**ABSTRACT**

Presently Gestures are used to control domestic appliances in a modern infrastructure. This project focusses on the current use of gestures in domestic appliances and possible usage for various other domains. Various kinds of appliances are used in households, industries and offices. These devices are mostly controlled by human hand with manual switches. A perfect example is the television control by hand gesture. This idea made it possible to switch channel, change the volume with the use of hand gestures. Present technologies available to recognize gestures in free air which uses common methods include cameras, depth sensors or capacitive systems. This work is focused on the study of electric field (E-field) for advanced proximity sensing which is distorted through hand movements. While compared to the other systems this technology can be employed unobtrusively, work through various materials and does not have a high computational burden also. It allows realization of new user interface applications by detection, tracking and classification of the user’s hand or finger motion in free space.

**CHAPTER: 1**

**INTRODUCTION**

**1.1OBJECTIVE OF THE PROJECT**

Due to advancement in technology, there is an improvement in living standard, by automating even simple task to complex tasks. User satisfaction and comfort is the main criteria for home automation. Most of the electrical appliances in every modern household are automated providing security, improving the quality of life and providing more comfort. Home automation is popular as the home appliances are controlled using automation technology, remote control and internet.

Home automation is also controlled using gesture, in which the person must be present to control the appliance. A predefined gesture is used to turn on/off the device. Gesture is defined as a motion of limbs or any other body part which are made to emphasize speech. It can also be defined as an act or a remark made as a sign of attitude. A gesture is scientifically categorized into two distinctive categories: dynamic and static. A waving hand means goodbye, is an example of dynamic gesture and the stop sign is an example of static gesture. It is necessary to explain all the static and dynamic gestures over a period of time in order to understand the message. Gesture recognition is interpretation of human motion by the computing device. Hand gesture can be detected by the controller that contains accelerometers to sense tilting and acceleration of movement.

**1.2 PROBLEM STATEMENT**

Remotely controlling system in home appliances is not that useful for elderly and disables people. This is because they might be illiterate which they don’t understand about the words that label in the remote control. Besides, if someone having eyes sight problem, they not able to read the small label that stick in the remote control. Thus, using voice controlling system will be the best way to solve those people and it able to make them control the home appliance more easily. Due to the physical device able to place in different height or position, the wireless voice controlling system is being implemented.

**1. 3 PROPOSED METHOD**

MEMS accelerometer based home appliances controlled system is designed for visually challenged and partially paralyzed persons. The system Consist of accelerometer, microcontroller. In this system the accelerometer senses different hand gestures and particular signal is transmitted to receiver section through RF transmitter. RF receiver module receives the transmitted signal compares it with the previously stored gestures and when hand gestures are matched with each other, then the home appliances are controlled.

**CHAPTER: 2**

**LITERATURE SURVEY**

In recent years, several remote hand-gesture control systems for home-media systems have become commercially available. Such systems aim to augment the living-room media experience and enhance user enjoyment. In this regard, scenarios that call for user identification include interface customization (i.e., facilitating a personalized gesture vocabulary), content adaptation, and parental control. Traditionally, TV-remote content adaptation and parental control are facilitated using numerical passwords.

Hand gesture recognition is implemented for traffic light control system (TLC). Thresholding Algorithm is used for recognition purpose. The accelerometer sensors are used to sense the gesture of hand using ARM based control unit. To indicate the traffic lights, LED's are used. ZigBee, SD card and speaker are used for transceiver, storage purpose and important announcement respectively.

Kishor P Jadhav, Santosh G Bari proposed system for Virtual Switch Control to remove the need to look into the hand held remote and to search for a specific key for specific function. Firstly, camera captures the image of user`s hand gesture & then sends it to PC for further image processing. Images are captured by using camera and are processed using MATLAB. The circuit consists of AVR microcontroller, relay driver IC, relay, camera, PC, USB to TTL converter and load. But it is costly and not user friendly.

A. Hayder, J. Dargham, A. Chekima, J. kela, and P. Korpiaa proposed Soapbox method of Sensing, Operating and Activating Peripheral Box, that has low power consumption. It is equipped with 3-axis accelerometer, an illumination sensor, an electronic compass and an optical proximity sensor. For communication purpose, it is wireless with RF technology and code is developed in C. In glove based device controlling approach, the exact location and orientation of the finger, coordinates of palm and hand configuration are proposed (Dan & Mohod, 2014).

The glove uses flex sensor that bends whenever the hand finger makes particular gesture and it gives variety of resistance. Then the accelerometer measures the motion of the hand, the flex sensor produces analogue output which are converted into digital output using ADC converter (Ghotkar et al., 2012). The converted digital signals are then sent to computer to process further the gestures into more meaningful information. Even though the glove based system is expensive it has advantages over the other systems because of its high accuracy and fast speed reaction.

**CHAPTER: 3**

**HARDWARE ASPECTS**

**Block Diagram**

ESP 32

POWER SUPPLY

LCD

LIGHT

RELAY1

MEMS

FAN

RELAY2

**2.1. POWER SUPPLY**

The input to the circuit is applied from the regulated power supply. The ac. input i.e., 230V from the mains supply is step down by the transformer to 12V and is fed to a rectifier. The output obtained from the rectifier is a pulsating dc voltage. So in order to get a pure dc voltage, the output voltage from the rectifier is fed to a filter to remove any ac components present even after rectification. Now, this voltage is given to a voltage regulator to obtain a pure constant dc voltage.

**230V AC**

**50Hz**

**D.C Output**

**Step down transformer**

# Filter

**Bridge Rectifier**

Regulator

Fig: 2.1 Block diagram of power supply.

**2.1.1 TRANSFORMER**

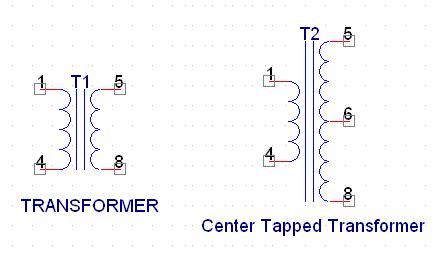


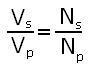
Fig 2.1.1 Transformer

A transformer consists of two coils also called as “WINDINGS” namely PRIMARY & SECONDARY.

They are linked together through inductively coupled electrical conductors also called as CORE. A changing current in the primary causes a change in the Magnetic Field in the core & this in turn induces an alternating voltage in the secondary coil. If load is applied to the secondary then an alternating current will flow through the load. If we consider an ideal condition then all the energy from the primary circuit will be transferred to the secondary circuit through the magnetic field.

So Image

Image



**2.1.2 Rectifier**

A rectifier is a device that converts an AC signal into DC signal. For rectification purpose we use a diode, a diode is a device that allows current to pass only in one direction i.e. when the anode of the diode is positive with respect to the cathode also called as forward biased condition & blocks current in the reversed biased condition.

 Rectifier can be classified as follows:

**1)      Half Wave rectifier**

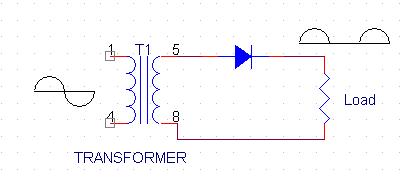
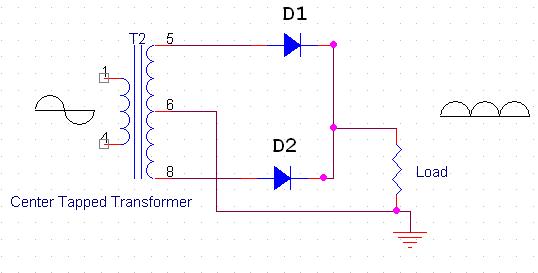


Fig.2.1.2.1 Half Wave Rectifier

This is the simplest type of rectifier as you can see in the diagram a half wave rectifier consists of only one diode. When an AC signal is applied to it during the positive half cycle the diode is forward biased & current flows through it. But during the negative half cycle diode is reverse biased & no current flows through it. Since only one half of the input reaches the output, it is very inefficient to be used in power supplies.

**2)  Full wave rectifier**

Fig.2.1.2.2.Full Wave Rectifier

Half wave rectifier is quite simple but it is very inefficient, for greater efficiency we would like to use both the half cycles of the AC signal. This can be achieved by using a center tapped transformer i.e. we would have to double the size of secondary winding & provide connection to the center. So during the positive half cycle diode D1 conducts & D2 is in reverse biased condition. During the negative half cycle diode D2 conducts & D1 is reverse biased. Thus we get both the half cycles across the load.

One of the disadvantages of Full Wave Rectifier design is the necessity of using a center tapped transformer, thus increasing the size & cost of the circuit. This can be avoided by using the Full Wave Bridge Rectifier.

**2.1.3 FILTER CAPACITOR**

Even though half wave & full wave rectifier give DC output, none of them provides a constant output voltage. For this we require to smoothen the waveform received from the rectifier. This can be done by using a capacitor at the output of the rectifier this capacitor is also called as “FILTER CAPACITOR” or “SMOOTHING CAPACITOR” or “RESERVOIR CAPACITOR”. Even after using this capacitor a small amount of ripple will remain.

We place the Filter Capacitor at the output of the rectifier the capacitor will charge to the peak voltage during each half cycle then will discharge its stored energy slowly through the load while the rectified voltage drops to zero, thus trying to keep the voltage as constant as possible.

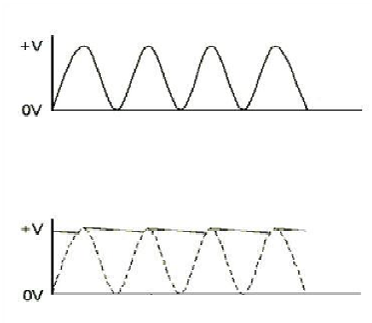


Fig 2.1.3 Waveforms of Filter Capacitor

If we go on increasing the value of the filter capacitor then the Ripple will decrease. But then the costing will increase. The value of the Filter capacitor depends on the current consumed by the circuit, the frequency of the waveform & the accepted ripple.



Where,

Vr= accepted ripple voltage.( should not be more than 10% of  the voltage)

I= current consumed by the circuit in Amperes.

F= frequency of the waveform. A half wave rectifier has only one peak in one cycle so

F=25hz

Whereas a full wave rectifier has Two peaks in one cycle so F=100hz.

**2.1.4 VOLTAGE REGULATOR**

A Voltage regulator is a device which converts varying input voltage into a constant regulated output voltage. Voltage regulator can be of two types

1)      Linear Voltage Regulator

  Also called as Resistive Voltage regulator because they dissipate the excessive voltage resistively as heat.

2)      Switching Regulators.

   They regulate the output voltage by switching the Current ON/OFF very rapidly. Since their output is either ON or OFF it dissipates very low power thus achieving higher efficiency as compared to linear voltage regulators. But they are more complex & generate high noise due to their switching action. For low level of output power switching regulators tend to be costly but for higher output wattage they are much cheaper than linear regulators.

The most commonly available Linear Positive Voltage Regulators are the 78XX series where the XX indicates the output voltage. And 79XX series is for Negative Voltage Regulators.

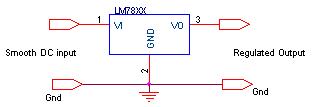


Fig.2.1.4 pin diagram of voltage regulator

After filtering the rectifier output the signal is given to a voltage regulator. The maximum input voltage that can be applied at the input is 35V.Normally there is a 2-3 Volts drop across the regulator so the input voltage should be at least 2-3 Volts higher than the output voltage. If the input voltage gets below the Vmin of the regulator due to the ripple voltage or due to any other reason the voltage regulator will not be able to produce the correct regulated voltage.

#### 2.1.4.1 Circuit diagram

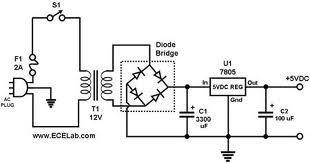


Fig 2.1.4.1 Circuit Diagram of power supply

#### IC 7805

7805 is an integrated three-terminal positive fixed linear voltage regulator. It supports an input voltage of 10 volts to 35 volts and output voltage of 5 volts. It has a current rating of 1 amp although lower current models are available. Its output voltage is fixed at 5.0V. The 7805 also has a built-in current limiter as a safety feature. 7805 is manufactured by many companies, including National Semiconductors and Fairchild Semiconductors.

The 7805 will automatically reduce output current if it gets too hot. The last two digits represent the voltage; for instance, the 7812 is a 12-volt regulator. The 78xx series of regulators is designed to work in complement with the 79xx series of negative voltage regulators in systems that provide both positive and negative regulated voltages, since the 78xx series can't regulate negative voltages in such a system.

The 7805 & 78 is one of the most common and well-known of the 78xx series regulators, as it's small component count and medium-power regulated 5V make it useful for powering TTL devices.

Specifications of IC7805

|  |  |
| --- | --- |
| **SPECIFICATIONS** | **IC 7805** |
| Vout | 5V |
| Vein - Vout Difference | 5V - 20V |
| Operation Ambient Temp | 0 - 125°C |
| Output Imax | 1A |

Table. 2.1.4.2Specifications of IC7805

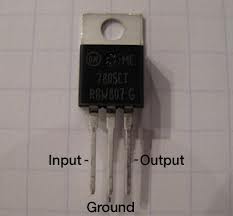
[](http://www.google.co.in/imgres?hl=en&sa=X&tbo=d&biw=1366&bih=625&tbm=isch&tbnid=D3tfeYa-QsyVcM:&imgrefurl=http://interactiondesign.sva.edu/classes/fall10/fundamentalsofphysicalcomputing/labs/lab-electronics/&docid=8q7wEdErZMG2VM&imgurl=http://interactiondesign.sva.edu/classes/physicalcomputing/files/vregulator7805.jpg&w=400&h=372&ei=eRKvUP6sF8-zrAfhjYGABg&zoom=1&iact=hc&vpx=136&vpy=31&dur=16&hovh=216&hovw=233&tx=157&ty=133&sig=104877907207258091705&page=1&tbnh=121&tbnw=125&start=0&ndsp=26&ved=1t:429,r:10,s:0,i:132)

Fig 2.1.4.3 Pin diagram of 7805

* **IC7812**

Here is a [7812 voltage regulator circuit](http://www.zimbio.com/go/U1ZL8PZvXe0/http:/powersupplycircuit.net/7812-voltage-regulator-circuit.html), but this is not a power supply with a 12V output voltage and load current 1A. IC LM7812 only serves as the input voltage of an LM723 regulator IC. So these power supply circuits with a larger load current capability with a variable voltage at the maximum voltage of 6V.

Output voltage range of the 7812 voltage regulator circuit is 2.5V-6V with 6A-8A load current. [Increasing load current](http://www.zimbio.com/go/8oQCDb46z1K/http:/powersupplycircuit.net/increase-regulator-current.html) through the transistor BD139 and TIP142 are sourced from the DC voltage of a transistor.

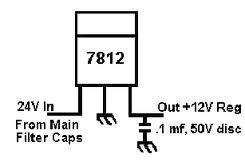
[](http://www.google.co.in/imgres?hl=en&sa=X&tbo=d&biw=1366&bih=625&tbm=isch&tbnid=0TfKdnfY6oiHEM:&imgrefurl=http://www.rangerusa.com/techtips.html&docid=mP3LmNe42Qpc1M&imgurl=http://www.rangerusa.com/photos/7812.gif&w=307&h=203&ei=GROvUKOvNcS8rAfU24CgAQ&zoom=1&iact=hc&vpx=149&vpy=157&dur=5531&hovh=162&hovw=245&tx=120&ty=108&sig=104877907207258091705&page=1&tbnh=116&tbnw=176&start=0&ndsp=22&ved=1t:429,r:1,s:0,i:90)

Fig 2.1.4.4 7812 Pin Connection

**ESP32 Pinout | ESP-WROOM-32 Pinout**

In this tutorial, we will learn about the pinout information of ESP32. The 30-pin ESP32 Development Board will be used to demonstrate the ESP32 Pinout. We will also learn about some important peripherals of ESP32 Microcontroller and their associated pins, what GPIO pins can be used in your project.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-Development-Board.jpg)

Outline

[](https://www.electronicshub.org/esp32-pinout/)

* [Introduction to ESP32 Pin Layout](https://www.electronicshub.org/esp32-pinout/#Introduction_to_ESP32_Pin_Layout)
* [ESP-WROOM-32 Pinout](https://www.electronicshub.org/esp32-pinout/#ESP-WROOM-32_Pinout)
* [ESP32 Pinout](https://www.electronicshub.org/esp32-pinout/#ESP32_Pinout)
* [Important ESP32 Peripherals](https://www.electronicshub.org/esp32-pinout/#Important_ESP32_Peripherals)
  + [GPIO](https://www.electronicshub.org/esp32-pinout/#GPIO)
  + [RTC GPIO](https://www.electronicshub.org/esp32-pinout/#RTC_GPIO)
  + [ADC](https://www.electronicshub.org/esp32-pinout/#ADC)
  + [DAC](https://www.electronicshub.org/esp32-pinout/#DAC)
  + [Capacitive Touch GPIOs](https://www.electronicshub.org/esp32-pinout/#Capacitive_Touch_GPIOs)
  + [SPI](https://www.electronicshub.org/esp32-pinout/#SPI)
  + [I2C](https://www.electronicshub.org/esp32-pinout/#I2C)
  + [PWM](https://www.electronicshub.org/esp32-pinout/#PWM)
* [What GPIO Pins to use?](https://www.electronicshub.org/esp32-pinout/#What_GPIO_Pins_to_use)
  + [GPIOs Connected to SPI Flash IC](https://www.electronicshub.org/esp32-pinout/#GPIOs_Connected_to_SPI_Flash_IC)
  + [Input only GPIO](https://www.electronicshub.org/esp32-pinout/#Input_only_GPIO)
  + [Interrupts](https://www.electronicshub.org/esp32-pinout/#Interrupts)
  + [Boot Strapping Pins](https://www.electronicshub.org/esp32-pinout/#Boot_Strapping_Pins)
  + [Conclusion](https://www.electronicshub.org/esp32-pinout/#Conclusion)

**Why do we need to learn ESP32 Pinout?**

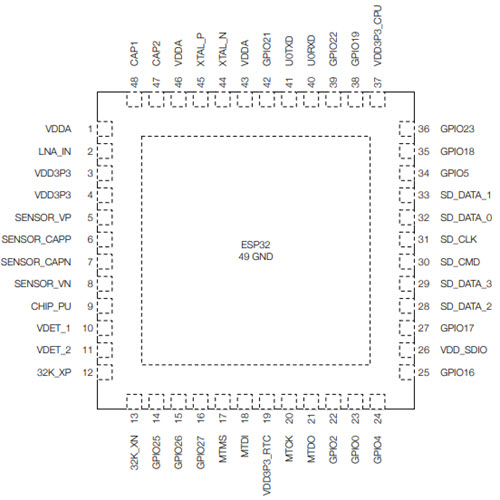
ESP32 based boards come in a variety of shapes and sizes and pinout of each board is different to other. Also, not all pins of the ESP32 Microcontroller SoC will be available on a development board as some pins might be permanently tied to a dedicated function.

One such case is the Flash Memory. We know that all ESP32 boards come with 4 MB of Flash Memory to store the programs. So, some of the GPIO Pins (6 to be specific) are connected to SPI Flash IC and those pins cannot be used as regular GPIO Pins.

Hence, it is important to understand the pinout of popular ESP32 boards so that you will know what pins are available for use in projects.

**Introduction to ESP32 Pin Layout**

The ESP32 Microcontroller IC is available in a 48-pin QFN package. Since it is a QFN (Quad Flat No Leads) package, it is difficult solder the IC on to the PCB, if you are interested in making an ESP32 Board by yourself.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-Pin-Layout.jpg)

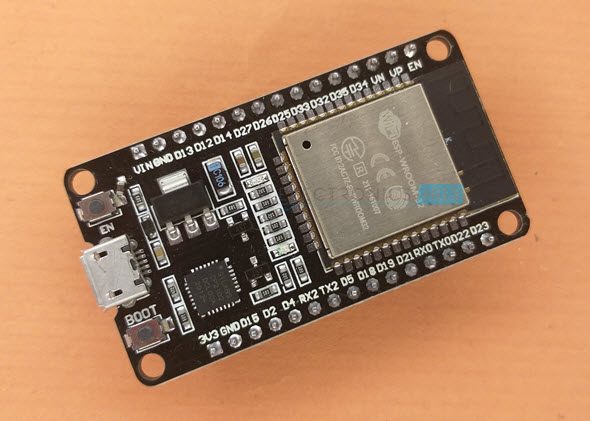
So, what Espressif Systems (the developers of ESP32) did is, they took the ESP32 IC and made a small module board with edge castellations. One popular version of such module board is called ESP-WROOM-32.

In addition to the ESP32 IC, the ESP-WROOM-32 also contain a 4 MB SPI Flash IC, a 40 MHz Crystal Oscillator, PCB Antenna and some discrete passive components to make a working system.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP-WROOM-32.jpg)

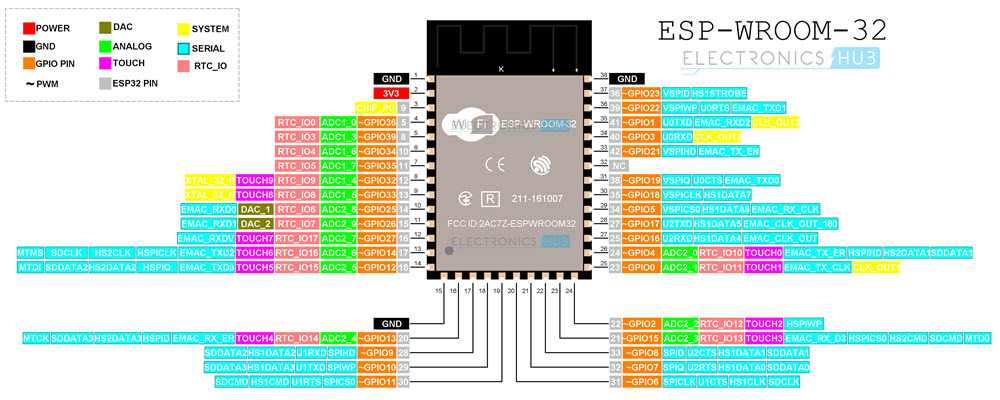
You can use this board in your hardware design (as it takes care of the complex RF section of the PCB) and make a development board or a breakout board or even a commercial product.

This is what third-party module manufacturers do. They take the ESP-WROOM-32 Module, design a break-out board based on this module with user friendly pins, USB Port, RESET and BOOT switches etc.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/Getting-Started-with-ESP32-1.jpg)

**ESP-WROOM-32 Pinout**

Let us start exploring the ESP32 Pinout by taking a look at the ESP WROOM 32 Pinout. The following image show the pinout of a typical ESP-WROOM-32 Module. It consists of 38-pin (14 pins on each long edge and 10 on bottom short edge).

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP-WROOM-32-Pinout.jpg)

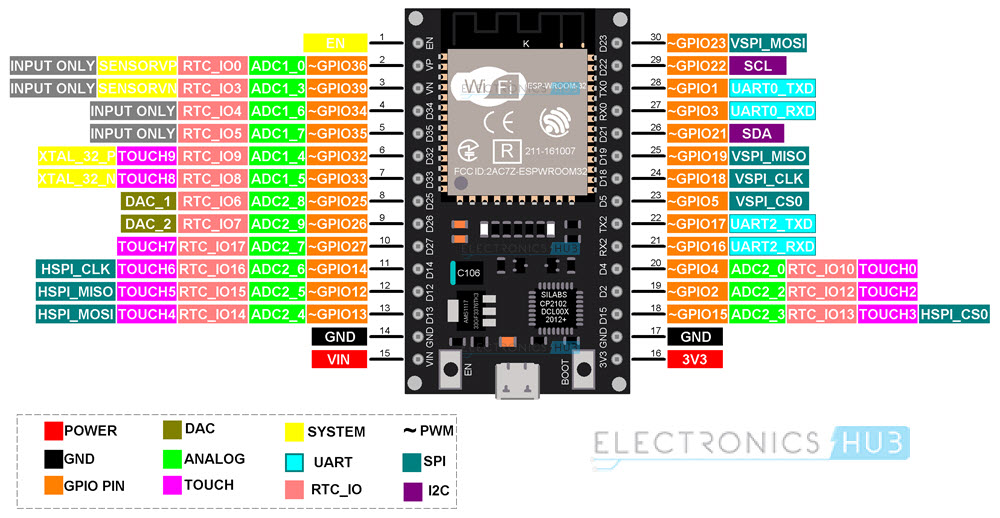
The other short edge is left for PCB Antenna. If you notice the previous ESP-WROOM-32 image, then the Microcontroller and other components are not visible as they placed under the RF Shield.

Use this pinout of ESP-WROOM-32 module as a reference if you are using such module in your hardware design.

**ESP32 Pinout**

One popular ESP32 Development Board available today is the 30-pin version shown in the above image. It consists of ESP-WROOM-32 as the baseboard and additionally few pins and components to easily interact with ESP32.

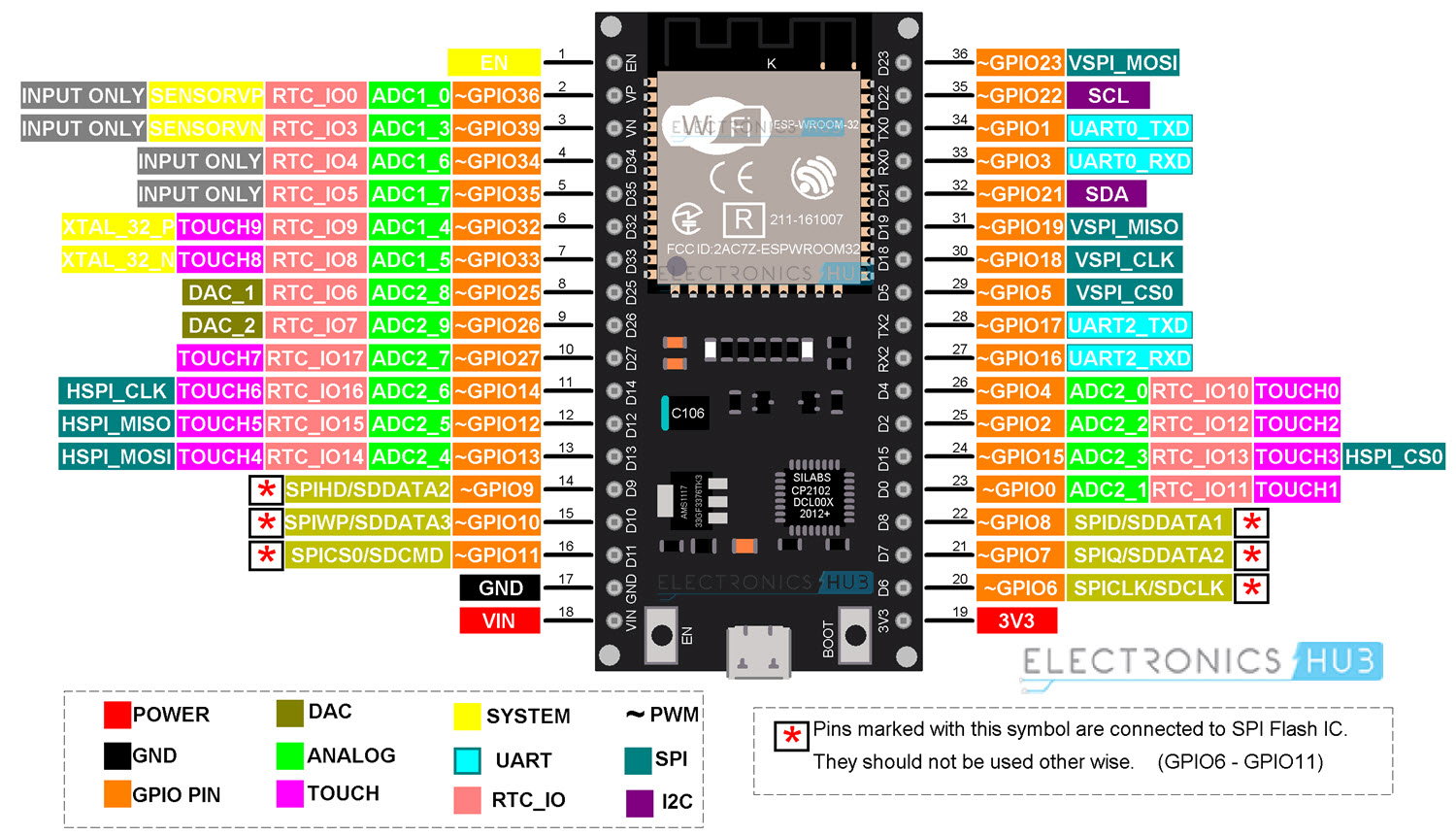
The following image shows the pinout of a 30-pin ESP32 DevKit Development Board.

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-Pinout-1.jpg)

As you can see from the image, each pin has more than one possible functionality and while using a pin for particular task, double check its alternative functions.

Another version of ESP32 DevKit Board comes with 36-pins. This version is not as popular as the 30-pin version. But if you happen to have a 36-pin ESP32 board, the following pinout will be very helpful.

Note that the pinout of both the 30-pin and the 36-pin versions of ESP32 Development Boards are very identical except for some pins at the bottom. In the 36-Pin version, 6 GPIO pins (GPIO6 to GPIO11) are used for SPI Flash IC. So, they shouldn’t be used for other purposes. Finally, you just get one extra pin (GPIO0 – Pin 23).

[](https://www.electronicshub.org/wp-content/uploads/2021/02/ESP32-36-Pin-Pinout.jpg)

**Important ESP32 Peripherals**

Now that we have seen a little bit about ESP32 Pinout. Let us now focus on some of the important peripherals of ESP32 and their associated pins. ESP32 Microcontroller has:

* 34 Programmable GPIOs
* 18 12-bit ADC Channels
* 2 8-bit DAC Channels
* 16 PWM Channels
* 3 UART Interfaces
* 3 SPI Interfaces
* 2 I2C Interfaces
* 2 I2S Interfaces
* 10 Capacitive Touch Sensing GPIOs
* 16 RTC GPIOs

**GPIO**

The most commonly used peripheral is the GPIO. ESP32 has 34 GPIO pins with each pin carrying out more than one function (only one will be active). You can configure a pin as either a GPIO or an ADC or an [UART](https://www.electronicshub.org/basics-uart-communication/) in the program.

ADC and DAC pins are predefined and you have to use the manufacturer specified pins. But other functions like PWM, SPI, UART, I2C etc. can be assigned to any GPIO pin through program.

**RTC GPIO**

ESP32 has 16 RTC GPIOs, which are part of the RTC Low-Power subsystem. These pins can be used to wake ESP32 from deep sleep as external wake-up source.

**ADC**

ESP32 has two 12-bit SAR [Analog to Digital Converter](https://www.electronicshub.org/analog-circuits-and-digital-circuits/) Modules with 8-channels and 10-channels each. So, ADC1 and ADC2 blocks combined together have 18 channels of 12-bit ADC.

With 12-bit resolution, the output Digital values will be in the range of 0 – 4093.

**DAC**

ESP32 Microcontroller has two independent 8-bit Digital to Analog Converter channels to convert digital values to analog voltage signals. The DAC has internal resistor network and uses power supply as input reference voltage.

The following two GPIO Pins are associated with DAC functionalities.

* DAC1 — GPIO25
* DAC2 — GPIO26

**Capacitive Touch GPIOs**

The ESP32 SoC has 10 capacitive-sensing GPIOs, which can detect variations in capacitance on a pin due to touching or approaching the GPIO Pin with a finger or stylus. These Touch GPIOs can be used in implementing capacitive touch pads, without any additional hardware.

**SPI**

The ESP32 Wi-Fi chip features three SPI blocks (SPI, HSPI and VSPI) in both master and slave modes. SPI is used to interface with Flash Memory. So, you have two SPI interfaces.

**I2C**

There are two I2C interfaces in ESP32 with complete flexibility on assigning pins i.e., SCL and SDA pins for both I2C interfaces can be assigned in the program by the user.

If you are using Arduino IDE, then the default I2C pins are:

* SDA – GPIO21
* SCL – GPIO22

**PWM**

The PWM Controller in ESP32 have 16 independent PWM waveform channels with configurable frequency and duty cycle. The PWM waveform can be used to drive motors and LEDs. You can configure the PWM signal frequency, channel, GPIO pin and also the duty cycle.

**What GPIO Pins to use?**

Let us now take a look at a table which specifies the GPIO pins and their input / output capabilities.

|  |  |  |
| --- | --- | --- |
| **GPIO Pin** | **Pin on ESP32** | **Information** |
| 0 | – | Pulled HIGH. Connected to BOOT Button |
| 1 | TX0 | Do not use while TXing |
| 2 | YES | Pulled LOW |
| 3 | RX0 | Do not use while RXing |
| 4 | D4 | Pulled LOW |
| 5 | D5 | Pulled HIGH |
| 6 | – | Connected to SPI Flash IC |
| 7 | – | Connected to SPI Flash IC |
| 8 | – | Connected to SPI Flash IC |
| 9 | – | Connected to SPI Flash IC |
| 10 | – | Connected to SPI Flash IC |
| 11 | – | Connected to SPI Flash IC |
| 12 | D12 | Pulled LOW. Boot fails if pulled HIGH as it sets voltage of internal voltage regulator. |
| 13 | D13 |  |
| 14 | D14 |  |
| 15 | D15 | Pulled HIGH |
| 16 | RX2 | UART2 RX |
| 17 | TX2 | UART2 TX |
| 18 | D18 |  |
| 19 | D19 |  |
| 21 | D21 | I2C SDA |
| 22 | D22 | I2C SCL |
| 23 | D23 |  |
| 25 | D25 |  |
| 26 | D26 |  |
| 27 | D27 |  |
| 32 | D32 |  |
| 33 | D33 |  |
| 34 | D34 | Digital Input Only. No Digital Output. |
| 35 | D35 | Digital Input Only. No Digital Output. |
| 36 | YES | Digital Input Only. No Digital Output. |
| 39 | YES | Digital Input Only. No Digital Output. |

**GPIOs Connected to SPI Flash IC**

If you take a look at the schematic of ESP-WROOM-32 Module, then you will see that GPIO6 to GPIO11 are connected to SPI Flash Memory IC. Even if these GPIO pins are accessible (which are not in 30-pin ESP32 Board), do not use them for any other purpose.

**Input only GPIO**

There are 4 GPIO pins which are capable of acting as Digital Input only pins. They are GPIO34, GPIO35, GPIO36 and GPIO39.

**Interrupts**

All GPIO pins are capable of interrupts.

**Boot Strapping Pins**

ESP32 SoC has 5 boot strapping pins. They are:

* GPIO0 (HIGH during BOOT)
* GPIO2 (LOW during BOOT)
* GPIO5 (HIGH during BOOT)
* GPIO12 (LOW during BOOT)
* GPIO15 (HIGH during BOOT)

These pins are used to put the microcontroller in to flashing mode or bootloader mode.

**MEMS:**

****

The term MEMS stands for micro-electro-mechanical systems. These are a set of devices, and the characterization of these devices can be done by their tiny size & the designing mode. The designing of these sensors can be done with the 1- 100-micrometer [components](https://www.elprocus.com/basic-components-used-electronics-electrical/). These devices can differ from small structures to very difficult electromechanical systems with numerous moving elements beneath the control of incorporated micro-electronics. Usually, these sensors include mechanical micro-actuators, micro-structures, micro-electronics, and micro-sensors in one package. This article discusses what is a MEMS sensor, working principle, advantages and it’s applications

**MEMS Sensor**

MEMS are low-cost, and high accuracy inertial sensors and these are used to serve an extensive range of industrial applications. This sensor uses a chip-based technology namely micro-electro-mechanical-system. These [sensors](https://www.elprocus.com/different-types-of-sensors-used-for-building-projects/) are used to detect as well as measure the external stimulus like pressure, after that it responds to the pressure which is measured pressure with the help of some mechanical actions. The best examples of this mainly include revolving of a motor for compensating the pressure change.

The [MEMS IC fabrication](https://www.elprocus.com/understanding-fabrication-mems/) can be done with silicon, whereby slight material layers are placed otherwise fixed onto a Si substrate. After that selectively fixed away to leave microscopic 3D structures like diaphragms, beams, levers, springs, and gears.

mems-ic

The MEMS fabrication needs many techniques which are used to construct other semiconductor circuits like oxidation process, diffusion process, ion implantation process, low-pressure chemical vapor deposition process, sputtering, etc. Additionally, these sensors use a particular process like micromachining.

### MEMS Sensor Working Principle

Whenever the tilt is applied to the MEMS sensor, then a balanced mass makes a difference within the electric potential. This can be measured like a change within capacitance. Then that signal can be changed to create a stable output signal in digital, 4-20mA or VDC.

These sensors are fine solutions to some applications which do not demand the maximum accuracy like industrial automation, position control, roll, and pitch measurement, and platform leveling.

### Types of MEMS

The common types of MEMS sensors are obtainable within the market are

* MEMS accelerometers
* MEMS gyroscopes
* MEMS pressure sensors
* MEMS magnetic field sensors

### MEMS Advantages

The advantages of MEMS sensor include the following.

* The manufacturing of MEMS is semiconductor IC manufacturing like low-cost mass invention, consistency is also essential to MEMS devices.
* The size of sensor sub-components will be within 1 to 100 micrometers range as well as the MEMS device size will determine 20 micro-meters to a millimeter range.
* Power consumption is very low.
* Simple to incorporate into systems or change
* The thermal constant is small
* These can be highly opposed to shock, radiation, and vibration.
* Better thermal development tolerance
* Parallelism

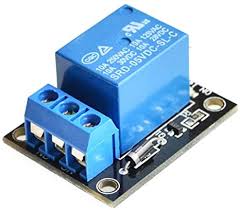
### Applications of MEMS

MEMS sensors are used in different domains which include [automotive](https://www.elprocus.com/power-electronics-in-automotive-applications/), consumer, industrial, military, biotechnology, space exploration, and commercial purposes which include inkjet printers, accelerometers within modern cars, consumer electronics, in personal computers, etc.

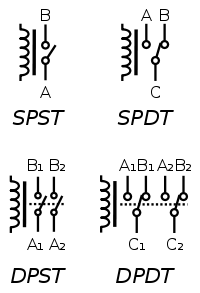
The best examples of MEMS devices mainly include adaptive optics, optical cross-connects, airbag [accelerometers](https://www.elprocus.com/accelerometer-sensor-working-and-applications/), mirror arrays for TVs & displays, steerable micromirrors, RF MEMS devices, not reusable medical devices, etc.

Thus, this is all about the [MEMS sensor](https://en.wikipedia.org/wiki/MEMS_sensor_generations). The main disadvantage of these sensors is, even though the making cost for each part is extremely low. But there is a huge investment associated while designing, manufacturing, and succeeding MEMS-based product. Consequently, designers are not likely to expand components for low volume applications.

**RELAY**

****

A relay is an [electrically](http://en.wikipedia.org/wiki/Electric) operated [switch](http://en.wikipedia.org/wiki/Switch). Many relays use an electromagnet to operate a switching mechanism mechanically, but other operating principles are also used. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits, repeating the signal coming in from one circuit and re-transmitting it to another. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

[](http://en.wikipedia.org/wiki/File:Relay_symbols.svg)

A type of relay that can handle the high power required to directly drive an electric motor is called a [contractor](http://en.wikipedia.org/wiki/Contactor). [Solid-state relays](http://en.wikipedia.org/wiki/Solid-state_relays) control power circuits with no [moving parts](http://en.wikipedia.org/wiki/Moving_parts), instead using a semiconductor device to perform switching. Relays with calibrated operating characteristics and sometimes multiple operating coils are used to protect electrical circuits from overload or faults; in modern electric power systems these functions are performed by digital instruments still called "[protective relays](http://en.wikipedia.org/wiki/Protective_relay)".



The two main types of relay are electromechanical and solid-state. Electromechanical relays have a moving plate with contacts on it, while solid-state relays work similar to transistors and have no moving parts.

Power relays use an electromagnet to move a set of contacts. The contacts are pulled towards the electromagnet while current is flowing it. The contacts are connected to a spring that pulls them back when the current stops flowing.

Latching relays acts like a two-pole switch in that the contacts stay in position when the current stops flowing. This is achieved by using a solenoid to move a ratchet and cam, or by using an electromagnet on either side.

Reed relays have only a set of ferro-metallic contacts inside a glass tube. The contacts close when an external magnetic field is applied, and open again when it is removed. Reed relays are commonly used as door sensors for alarm systems.

Overload relays are similar to reed relays but use a bimetallic strip to move the contacts. High temperature causes the strip to bend, bringing the contacts together.

## Applications:

Relays are used to control the flow of large currents using a small current. In the past, they were widely used in telephone exchanges to switch calls, but have been largely replaced by digital equipment. They are used to switch motors on and off, and to protect them from overheating. Thermostats use an overload relay to shut off the current when the temperature rises above the required setting.

**RELAY DRIVER DESCRIPTION:**

The ULN2003 is a monolithic high voltage and high current Darlington transistor arrays. It consists of seven NPN Darlington pairs that features high-voltage outputs with common-cathode clamp diode for switching inductive loads. The collector-current rating of a single Darlington pair is 500mA. The Darlington pairs may be paralleled for higher current capability. Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED gas discharge), line drivers, and logic buffers.

The ULN2003 has a 2.7kW series base resistor for each Darlington pair for operation directly with TTL or 5V CMOS devices.

**FEATURES:**

\* 500mA rated collector current (Single output)

\* High-voltage outputs: 50V

\* Inputs compatible with various types of logic.

\* Relay driver application

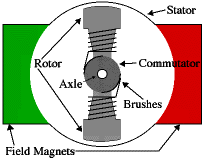
**DC MOTOR**

****

## An electric motor is a machine which converts electrical energy into mechanical energy.

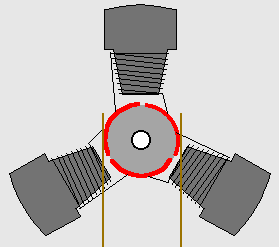
**Principles of operation**

In any electric motor, operation is based on simple electromagnetism. A current-carrying conductor generates a magnetic field; when this is then placed in an external magnetic field, it will experience a force proportional to the current in the conductor, and to the strength of the external magnetic field. As you are well aware of from playing with magnets as a kid, opposite (North and South) polarities attract, while like polarities (North and North, South and South) repel. The internal configuration of a DC motor is designed to harness the magnetic interaction between a current-carrying conductor and an external magnetic field to generate rotational motion.

 Let's start by looking at a simple 2-pole DC electric motor (here red represents a magnet or winding with a "North" polarization, while green represents a magnet or winding with a "South" polarization).

Every DC motor has six basic parts -- axle, rotor (a.k.a., armature), stator, commutator, field magnet(s), and brushes. In most common DC motors (and all that BEAMers will see), the external magnetic field is produced by high-strength permanent magnets1. The stator is the stationary part of the motor -- this includes the motor casing, as well as two or more permanent magnet pole pieces. The rotor (together with the axle and attached commutator) rotates with respect to the stator. The rotor consists of windings (generally on a core), the windings being electrically connected to the commutator. The above diagram shows a common motor layout -- with the rotor inside the stator (field) magnets.

The geometry of the brushes, commutator contacts, and rotor windings are such that when power is applied, the polarities of the energized winding and the stator magnet(s) are misaligned, and the rotor will rotate until it is almost aligned with the stator's field magnets. As the rotor reaches alignment, the brushes move to the next commutator contacts, and energize the next winding. Given our example two-pole motor, the rotation reverses the direction of current through the rotor winding, leading to a "flip" of the rotor's magnetic field, driving it to continue rotating.



In real life, though, DC motors will always have more than two poles (three is a very common number). In particular, this avoids "dead spots" in the commutator. You can imagine how with our example two-pole motor, if the rotor is exactly at the middle of its rotation (perfectly aligned with the field magnets), it will get "stuck" there. Meanwhile, with a two-pole motor, there is a moment where the commutator shorts out the power supply (i.e., both brushes touch both commutator contacts simultaneously). This would be bad for the power supply, waste energy, and damage motor components as well. Yet another disadvantage of such a simple motor is that it would exhibit a high amount of torque "ripple" (the amount of torque it could produce is cyclic with the position of the rotor).

|  |  |
| --- | --- |
| motor | motor |
| **Figure 1: Force in DC Motor** | **Figure 2 : Magnetic Field in DC Motor** |
| motor | motor |
| **Figure 3 :  Torque in DC Motor** | **Figure 4 : Current Flow in DC Motor** |

If an Electric current flows through two copper wires that are between the poles of a magnet, an upward force will move one wire up and a downward force will move the other wire down. The loop can be made to spin by fixing a half circle of copper which is known as COMMUTATOR to each end of the loop. Current is passed into and out of the loop by brushes that press onto the strips. The BRUSHES do not go round so the wire does not get twisted. This arrangement also makes sure that the current always passes down on the right and back on the left so that the rotation continues. This is how a simple Electric motor is made.

**APPLICATIONS:**

1. Machines.
2. Material handler conveyors
3. System and gear drivers
4. Elevators
5. mixers

**CHAPTER: 3**

**SOFTWARE DEVELOPMENT**

**3.1 INTRODUCTION**

This tutorial will walk you through downloading, installing, and testing the [Arduino software](http://arduino.cc/en/Main/Software) (also known as the Arduino IDE - short for Integrated Development Environment). Before you jump to the page for your operating system, make sure you’ve got all the right equipment.

What you will need:

* A computer (Windows, Mac, or Linux)
* An Arduino-compatible microcontroller (anything from [this guide](https://www.sparkfun.com/pages/arduino_guide) should work)
* A USB A-to-B cable, or another appropriate way to connect your Arduino-compatible microcontroller to your computer (check out this [USB buying guide](https://www.sparkfun.com/pages/USB_Guide) if you’re not sure which cable to get).

An Arduino Uno

[](https://cdn.sparkfun.com/assets/0/5/8/5/c/516f152ace395fd874000000.png)

An A-to-B USB Cable

### Suggested Reading

If you’re new to Arduino in general, you want to check out this tutorial to familiarize yourself with everyone’s favorite microcontroller platform.

* [What is an Arduino?](https://learn.sparkfun.com/tutorials/what-is-an-arduino)

If you’re ready to get started, click on the link in the column on the left that matches up with your operating system, or you can jump to your operating system here.

* [Windows](https://learn.sparkfun.com/tutorials/installing-arduino/windows)
* [Mac](https://learn.sparkfun.com/tutorials/installing-arduino/mac)
* [Linux](https://learn.sparkfun.com/tutorials/installing-arduino/linux)

## Windows

This page will show you how to install and test the Arduino software with a Windows operating system (Windows 8, Windows 7, Vista, and XP).

### Windows 8, 7, Vista, and XP

* Go to the Arduino [download page](http://arduino.cc/en/Main/Software) and download the latest version of the Arduino software for Windows.
* When the download is finished, un-zip it and open up the Arduino folder to confirm that yes, there are indeed some files and sub-folders inside. The file structure is important so don’t be moving any files around unless you really know what you’re doing.
* Power up your Arduino by connecting your Arduino board to your computer with a USB cable (or FTDI connector if you’re using an Arduino pro). You should see the an LED labed ‘ON’ light up. ([This diagram](https://learn.sparkfun.com/tutorials/what-is-an-arduino/whats-on-the-board) shows the placement of the power LED on the UNO).
* If you’re running Windows 8, you’ll need to disable driver signing, so go see the Windows 8 section. If you’re running Windows 7, Vista, or XP, you’ll need to install some drivers, so head to the Windows 7, Vista, and XP section down below.

### Windows 8

Windows 8 comes with a nice little security ‘feature’ that ‘protects’ you from unsigned driver installation. Some older versions of Arduino Uno come with unsigned drivers, so in order to use your Uno, you’ll have to tell Windows to disable driver signing. This issue has been addressed in newer releases of the Arduino IDE, but if you run into issues, you can try this fix first.

For a nice, step-by-step tutorial with pictures [click here](https://learn.sparkfun.com/tutorials/disabling-driver-signature-on-windows-8/overview), otherwise the steps are outlined below.

To temporarily disable driver signing:

* From the Metro Start Screen, open Settings (move your mouse to the bottom-right-corner of the screen and wait for the pop-out bar to appear, then click the Gear icon)
* Click ‘More PC Settings’
* Click ‘General’
* Scroll down, and click ‘Restart now’ under ‘Advanced startup’.
* Wait a bit.
* Click ‘Troubleshoot’.
* Click ‘Advanced Options’
* Click ‘Windows Startup Settings’
* Click Restart.
* When your computer restarts, select ‘Disable driver signature enforcement‘ from the list.

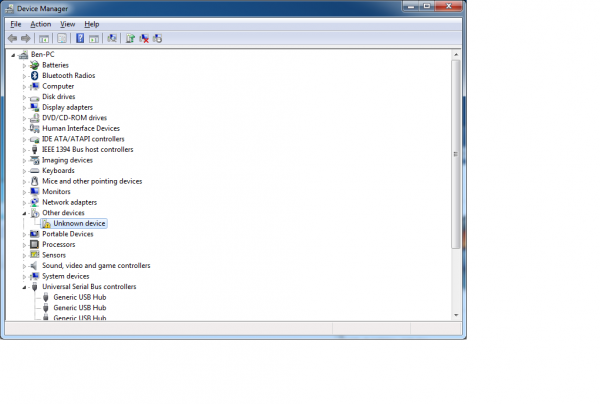
To permanently disable driver signing (recommended, but has some minor security implications):

* Go to the metro start screen
* Type in “cmd”
* Right click “Command Prompt” and select “Run as Administrator” from the buttons on the bottom of your screen
* Type/paste in the following commands: bcdedit -set loadoptions DISABLE\_INTEGRITY\_CHECKS bcdedit -set TESTSIGNING ON
* Reboot!

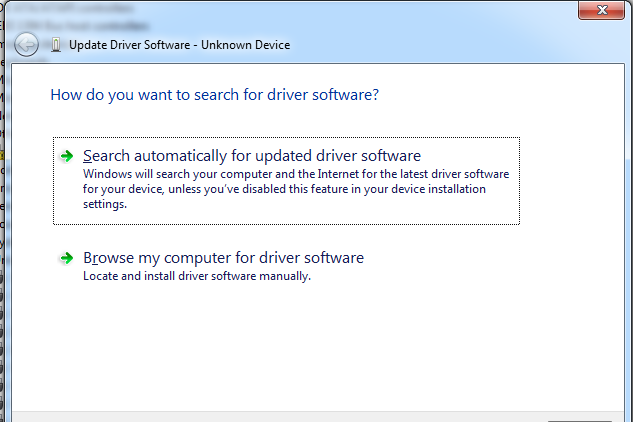
### Windows 7, Vista, and XP

Installing the Drivers for the Arduino Uno (from Arduino.cc)

* Plug in your board and wait for Windows to begin it’s driver installation process
* After a few moments, the process will fail, despite its best efforts
* Click on the Start Menu, and open up the Control Panel
* While in the Control Panel, navigate to System and Security. Next, click on System
* Once the System window is up, open the Device Manager
* Look under Ports (COM & LPT). You should see an open port named “Arduino UNO (COMxx)”. If there is no COM & LPT section, look under ‘Other Devices’ for ‘Unknown Device’

[](https://cdn.sparkfun.com/assets/b/9/0/7/b/51795da2ce395f7156000001.png)

* Right click on the “Arduino UNO (COMxx)” or “Unknown Device” port and choose the “Update Driver Software” option
* Next, choose the “Browse my computer for Driver software” option

[](https://cdn.sparkfun.com/assets/6/9/5/7/2/51795da2ce395f0d56000000.png)

* Finally, navigate to and select the Uno’s driver file, named “ArduinoUNO.inf”, located in the “Drivers” folder of the Arduino Software download (not the “FTDI USB Drivers” sub-directory). If you cannot see the .inf file, it is probably just hidden. You can select the ‘drivers’ folder with the ‘search sub-folders’ option selected instead.
* Windows will finish up the driver installation from there

For earlier versions of the Arduino boards (e.g.Arduino Duemilanove, Nano, or Diecimila) check out [this page](http://arduino.cc/en/Guide/Windows) for specific directions.

There are a variety of development platforms that can be equipped to program the ESP8266. You can go with [Espruino](https://www.espruino.com/) – JavaScript SDK and firmware closely emulating Node.js, or use [Mongoose OS](https://mongoose-os.com/) – An operating system for IoT devices (recommended platform by Espressif Systems and Google Cloud IoT) or use a software development kit (SDK) provided by Espressif or one of the platforms listed on [WiKiPedia](https://en.wikipedia.org/wiki/ESP8266" \l "SDKs).

Fortunately, the amazing ESP8266 community took the IDE selection a step further by creating an Arduino add-on. If you’re just getting started programming the ESP8266, this is the environment we recommend beginning with, and the one we’ll document in this tutorial.

This ESP8266 add-on for Arduino is based on the amazing work by [Ivan Grokhotkov](https://github.com/igrr) and the rest of the ESP8266 community. Check out the [ESP8266 Arduino GitHub repository](https://github.com/esp8266/Arduino) for more information.

Installing the ESP8266 Core on Windows OS

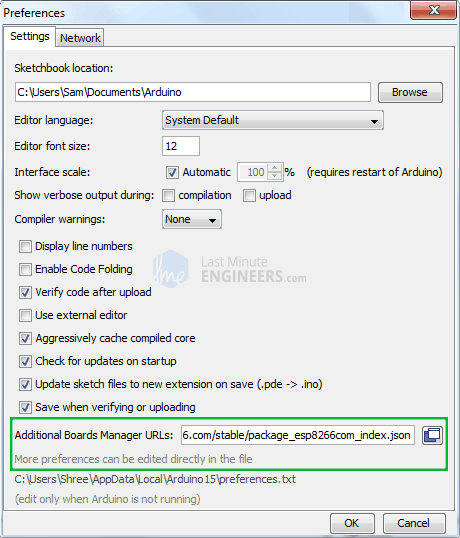
Let’s proceed with installing ESP8266 Arduino core.

The first thing is having latest Arduino IDE (Arduino 1.6.4 or higher) installed on your PC. If don’t have it, we recommend upgrading now.

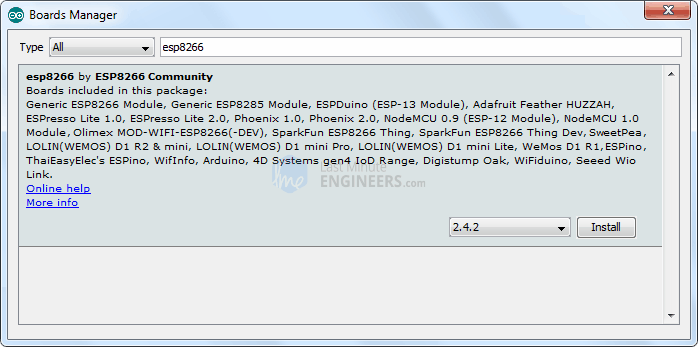
[Latest Arduino IDE](https://www.arduino.cc/en/Main/Software)

To begin, we’ll need to update the board manager with a custom URL. Open up Arduino IDE and go to File > Preferences. Then, copy below URL into the Additional Board Manager URLs text box situated on the bottom of the window:

<http://arduino.esp8266.com/stable/package_esp8266com_index.json>



Hit OK. Then navigate to the Board Manager by going to Tools > Boards > Boards Manager. There should be a couple new entries in addition to the standard Arduino boards. Filter your search by typing esp8266. Click on that entry and select Install.



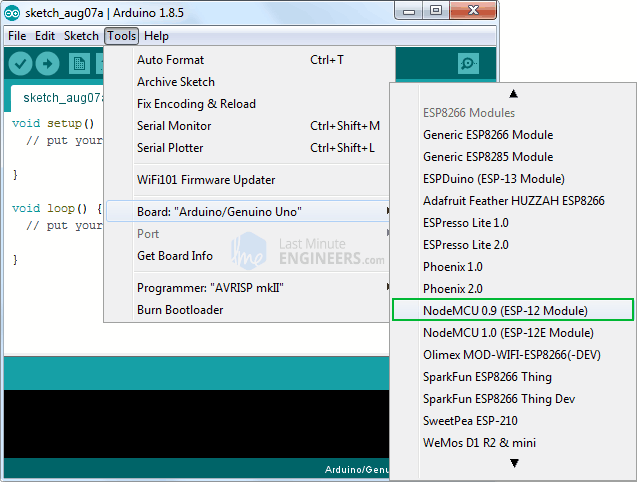
The board definitions and tools for the ESP8266 include a whole new set of gcc, g++, and other reasonably large, compiled binaries, so it may take a few minutes to download and install (the archived file is ~110MB). Once the installation has completed, a small *INSTALLED* text will appear next to the entry. You can now close the Board Manager.

Arduino Example: Blink

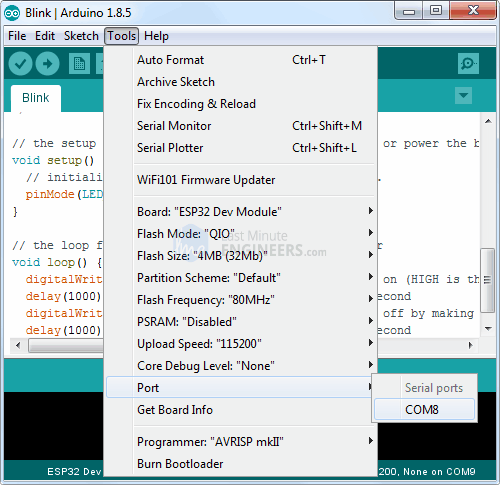
To make sure ESP8266 Arduino core and the NodeMCU are properly set up, we’ll upload the simplest sketch of all – The Blink!

We will use the on-board LED for this test. As mentioned earlier in this tutorial, D0 pin of the board is connected to on-board Blue LED & is user programmable. Perfect!

Before we get to uploading sketch & playing with LED, we need to make sure that the board is selected properly in Arduino IDE. Open Arduino IDE and select NodeMCU 0.9 (ESP-12 Module) option under your Arduino IDE > Tools > Board menu.



Now, plug your ESP8266 NodeMCU into your computer via micro-B USB cable. Once the board is plugged in, it should be assigned a unique COM port. On Windows machines, this will be something like COM#, and on Mac/Linux computers it will come in the form of /dev/tty.usbserial-XXXXXX. Select this serial port under the Arduino IDE > Tools > Port menu. Also select the Upload Speed : 115200



Warning:

More attention needs to be given to selecting board, choosing COM port and selecting Upload speed. You may get espcomm\_upload\_mem error while uploading new sketches, if failed to do so.

Once you are done, try the example sketch below.

void setup()

{

pinMode(D0, OUTPUT);

}

void loop()

{

digitalWrite(D0, HIGH);

delay(500);

digitalWrite(D0, LOW);

delay(500);

}

Once the code is uploaded, LED will start blinking. You may need to tap the RST button to get your ESP8266 to begin running the sketch.

**CHAPTER-4**

**EXPERIMENTAL RESULT**

Once the device is worn and connected, rest the finger on a surface if it is required, and switch ON the device. Move the finger tip for the control of devices. The graph(figure 7), table (figure 8) illustrates the required output obtained from the device, i.e. change in axis provides different voltage values which are transmitted using Mems. By moving the index finger on transmitter end the latch can be controlled on the receiver end.

**CHAPTER-5**

**ADVANTAGES AND APPLICATIONS**

* 1. **ADVANTAGES**
* Low cost
* Easy to implement

**5.2 APPLICATIONS**

* Commercial
* Household
* Useful for patients and handicapped

**CHAPTER-6**

**CONCLUSION AND FUTURESCOPE**

**CONCLUSION**

Technologies developed based on gesture are now really affordable and converged with familiar and popular technologies. This project has been done to control a device based on free air hand gesture motion technology. It uses the Hand gesture pad which works on a principle of e-field distortion. Hand gesture pad works based on electric field variation. With this technology, any device can be controlled from the user end. It is able to bring a reliable assistance and security in electronic sector.

**FUTURESCOPE**

In future more, home appliances can be controlled by incorporating those devices with newer versions of gestures, also implemented in every home at low cost. The device helps the aged people who have less mobility.

**REFERENCES**

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller,and T. M. Vaughan, “Brain-computer interfaces for communication and control,” Clin Neu rophysiol , vol. 113, no. 6, pp. 767–791,Jun. 2002.

[2] Omri Mendels, Helman Stern and Sigal Berman(November2014) User Identification for Home Entertainment Based on Free-Air Hand Motion Signatures IEEE Transactions On Systems, Man, And Cybernetics: Systems, Vol. 44, No. 11

[3] A. Hayder, J. Dargham, A. Chekima, and G. M. Ervin, Person identification using gait, Int. J. Comput. Electr. Eng., vol. 3, no. 4, pp. 477–482, 2011

[4] J. Kela, P. Korpipaa, J. Maentyjaervi, S. Kallio, G. Savino, L. Jozzo, and S. Marca. Accelerometer-based gesture control for a design environment. Springer. Personal And Ubiquitous Computing,10(5):285–299,

[5] J. Maentyjaervi, J. Kela, P. Korpipaa, and S. Kallio. Enabling fast and effortless customisation in accelerometer based gesture interaction. In Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia, pages 25–31. ACM, 2006.

[6] A. Kale et al(Sep. 2004.)., Identification of humans using gait, IEEE Trans. Image Process., vol. 13, no. 9, pp. 1163–1173

[7] A. Iosifidis, A. Tefas, and I. Pitas,(Apr. 2007) Activity-based person identification using fuzzy representation and discriminant learning, IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 530–5A.

[8] A. E. Ahmed and I.Traore,( Jul./Sep. 2007) A new biometric technology based on mouse dynamics, IEEE Trans. Dependable Secure Comput., vol. 4, no. 3, pp. 165– 17

[9] C. Feher et al., Identity theft, computers and behavioral biometrics, in Proc. IEEE Int.Conf. Intell. Secur. Inf. (ISI),2009, pp. 8–

[10] D. E. L. Menshawy, H. M. O. Mokhtar, and O. Hegazy(2012) Enhanced authentication mechanisms for desktop platform and smart phones, Int. J. Adv. Comput. Sci. Appl.,vol. 3, no. 10, pp. 100–106.

[11] A. K. Jain, F. D. Griess, and S. D. Connell,( 2002) On-line signature verification, Pattern Recognit., vol, 35, no, 12, pp. 2963–297

[12] J. Guo, D. Doermann, and A. Rosenfeld,( 1997) Local correspondence for detecting random forgeries, in Proc. Int. Conf. Doc. Anal. Recognit., Ulm, Germany,p.319