**HARDWARE DESCRIPTION:**

**4.1 ARDUINO (ATMEGA 328P)**



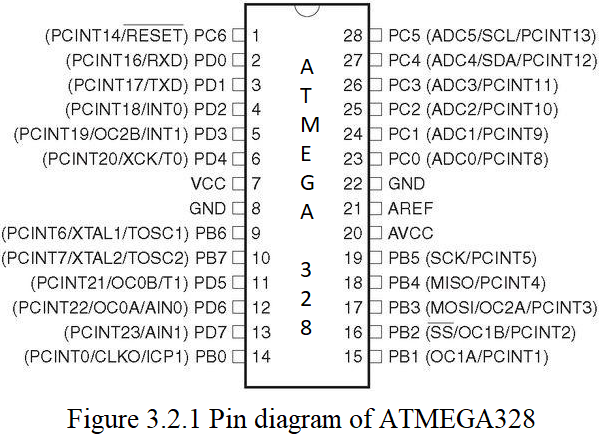
Figure 3.2 Aurdino

The Atmel AVR® core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in a single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The ATmega328/P provides the following features: 32Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 1Kbytes EEPROM, 2Kbytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 1 serial programmable USARTs , 1 byte-oriented 2-wire Serial Interface (I2C), a 6- channel 10- bit ADC (8 channels in TQFP and QFN/MLF packages) , a programmable Watchdog Timer with internal Oscillator, an SPI serial port, and six software selectable power saving modes.

This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main oscillator and the asynchronous timer continue to run. Atmel offers the QTouch® library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS™) technology for unambiguous detection of key events. The easy-to-use Q Touch Suite toolchain allows you to explore, develop and debug your own touch applications. The device is manufactured using Atmel’s high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core.

Pin diagram of ATMEGA328

The ATmega328/P is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, and Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits. Below figure 3.2.1 pin diagram of ATMEGA328.



FEATURES OF ATMEGA328

28-pin AVR Microcontroller

Flash Program Memory: 32 kilo bytes

EEPROM Data Memory: 1 kilo bytes

SRAM Data Memory: 2 kilo bytes

I/O Pins: 23

Timers: Two 8-bit / One 16-bit

A/D Converter: 10-bit Six Channel

PWM: Six Channels

RTC: Yes with Separate Oscillator

MSSP: SPI and I²C Master and Slave Support

USART: Yes

External Oscillator: up to 20MHz

ADVANTAGES/ IMPROVEMENTS IN ATMEGA328

Still runs on 5 V, so legacy 5 V stuff interfaces cleaner

Even though it's 5 V capable, newer parts can run to 1.8 V. This wide range is very rare.

Nice instruction set, very good instruction throughput compared to other processors (HCS08, PIC12/16/18).

High quality GCC port (no proprietary crappy compilers!)

"PA" variants have good sleep mode capabilities, in micro-amperes.

Well rounded peripheral set

Q Touch capability

Pin Descriptions table

VCC

Digital supply voltage.

GND

Ground.

Port B (PB [7:0]) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up

resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB [7:6] is used as TOSC [2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

Port C (PC [5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC [5:0] output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port ‘C’ are elaborated in the Alternate Functions of Port C

section.

Port D (PD [7:0])

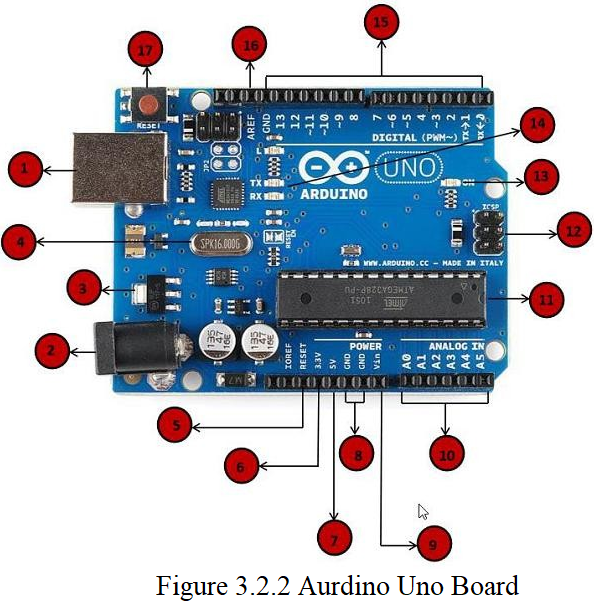
Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

AVCC

AVCC is the supply voltage pin for the A/D Converter, PC [3:0], and PE [3:2]. It should be externally connected to VCC, even if the ADC is not used. If the ADC is used, it should be connected to VCC through a low-pass filter. Note that PC [6:4] use digital supply voltage, VCC.

Arduino Uno Board Description

We will learn about the different components on the Arduino board. We will study the Arduino UNO board because it is the most popular board in the Arduino board family. In addition, it is the best board to get started with electronics and coding. Some boards look a bit different from the one given below figure 3.2.2, but most Arduinos have majority of these components in common.



Arduino board can be powered by using the USB cable from computer. All we need to do is connect the USB cable to the USB connection (1).

Power (Barrel Jack)

Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.

Voltage Regulator

The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.

Crystal Oscillator

The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.

Arduino Reset

We can reset wer Arduino board, i.e., start wer program from the beginning. We can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, we can connect an external reset button to the Arduino pin labelled RESET (5).

Pins (3.3, 5, GND, Vin)

3.3V (6) − Supply 3.3 output volt

5V (7) − Supply 5 output volt

Most of the components used with Arduino board works fine with 3.3 volt and 5 volt.

GND (8) (Ground) − There are several GND pins on the Arduino, any of which can be used to ground wer circuit.

Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.

Analog pins

The Arduino UNO board has five analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.

**INTERFACING ESP8266 WITH ARDUINO:**

ESP8266 (technically ESP8266EX) is a WiFi Module based on Cadence Tensilica L106 32-bit MCU manufactured by Espressif Systems. The ESP8266 SoC contains a fully functional WiFi Stack and TCP/IP Stack that allows any Microcontroller to get connected to WiFi Network.

With Software Development Kits (SDKs), you can directly program the ESP8266’s on-chip Microcontroller, without the need for an external Microcontroller.

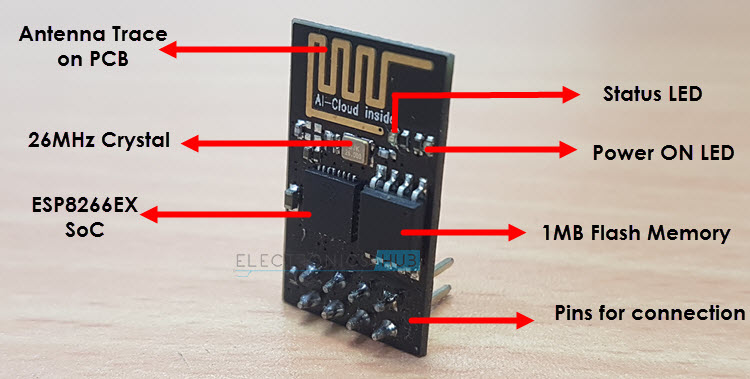
Based on the ESP8266 SoC, several third party manufacturers started manufacturing custom boards and one such manufacturer is Ai-Thinker. The first board manufactured by Ai-Thinker is ESP-01 (which is the same board used in this project) and it became quite popular.

Based on the success of the ESP-01 Module, several other modules like ESP-02, ESP-07, ESP-12, etc. were released by Ai-Thinker. All these boards are based on ESP8266 SoC but the main difference is the number of GPIO Pins.

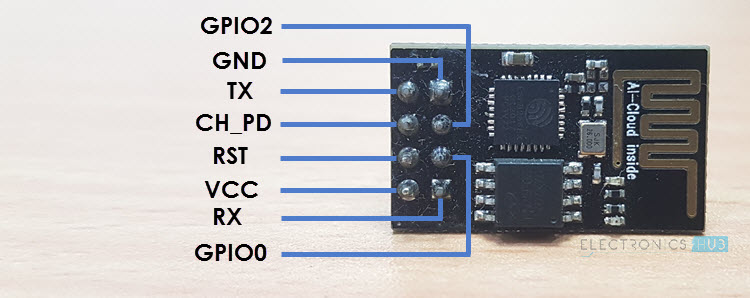
There are other modules like ESP-WROOM by Espressif Systems, NodeMCU, WeMOS, SparkFun ESP8266, etc.

### ESP8266 ESP-01

In this project, we will be using the Ai-Thinker’s ESP-01 Module. It consists of 8 pins and the following image shows the different components of the board.



Coming to the pin configuration, as mentioned above, the ESP-01 module consists of 8 pins and these pins are VCC, GND, TX, RX, RST, CH\_PD, GPIO0 and GPIO2. The following image shows the pin diagram of the ESP-01 Module.



#### Pin Description of ESP8266 ESP-01 Module

* **VCC**: It is the power pin through which 3.3V is supplied.
* **GND**: It is the ground pin.
* **TX**: This pin is used to transmit serial data to other devices.
* **RX**: The RX pin is used to receive serial data from other devices.
* **RST**: It is the Reset Pin and it is an active LOW Pin. (ESP8266 will reset if the RST pin receives LOW signal).
* **CH\_PD**: This is the chip enable pin and it is an active HIGH Pin. It is usually connected to 3.3V.
* **GPIO0**: The GPIO0 (General Purpose I/O) Pin has dual functions – one for normal GPIO Operation and other for enabling the Programming Mode of ESP8266.
* **GPIO2**: This is GPIO Pin.

**IMPORTANT NOTE**: The ESP8266 is not compatible with 5V and the ESP-01 Module does not have any voltage regulators on-board. Make sure that the power supply to the ESP8266 is 3.3V, preferably from a dedicated power supply rather than taking it from the 3.3V Pin of the Arduino.

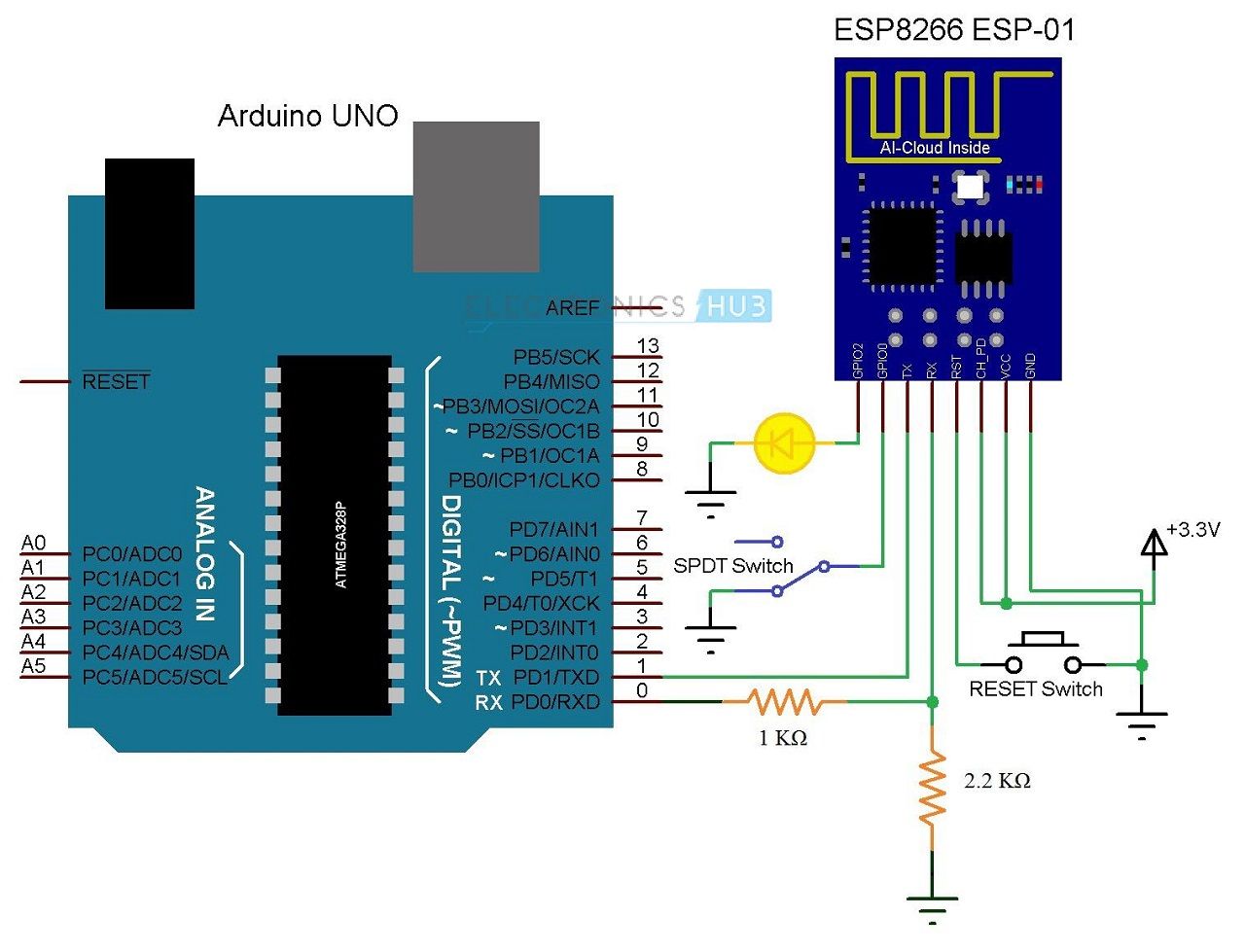
### ESP8266 Arduino Interface

Before seeing the ESP8266 Arduino Interface, you need to know a few things about the ESP8266 Module. The ESP8266 WiFi Module comes with default firmware which supports AT commands.

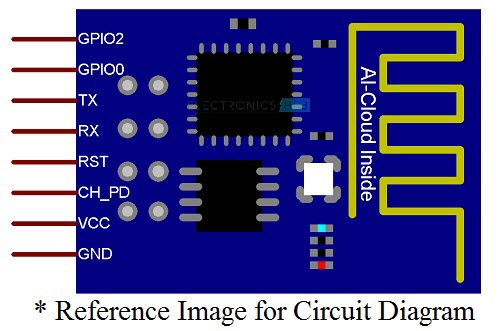
After interfacing the ESP8266 WiFi Module with Arduino and uploading our own program, the original firmware will be erased. We will see in a separate project on how to interface ESP8266 Module for AT Commands and also how to flash the original firmware using Arduino.

Now, we will see how to program ESP8266 using Arduino and access its GPIO pins. First, we will see the circuit diagram of the interface.

#### Circuit Diagram of ESP8266 Arduino Interface



If the ESP8266 Module in the circuit diagram is not clear, the following image might help you. It is just a personal representation for circuit diagram. You have already seen actual pin diagram in the previous section.

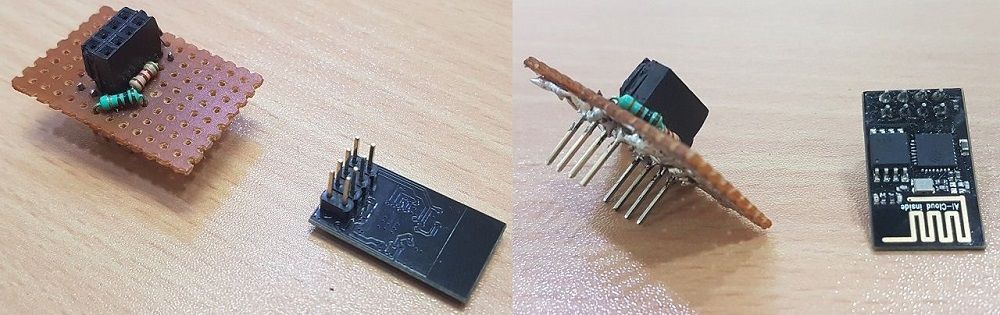


#### Components Required

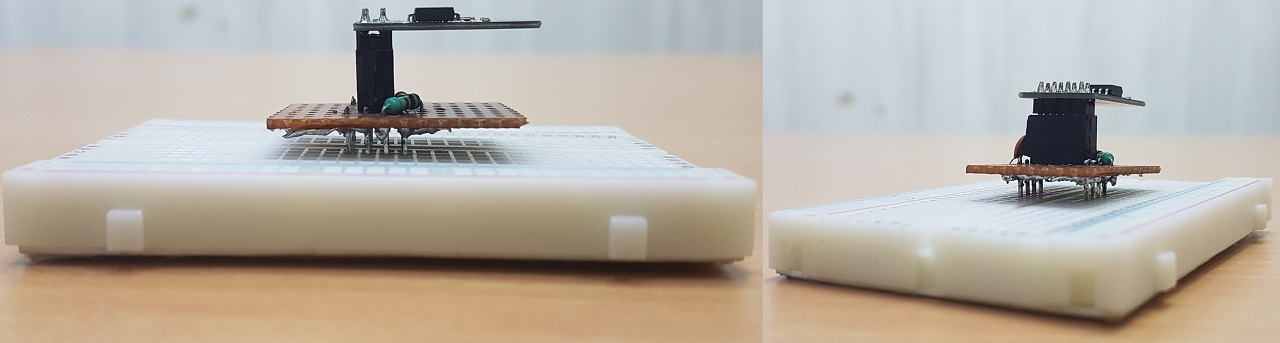
* Arduino UNO  [[Buy Here](https://amzn.to/2CcrrQh)]
* ESP8266 ESP-01
* 1 KΩ Resistor
* 2.2 KΩ Resistor
* 100 pF Capacitor (Capacitor Code – 104)
* Mini Breadboard
* Connecting Wires

### Getting the ESP8266 ESP-01 Module Ready for Breadboard Mount

If you take a look at the Pins of the ESP8266 ESP-01 Module, you can observe that it is not breadboard friendly. So, I have made a small perf board with breadboard friendly pins on the bottom and female headers to mount the ESP8266 ESP-01 Module on it.



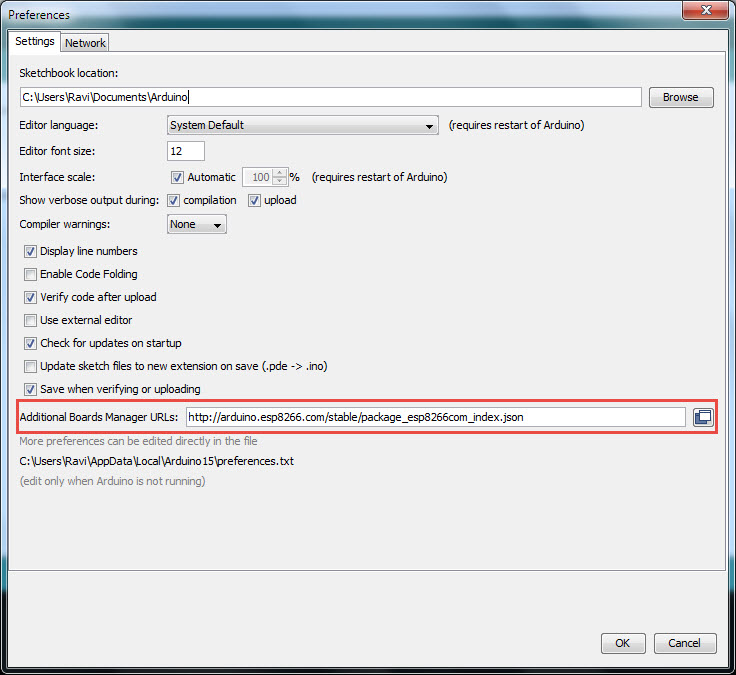
On this perf board, I’ve also connected the level converter resistors for RX Pin of the ESP8266 and also a 100 pF Bypass Capacitor between VCC (3.3V) and GND. The following image shows the mounting of ESP8266 ESP-01 Module on a mini breadboard.



### Getting Arduino IDE Ready for Programming ESP8266

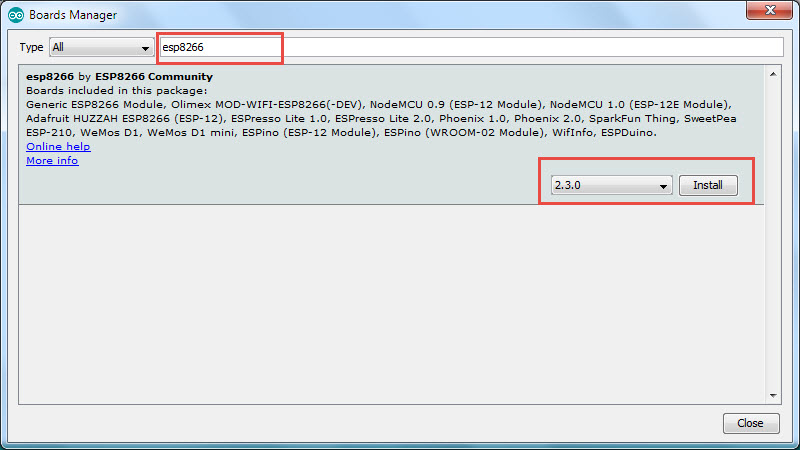
ESP8266 WiFi Module can be programmed using Arduino IDE and in order to do that you need to make a few changes to the Arduino IDE. First, go to File –> Preferences in the Arduino IDE and in the Additional Boards Manager URLs Section, enter the following URL.

http://arduino.esp8266.com/stable/package\_esp8266com\_index.json



**NOTE**: You can add many such URLs but they must be separated with commas.

Now, go to Tools –> Board –> Boards Manager and search for ESP8266 in the search field. Select the ESP8266 by ESP8266 Community and click on Install.



**NOTE**: This feature of adding third-party boards through board manager is available for Arduino IDE Version 1.6.4 and higher. So, make sure that you have the latest version of Arduino IDE.

### Getting Arduino UNO Ready for Programming ESP8266

In order to Program ESP8266 Module, we need to connect it to a computer. Since Serial Communication is the only available communication on the ESP8266 ESP-01 Module, we need an USB to Serial Adapter like an FTDI, CH340 or FT232RL.

If you do not have a dedicated USB to Serial Adapter, do not worry. The Arduino UNO has an on-board USB to Serial Adapter (which is used to program the Arduino). We are going to use this for programming the ESP8266.

We will be using the TX and RX Pins of the Arduino to connect to the ESP8266 Module and in order to make sure that Arduino isn’t using those pins, we can upload a bare minimum sketch to Arduino.

**NOTE**: Bare minimum sketch consists of just the setup and loop functions without any data in them.

In my case, I have an extra Arduino UNO Board with a non-functioning ATmega328p IC. So, I removed the Microcontroller IC from the Arduino UNO and started using it as an USB to Serial Converter.

### Circuit Design for Programming ESP8266 using Arduino

You have already seen the required components and the circuit diagram of the project. Now, let us try to understand the design of the circuit.

First and foremost, the ESP8266 Module works on 3.3V Power Supply and anything greater than that, like 5V for example, will kill the SoC. So, the VCC Pin and CH\_PD Pin of ESP8266 ESP-01 Module are connected to a 3.3V Supply.

Next important thing to remember is that the ESP8266 WiFi Module has two modes of operation: Programming Mode and Normal Mode.

In Programming Mode, you can upload the program or firmware to the ESP8266 Module and in Normal Mode, the uploaded program or firmware will run normally.

In order to enable the Programming Mode, the GPIO0 pin must be connected to GND. In the circuit diagram, I’ve connected a SPDT switch to the GPIO0 pin. Toggling the lever of SPDT will switch the ESP8266 between Programming mode (GPIO0 is connected to GND) and normal mode (GPIO0 acts as a GPIO Pin).

Also, the RST (Reset) will play an important role in enabling Programming Mode. The RST pin is an active LOW pin and hence, it is connected to GND through a Push Button. So, whenever the button is pressed, the ESP8266 Module will reset.

The RX and TX pins of the ESP8266 Module are connected to RX and TX Pins on the Arduino board. Since the ESP8266 SoC cannot tolerate 5V, the RX Pin of Arduino is connected through a level converter consisting of a 1KΩ and a 2.2KΩ Resistor.

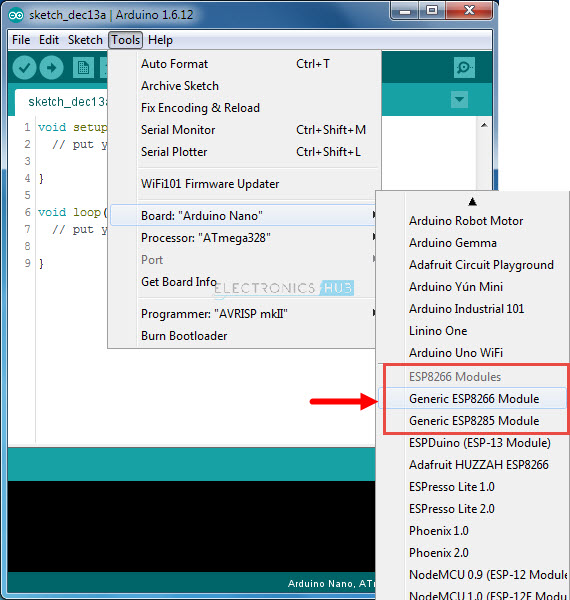
Finally the GPIO2 pin is connected to an LED to test the working of the program. All the necessary connections for enabling the Programming Mode in ESP8266 are mentioned below.

1. VCC – – > 3.3V
2. GND – – > GND
3. CH\_PD – – > 3.3V
4. RST – – > Normally Open; GND to Reset
5. GPIO0 – – > GND
6. TX – – > TX of Arduino
7. RX – – > RX of Arduino (through level converter)

### Working of ESP8266 Arduino Interface

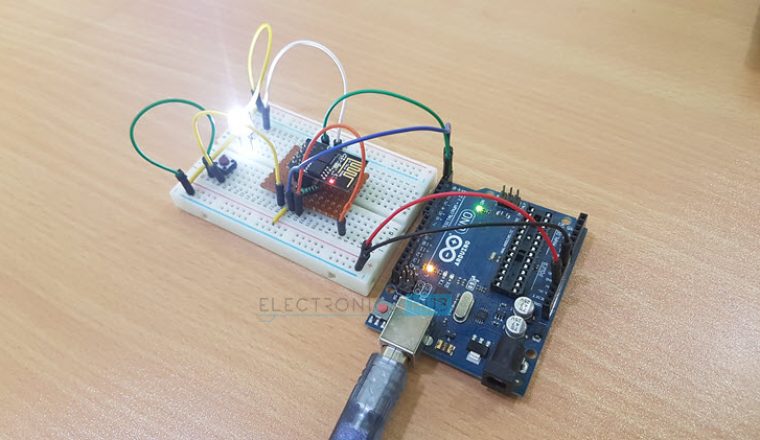
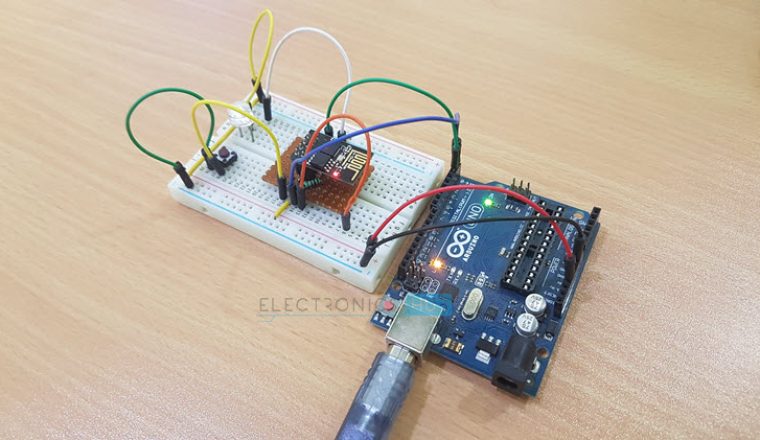
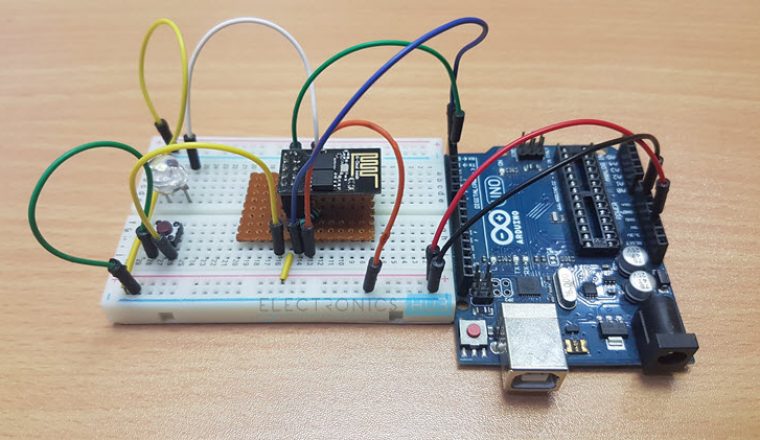
Make sure that all the above mentioned connections are properly made. After connecting and configuring the ESP8266 in Programming Mode (GPIO0 is connected to GND), connect the Arduino to the system.

Once the ESP8266 Module is powered ON, Push the RST button and open the Arduino IDE. In the Board options (Tools –> Board), select the “Generic ESP8266” Board. Select the appropriate port number in the IDE.



Now, open the Blink Sketch and change the LED Pin to 2. Here, 2 means GPIO2 pin of the ESP8266 Module. Before you hit the upload make sure that GPIO0 is connected to GND first and then press the RST button.

Hit the upload button and the code will take a while to compile and upload. You can see the progress at the bottom of the IDE. Once the program is successfully uploaded, you can remove the GPIO0 from GND. The LED connected to GPIO2 will blink.



**MOISTURE SENSOR:**

**Working of Sensor**

The soil moisture sensor consists of two probes which are used to measure the volumetric content of water. The two probes allow the current to pass through the soil and then it gets the resistance value to measure the moisture value.

When there is more water, the soil will conduct more electricity which means that there will be less resistance. Therefore, the moisture level will be higher. Dry soil conducts electricity poorly, so when there will be less water, then the soil will conduct less electricity which means that there will be more resistance. Therefore, the moisture level will be lower.

This sensor can be connected in two modes; Analog mode and digital mode. First, we will connect it in Analog mode and then we will use it in Digital mode.

#### **Specifications**

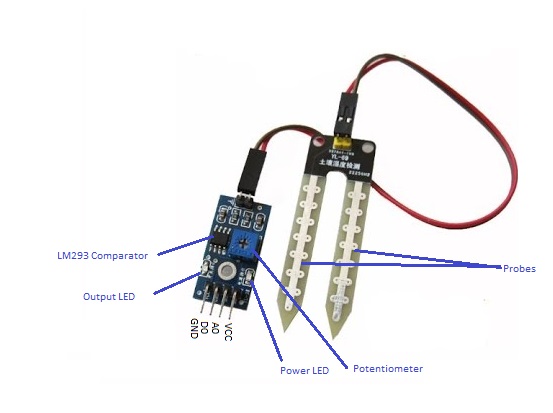
The specifications of the soil moisture sensor FC-28 are as follows

|  |  |
| --- | --- |
| Input Voltage | 3.3 – 5V |
| Output Voltage | 0 – 4.2V |
| Input Current | 35mA |
| Output Signal | Both Analog and Digital |

#### **Pin Out – Soil Moisture Sensor**

The soil Moisture sensor FC-28 has four pins

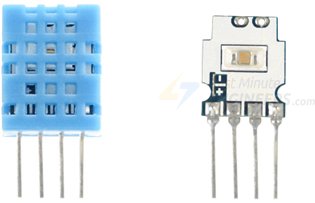
* VCC: For power
* A0: Analog output
* D0: Digital output
* GND: Ground
* The Module also contains a potentiometer which will set the threshold value and then this threshold value will be compared by the LM393 comparator. The output LED will light up and down according to this threshold value.

[](https://www.circuitstoday.com/arduino-soil-moisture-sensor/pin-out)Pin Out – Diagram

### ****Analog Mode – Interfacing Soil Moisture Sensor and Arduino****

* To connect the sensor in the analog mode, we will need to use the analog output of the sensor. When taking the analog output from the soil moisture sensor FC-28, the sensor gives us the value from 0-1023. The moisture is measured in percentage, so we will map these values from 0 -100 and then we will show these values on the serial monitor.

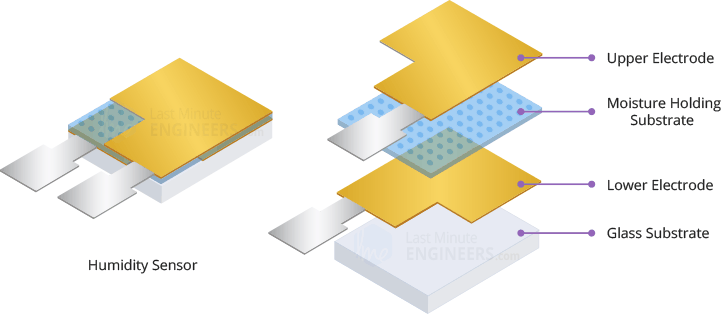
Inside the DHT11, there is a humidity sensing component along with a Thermistor.



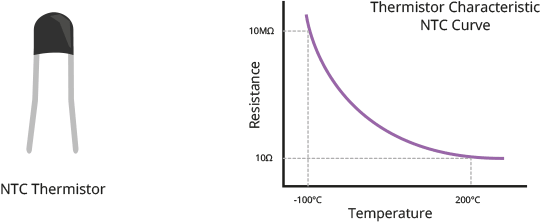
Humidity sensing component has two electrodes with moisture holding substrate sandwiched between them.

The ions are released by the substrate as water vapor is absorbed by it, which in turn increases the conductivity between the electrodes.

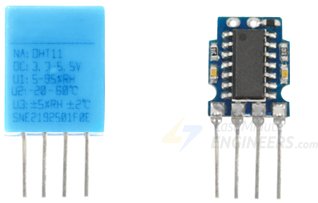
The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.



DHt11 also contains a NTC/Thermistor to measure temperature. A thermistor is a thermal resistor whose resistance changes drastically with temperature. The term “NTC” means “Negative Temperature Coefficient”, which means that the resistance decreases with increase of the temperature.



On the other side, there is a small PCB with an 8-bit SOIC-14 packaged IC. This IC measures and processes the analog signal with stored calibration coefficients, does analog to digital conversion and spits out a digital signal with the temperature and humidity.

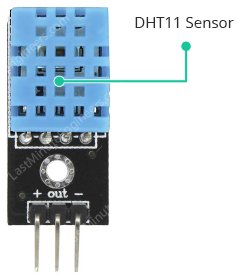


DHT11 Module Hardware Overview

At the heart of the module is the digital temperature & humidity sensor manufactured by AOSONG – DHT11.

DHT11 Sensor

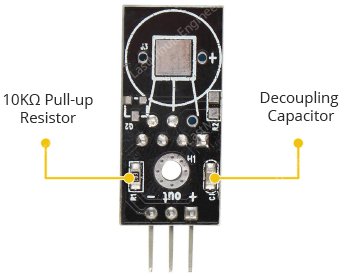
DHT11 can measure temperature from 0°C to 50°C with ±2.0°C accuracy, and humidity from 20 to 80% with 5% accuracy.



Note that the sampling rate of the DHT11 is 1Hz, meaning you can get new data from it once every second.

Supporting Circuitry

The module comes with all the essential supporting circuitry, so it should be ready to run without any extra components.

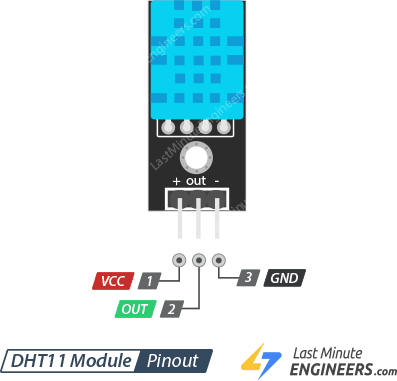


The DHT11 sensors usually require external pull-up resistor of 10KΩ between VCC and Out pin for proper communication between sensor and the Arduino. However, the module has a built-in pull-up resistor, so you need not add it.

The module also has a decoupling capacitor for filtering noise on the power supply.

DHT11 Module Pinout

The DHT11 module is fairly easy to connect. It has only three pins:



+ (VCC) pin supplies power for the sensor. 5V supply is recommended, although the supply voltage ranges from 3.3V to 5.5V. In case of 5V power supply, you can keep the sensor as long as 20 meters. However, with 3.3V supply voltage, cable length shall not be greater than 1 meter. Otherwise, the line voltage drop will lead to errors in measurement.

Out pin is used to communication between the sensor and the Arduino.

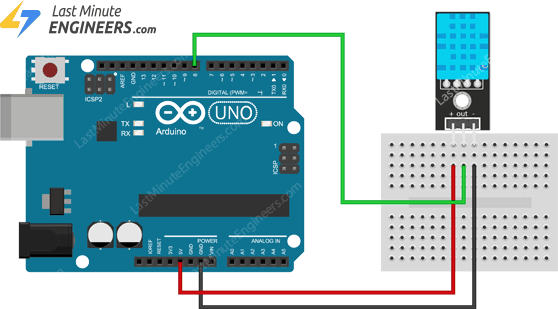
– (GND) should be connected to the ground of Arduino.

Wiring DHT11 Module to Arduino

Let’s hook the DHT11 module up to the Arduino.

Connections are fairly simple. Start by connecting + (VCC) pin to the 5V output on the Arduino and connect – (GND) to ground. Finally, connect the Out pin to the digital pin #8.

The following diagram shows you how to wire everything.



MQ-135:

Sensitive material of MQ135 gas sensor is SnO2, which with lower conductivity in clean air. When target pollution gas exists, the sensor’s conductivity gets higher along with the gas concentration rising. Users can convert the change of conductivity to correspond output signal of gas concentration through a simple circuit. MQ135 gas sensor has high sensitivity to ammonia gas, sulfide, benzene series steam, also can monitor smoke and other toxic gases well. It can detect kinds of toxic gases and is a kind of low-cost sensor for kinds of applications.

Sensitive material of MQ135 gas sensor is SnO2, which with lower conductivity in clean air. When the target combustible gas exist, The sensor’s conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration. MQ135 gas sensor has high sensitity to Ammonia, Sulfide and Benze steam, also sensitive to smoke and other harmful gases. It is with low cost and suitable for different application.



Sensitive material of MQ135 gas sensor is SnO2, which with lower conductivity in clean air. When thetarget combustible gas exist, The sensors conductivity is more higher along with the gas concentrationrising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gasconcentration. MQ135 gas sensor has high sensitityto Ammonia, Sulfide and Benze steam, alsosensitive to smoke and other harmful gases.It is with low cost and suitable for different application.Used for family,Surrounding environment noxious gas detection device, Apply to ammonia, aromatics, sulfur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentration range: 10 to1000ppm

Specification

•Working voltage: DC 5V

•Working Current: 150mA

•DOUT: TTL output

•AOUT: Analog output

•Preheat time: Over 20s

•Dimension: 32mm x 22m x 27mm(HIGH 27mm)

MQ135 Gas Sensor is an air quality sensor for detecting a wide range of gases, including NH3,NOx, alcohol, benzene, smoke and CO2. Ideal for use in office or factory. MQ135 gas sensor hashigh sensitivity to Ammonia, Sulfide and Benzene steam, also sensitive to smoke and otherharmful gases. It is with low cost & particularly suitable for Air quality monitoring application.

Features of MQ135 Gas Sensor:-

•High sensitivity to Ammonia, Sulfide and Benzene

•Stable and Long Life

•Detection Range: 10 - 300 ppm NH3, 10 - 1000 ppm Benzene, 10 - 300 ppm Alcohol

•Heater Voltage: 5.0V

•Dimensions: 18mm Diameter, 17mm High excluding pins, Pins - 6mm High

•Long life and low cost

Applications of MQ135 Gas Sensor:-

•Domestic air pollution detector

•Industrial air pollution detector

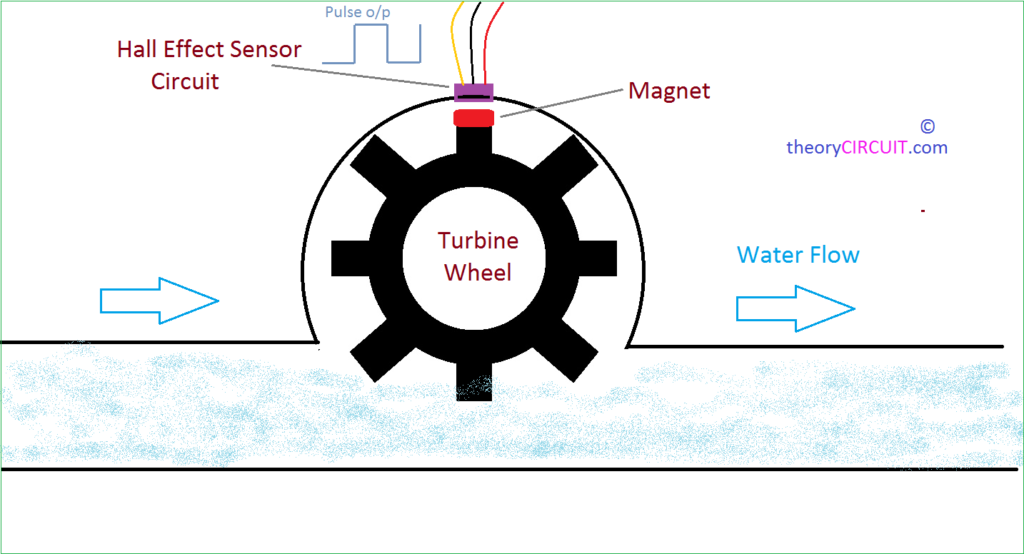
•Portable air pollution detect

FLOW SENSOR:

**Flowsensor**

It is Very simple to Measure the water or liquid flow by using water flow sensor YF-S201 with Arduino, this Article describes about the water flow sensor and How the water flow sensor works then how to interface water flow sensor with Arduino.  
  
  
  
  
To take control on volume we need to measure, water is essential to every thing, here this article helps you to built water flow meter to measure the volume of water flow through pipelines.

### Flow Sensor Working:

[](http://www.theorycircuit.com/wp-content/uploads/2017/11/how-water-flow-sensor-works.png)

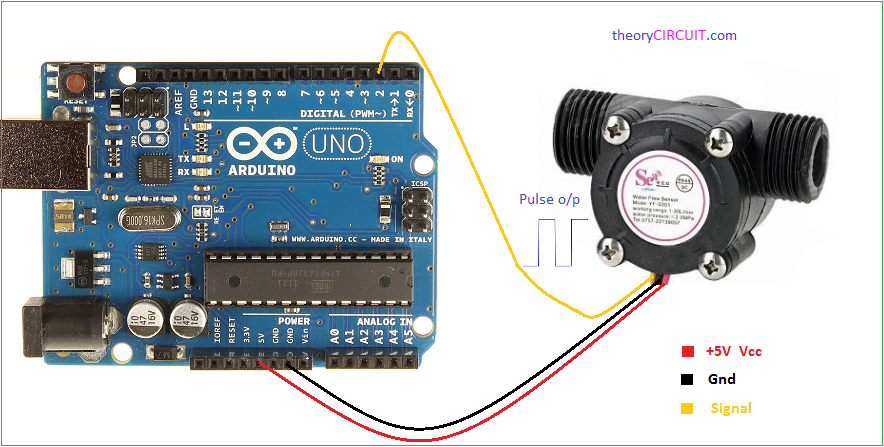
This illustration gives detailed working method of hall effect sensor based water flow sensor, a turbine wheel embed with magnet is placed on a closed plastic envelop and a Hall effect sensor placed, When the water flows through the pipeline, it makes the turbine wheel to rotate and hence the magnet flux interferes the hall sensor, the rate of interference is depends on the speed of water flow, so the hall effect sensor produce pulse signal output, this pulse output can be calculated as water volume.

## YF-S201 flow sensor

[](http://www.theorycircuit.com/wp-content/uploads/2017/11/water-flow-sensor-pin-configuration.png)

This water flow sensor has only three wires and it can be easily interfaced between any microcontroller and Arduino board. It requires only +5V Vcc and gives pulse output, the sensor needs to be tightly fitted between water pipeline.

**Arduino Hookup**

[](http://www.theorycircuit.com/wp-content/uploads/2017/11/arduino-water-flow-sensor-interface.png)

Connect the +5V wire to Arduino power pin 5V and Ground pin to Gnd then connect Signal pin to Digital pin D2, this sensor has control circuit hence there is no need for pull up resistor, some sensor requires pull up resistors refer datasheet of water flow sensor before concluding hookup.

POWER SUPPLY:

TRANSFORMER:

This document presents the solution for a 12V 1A flyback converter based on the Infineon OPTIREG™ TLE8386-2EL controller and IPD50N08S4-13 OptiMOS™-T2. The user is guided through the component selections, the circuit design and, finally, an overview of the experimental results are presented. The TLE8386-2EL is part of the Automotive OPTIREG™ family and it implements a low-side-sense current mode controller with built in protection features. The device is AECQ-100 qualified. The IPD50N08S4-13 is an AEC-Q101 qualified 80V N-channel enhanced mode MOSFET, it is part of the OptiMOS™-T2 family. Intended audience This document is intended for power supply design engineers, application engineers, students, etc., who need to design a Flyback converter for automotive power applications where a galvanic isolation between two voltage domains is required. In particular the focus is on a battery connected flyback that delivers up to 12W at 12V output voltage; the intention is to provide the user with all of the needed information to fully design and characterize the SMPS bringing it from an engineering concept to its production. Specific features and applications are: - 48V to 12V Automotive applications - Isolated current mode SMPS - Flyback regulators with auxiliary sensing

### **Centre Tapped Transformer Specifications**

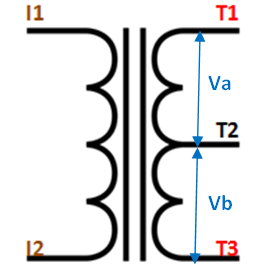
* Step-down Centre tapped Transformer
* Input Voltage: 220V AC at 50Hz
* Output Voltage: 24V, 12V or 0V
* Output Current: 1A
* Vertical mount type
* Low cost and small package

A **centre-tapped transformer** also known as **two phase three wire transformer** is normally used for rectifier circuits. When a digital project has to work with AC mains a Transformer is used to step-down the voltage (in our case, to 24V or 12V) and then convert it to DC by using a rectifier circuit. In a center-tapped transformer the peak inverse voltage is twice as in bridge rectifier hence this transformer is commonly used in full wave rectifier circuits.

The operation and theory behind a Center tapped transformer is very similar to a normal secondary transformer. A primary voltage will be induced in the primary coil (I1 and I3) and due to magnetic induction the voltage will be transferred to the secondary coil. Here in the secondary coil of a centre tapped transformer, there will be an additional wire (T2) which will be placed exactly at the center of the secondary coil, hence the voltage here will always be zero.

If we combine this zero potential wire (T2) with either T1 or T2, we will get a voltage of 12V AC. If this wire is ignored and voltage across T1 and T2 is considered then we will get a voltage of 24V AC. This feature is very useful for the function of a full wave rectifier.

Let us consider the voltage given by the first half of the secondary coil as Va and the voltage across the second half of the secondary coil as Vb as shown



RECTIFER CIRCUIT:

We have learnt in rectifier circuits about converting a sinusoidal ac voltage into its corresponding pulsating dc. Apart from the dc component, this pulsating dc voltage will have unwanted ac components like the components of its supply frequency along with its harmonics (together called ripples). These ripples will be the highest for a single-phase half wave rectifier and will reduce further for a single-phase full wave rectifier. The ripples will be minimum for 3-phase rectifier circuits. Such supply is not useful for driving complex electronic circuits. For most supply purposes constant dc voltage is required than the pulsating output of the rectifier. For most applications the supply from a rectifier will make the operation of the circuit poor. If the rectifier output is smoothened and steady and then passed on as the supply voltage, then the overall operation of the circuit becomes better. Thus, the output of the rectifier has to be passed though a filter circuit to filter the ac components. The filter is a device that allows passing the dc component of the load and blocks the ac component of the rectifier output. Thus the output of the filter circuit will be a steady dc voltage. The filter circuit can be constructed by the combination of components like capacitors, resistors, and inductors. Inductor is used for its property that it allows only dc components to pass and blocks ac signals. Capacitor is used so as to block the dc and allows ac to pass. All the combinations and their working are explained in detail below. Series Inductor Filter The circuit diagram of a full wave rectifier with a series inductor filter is given below. As the name of the filter circuit suggests, the Inductor L is connected in series between the rectifier circuit and the load. The inductor carries the property of opposing the change in current that flows through it. In other words, the inductor offers high impedance to the ripples and no impedance to the desired dc components. Thus the ripple components will be eliminated. When the rectifier output current increases above a certain value, energy is stored in it in the form of a magnetic field and this energy is given up when the output current falls below the average value. Thus all the sudden changes in current that occurs in the circuit will be smoothened by placing the inductor in series between the rectifier and the load. The waveform below shows the use of inductor in the circuit. From the circuit, for zero frequency dc voltage, the choke resistance Ri in series with the load resistance RL forms a voltage divider circuit, and thus the dc voltage across the load is Vdc = RL/(Ri + RL) Vdc is the output from a full wave rectifier. In this case, the value of Ri is negligibly small when compared to RL. The effect of higher harmonic voltages can be easily neglected as better filtering for the higher harmonic components take place. This is because of the fact that with the increase in frequency, the reactance of the inductor also increases. It should be noted that a decrease in the value of load resistance or an increase in the value of load current will decrease the amount of ripples in the circuit. So, the series inductor filter is mostly used in cases of high load current or small load resistance. A simple series inductor filter may not be properly used. It is always better to use a shunt capacitor (C) with series inductor (L) to form an LC Filter. Shunt Capacitor Filter As the name suggests, a capacitor is used as the filter and this high value capacitor is shunted or placed across the load impedance. This capacitor, when placed across a rectifier gets charged and stores the charged energy during the conduction period. When the rectifier is not conducting, this energy charged by the capacitor is delivered back to the load. Through this energy storage and delivery process, the time duration during which the current flows through the load resistor gets increased and the ripples are decreased by a great amount. Thus for the ripple component with a frequency of ‘f’ megahertz, the capacitor ‘C’ will offer a very low impedance. The value of this impedance can be written as: Shunt Capacitor Impedance = 1/2 fC Thus the dc components of the input signal along with the few residual ripple components, is only allowed to go through the load resistance RLoad. The high amount of ripple components of current gets bypassed through the capacitor C. Now let us look at the working of Half-wave rectifier and Full-wave rectifier with Capacitor filters, their output filtered waveform, ripple factor, merits and demerits in detail.