents, and improve the efficiency of a facility.

Welding

Welding is a task that can be performed by robots in advanced industrial settings such as automotive manufacturing. Given its critical impact on product quality, welding is an excellent candidate for advanced robotics with vision and AI augmentation for inline quality inspection.

Inspection

Performing quality inspection is typically completed at the end of a production line, which delays the detection of production quality issues. By enhancing robots with vision and AI systems, businesses can benefit from real-time inspection, helping to reduce waste and downtime.

Pick and Place

Pick-and-place robots are typically used in modern manufacturing and logistics. They are equipped with advanced machine vision systems to identify an object, grasp it, and move it from one location to another —quickly and efficiently— to increase speed of production and distribution of goods.

2.6 LEAD ACID BATTERY

The lead–acid battery consists of two electrodes submerged in an electrolyte of sulfuric acid. The positive electrode is made of grains of metallic lead oxide, while the negative electrode is attached to a grid of metallic lead. Lead–acid batteries are classified into two types: flooded and valve-regulated. Flooded lead–acid batteries are less expensive but require more maintenance and ventilation than the valve-regulated lead–acid batteries. Lead–acid batteries are commonly used for renewable energy systems, largely because they can be easily transported and they have

relatively low costs. However, these batteries have some challenges, including low numbers of charging–discharging cycles over their lifetimes, low discharge intensities, restricted lifetimes, and slow charging rates . More advanced lead–acid batteries are continually being introduced. New lead–acid batteries utilize carbon on the negative electrode to create a supercapacitor negative electrode. In these batteries, the positive electrode does not undergo any change in its chemical process, and no chemical process occurs at the negative electrode. As a result, the positive electrode is less subject to corrosion, leading to longer lifetimes and higher efficiencies than conventional lead–acid batteries.

Fig 2.7 Lead Acid Battery

2.7 L298 MOTOR DRIVER IC

An **L298 Motor Driver IC** is a monolithic chip, used in motor driver modules to control the speed of a DC motor. At present, the most frequently used motor driver ICs as compared to L298 is; L293D & L2938N. This IC is frequently used in RC cars & autonomous robots. The input provided to a motor driver module is taken from a controller like Arduino.

So this logic input is simply used to control the direction of the motor which is connected to the motor driver IC. The motor driver module mainly includes a motor driver IC, which is an essential component in this module. This single IC can control the motor alone but using the motor driver module by interfacing it with Arduino can make it easy. This article discusses an overview of L298 motor driver IC, working with applications.

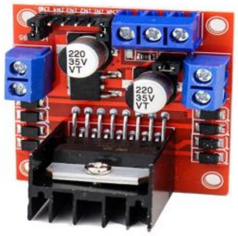


Fig 2.8 L298 Motor Driver IC

*2.7.*1 About L298 Motor Driver IC

The high-power version of the L298 IC is an L293 motor driver IC. It is a dual full-bridge driver IC with high current & voltage, mainly designed to allow typical TTL logic levels to control different inductive loads like DC motors, solenoids, relays, stepper motors, etc. The motor driver is a small current amplifier that uses a low current signal to provide a high current signal for driving an electric motor.

L298 IC includes four separate power amplifiers where two amplifiers can form H-bridge A and the other two types of amplifiers can form H-bridge B. Here, One H Bridge is used to switch the polarity to control the motor direction whereas pair of H bridges are used for controlling a bipolar stepper motor

Each bridge in this IC includes two current sense pins like CSA & CSB and enable pins like ENA & ENB. Here, current sense pins are connected to the ground terminal but we can also include a low-value resistor where its voltage reading is relative to the current. Similarly, enable pins can also be used to make all the outputs active simultaneously. All the enable and input pins in this IC work with 5V TTL logic to make the connection simple with different types of microcontrollers

2.7.2 PIN CONFIGUATION

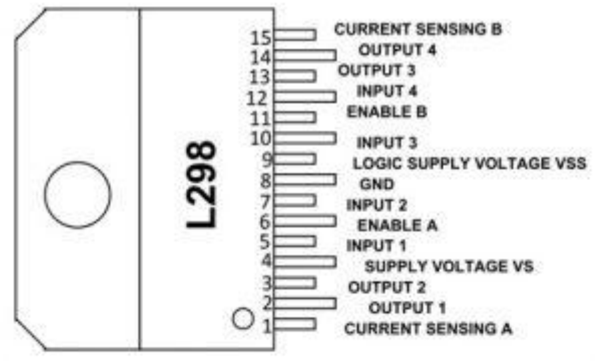


Fig 2.9 Pin Configuation of L298 motor Driver IC

- **Pin1 (Current Sensing A):** This pin is used to control the flow of the current of the load.
- **Pin2 & 3(Output 1 & 2):** These two pins are output pins of the H- Bridge A where the current supplies throughout the load which is monitored at pin-1
- **Pin4 (VS):** This is a voltage supply pin, connected to +5V.
- **Pin5 & 7 (Inputs):** Control Inputs of the Bridge A & compatible with TTL
- **Pin6 (Enable A):** This is an enable input with TTL compatible
- **Pin8 (GND):** This is a GND pin

- **Pin9 (Logic Voltage Supply):** This pin provides voltage supply for the logic blocks.
- **Pin10 & 12 (Inputs3 & 4):** These are control inputs of bridge-B and compatible with TTL
- **Pin11 (Enable B):** This is an enable Input with TTL compatible
- **Pins 13 & 14 (Output 3 & 4):** These are output pins of the H- Bridge B where the flow of current throughout the load is monitored at pin15.

2.7.3 Specifications of L298 Motor Driver IC

- The operating voltage supply is up to 46 v.
- Total DC is up to 4A.
- The saturation voltage is less.
- Protection from over temperature.
- Power dissipation is 25w.
- Operating voltage ranges from +5 to +46v
- The maximum voltage supply voltage is 50v.
- Maximum input & enable voltage is +7v.
- TTL controlled inputs.
- Storage temperature ranges from -40°c – 150°c.
- Operating temperature ranges from -23°c to – 130°c.
- The maximum allowed the current flow to draw through every output is 3

2.7.4 Applications

- L298N motor driver IC is used in different fields like robotics, embedded, etc. L298 motor driver is applicable where H- BRIDGE is used.
- This motor driver is used in high power-based applications.
- This IC is used where current control & PWM operable IC is required.
- This IC is used where the control unit provides only TTL outputs.

2.8 Servo Motor

Servo motor is a rotatory actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with the servo motors. It is not a specific class of motor, although the term servomotor is often used to refer to a motor suitable for use in a closed loop control system.

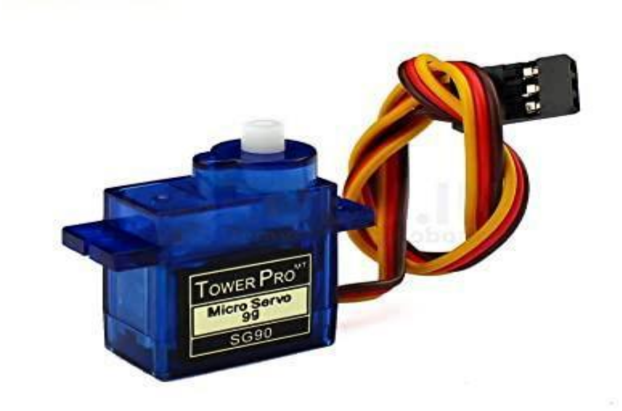
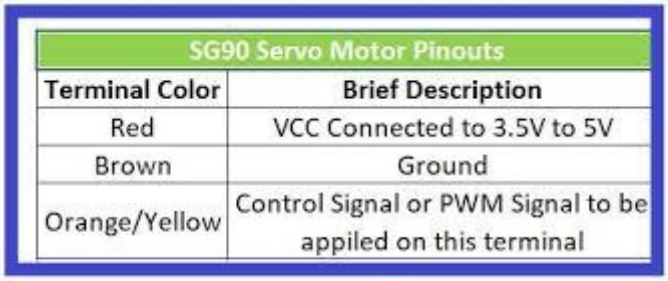


Fig 2.10 Servo Motor



SG90 Servo Motor Pinouts	
Terminal Color	**Brief Description**
Red	VCC Connected to 3.5V to 5V
Brown	Ground
Orange/Yellow	Control Signal or PWM Signal to be appiled on this terminal

Fig 2.11 Servo Motor Pin Description

A servo motor is a motor with a built-in “servomechanism”. The servomechanism uses a sensor to monitor the motor shaft position and a controller to control the motor. It is fed a signal that indicates the position that the shaft should be set to. It then moves the motor into the required position. In the analog servo motors we will be working with the control signal which is a PWM signal whose pulse width determines the angle the motor shaft is to be positioned at. The motor itself is a simple DC motor with a lot of gearing to slow down its speed and to increase its torque. In order to function properly the servo motor needs a sensor that can accurately measure its shaft position. On some industrial and high-end hobby servos this is done using an optical interrupter disc, but in most standard hobby servo motors the sensor is a potentiometer. This works well as these servos typically travel 180 to 270 degrees, well within the range of a potentiometer

CHAPTER 3

SOFTWARE REQUIRED

3.1 BLYNK CLOUD

3.1.1 About Blynk Cloud

Blynk is an IoT platform for iOS or Android smartphones that is used to control Arduino, Raspberry Pi and NodeMCU via the Internet.

3.1.2 Block Diagram of Blynk IOT

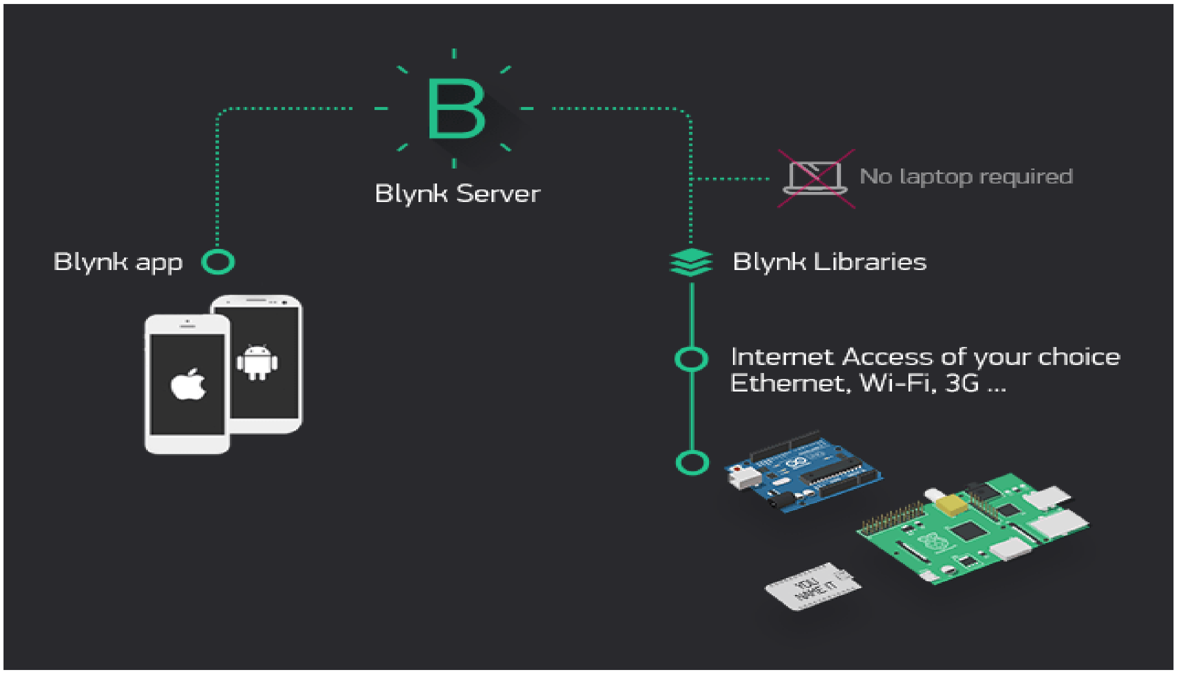


Fig 3.1 Block Diagram of Blynk IOT

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.

3.1.3 COMPONENTS OF BLYNK IOT

There are three major components in blynk IOT

1. Blynk App
2. Blynk Server
3. Blynk Libraries

Blynk App: – It allows you to create amazing interfaces for your projects using various widgets which are provided.

Blynk Server: – It is responsible for all the communications between the smartphone and hardware.

Blynk Libraries: – It enables communication, for all the popular hardware platforms, with the server and process all the incoming and outcoming commands.

TEMPLATE: Template is a set of elements and configurations, used to display all Devices of a particular type. Templates are created by Developers and can be published to be accessible globally.

CREATE PROJECT

1. Go to create project
2. Enter the device name
3. Select the hardware device and connection type
4. Then the project is created

ELEMENTS OF TEMPLATE:

1. Template ID
2. Datastreams
3. Events and notification setup
4. Metadata
5. Web Dashboard
6. Mobile Dashboard

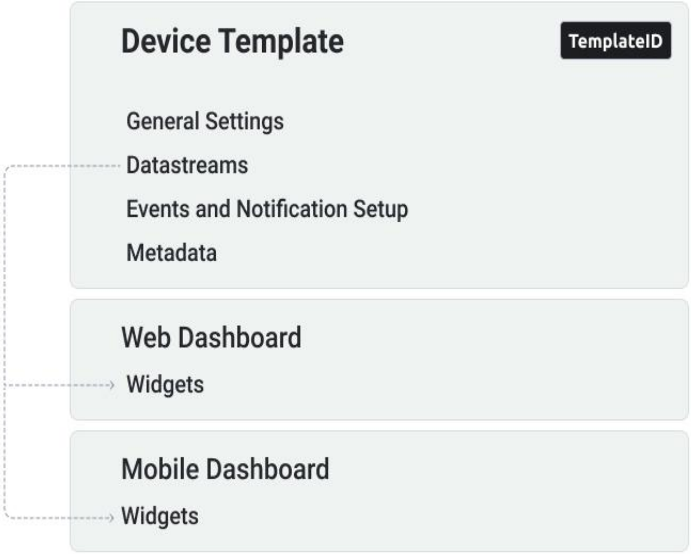


Fig 3.2 Elements of Template

TemplateID

The most important element of the template is TemplateID. This is a unique identifier of every template which should be specified in the code on your device. If you check the code we generated for your device.

Datastreams
Datastreams is a way to structure data that regularly flows in an out from device. Use it for sensor data, any telemetry, or actuators.

There are 3 pins in Datastreams

1. Virtual pin
2. Location
3. Enumerable

Virtual Pin: Virtual Pin is a concept invented to provide exchange of any data between hardware, web and mobile app. Virtual pins allow you to interface with any sensor, any library, any actuator.
it's a very powerful feature to display and send any data from your hardware to the application.

Location:
It can be useful to collect Device coordinates data to view the places it was working at or to represent movements detailed data by building a GPS track in Map Widget.

Enumerable:

This datastream accepts a fixed set of values and transforms them into desired outcomes.

WEB DASHBOARD

a set of UI elements (widgets) to visualize the data from the device accessible for the users in Blynk.Console – a web-based application.

MOBILE DASHBOARD

a set of UI elements (widgets) to visualize the data in Blynk mobile apps for iOS and Android. Mobile apps also contain a template of how device is represented in the list of devices (tiles).

METADATA
Metadata is a set of key:value data attached to every device. Keys are static, and values are related for each device.

Metadata can be edited by users based on their access level.
Default value can be set to each Metadata, be changed during Add Device procedure and edited any time later.
There are 5 info present in metadata

1. D**evice Name** – here you can type default Device name.
2. **Device Owner** – here you can type default Device Owner name or Email.
3. **Location** – here you can type default Location name.
4. **Device Timezone** – select actual or preferred Timezone from the dropdown list

5. **Hotspot Name** – here you can type default Hotspot name.

3.2 INSTALLING RASPBIAN OS IN RASPBERRY PI:

Step 1 : Download the Required Software and Files

You need to download 2 software and 1 OS i.e. Raspbian for this complete process.

1st software: The first software is Win32 Disk Imager.
https://sourceforge.net/projects/win32diskimager/

2nd software: Second software is SD Card Formatter.
https://www.sdcard.org/downloads/formatter_4/

Raspbian OS: This is the Main operating system of the Pi.
https://www.raspberrypi.org/downloads/raspbian/

Extract all files to the desktop.

Step 2 : Get the SD Card and the Card Reader

Get a minimum **8GB class 10** SD card with a card reader. Insert that card into the card reader and plug that to the USB port.

Step 3: Check the Drive in Which the SD Card Is Mounted

Go to my computer or **My PC** and find the drive name where the SD card is**mounted**.

Step 4: Format the SD Card

Open SD Card Formatter and select the drive you noticed in the previous step.

Click on format and **don't alter any other options**.

When formatting is completed, click on OK.

Step 5: Write the OS on the SD Card

Open **win32diskimager**.

Browse the .img file of Raspbian OS that was extracted from the downloaded file.

Click on open and then click on Write. **If any warning pops up then ignore those by clicking OK.**

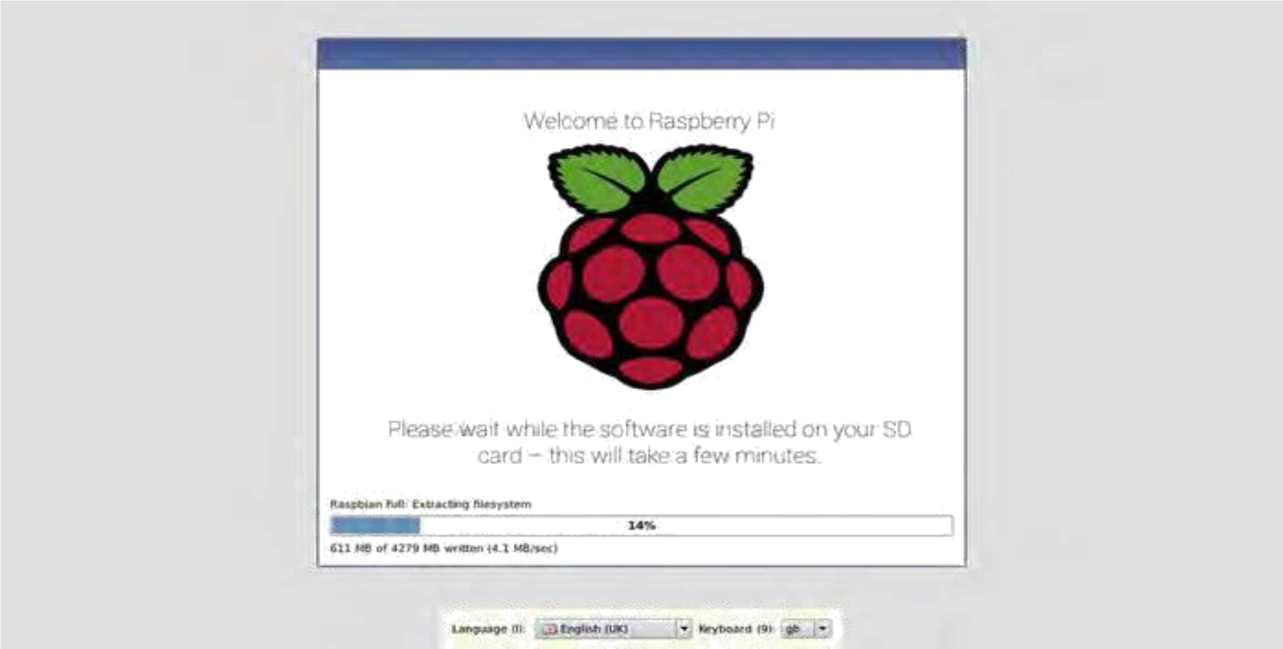
Wait for the write to be completed and it may take some minutes.

Step 6: Eject the SD Card

Step7: Keep the SD card into the SD card connector in the Raspberry Pi. .

3.2.1 Setting up the software:

.



Welcome to Raspberry Pi

Please wait while the software is installed on your SD card – this will take a few minutes.

Raspbian Full: Extracting filesystem

14%

611 MB of 4279 MB written (4.1 MB/sec)

Language (l): English (UK) Keyboard (9): gb

Fig 3.3 Installing Raspbian OS

After installing Finally, you’ll briefly see a window with the Raspberry Pi logo on it before the Raspbian desktop and setup wizard appears, as. Your operating system is now fully installed and ready to be configured,

Now your OS is installed on your Raspberry Pi.

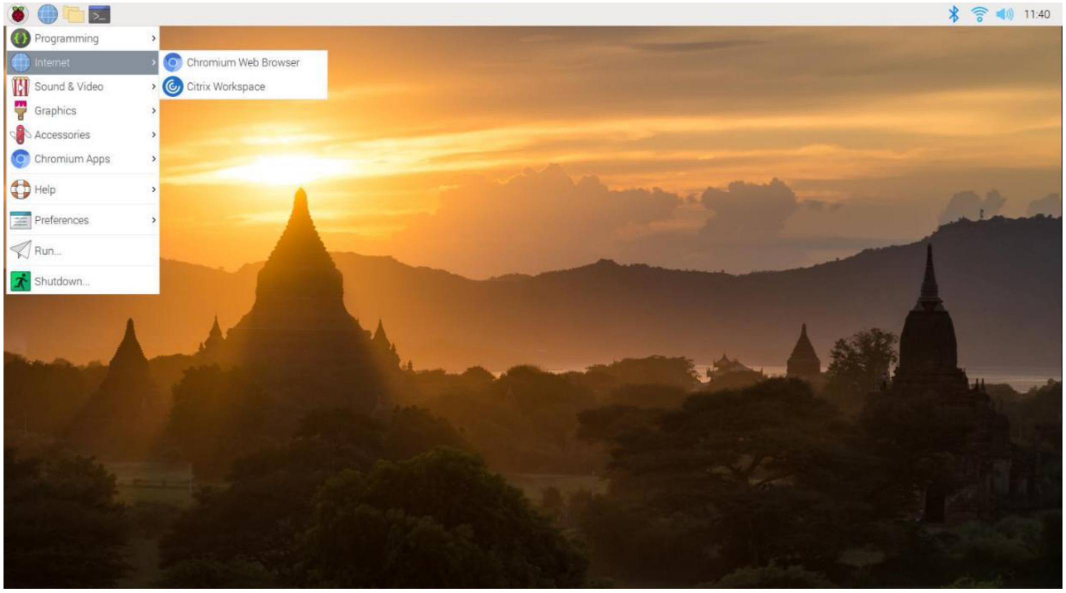


Fig 3.7 Raspbian Desktop

3.3 Arduino sketch for Node_MCU:

3.3.1 About Arduino IDE:

Arduino IDE is software for progaming Arduino Board / Node_MCU. This software is used as a **text editor** to create, open, edit, and validate Code for Arduino. The Code or Program in Arduino is called "**sketch**".

3.3.2 INSTALLING ARDUINO IDE IN WINDOWS 10:

Step 1: Download File Arduino IDE

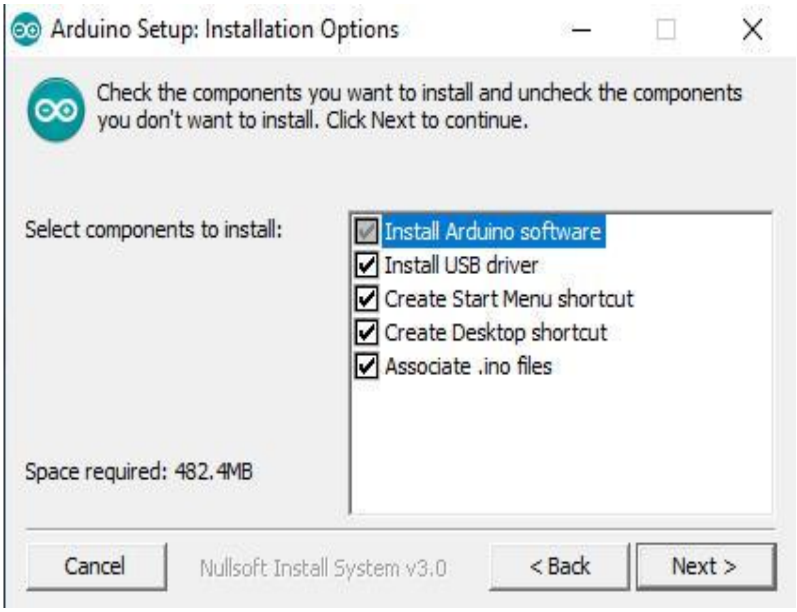
Click the link below to download the Software :https://www.arduino.cc/en/software

On that page, there are 3 download options for Windows.

1. **Windows Installer** : The software will be installed in Windows operating system and required admin access.

2. **Windows Zip file** : To make a portable installation.
3. **Windows App** : for Windows 8.1 or 10.

Step 2: Installation Option



Arduino Setup: Installation Options

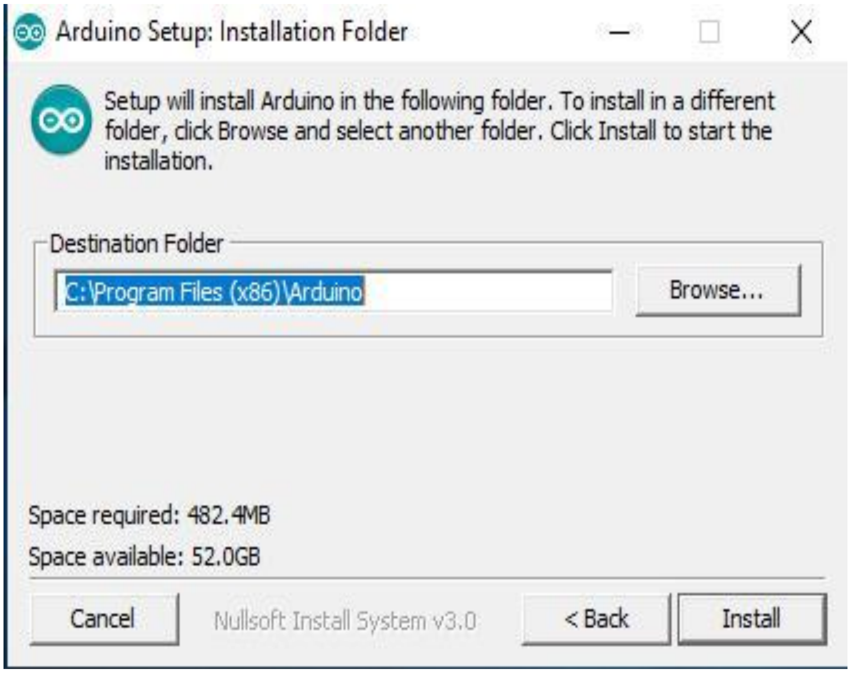
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Install Arduino software
☑ Install USB driver
☑ Create Start Menu shortcut
☑ Create Desktop shortcut
☑ Associate .ino files

Space required: 482.4MB

Step 3: Installation Folder



Arduino Setup: Installation Folder

Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino

Space required: 482.4MB

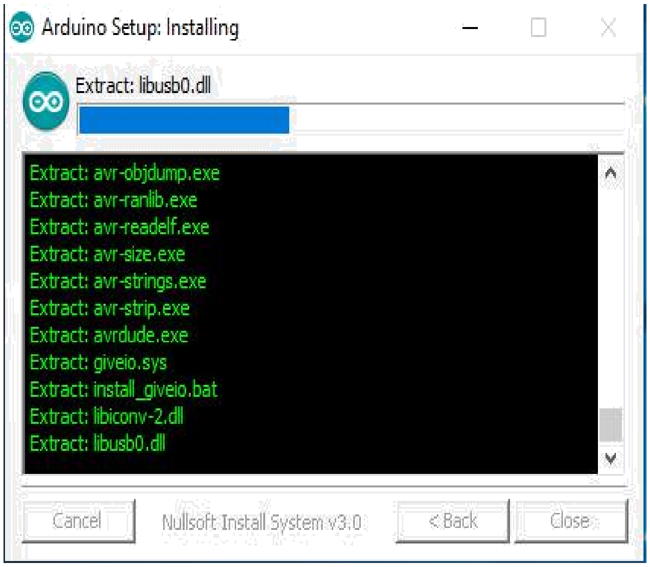
Space available: 52.0GB

1. Arduino will automatically be installed in "**C:\Program Files (x86)\Arduino**".

2. If you want to change the folder, click "*Browse*" and select the desired folder.

3. Click install to start the installation.

Step 4: Installing Process

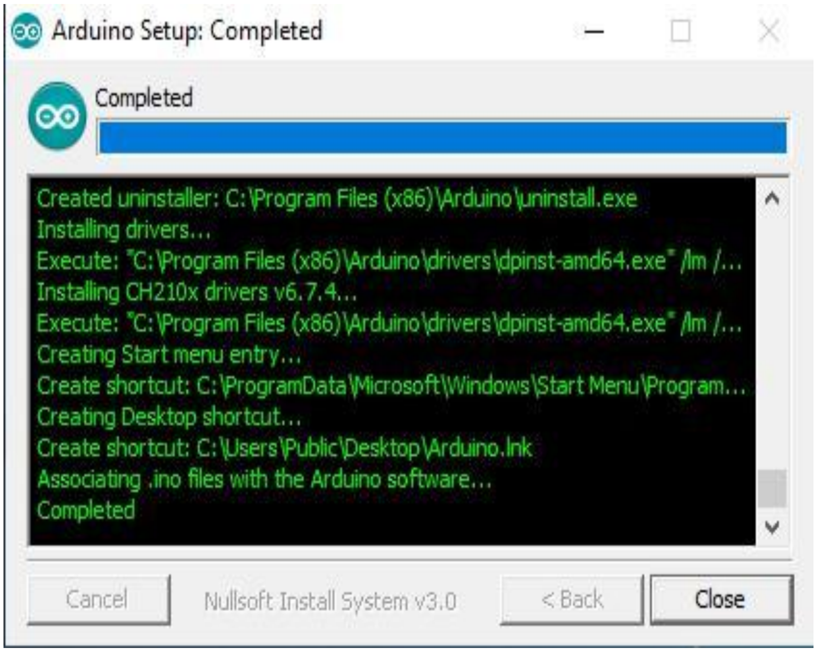


Extract: libusb0.dll

```
Extract: avr-objdump.exe
Extract: avr-ranlib.exe
Extract: avr-readelf.exe
Extract: avr-size.exe
Extract: avr-strings.exe
Extract: avr-strip.exe
Extract: avrdude.exe
Extract: giveio.sys
Extract: install_giveio.bat
Extract: libiconv-2.dll
Extract: libusb0.dll
```

Fig 3.4 Arduino Installation process window

Step 5: Installation Complete

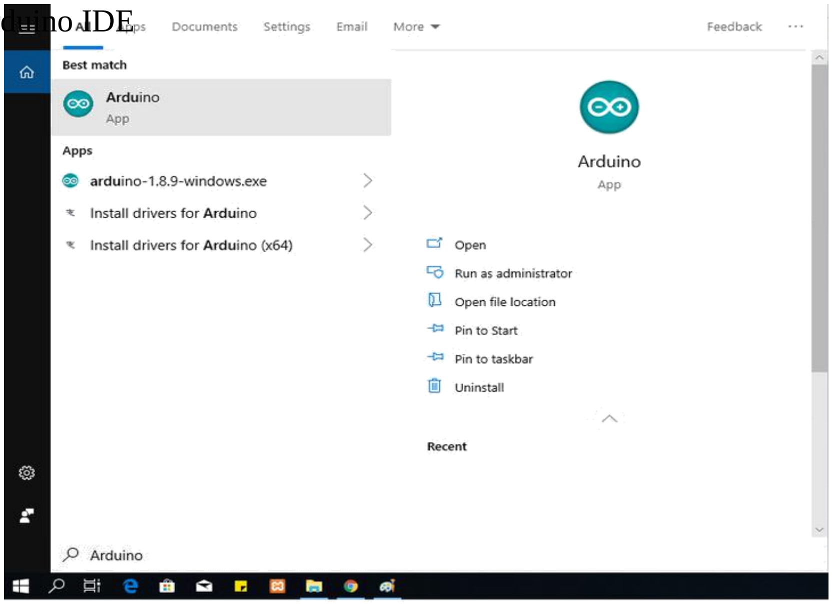


```
Created uninstaller: C:\Program Files (x86)\Arduino\uninstall.exe
Installing drivers...
Execute: "C:\Program Files (x86)\Arduino\drivers\dpinst-amd64.exe" /lm /...
Installing CH210x drivers v6.7.4...
Execute: "C:\Program Files (x86)\Arduino\drivers\dpinst-amd64.exe" /lm /...
Creating Start menu entry...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Program...
Creating Desktop shortcut...
Create shortcut: C:\Users\Public\Desktop\Arduino.lnk
Associating .ino files with the Arduino software...
Completed
```

Fig 3.5 Arduino Installation complete window

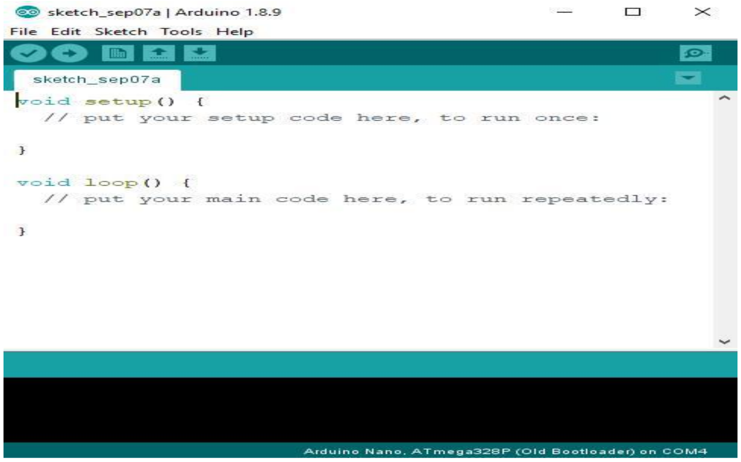
If there is written "*complete*", it means that the installation process is complete. click "*Close*".

Step 6: Open Arduino IDE



After the installation process is complete, there will be an Arduino icon on the Desktop. Or check on the search icon and write "arduino". If you have found the arduino icon, run the application.

Step 7: Display Arduino IDE



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

}

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

}
```

Arduino Nano, ATmega328P (Old Bootloader) on COM4

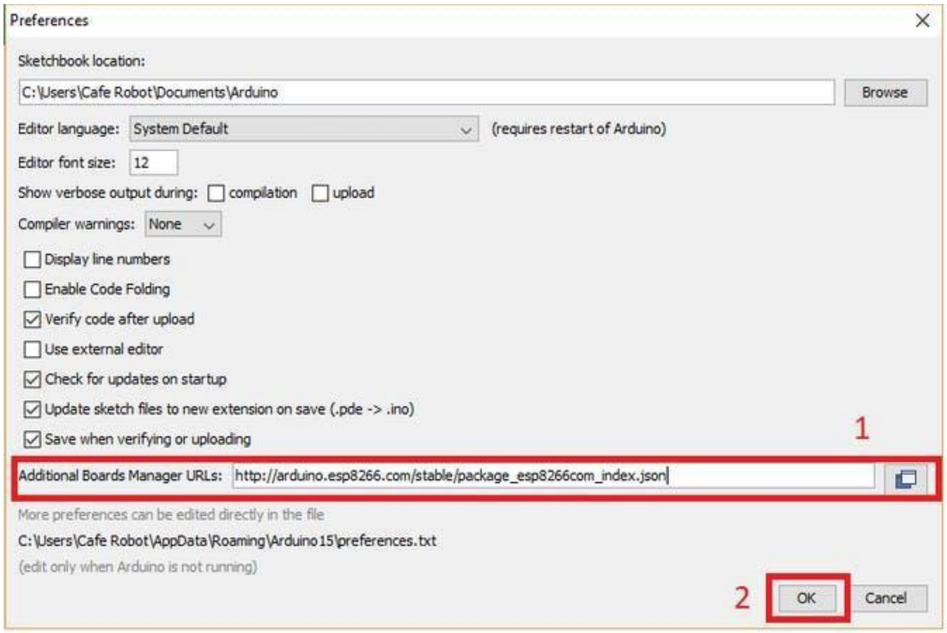
This is a display of the Arduino IDE Software.

3.3.3 How to program NodeMCU using Arduino IDE

In order to use Arduino IDE to program the NodeMCU, you have to introduce it to the software at first.

To do this copy the following code and follow the steps below:
http://arduino.esp8266.com/stable/package_esp8266com_index.json

Step1. Choose Preferences in the File menu and enter the copied code in Additional Board Manager URLs part. Then press OK.



Preferences

Sketchbook location:
C:\Users\Cafe Robot\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
□ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json

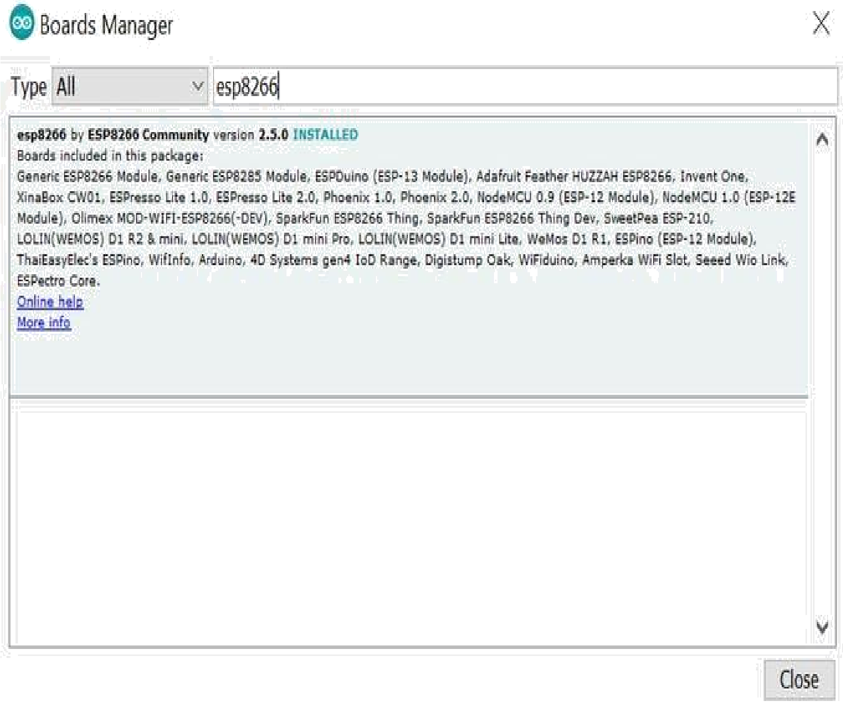
More preferences can be edited directly in the file
C:\Users\Cafe Robot\AppData\Roaming\Arduino15\preferences.txt
(edit only when Arduino is not running)

Step2:

1. Search the word ESP8266 in Boards>boards manager from Tools menu.

2. Then install ESP8266 boards.

3. After complete installation, you will see the INSTALLED label on ESP8266 boards.



esp8266 by **ESP8266 Community** version **2.5.0** INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

After these two steps, you can see ESP8266 based boards such as NodeMCU in your Arduino IDE boards list, and you can choose your desired board to upload the code. In order to use digital pins, you should select GPIO numbers

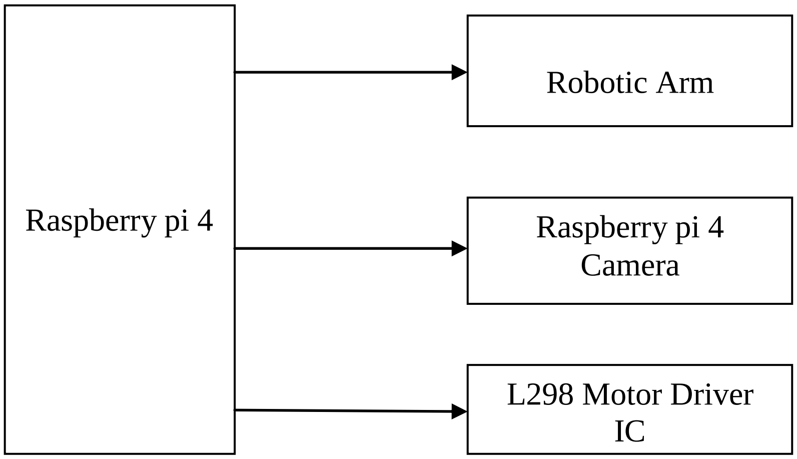
3.4 VNC Viewer

VNC Viewer turns your phone into a remote desktop, giving you instant access to your Mac, Windows and Linux computers from anywhere in the world. You can view your computer's desktop remotely, and control its mouse and keyboard as though you were sitting down in front of it. Use VNC Server to look up the private (internal) IP address of the computer. Download VNC Viewer to the device you want to control from. Enter the private IP address in VNC Viewer to establish a direct connection. Enter the user name and password you typically use to log on to the VNC Server computer

CHAPTER 4

METHODOLOGY

4.1 Block Diagram of Robotic Arm



4.1.2Working Principle

The working of the robatic arm , the chassis and raspberry camera respectively in the form of flow chart.Intially format using SD card formatter and install Raspbian buster software into the new SD card using eject flasher and flash it.Insert SD card into the holder in rapberry pi and power it using universal power cable and connect it to WI-FI router using etherent cable.now HDMI to HDMI connector is used to connect raspberry pi to HDMI display in desktop using the VNC server.Install VNC viewer app in the laptop to navigate your raspberry pi files with ease.The VMC viewer comes in the form of a prompt window that enables the inputting of the unique IP address,username,and password for your raspberry pi.Click ok and the VNC window to access the graphicaluser interface of your raspberryd pi.That’s just about it the raspberry pi has been successfully connected to your lapto via HDMI.we can now make use of a laptop to view the files on your raspberry pi successfully.

The Raspberry pi 4 is connected with Robotic Arm,L298 Motor Driver IC, camera.

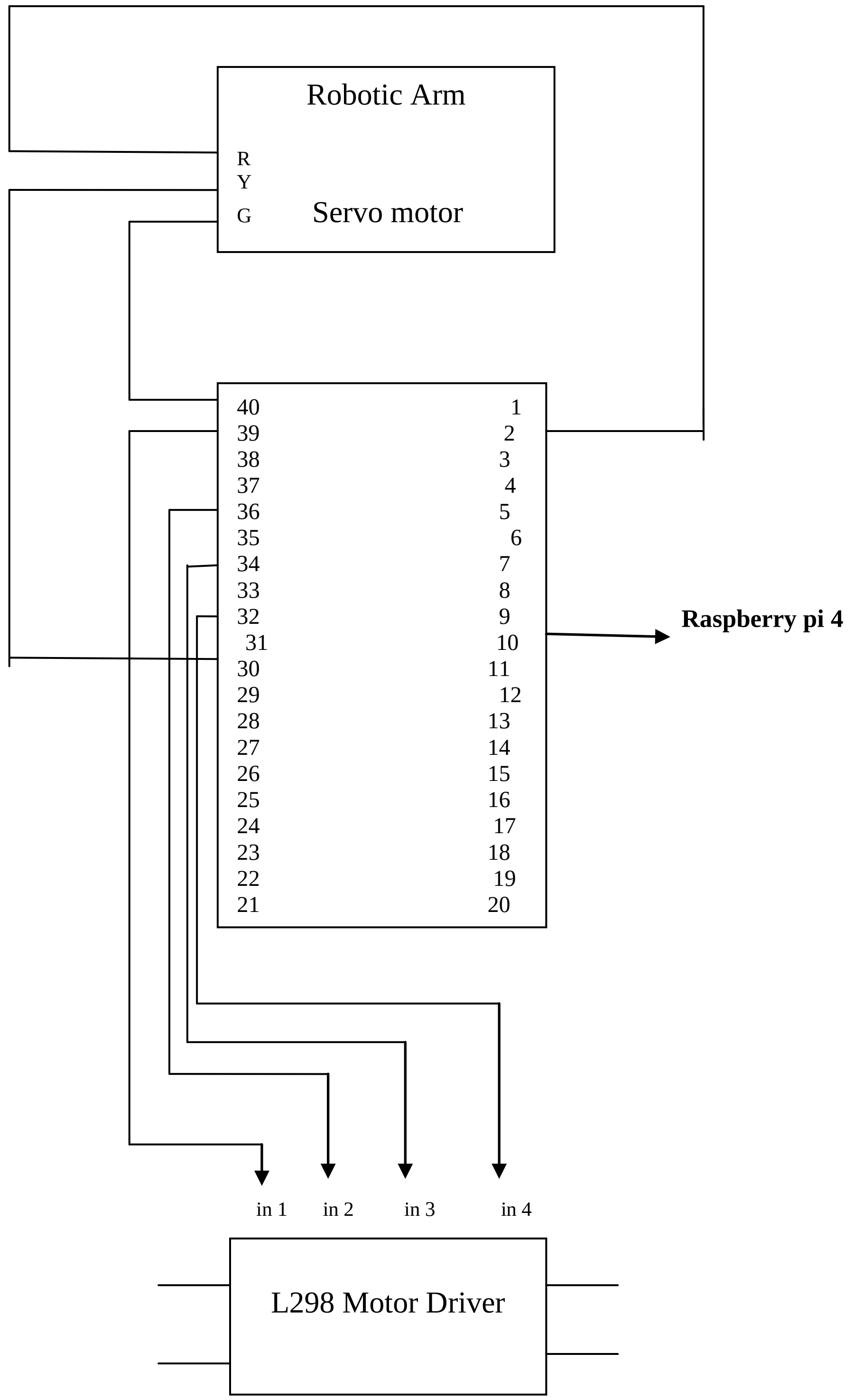
The Robotic Arm is controlled by Raspberry pi 4 with the help of IOT. To move the robotic arm claws we should pass the instructions to Raspberry pi 4 using Blynk App with help of IOT.

We use L298 Motor Driver IC for movement of robotic arm from one place to another place

The signals from L298 Motor Driver IC will go to the motors of wheels and this help in the movement of robotic chasis to move forward ,backward left and right as per our instructions.

The Raspberry pi 4 camera is intefaced with Raspberry pi 4 with help of this camera we can see the visuals of the old age persons in our mobile phone using Blynk App from anywhere.

4.1.3Circuit Diagram of Robotic Arm



4.1.4 Prototype of Roboti Arm

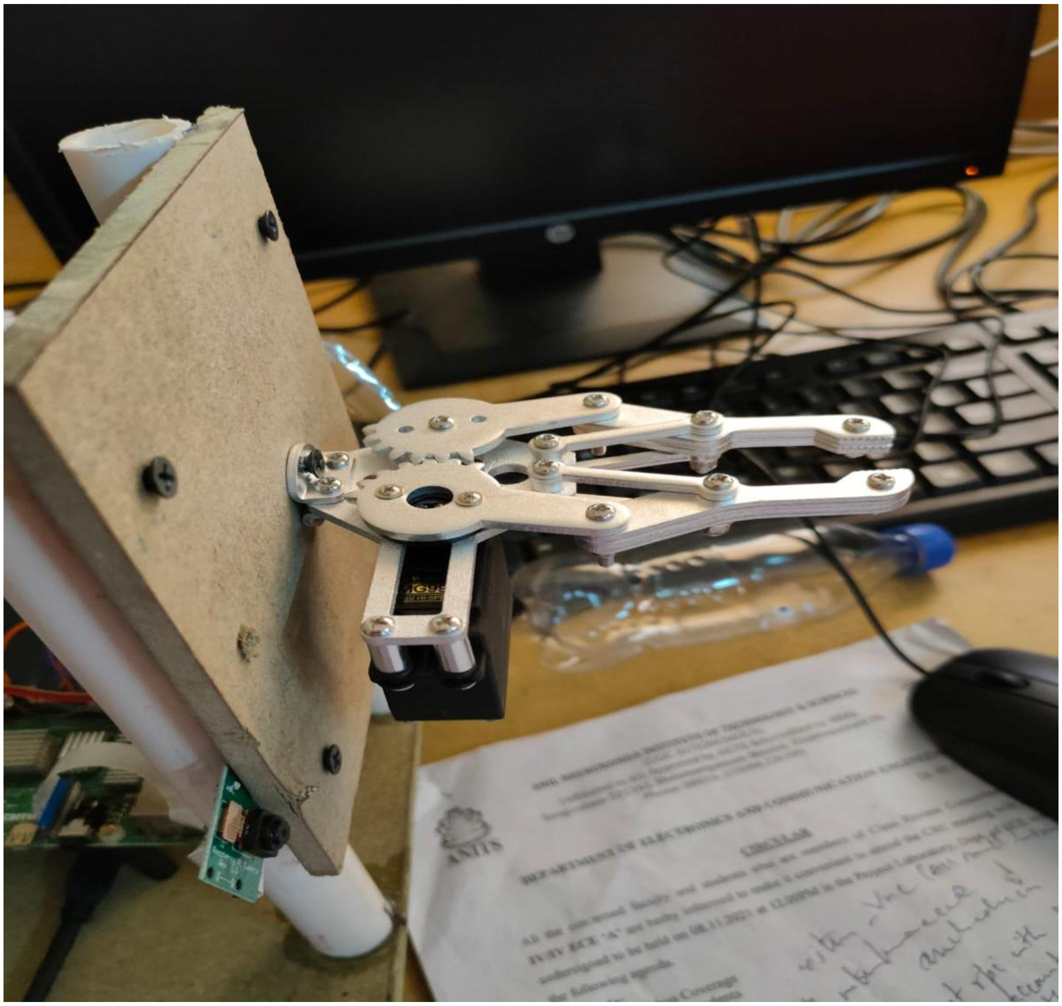


Fig 4.1Working Module of Robotic Arm

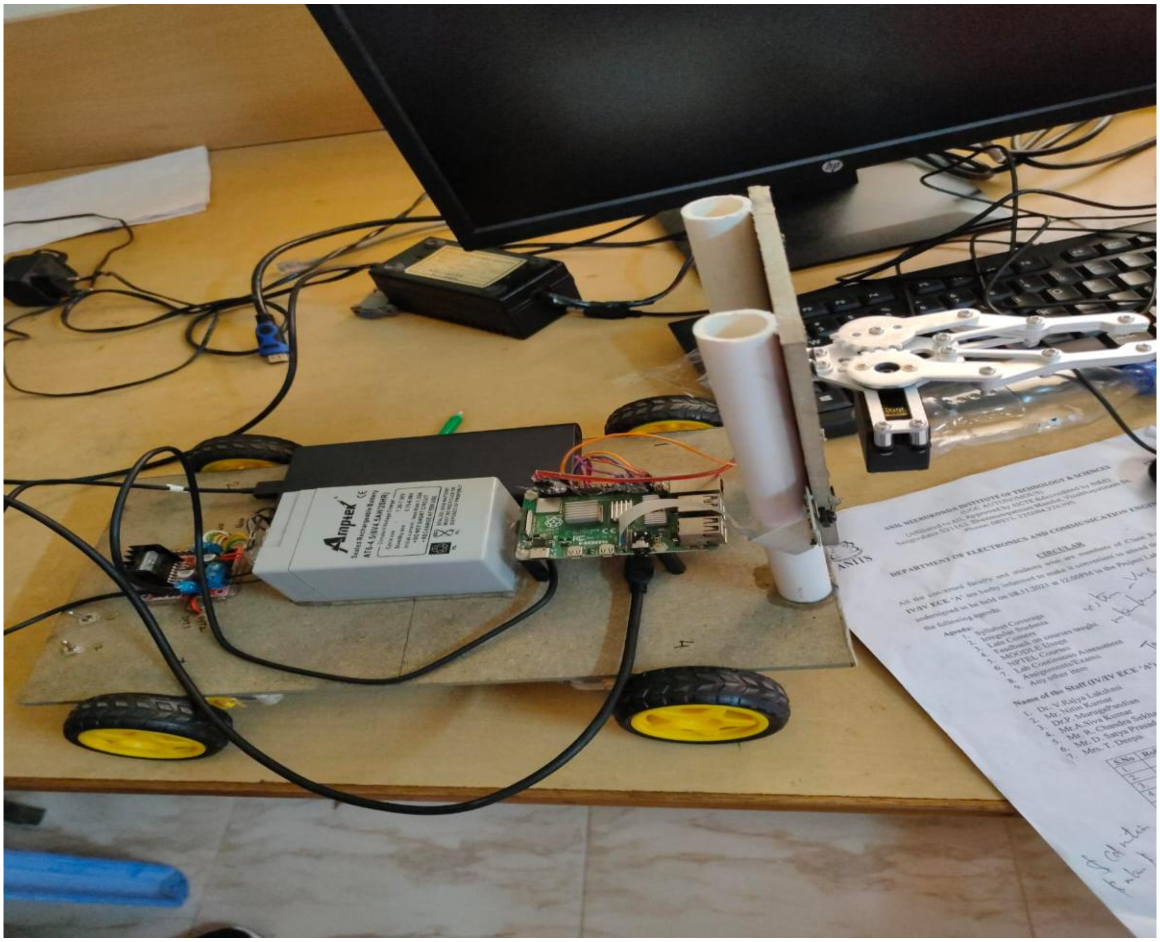
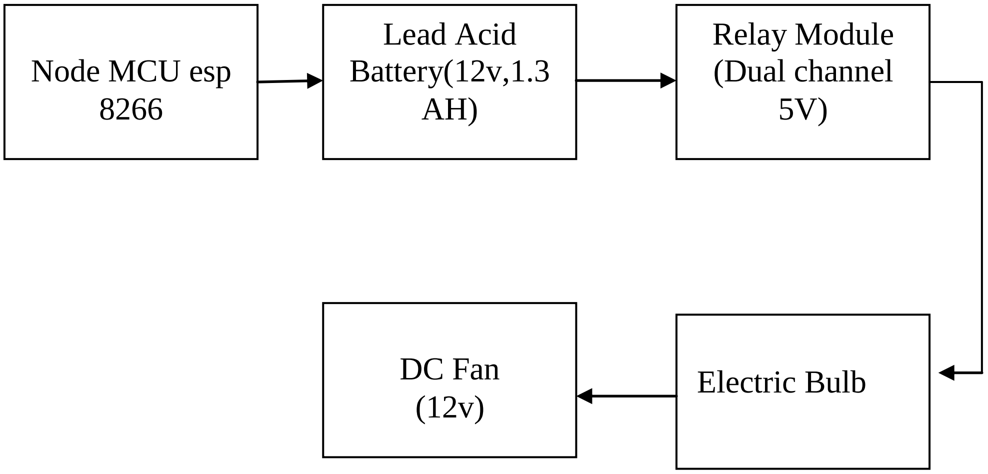
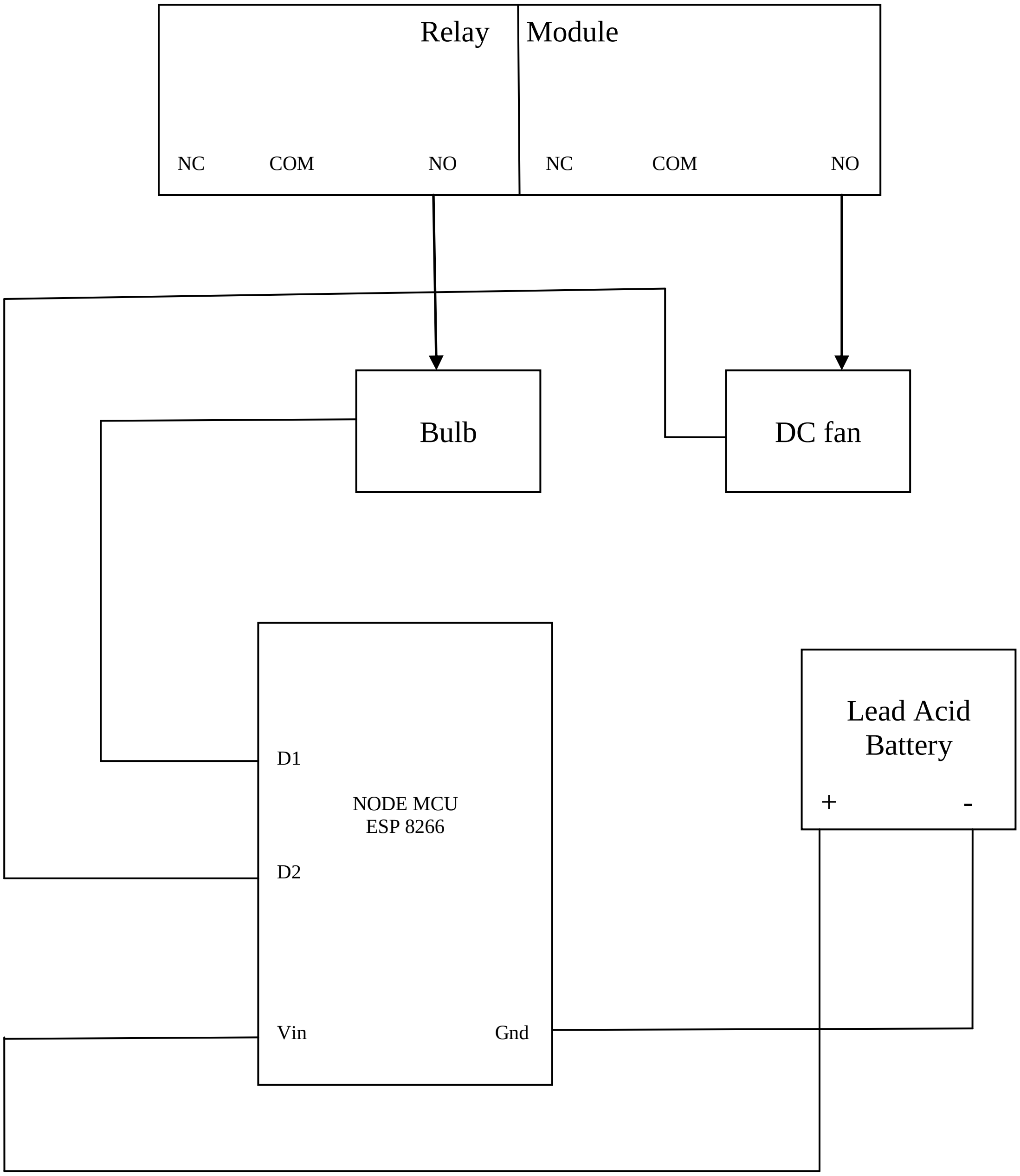


Fig 4.2 Controlling of Robotic Arm

4.2 Block Diagram of Smart Home Automation System



4.2.1 Circuit Diagram



4.2.2 Working of Circuit

This Circuit belongs to Smart Home Automation System. The Circuit consists of Node MCU esp 8266, Relay Module(dual Channel,5V), Lead Acid Battery, Electric Bulb, DC fan. The Relay Module consists of six pins i.e NO,NC,COM(Channel1) NO,NC,COM(Channel2). Here we connect channel1 of NO with the bulb and the channel 2 of NO with dc fan. The pins we used in Node MCU are D1,D2,Vcc,Gnd. The bulb is connected to D1 pin of Node MCU and the DC fan is connected to D2 pin of Node MCU. The Lead Acid Battery has Two Terminals i.e positive and Negative. The positive terminal is connected with Vcc of Node MCU and Negative terminal is connected with Gnd pin of Node MCU. We can control this model by using Blynk App with the help of IOT. Whenever a person wants to control fan and bulb. He should open his blynk application in mobile phone. with help of this Blynk App signal passes through the node MCU from the Cloud. According to this signal the supply from lead acid battery will go to bulb and the bulb will glow. In this Way we can control bulb and fan etc.

4.2.3 Implementation of Smart Home Automation System

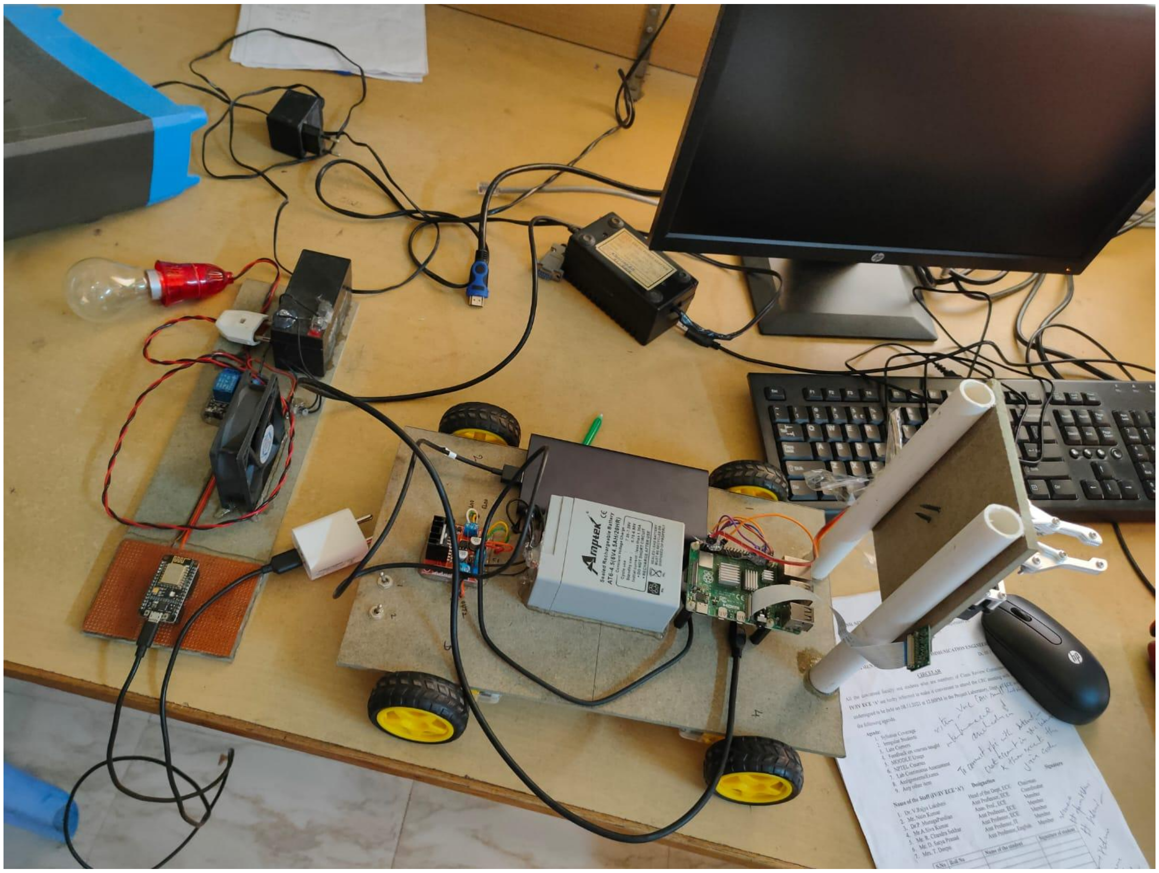


Fig 4.3 Sample of Home Automation System

CHAPTER 5

CONCLUSION

Although research on robotic arm has been conducted in the past, the lack of technology applied has given the chance to conduct this research by implementing IoT for the robotic arm. Android mobile device is utilized as the platform to remotely controlled this robotic arm to perform the pick and place operations. Arduino Yun is used as the brain of the system, which will process the control signal and output modulation signal to the servo motor.

Future research needs to examine the effects of other types of sensor and actuator. Radio control servo motor with greater angle of rotation can be used to improve the reachability of the robotic arm. Moreover, more advance motor such as stepper motor can be used to offer more accurate and precise movement. This future research could help in the review of available robotic arm in the industries and make recommendations for future editions. Finally, future research and development in this research is greatly encourage to bring benefits and advancement in industrial and smart home automation system which we can operate from our phone in blynk app with help of IoT

CHAPTER 6

REFERENCES

- REFERENCESPinheiro, P.R., Pinheiro, P.G.C.D., HolandaFilho, R., Barrozo, J.P., Rodrigues, J.J., Pinheiro, L.I. and Pereira, M.L., 2020. Integration of the Mobile Robot and Internet of Things to Monitor Older People. *IEEE Access*, *8*, pp.138922-138933.
- Ibrahim, K., 2019, December. Optimizing Breadth-First Search at Scale Using Hardware-Accelerated Space Consistency. In *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)* (pp. 23-33). IEEE.
- Chen, Y.Z., Shen, S.F., Chen, T. and Yang, R., 2014. Path optimization study for vehicles evacuation based on Dijkstra algorithm. *Procedia Engineering*, *71*, pp.159-165.