1. **INTRODUCTION**

Adequate fertilizer can help plants produce good yields and quantities, to meet the needs of a world that is increasingly rising in need of food and food production. To improve the quality and quantity of crops, measurement of N (nitrogen), P (phosphorous) and K (potassium) contents of soil is necessary to determine how much fertilizer is needed to add to the soil to increase crop fertility. This improves the quality of the soil which in turn yields a good quality crop. It also reduces the addition of undesired fertilizers of soil. The pH sensor developed can identify the levels of NPK in the soil and then accordingly required fertilizers can be added to the soil. The sensor is thus developed to detect the deficiency of the nutrients N, P or K in the soil. The sensor is fabricated which has concentric arrangement of source and receiving fibres. These three elements nutrients promote the growth of the plant in different ways -Nitrogen promotes the growth of leaves and vegetation, Phosphorus promotes root and growth and Potassium promotes flowering, fruiting and keeps regulation of nutrient and water in plant cell.

Soil can be acid, neutral or alkaline, according to its pH value. Most plants prefer a pH range from 5.5 to 7.5; but some species prefer more acid or alkaline soils. Nevertheless, every plant requires a particular range of pH, for optimum growth.

pH strongly influences the availability of nutrients and the presence of microorganisms and plants in the soil. For example, fungi prefer acidic conditions whereas most bacteria, especially those supplying nutrients to the plants, have a preference for moderately acidic or slightly alkaline soils. In fact, in strongly acidic conditions, nitrogen fixing and the mineralization of vegetable residual is reduced. Plants absorb the nutrients dissolved in the soil water and the nutrient solubility depends largely on the pH value. Hence, the Soil can be acid, neutral or alkaline, according to its pH value. Most plants prefer a pH range from 5.5 to 7.5; but some species prefer more acid or alkaline soils. Nevertheless, every plant requires a particular range of pH, for optimum growth.

pH strongly influences the availability of nutrients and the presence of microorganisms and plants in the soil. For example, fungi prefer acidic conditions whereas most bacteria, especially those supplying nutrients to the plants, have a preference for moderately acidic or slightly alkaline soils. In fact, in strongly acidic conditions, nitrogen fixing and the mineralization of vegetable residual is reduced. Plants absorb the nutrients dissolved in the soil water and the nutrient solubility depends largely on the pH value. Hence, theavailability of elements is different at different pH levels.

Each plant needs elements in different quantities and this is the reason why each plant requires a particular range of pH to optimize its growth. For example, iron, copper and manganese are not soluble in an alkaline environment. This means that plants needing these elements should theoretically be in an acidic type of soil. Nitrogen, phosphorus, potassium and sulfur, on the other hand, are readily available in a pH range close to neutrality. Furthermore, abnormal pH values, increase the concentration of toxic elements for plants. For example, in acid conditions, there can be an excess of aluminum ions in such quantities that the plant cannot tolerate. Negative effects on chemical and physical structure are also present when pH values are too far from neutral conditions (break up of aggregates, a less permeable and more compact soil).

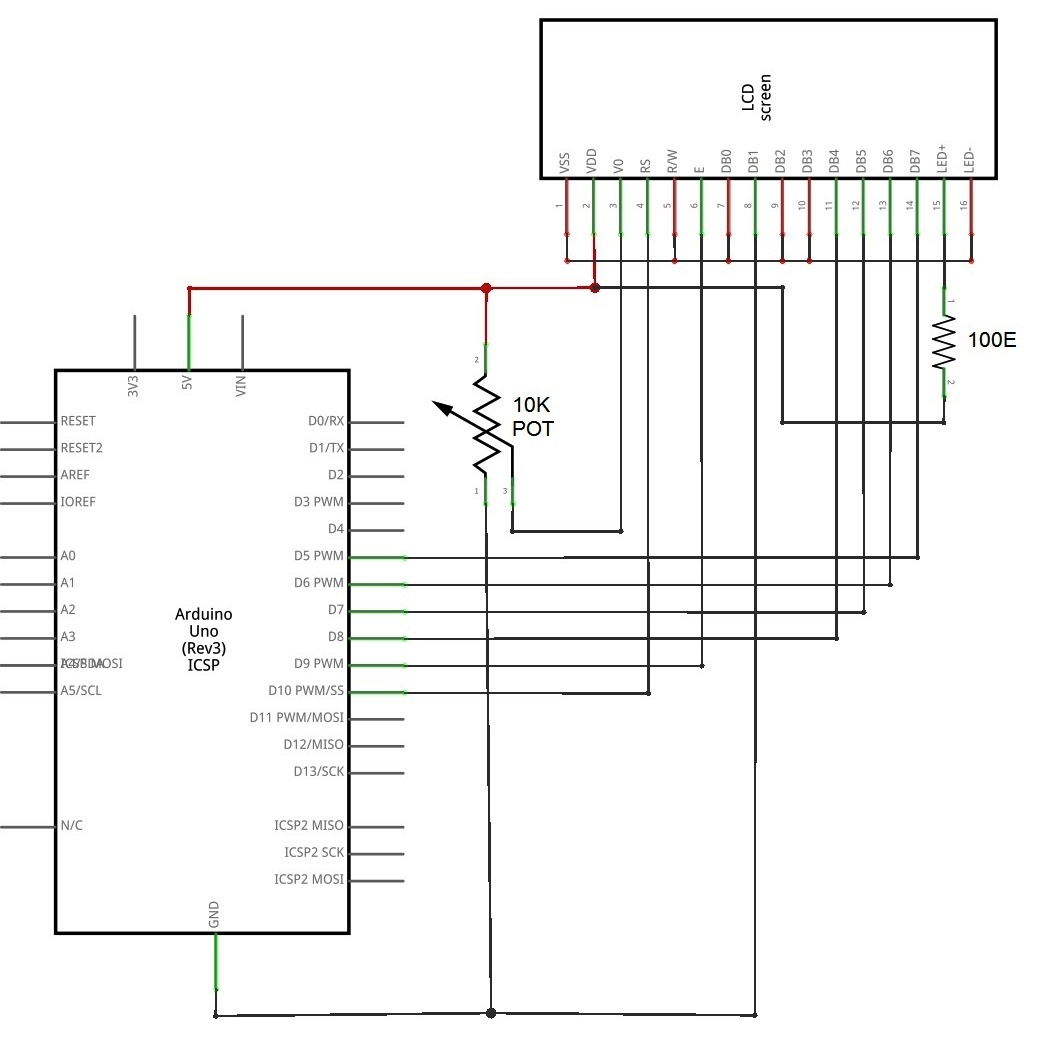
Each plant needs elements in different quantities and this is the reason why each plant requires a particular range of pH to optimize its growth. For example, iron, copper and manganese are not soluble in an alkaline environment. This means that plants needing these elements should theoretically be in an acidic type of soil. Nitrogen, phosphorus, potassium and sulfur, on the other hand, are readily available in a pH range close to neutrality. Furthermore, abnormal pH values, increase the concentration of toxic elements for plants. For example, in acid conditions, there can be an excess of aluminum ions in such quantities that the plant cannot tolerate. Negative effects on chemical and physical structure are also present when pH values are too far from neutral conditions (break up of aggregates, a less permeable and more compact soil).

* 1. **DETAILED PROBLEM DEFINITION**

To identify the levels of NPK nutrients of soil using pH of the soil. This project can reduce the problems in determining the amount of nutrients in soil with a cheaper cost with other technology. It can also reduce the undesired use of fertilizers to be added to the soil which can cause dead plants and reduce plant quality and quantity.

In India, where the economy is mainly base on agriculture and the climatic conditions are isotropic and are not able to make full use of agricultural resources. The main reason is the lack of rains and scarifies of land reservoir water and overused of fertilizers so we need to control these parameters. This system detected NPK levels in soil using pH sensors. By using pH sensors we get the information about the soil and analyse the acid level of the soil. By which we can apply fertilizer to the place where it needs, also we can avoid over fertilization of the crops. We used humidity sensor to sense the weather. By this the farmer can get idea about the climate. If there is any chance for rainfall; the farmer need not irrigate the crop field.

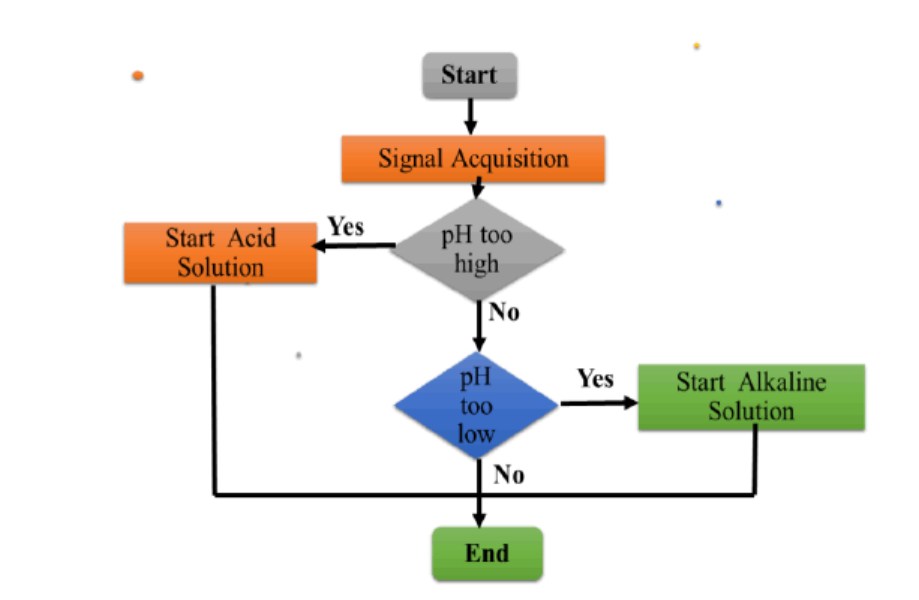
* 1. **BLOCK DIAGRAM**

****

**Fig 1.1. Block Diagram**

1. **SYSTEM ANALYSIS**

* 1. **SYSTEM ARCHITECTURE**



**Fig 2.1 Flowchart of detecting pH value**

* 1. **FUNCTIONAL REQUIREMENTS**

Initially the system starts and pH sensor electrode measures the pH value of the water of the soil . The water of the soil is being put in the container of the sensor. The pH value of the water of the particular soil is measured and the results are to be displayed on the display board. Through the results displayed, one can detect the Nitrogen, Potassium and Phosphorous (NPK) level in the soil and thus add required fertilizers to the soil to increase the efficiency of the soil.

* 1. **REQUIREMENT SPECIFICATION**

Some of the requirements of the system are-

* + - Arduino Uno Board
    - ph Sensor Board
    - ph sensor electrode
    - Water level sensor (Probs)
    - Serial Monitor
  1. **SCHEDULE AND ESTIMATION**

The total cost of the project is Rs.2000/-. The project was started in the month of Novemberand planning to be done by end of February.

Schedule for the project:-

Week 1: Synopsis details.

Hardware details.

Circuit diagram.

Week 2: Detailed problem analysis and literature survey.

Hardware details.

Hardware tools specification.

Week 3: The working of design.

Components requirement study.

Discuss about the Connection.

Week 4: User interface code.

Week 5: Components for soldering.

Design platform.

Soldering.

Week 6: Serial input/output

Paper work.

Report. **Week 7:** Testing of user interface code.

Design report.

**Week 8:** Validation and testing.

Draft report.

**Week 9:** Enhancement of design and serial interface.

Complete working project.

1. **SYSTEM DESIGN**

**3.1 MODULAR SPECIFICATION**

**Module 1 – Collecting different samples of the soil and extracting water out of it**

We collected different soil samples and extracted water out of it so that we can test the pH of the water of all different types of soil collected.

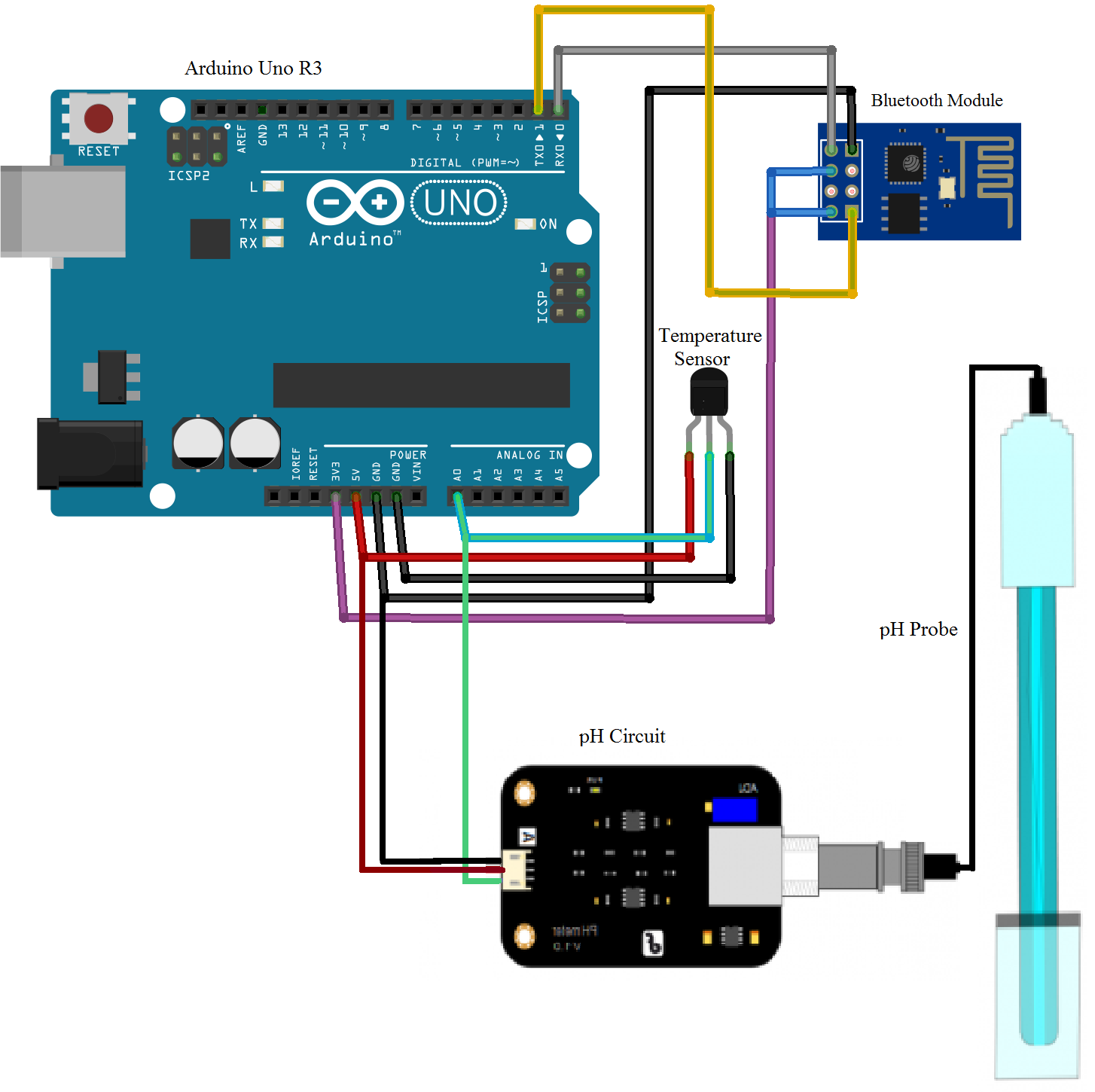
**Module 2 – Determining the pH value of the soil**

This module is determines the pH value of the soil by using pH sensor electrode.

**Module 3 – Determining NPK value of the soil**

This module is determines the NPK value of the soil by using the pH values determined..

**3.2 HARDWARE CIRCUIT DESIGN**

****

**Fig 3.1. Circuit Design**

* 1. **COMPONENT DESCRIPTION**

1. **Arduino Uno Board**-

The Arduino UNO is an open-source microcontroller board based on the [Microchip](https://en.wikipedia.org/wiki/Microchip_Technology) [ATmega328P](https://en.wikipedia.org/wiki/ATmega328P) microcontroller and developed by [Arduino.cc](https://en.wikipedia.org/wiki/Arduino). The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.[[1]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-Makerspace-1) The board has 14 Digital pins, 6 Analog pins, and programmable with the [Arduino IDE](https://en.wikipedia.org/wiki/Arduino#Software) (Integrated Development Environment) via a type B USB cable.[[4]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-priceton-4) It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

General pin functions of Arduino Uno are-

* + - **LED**: There is a built-in LED driven by digital pin 13. When the pin is

HIGH value, the LED is on, when the pin is LOW, it's off.

* + - * **VIN**: The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
      * **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
      * **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
      * **GND**: Ground pins.
      * **IOREF**: This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
    - **Reset**: Typically used to add a reset button to shields which block the one on the board.

Special pin functions of Arduino Uno-

Each of the 14 digital pins and 6 Analog pins on the Uno can be used as an input or output, using pinMode(),digitalWrite(), and digitalRead() functions2. They operate at 5 volts. Each pin can provide or receive 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50k ohm. A maximum of 40mA is the value that must not be exceeded on any I/O pin to avoid permanent damage to the microcontroller. The Uno has 6 analog inputs, label led A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.[[7]](https://en.wikipedia.org/wiki/Arduino_Uno#cite_note-website-7)

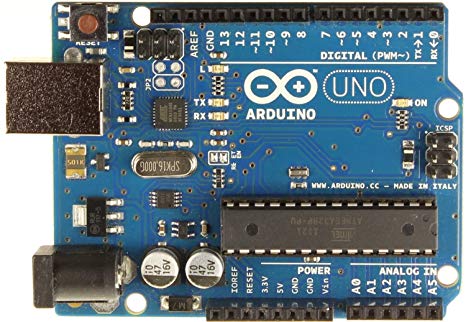
In addition, some pins have specialized functions:

* **Serial** / [**UART**](https://en.wikipedia.org/wiki/UART): pins 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
* **External Interrupts**: pins 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
* **PWM** (**P**ulse **W**idth **M**odulation): 3, 5, 6, 9, 10, and 11 Can provide 8-bit PWM output with the analogWrite() function.
* **SPI** (**S**erial **P**eripheral **I**nterface): 10 (SS), 11 (MOSI), 12 (MISO), 13

(SCK). These pins support SPI communication using the SPI library.

* **TWI** (**T**wo **W**ire **I**nterface) / [I²C](https://en.wikipedia.org/wiki/I%C2%B2C): A4 or SDA pin and A5 or SCL pin.

* **AREF** (**A**nalog **REF**erence): Reference voltage for the analog inputs.



**Fig. Arduino Uno Board**

1. **ph Sensor Board-**

The analog pH sensor, specially designed for Arduino controllers is easy to use and can be used as a plug and play solution to measure pH value of a solution without any additional circuit required.

It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface.

To use it, we just connect the pH sensor with BNC connector, and plug the PH2.0 interface into the analog input port of any Arduino controller. With a simple program to ready analog voltage, and we will get the pH value easily.

**Working of pH Sensor**

The pH is a measure of the acidity or alkalinity of a solution, the pH scale varies from 0 to 14.

The pH indicates the concentration of hydrogen ions [H]+ present in certain solutions.

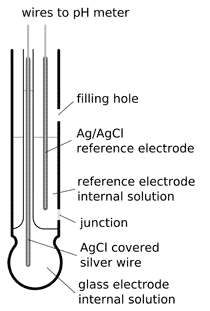
It can be quantified accurately using a sensor that measures the difference of potential between two electrodes: a reference electrode (silver/silver chloride) and a glass electrode is sensitive to hydrogen ion. This is what will form the probe. In addition there are that use an electronic circuit to condition the signal appropriately, and that we can use this sensor with a microcontroller.



**Fig. ph Sensor Board**

1. **ph Sensor Electrode-**

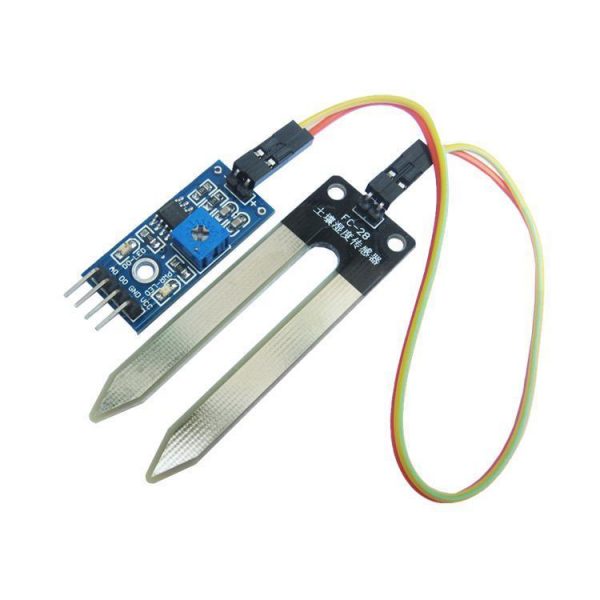
pH electrodes are constructed from a special composition glass which senses the hydrogen ion concentration.  This glass is typically composed of alkali metal ions that undergo an ion exchange reaction with the hydrogen ions in the test solution to generate a potential difference.  The bulb is filled with an acid solution (e.g. 0.1 molL-1 HCl).



**Fig. ph Sensor Electrode**

1. **Water Level Sensor (Probes)-**

A  water level probe is a specific type of level sensor used to measure the level or vertical height water in a water reservoir, tank or vessel. ... Thus, using a water  level probe, the water level of e.g. an underground reservoir can directly be monitored by measuring the pressure above any convenient level measurement point.



**Fig. Water Level Sensor (Probes)**

1. **Serial Monitor-**

The serial monitor is the 'tether' between the computer and your Arduino - it lets you send and receive text messages, handy for debugging and also controlling the Arduino from a keyboard! For example, you will be able to send commands from your computer to turn on LEDs.

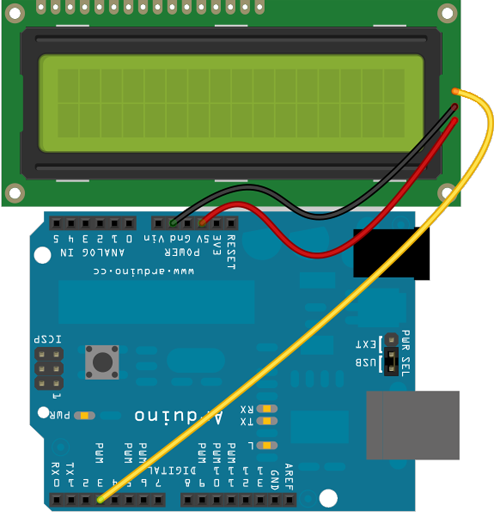


Fig. Serial Monitor

1. **Jumper Wires-**

A **jump wire** (also known as jumper wire, or jumper) is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.[[1]](https://en.wikipedia.org/wiki/Jump_wire#cite_note-1)

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

There are different types of jumper wires. Some have the same type of electrical connector at both ends, while others have different connectors. Some common connectors are:

* Solid tips – are used to connect on/with a breadboard or female header connector. The arrangement of the elements and ease of insertion on a breadboard allows increasing the mounting density of both components and jump wires without fear of short-circuits. The jump wires vary in size and colour to distinguish the different working signals.
* Crocodile clips – are used, among other applications, to temporarily bridge sensors, buttons and other elements of prototypes with components or equipment that have arbitrary connectors, wires, screw terminals, etc.
* Banana connectors – are commonly used on test equipment for DC and low-frequency AC signals.
* Registered jack (RJnn) – are commonly used in telephone (RJ11) and computer networking (RJ45).
* RCA connectors – are often used for audio, low-resolution composite video signals, or other low-frequency applications requiring a shielded cable.
* RF connectors – are used to carry radio frequency signals between circuits, test equipment, and antennas.



Fig . Jumper Wires

**3.4 TEST PLAN**

The objectives of testing includes-

* To make sure that the pH sensor electrode is working fine and the pH values of the soil are getting determined.
* To determine the correct pH values of the soil as the determination of the NPK levels in the soil depends on the levels of the pH value of the soil.
* To ensure that pH value of the soil lies between the range of 0-14 as the standard scale of pH is from 0-14.

1. **IMPLEMENTATION**
   1. **TOOL/SOFTWARE DESCRIPTION**

[Arduino](http://arduino.cc/) is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a [microcontroller](http://en.wikipedia.org/wiki/Microcontroller)) and a piece of [software](http://arduino.cc/en/Main/Software), or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

A program for Arduino hardware may be written in any [programming language](https://en.wikipedia.org/wiki/Programming_language) with compilers that produce binary machine code for the target processor. Atmel provides a development environment for their 8-bit [AVR](https://en.wikipedia.org/wiki/Atmel_AVR) and 32-bit [ARM Cortex-M](https://en.wikipedia.org/wiki/ARM_Cortex-M) based microcontrollers: AVR Studio (older) and Atmel Studio (newer).

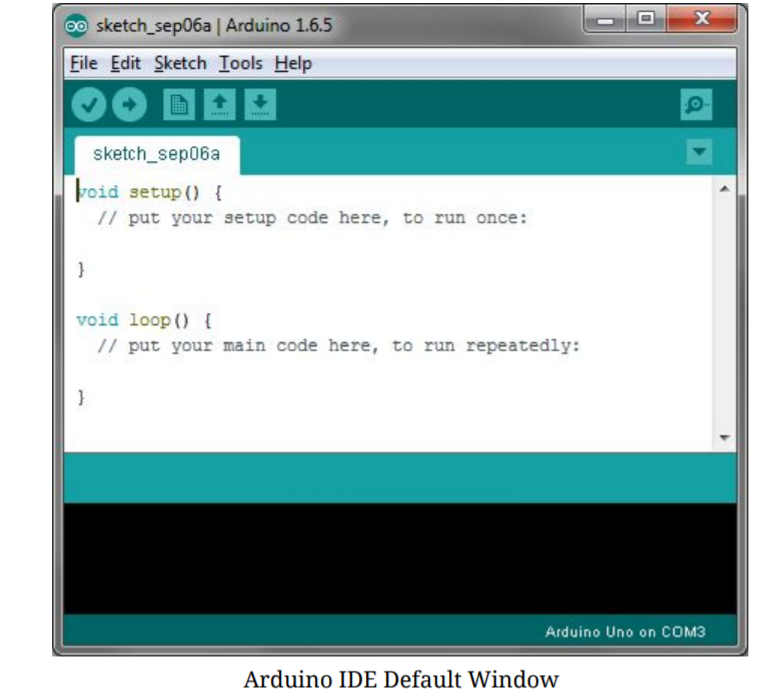
**IDE**

The Arduino [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) application (for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), [Linux](https://en.wikipedia.org/wiki/Linux)) that is written in the programming language [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It originated from the IDE for the languages *[Processing](https://en.wikipedia.org/wiki/Processing_(programming_language)" \o "Processing (programming language))*and [*Wiring*](https://en.wikipedia.org/wiki/Wiring_(development_platform)). It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, [brace matching](https://en.wikipedia.org/wiki/Brace_matching), and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting), and provides simple *one-click* mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License), version 2.[[56]](https://en.wikipedia.org/wiki/Arduino#cite_note-59)

The Arduino IDE supports the languages [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) using special rules of code structuring. The Arduino IDE supplies a [software library](https://en.wikipedia.org/wiki/Software_library) from the [Wiring](https://en.wikipedia.org/wiki/Wiring_(development_platform)) project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable [cyclic executive](https://en.wikipedia.org/wiki/Cyclic_executive) program with the [GNU toolchain](https://en.wikipedia.org/wiki/GNU_toolchain), also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

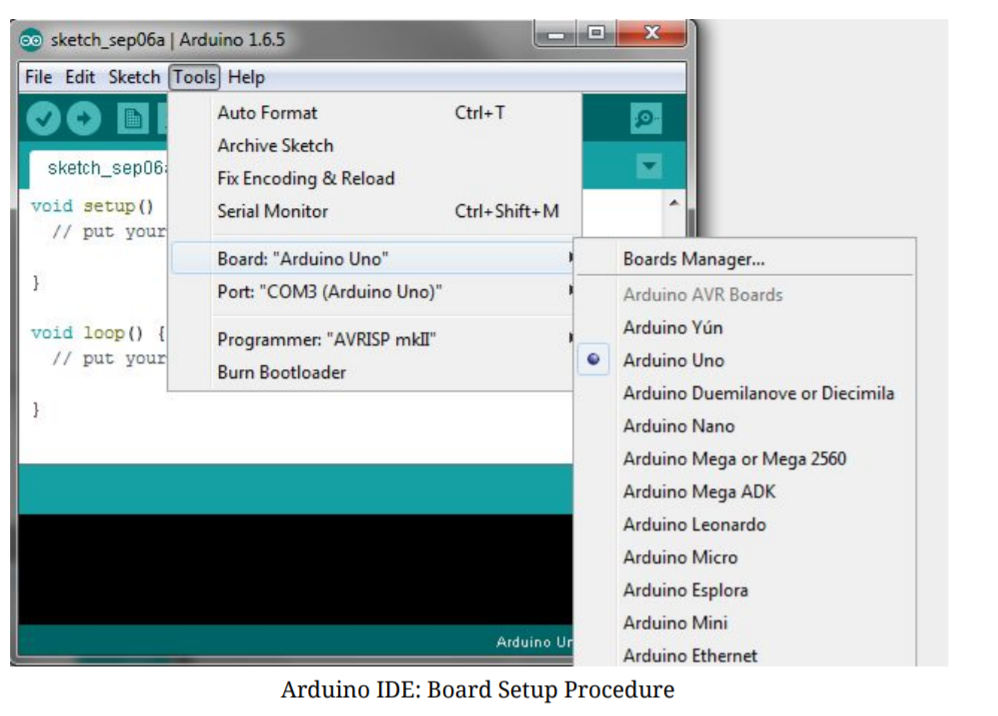
**Arduino IDE: Initial Setup**

Download Arduino Integrated Design Environment (IDE) here (Most recent version: 1.6.5): https://www.arduino.cc/en/Main/Software This is the Arduino IDE once it’s been opened. It opens into a blank sketch where you can start programming immediately. First, we should configure the board and port settings to allow us to upload code. Connect your Arduino board to the PC via the USB cable.



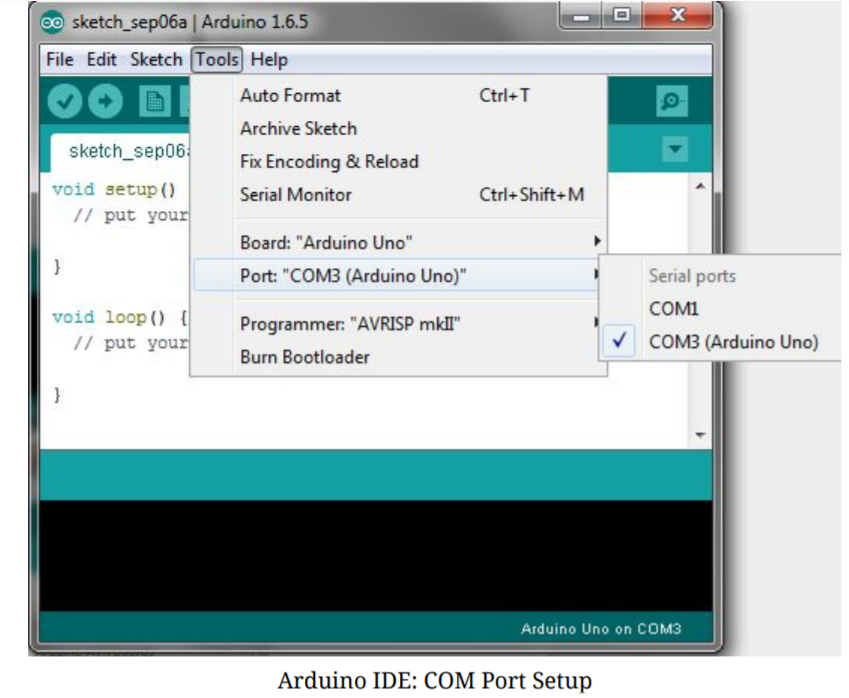
**IDE: Board Setup**

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



**IDE: COM Port Setup**

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

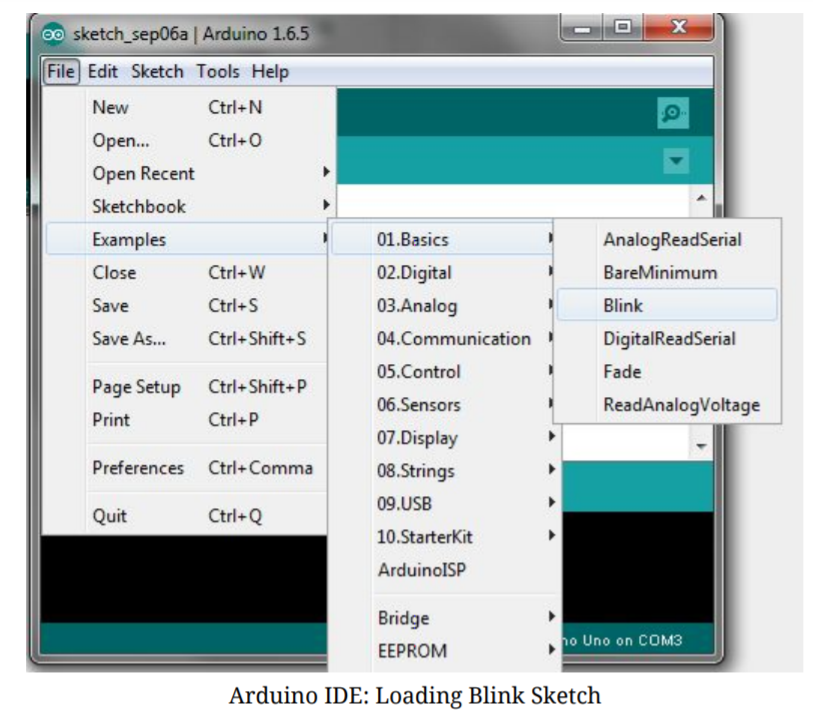
****

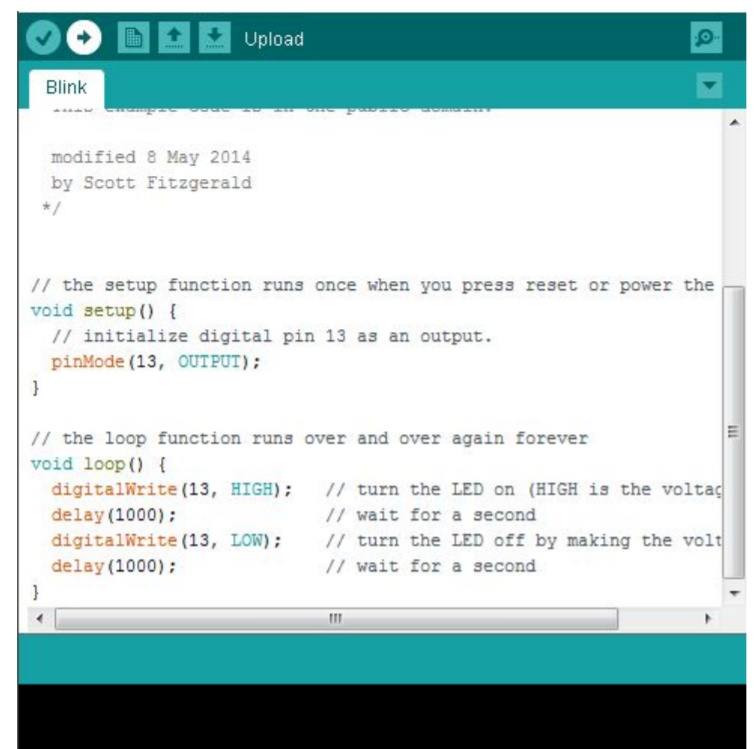
At this point, your board should be set up for programming, and you can begin writing and uploading code.

**Testing Your Settings: Uploading Blink**

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



****



**SKETCH**

A *sketch* is a program written with the Arduino IDE. Sketches are saved on the development computer as text files with the file extension **.ino**. Arduino Software (IDE) pre-1.0 saved sketches with the extension **.pde**.

A minimal Arduino C/C++ program consist of only two functions:

**setup( ):** This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

**loop( ):** After **setup()**  function exits (ends), the **loop( )** function is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

Most Arduino boards contain a [light-emitting diode](https://en.wikipedia.org/wiki/Light-emitting_diode) (LED) and a current limiting resistor connected between pin 13 and ground, which is a convenient feature for many tests and program functions.[[61]](https://en.wikipedia.org/wiki/Arduino#cite_note-Blink_Tutorial-64) A typical program used by beginners, akin to [Hello, World!](https://en.wikipedia.org/wiki/Hello,_World!), is "blink", which repeatedly blinks the on-board LED integrated into the Arduino board. This program uses the functions pinMode( ), digitalWrite()and delay(), which are provided by the internal libraries included in the IDE environment. This program is usually loaded into a new Arduino board by the manufacturer.

#define LED\_PIN 13 *// Pin number attached to LED.*

**void** setup() {

pinMode(LED\_PIN, **OUTPUT**); *// Configure pin 13 to be a digital output.*

}

**void** loop() {

digitalWrite(LED\_PIN, **HIGH**); *// Turn on the LED.*

delay(1000); *// Wait 1 second (1000 milliseconds).*

digitalWrite(LED\_PIN, **LOW**); *// Turn off the LED.*

delay(1000); *// Wait 1 second.*

}

### Libraries

The open-source nature of the Arduino project has facilitated the publication of many free software libraries that other developers use to augment their projects.

* 1. **SAMPLE SOURCE CODE**

**#define SensorPin A0 //pH meter Analog output to Arduino Analog Input 0**

**static unsigned long samplingTime = millis();**

**#define Offset 0.00 //deviation compensate**

**#define LED 13**

**#define samplingInterval 20**

**#define printInterval 800**

**#define ArrayLenth 40 //times of collection**

**int pHArray[ArrayLenth]; //Store the average value of the sensor feedback**

**int pHArrayIndex = 0;**

**void setup() {**

**pinMode(LED, OUTPUT);**

**Serial.begin(9600);**

**Serial.println("pH meter experiment!"); //Test the serial monitor**

**}**

**void loop() {**

**static unsigned long printTime = millis();**

**static float pHValue, voltage;**

**if (millis() - samplingTime > samplingInterval) {**

**pHArray[pHArrayIndex++] = analogRead(SensorPin);**

**if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;**

**voltage = avergearray(pHArray, ArrayLenth) \* 5.0 / 1024;**

**pHValue = 3.5 \* voltage + Offset;**

**samplingTime = millis();**

**}**

**if (millis() - printTime > printInterval) {**

**Serial.print("Voltage:");**

**Serial.print(voltage, 2);**

**Serial.print(" pH value: ");**

**Serial.println(pHValue, 2);**

**digitalWrite(LED, digitalRead(LED) ^ 1);**

**printTime = millis();**

**}**

**}**

**double avergearray(int\* arr, int number) {**

**int i;**

**int max, min;**

**double avg;**

**long amount = 0;**

**if (number <= 0) {**

**Serial.println("Error number for the array to avraging!/n");**

**return 0;**

**}**

**if (number < 5) { //less than 5, calculated directly statistics**

**for (i = 0; i < number; i++) {**

**amount += arr[i];**

**}**

**avg = amount / number;**

**return avg;**

**}**

**else {**

**if (arr[0] < arr[1]) {**

**min = arr[0]; max = arr[1];**

**}**

**else {**

**min = arr[1]; max = arr[0];**

**}**

**for (i = 2; i < number; i++) {**

**if (arr[i] < min) {**

**amount += min; //arr<min**

**min = arr[i];**

**}**

**else {**

**if (arr[i] > max) {**

**amount += max; //arr>max**

**max = arr[i];**

**}**

**else {**

**amount += arr[i]; //min<=arr<=max**

**}**

**}**

**}**

**avg = (double)amount / (number - 2);**

**}**

**return avg;**

**}**

* 1. **SCREENSHOTS**

1. **TESTING**
   1. **TESTING ENVIRONMENT**

A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured.

Test bed or test environment is configured as per the need of the Application Under Test. On a few occasion, test bed could be the combination of the test environment and the test data it operates.

Setting up a right test environment ensures software testing

success. Any flaws in this process may lead to extra cost and

time to the client.

Test Environment consists of elements that support test execution with software, hardware and network configured. Test environment configuration must mimic the production environment in order to uncover any environment/configuration related issues.

**Factors for designing Test Environment**

* Determine if test environment needs archiving in order to take back-ups.
* Verify the network configuration.
* Identify the required server operating system, databases and other components.
* Identify the number of license required by the test team.

The project is being tested for various soil conditions. Soil is a living community that requires a balance of components. Soil has microorganisms, nutrients, minerals, water and oxygen. All of these elements contribute to healthy plant growth. Good soil means well-fed plants that bloom more abundantly and produce bigger and longer-lasting flowers. Hence the quality of soil is being tested for conditions-

* The area from where soil belongs to.
* The pH level of the soil from which NPK levels can be determined

We should take some factors in consideration-

* The nutrients that are required by crops in the largest amounts are N, P and K. For that reason, they are often considered as the most important nutrients.
* The main functions of N and P are that they are constituents of proteins and nucleic acids, which are important components of plant tissue. K is the only nutrient that is not a constituent of organic plant compounds, but is mainly of importance in the regulation of processes in the plant, such as osmosis and enzyme activities. K is generally playing an important role for the quality of harvested plant products.
* For the optimal growth of crops, sufficient amounts of nutrients should be available in the root zone of the crops. Those nutrients can be partly supplied by the soil and should be partly added with organic manures and fertilizers. Soils will contain different amounts of available nutrients, depending of the parent material (e.g. sand, clay, peat), and differences in the management history such as preceding crops, management of crop residues and use of manure and fertilizers in the past. Also differences in climatic conditions may alter the available nutrients. For that reason, it is of importance for farmers to know the NPK content of their soil, so that they know how much N, P and/or K they should add with organic or mineral fertilizers, to optimize crop growth, production and yield.
* Nutrients are present in the soil in different forms, which differ in its availability for plants. For example, most nitrogen is present in the soil in organic form as part of organic matter, while it can be taken up only in mineral forms (ammonium and nitrate). The organic nitrogen should be mineralized into mineral forms before plant roots can take it up. Phosphorus in the soil is also present in organic matter, but often mainly in chemical forms, which differ in solubility and plant availability. Potassium is mainly present in the soil solution and adsorbed to soil particles, such as clay and organic matter, from which it can desorb relatively easily by changes in equilibrium between the surface of soil particles and the soil solution.
* The crop roots take up the available nutrients from the top layer of the soil. Despite differences in plant root systems, which vary from shallow (e.g. grass) to deep (most tree crops, sugar beet, maize, cereals), all crops take up their nutrients from the top soil.
* The mobility of nutrients in soils strongly differs: N and K dissolve in water quite well and are very mobile in soil, while P is rather immobile in soil. The consequence is that the supply of N and K to plant roots is mostly sufficient, provided that the amount in soil is high enough, while the P supply to plant roots, especially in the first stages after sowing or planting, may be difficult.
* The availability of N, P and K in soil should be sufficient, but not too high. Too low availabilities will lead to hampered growth and low yields, while too high availabilities of one or more nutrients may lead to disturbed plant growth and adverse effects for yield and/or quality of harvested products. Moreover, the N, P and K availability should be balanced, so the availability of the other nutrients should be taken into account while the availability of the considered nutrient is adjusted.
  1. **TEST PLAN**

Test Plan is a dynamic document. The success of a testing project depends upon a well-written test plan document that is current at all times. Test Plan is more or less like a blueprint of how the testing activity is going to take place in a project.

Given below are few pointers on a Test Plan:

* + - Test Plan is a document that acts as a point of reference and only based on that testing is carried out within the QA team.
    - It is also a document that we share with the Business Analysts, Project Managers, Dev team and the other teams. This helps to enhance the level of transparency of the QA team’s work to the external teams.
    - It is documented by the QA manager/QA lead based on the inputs from the QA team members.
    - Test Planning is typically allocated with 1/3rd of the time that takes for the entire QA engagement.  The other 1/3rd is for Test Designing and the rest is for Test Execution.
    - This plan is not static and is updated on an on-demand basis.
    - The more detailed and comprehensive the plan is, the more successful will be the testing activity.

The test plan is that the project should be checked for all types of soils. The pH level determined from the soil can be used to determine NPK level of the soil. By determining the NPK level of the soil, we can actually understand the fertility of the soil and accordingly crops can be grown on the field. The crops grown on the field basically depends on the type of soil and the weather conditions best suited for the crop to be grown.

* 1. **TEST CASES AND RESULTS**

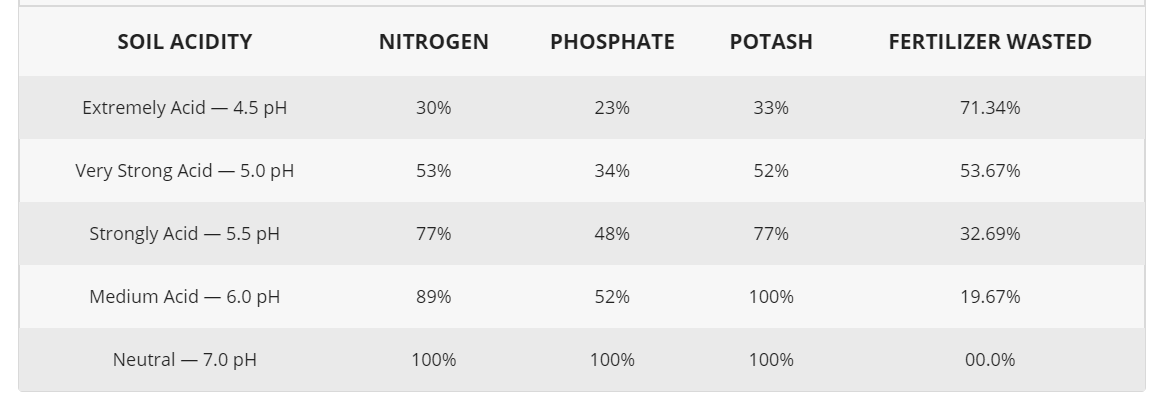
A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

## **Typical Test Case Parameters:**

* Test Case ID
* Test Scenario
* Test Case Description
* Test Steps
* Prerequisite
* Test Data
* Expected Result
* Test Parameters
* Actual Result
* Environment Information
* Comments

The tests when performed on different types of soil depicted different results. Basically, we cannot predict the pH level of soil as soil differs from region to region. But most commonly, it is found as-



1. **CONCLUSIONS and FUTURE SCOPE**

Determining the nutrient concentrations for nitrate, phosphates, and potassium can reveal how a soil is functioning in regards to its intended use and how nutrients are cycling through a soil. A nutrient test provides a report of average nutrient concentration (mg/L) for all nutrients tested.  In an agricultural setting, knowing the concentration of nutrients can help food producers know when to add fertilizer, how much to add, and which nutrients need supplemented and in what amount. Consistently high nitrogen soils, for instance, would be good for growing nitrogen-demanding crops such as soy and corn. High nitrogen levels are also particularly useful for non-flowering plants because nitrogen is required for any green part of plants. High nitrogen levels can suppress flowering however, if they remain higher than phosphorus levels. Phosphorus controls flowering in plants and is important to any plant production involving flowering or fruiting plants and phosphorus is often added to soils or directly to plants before and during flowering and fruiting life-cycle stages to increase agricultural yields in larger crop size and increased amounts of fruit production per plant. Potassium is involved in catalyzing many chemical reactions required to support plant life including drought tolerance and moisture regulation. Low potassium soils will likely need to be irrigated if soil amendment is not possible. Nutrient concentration can also inform of nutrient deficiencies or surpluses that can be detrimental to plant growth.  If a nutrient is too high, amendments can be performed to reduce a surplus, such as adding mulch or tilling the soil. If nutrients are too low to support plant production, fertilization can be used to add nutrients in an amount needed for a specific crop. Low nutrient soil may also have more applicable uses to land managers for recreational or developed (paved surfaces or building construction) spaces.

**BIBLIOGRAPHY**