**Smart Agriculture: Crop Recommendation System with Python and Embedded Technology**

**Abstract:**

The increasing demand for optimized agricultural practices has driven the development of intelligent systems that can recommend suitable crops based on real-time soil data. This project presents a crop suggestion system that integrates embedded systems and machine learning for enhanced decision-making. Using an ESP32 microcontroller, the system gathers soil data from multiple sensors: a soil moisture sensor to measure the water content, a pH sensor to determine soil acidity or alkalinity, and an NPK sensor to assess nutrient levels (Nitrogen, Phosphorus, and Potassium). Data from these sensors are displayed in real-time using an I2C-enabled display and uploaded to a cloud platform for further analysis. A machine learning algorithm processes the sensor data to provide accurate crop recommendations based on the specific conditions of the soil. This approach enables farmers to make informed decisions, increasing crop yield and improving agricultural sustainability.

**Introduction:**

In modern agriculture, optimizing crop selection based on soil conditions is crucial for improving productivity and sustainability. This project introduces a crop suggestion system that combines embedded technology with machine learning to offer precise recommendations tailored to specific soil conditions. The system uses an ESP32 microcontroller to interface with various sensors, including a soil moisture sensor to monitor water content, a pH sensor to assess the soil's acidity or alkalinity, and an NPK sensor to measure key nutrient levels—Nitrogen, Phosphorus, and Potassium. These sensors provide vital soil data, which is displayed in real-time using an I2C interface and then transmitted to a machine learning model for further analysis. By processing this data, the system suggests crops that are best suited to the current soil conditions, helping farmers make informed decisions and improve their crop yields. This smart, data-driven approach to farming enhances resource efficiency, promoting sustainable agricultural practices.

**Literature Survey:**

1. **Soil-Based Crop Recommendation Systems**

In recent years, advancements in soil-based crop recommendation systems have significantly contributed to precision agriculture. Researchers have developed systems that use soil properties, such as moisture, pH, and nutrient levels, to suggest suitable crops. For instance, Subash et al. (2020) designed a crop recommendation system that utilized soil pH and moisture content to provide crop suggestions. By analyzing historical data and using machine learning models, their system helped farmers make better decisions, thus increasing agricultural productivity. This approach demonstrated the importance of real-time soil data for improving crop selection accuracy.

1. **Sensor Integration in Smart Agriculture**

The integration of sensors with microcontrollers has revolutionized smart agriculture by enabling the real-time monitoring of essential soil parameters. Mahajan et al. (2019) implemented a system using soil moisture and NPK sensors to gather vital data on soil conditions. Their research highlighted the importance of accurate soil data in optimizing irrigation and fertilization practices. The system, which employed Arduino and GSM for data transmission, emphasized the potential of IoT-based agriculture systems in minimizing resource wastage while maximizing crop yields.

1. **Machine Learning for Agricultural Decision-Making**

The application of machine learning in agriculture has opened new avenues for data-driven decision-making. Patel et al. (2021) proposed a crop recommendation system using supervised machine learning models trained on soil data, including pH, temperature, and nutrient content. Their study demonstrated that machine learning algorithms, such as Random Forest and Support Vector Machine (SVM), could predict the most suitable crops for specific soil conditions with high accuracy. This highlights the role of machine learning in advancing precision farming by automating crop selection processes.

1. **IoT in Agriculture for Remote Monitoring and Data Analytics**

IoT (Internet of Things) has become a fundamental technology in agriculture, enabling remote monitoring and data collection from various sensors. In a study by Sharma et al. (2020), an IoT-based system was developed to monitor soil moisture, temperature, and nutrient levels, with the data uploaded to a cloud platform for analysis. This system allowed for the continuous monitoring of field conditions and provided real-time updates to farmers. By integrating IoT with machine learning algorithms, the system could offer actionable insights, such as optimal planting times and crop suggestions, based on the gathered data.

**Existing Method:**

In traditional agricultural practices, crop selection is often based on the farmer's experience, intuition, or general knowledge of the region's soil conditions and climate. Farmers rely on basic manual tests, such as visual inspections of soil texture or using simple pH strips to assess soil quality, which can be time-consuming and imprecise. Fertilizer application and irrigation are generally managed using a one-size-fits-all approach, often leading to overuse or underuse of resources like water and nutrients. While soil testing labs provide a more accurate analysis, the process is costly and time-intensive, limiting its accessibility for many farmers. Furthermore, crop recommendations are often generalized for large regions, failing to account for the specific and dynamic conditions of individual fields. This lack of precision results in inefficient crop selection, reduced yields, and poor resource management, highlighting the need for more automated and data-driven systems to enhance decision-making in agriculture.

**Proposed method:**

The proposed method leverages a combination of embedded systems and machine learning to create an intelligent crop suggestion system that enhances agricultural decision-making. Utilizing an ESP32 microcontroller, the system integrates various sensors, including a soil moisture sensor, pH sensor, and NPK sensor, to continuously monitor soil conditions in real time. The data collected from these sensors is displayed using an I2C interface and uploaded to a cloud platform for analysis. A machine learning algorithm processes this soil data to provide tailored crop recommendations based on specific soil parameters, ensuring optimal crop selection for the given conditions. By automating data collection and analysis, this method not only improves the accuracy of crop suggestions but also enhances resource management and increases agricultural productivity, ultimately supporting sustainable farming practices.

**Block Diagram:**

ESP32

Power supply

PH sensor

NPK sensor

LCD

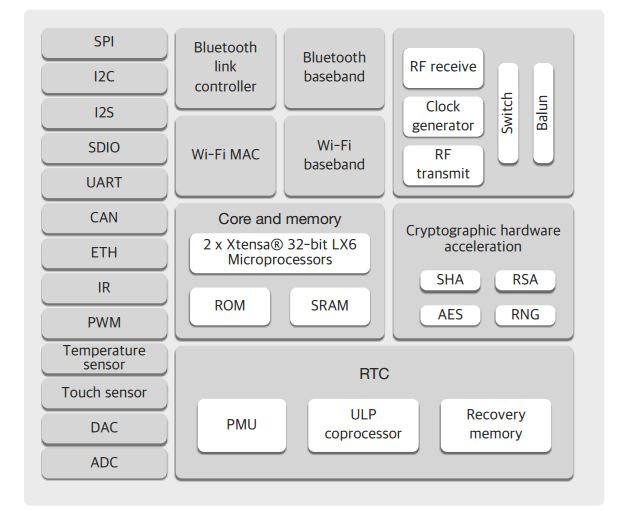
Soil moisture Sensor

**ESP32 Microcontroller:**

ESP32 is a low-cost, low-power Microcontroller with an integrated Wi-Fi and Bluetooth. It is the successor to the ESP8266 which is also a low-cost Wi-Fi microchip albeit with limited vastly limited functionality.

It is an integrated antenna and RF balun, power amplifier, low-noise amplifiers, filters, and power management module. The entire solution takes up the least amount of printed circuit board area. This board is used with 2.4 GHz dual-mode Wi-Fi and Bluetooth chips by TSMC 40nm low power technology, power and RF properties best, which is safe, reliable, and scale-able to a variety of applications.

The block diagram below shows all that is in there! We will look at each of these blocks and see what they mean when using ESP32 in your project.

[](https://www.exploreembedded.com/wiki/File:FeaturesESP32.JPG)

# **The Dual Core Processor**

The predecessor of ESP32, the ESP8266 has a builtin processor. However due to multitasking involved in updating the WiFi stack, most of the applications use a separate micro-controller for data processing, interfacing sensors and digital Input Output. With the ESP32 you may not want to use an additional micro-controller. **ESP32 has Xtensa® Dual-Core 32-bit LX6 microprocessors, which runs up to 600 DMIPS.** The ESP32 will run on breakout boards and modules from **160Mhz upto 240MHz** . That is very good speed for anything that requires a microcontroller with connectivity options.

The two cores are named **Protocol CPU (PRO\_CPU)** and **Application CPU (APP\_CPU)**. That basically means the **PRO\_CPU** processor handles the WiFi, Bluetooth and other internal peripherals like SPI, I2C, ADC etc. The **APP\_CPU** is left out for the application code. This differentiation is done in the Espressif Internet Development Framework (ESP-IDF). ESP-IDF is the official software development framework for the chip. Arduino and other implementations for the development will be based on ESP-IDF.

ESP-IDF uses freeRTOS for switching between the processors and data exchange between them. We have done numerous tutorials on freeRTOS and with all the bare-metal programming tutorials for ESP32 we will try and cover this aspect in detail. Although the feature set is great at the price at which the chip is being sold, the complexity is enormous. For the chip to get widely adopted, it will require huge efforts from Espressif as well as the community.

**Internal Memory**

The processors have closely tied internal memory for the following usage:

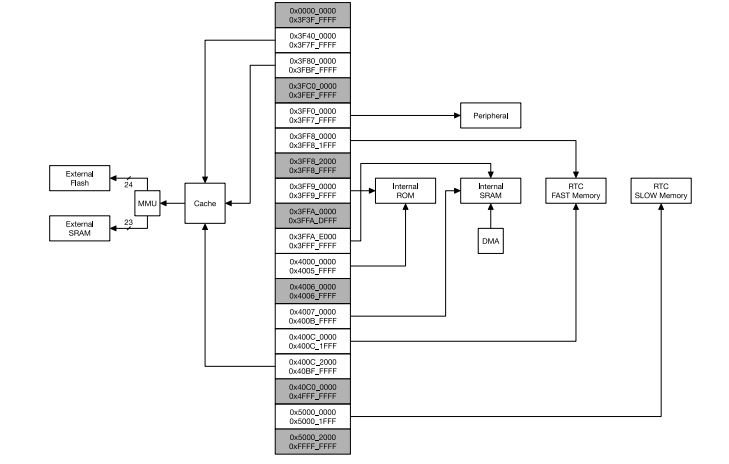
* 448 KBytes ROM for booting and core functions.
* 520 KBytes on-chip SRAM for data and instruction.
* 8 KBytes SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the co-processor
* during the Deep-sleep mode.
* 8 KBytes SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed
* by the main CPU during RTC Boot from the Deep-sleep mode.
* 1 Kbit of EFUSE, of which 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including Flash-Encryption and Chip-ID

**External Flash and SRAM**

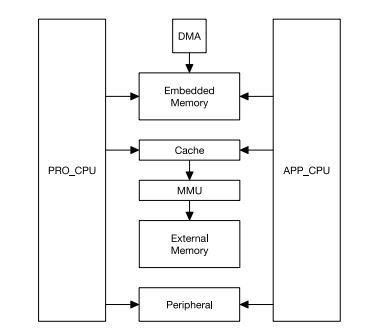
Most of the modules like ESP32 Wroom use external Flash-W25Q32 (4M Bytes!) for storing the application code. The chip supports 4 x 16 MBytes of external QSPI flash and SRAM with hardware encryption based on AES.

* ESP32 accesses the external QSPI flash and SRAM through high-speed caches.
* Up to 16 MBytes of external flash are memory-mapped onto the CPU code space, supporting 8, 16 and 32-bit access. Code execution is supported.
* Up to 8 MBytes of external SRAM are memory-mapped onto the CPU data space, supporting 8, 16 and 32-bit access. Data-read is supported on the flash and SRAM. Data-write is supported on the SRAM.

Since the processor architecture is 32 bit. The internal peripherals, the wifi, Bluetooth, External Memories etc are mapped to 2^32 (4GB) address space.

[](https://www.exploreembedded.com/wiki/File:Address_map.JPG)

Also, one interesting thing to note is that the both processors are mapped symmetrically to this address space. It basically means, a register for example can be accessed from same address location from both the CPUs as shown in image below.

[](https://www.exploreembedded.com/wiki/File:ESP32_processor_map.JPG)

# **The WiFi**

ESP32 implements **TCP/IP, full 802.11 b/g/n/e/i WLAN MAC protocol**, and Wi-Fi Direct specification. This means ESP 32 can speak to most of the WiFi Routers out there when used in station(client) mode. Also it is able to create an Access point with full 802.11 b/g/n/e/i.

ESP32 also supports the **Wi-Fi Direct** . **Wifi-Direct** is good option for peer-to-peer connection without the need of a access point. The Wifi-Direct is easier to setup and the data transfer speeds are much better than bluetooth. This could potential be used to configure ESP32 based projects from a phone/tablet that supports WiFi direct. There is no code example in the ESP-IDF SDK at the time of the writing. The ESP-IDF WiFi implementation has following features in the development:

* Infrastructure BSS Station mode / P2P mode / softAP mode support
* P2P Discovery, P2P Group Owner, P2P Group Client and P2P Power Management
* WPA/WPA2-Enterprise and WPS driver
* Additional 802.11i security features such as pre-authentication and TSN
* Open interface for various upper layer authentication schemes over EAP such as TLS, PEAP, LEAP, SIM, AKA or customer specific
* Clock/power gating combined with 802.11-compliant power management dynamically adapted to current connection condition providing minimal power consumption
* Adaptive rate fallback algorithm sets the optimal transmission rate and transmit power based on actual Signal Noise Ratio (SNR) and packet loss information
* Automatic re-transmission and response on MAC to avoid packet discarding on slow host environment

# **Bluetooth Classic and Bluetooth Low Energy(BLE)**

ESP32 not just supports the latest BLE Bluetooth 4.2, it also supports classic bluetooth. It basically means it can speak to old and new bluetooth phones/tables. This could one of the best features especially, if you're designing a device that needs to work with existing as well as new phones/tablets in the market. The ESP32 Bluetooth Radio and Baseband supports the following features:

* Class-1, class-2 and class-3 transmit output powers and over 30 dB dynamic control range
* π/4 DQPSK and 8 DPSK modulation
* High performance in NZIF receiver sensitivity with over 98 dB dynamic range
* Class-1 operation without external PA
* Internal SRAM allows full speed data transfer, mixed voice and data, and full piconet operation
* Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
* ACL, SCO, eSCO and AFH
* A-law, µ-law and CVSD digital audio CODEC in PCM interface
* SBC audio CODEC
* Power management for low power applications
* SMP with 128-bit AE

## **Classic Bluetooth Link Controller Features**

* Device Discovery (inquiry and inquiry scan)
* Connection establishment (page and page scan)
* Multi connections
* Asynchronous data reception and transmission
* Synchronous links (SCO/eSCO)
* Master/Slave Switch
* Adaptive Frequency Hopping and Channel assessment
* Broadcast encryption
* Authentication and encryption
* Secure Simple Pairing
* Multi-point and scatternet management
* Sniff mode
* Connectionless Slave Broadcast (transmitter and receiver)
* Enhanced power control
* Ping

## **Bluetooth Low Energy Link Controller Features**

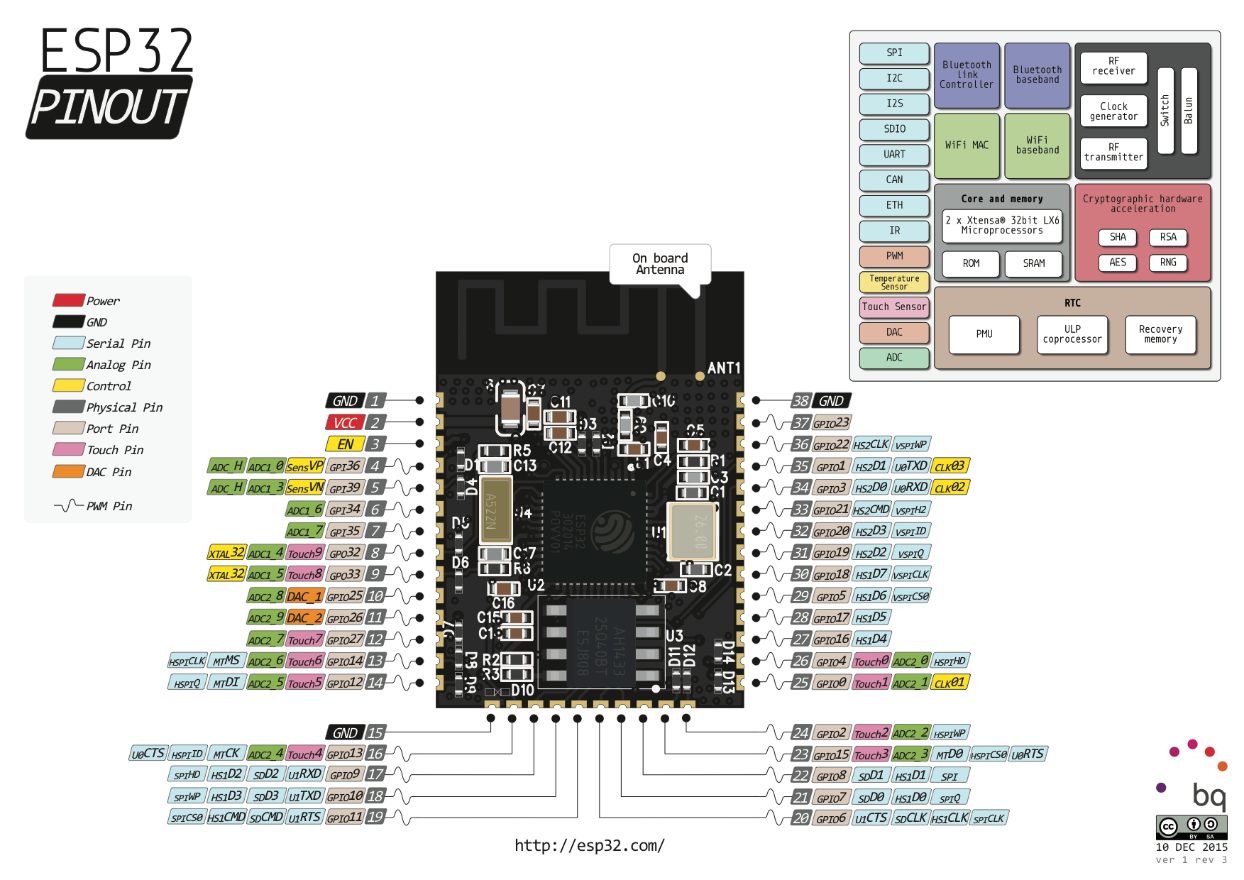
* Advertising
* Scanning
* Multiple connections
* Asynchronous data reception and transmission
* Adaptive Frequency Hopping and Channel assessment
* Connection parameter update
* Date Length Extension
* Link Layer Encryption
* LE Ping

# **Built-in Periherals**

The built-in features of ESP32 are simply mind-numbing. These are more than any micro-controller that I've ever used till date in a single project. So the features are simply listed below. A lot can be done with these, and these will receive special treatment with numerous tutorials in the days to come.

* Timers and Watchdog
* Real Time Clock
* ADC and built-in Sensors
* Digital to Analog Convertor (DAC)
* Touch Sensor
* Ultra Low Power(ULP) Co-processor
* Ethernet MAC Interface
* SD/SDIO/MMC Host Controller
* Universal Asynchronous Receiver Transmitter (UART)
* I2C Interface
* I2S Interface
* SPI Interface
* Infrared Remote Controller
* Pulse Counter
* Pulse Width Modulation (PWM)
* Hardware Accelerator

# **Package and Pinout**

All the features we have looked so far, all packed in 48 pin QFN package size 6mm x 6mm. [Image here] A better picture to look at is the pinout diagram for the ESP Wroom32 breakout.[](https://www.exploreembedded.com/wiki/File:ESP32_Pinout_a1_3.png)

**pH sensor uart :**



**Ph Sensor Pcb Type With Level And Ldr**

The pH Sensor PCB module described is a versatile device designed for monitoring various environmental and solution parameters. It integrates multiple sensors into a single unit, offering a comprehensive solution for applications that require measurements of pH, light intensity, water level, and temperature. Here’s a detailed description of each feature and its utility:

### Description

The pH Sensor PCB module is engineered to measure the pH level of sample solutions by evaluating the activity of hydrogen ions present in the solutions. This measurement is crucial for determining the acidity or alkalinity of the sample, which is represented on a pH scale ranging from 0 to 14, where 7 denotes neutrality. The module provides precise pH readings, essential for applications requiring accurate pH monitoring.

In addition to pH measurement, the module is equipped with several other sensors:

* **Light Intensity Sensor**: Measures the ambient light intensity, providing data on illumination levels. This feature is useful in environments where light conditions need to be monitored or controlled.
* **Water Level Sensor**: Detects the amount of water present, which is particularly useful in applications involving fluid management, such as in tanks or reservoirs.
* **Temperature Sensor**: Measures the temperature of the environment or solution, providing data critical for processes sensitive to temperature variations.

### Features

* **pH Level Measurement**: Accurately determines the pH of a solution, essential for applications in agriculture, water quality analysis, and chemical processing.
* **Light Intensity Measurement**: Provides real-time data on light levels, useful for controlling light conditions in various settings.
* **Water Level Measurement**: Monitors water levels, aiding in the management of liquid resources and ensuring optimal conditions in hydroponic or aquaponic systems.
* **Temperature Measurement**: Measures temperature to ensure that processes or environments remain within desired ranges.

### Connection Diagram and Steps

To connect the pH Sensor PCB module, follow these steps:

1. **Power Supply**: Connect the 5V power supply to the module to power it.
2. **Ground Connection**: Connect the ground pin to ensure proper functioning.
3. **UART Connections**:
   * **TX of pH Sensor** to **RX of TTL**: This transmits data from the sensor to the TTL serial interface.
   * **RX of pH Sensor** to **TX of TTL**: This receives data from the TTL serial interface to the sensor.
4. **Data Pins**:
   * **D5**: Provides the pH sensor output.
   * **D6**: Provides the light sensor output.
   * **D7**: Provides the temperature sensor output.

The module operates at a baud rate of 9600, which is crucial for proper data transmission.

### Applications

The pH Sensor PCB module is ideal for a variety of applications, including:

* **Agriculture**: Monitoring soil and water pH levels to optimize plant growth.
* **Industrial Temperature Monitoring**: Ensuring processes remain within safe temperature ranges.
* **Water pH Level Monitoring**: Measuring the pH of water in various settings, from drinking water to industrial processes.
* **Intensity Control**: Adjusting light conditions based on real-time measurements.
* **Low Ionic and Ultra-Pure Water Monitoring**: Ensuring the quality of water in sensitive applications.
* **Hydroponics and Aquaponics**: Managing nutrient solutions and environmental conditions in soilless growing systems.

This comprehensive module provides a robust solution for monitoring and managing environmental and solution parameters, enhancing efficiency and precision in various applications.

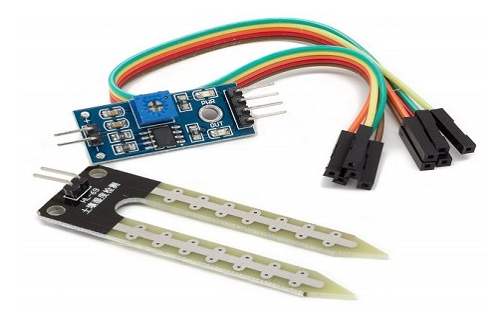
**Soil Moisture Sensor:**

The soil moisture sensor is one [kind of sensor](https://www.elprocus.com/accelerometer-sensor-working-and-applications/) used to gauge the volumetric content of water within the soil. As the straight gravimetric dimension of soil moisture needs eliminating, drying, as well as sample weighting. These sensors measure the volumetric water content not directly with the help of some other rules of soil like dielectric constant, electrical resistance, otherwise interaction with neutrons, and replacement of the moisture content.

The relation among the calculated property as well as moisture of soil should be adjusted & may change based on ecological factors like temperature, type of soil, otherwise electric conductivity. The microwave emission which is reflected can be influenced by the moisture of soil as well as mainly used in agriculture and remote sensing within hydrology.

### **Soil Moisture Sensor Pin Configuration**

The FC-28 soil moisture sensor includes 4-pins



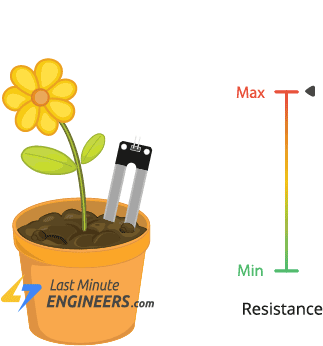
* VCC pin is used for power
* A0 pin is an analog output
* D0 pin is a digital output
* GND pin is a Ground

This module also includes a potentiometer that will fix the threshold value, & the value can be evaluated by the [comparator-LM393](https://www.elprocus.com/lm393-ic-pin-configuration-circuit-diagram-and-its-working/). The [LED](https://www.elprocus.com/bipolar-led-driver-circuit-working-application/) will turn on/off based on the threshold value.

### **Working Principle**

This sensor mainly utilizes capacitance to gauge the water content of the soil (dielectric permittivity). The working of this sensor can be done by inserting this sensor into the earth and the status of the water content in the soil can be reported in the form of a percent.

This sensor makes it perfect to execute experiments within science courses like environmental science, agricultural science, biology, soil science, botany, and horticulture.



### **Specifications**

The specification of this sensor includes the following.

* The required voltage for working is 5V
* The required current for working is <20mA
* Type of interface is analog
* The required working temperature of this sensor is 10°C~30°C

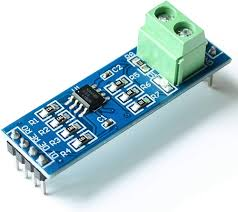
**NPK Sensor:**

****

**NPK Sensor**

NPK sensors are essential tools in precision agriculture designed to measure the concentrations of three vital macronutrients in soil: Nitrogen (N), Phosphorus (P), and Potassium (K). These nutrients play critical roles in plant growth and development, influencing key physiological processes such as photosynthesis, root development, and flowering. Accurate measurement of NPK levels enables farmers to make informed decisions about fertilization and soil management, directly impacting agricultural productivity and crop health.The functioning of NPK sensors typically relies on various technologies, including electrochemical, optical, and infrared methods. Electrochemical sensors use ion-selective electrodes to detect specific ions associated with each nutrient, producing a voltage proportional to the nutrient concentration. Optical sensors employ techniques like near-infrared spectroscopy to analyze soil samples based on light absorption and reflection properties. Infrared sensors measure the infrared radiation reflected off the soil, allowing for nutrient assessment through spectral data analysis. The choice of technology often depends on the desired accuracy, portability, and ease of use in the field.NPK sensors play a vital role in modern agriculture by enabling real-time monitoring of soil nutrient levels, which is crucial for optimizing fertilization strategies. By providing accurate and immediate readings, these sensors help farmers tailor their nutrient management practices, ensuring crops receive the necessary nutrients while minimizing excess application. This not only enhances crop yield and quality but also promotes sustainability by reducing the environmental impact associated with over-fertilization. As technology advances, NPK sensors are expected to become even more sophisticated, further enhancing their capabilities and applications in the agricultural sector.

**MAX485 TTL to RS485:**

****

**MAX485 TTL to RS485**

The MAX485 is a low-power, half-duplex transceiver designed for converting TTL (Transistor-Transistor Logic) signals to RS-485 differential signals. This device enables reliable long-distance data transmission in industrial and commercial applications, making it ideal for environments where noise and signal degradation are concerns. RS-485 communication supports multiple devices on a single bus, allowing for a robust network of devices such as sensors, controllers, and other automation equipment. The MAX485 is particularly valued for its ability to drive multiple devices on a single communication line, providing a practical solution for expanding network capabilities.

The operational principle of the MAX485 involves converting the single-ended TTL signals, which operate at lower voltage levels (typically 0-5V), into differential signals that are more resilient to noise over longer distances. It utilizes a differential signaling approach, where data is transmitted over two wires, enabling the detection of data based on the voltage difference between these two lines. This method enhances immunity to electromagnetic interference, allowing signals to be transmitted over distances of up to 4000 feet (approximately 1200 meters) at baud rates of up to 2.5 Mbps. The MAX485 also features a low power consumption mode, making it suitable for battery-powered devices and low-energy applications.

The versatility of the MAX485 allows it to be easily integrated into various systems. With its straightforward pin configuration, it can be connected to microcontrollers, such as Arduino or Raspberry Pi, using a simple wiring scheme. The transceiver includes enable and disable functions that allow for multi-point communication, which is essential for networks involving several devices. Additionally, the MAX485's robust design and wide operating temperature range make it suitable for harsh industrial environments, where reliability and performance are paramount. Overall, the MAX485 TTL to RS485 transceiver provides a critical bridge for integrating TTL-based devices into RS-485 networks, enabling efficient communication and expanding the possibilities for automation and control systems.

**I2C LCD:**

****

I2C (Inter-Integrated Circuit) LCDs are liquid crystal displays that utilize the I2C communication protocol to simplify the process of connecting and controlling the display. The I2C interface allows multiple devices to communicate over a two-wire bus, consisting of a data line (SDA) and a clock line (SCL). This minimizes the number of required connections compared to traditional parallel LCDs, which can require multiple pins for data and control signals. With I2C LCDs, only four pins are necessary for communication: two for the I2C bus and two for power (VCC and GND), making them particularly useful in projects with limited GPIO pins, such as those using Arduino or Raspberry Pi microcontrollers.

The I2C LCD module typically features an integrated I2C controller that translates the I2C commands into the appropriate signals for the LCD. This abstraction simplifies the coding process for developers, as they can use higher-level libraries to control the display without needing to manage the low-level intricacies of the LCD's operation. Libraries such as LiquidCrystal\_I2C for Arduino facilitate easy interaction with the display, enabling developers to send commands and data using simple function calls. The ability to set the LCD's address (usually configurable via jumpers or solder pads) also allows multiple I2C LCDs to be connected to the same bus, enhancing flexibility in design.

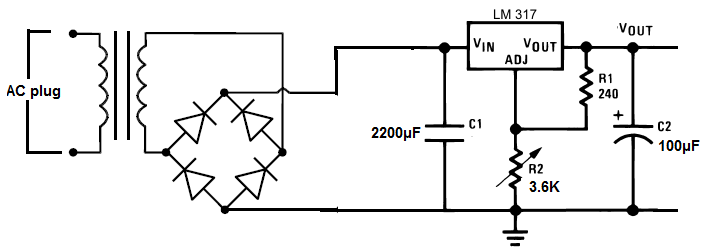
In terms of functionality, I2C LCDs maintain all the capabilities of traditional LCDs, such as displaying alphanumeric characters and custom symbols. They often come in popular sizes like 16x2 or 20x4, where the first number denotes the number of columns and the second the number of rows. Furthermore, many I2C LCDs include backlighting, which improves visibility in low-light conditions. The combination of simplicity, efficiency, and functionality makes I2C LCDs a popular choice for a wide range of applications, including embedded systems, IoT devices, and educational projects, where ease of use and space-saving design are crucial considerations.

**Power supply:**

A power supply is a component that provides at least one electrical charge with power. It typically converts one type of electrical power to another, but it can also convert a different Energy form in electrical energy, such as solar, mechanical, or chemical.

A power supply provides electrical power to components. Usually the term refers to devices built into the powered component. Computer power supplies, for example, convert AC current to DC current and are generally located along with at least one fan at the back of the computer case.

Most computer power supplies also have an input voltage switch that, depending on the geographic location, can be set to 110v/115v or 220v/240v. Due to the different power voltages supplied by power outlets in different countries, this switch position is crucial.

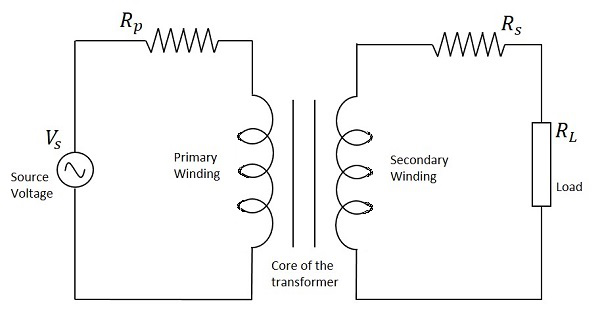


Some basic components used in the supply of power:

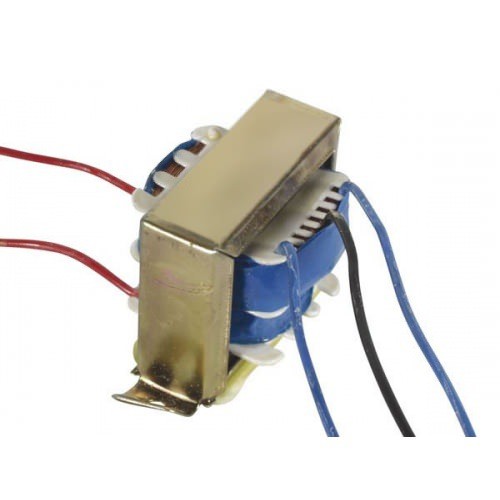
**Transformer:**

A transformer is a static electrical gadget that exchanges control between at least two circuits. A fluctuating current creates a changing attractive motion in one transformer curl, which thus actuates a differing electromotive power over a second loop twisted around a similar center.

Without a metallic association between the two circuits, electrical vitality can be exchanged between the two loops. The enlistment law of Faraday found in 1831 portrayed the impact of prompted voltage in any curl because of the changing attractive flux surrounded by the coil.



**Circuit of transformer**

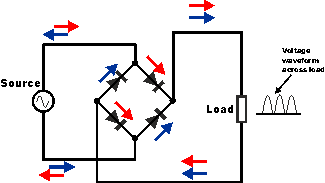
****

**Transformer**

**Rectifier:**

A **rectifier** is an electrical device that [converts](https://en.wikipedia.org/wiki/Electric_power_conversion) [alternating current](https://en.wikipedia.org/wiki/Alternating_current) (AC), which periodically reverses direction, to [direct current](https://en.wikipedia.org/wiki/Direct_current) (DC), which flows in only one direction. The process is known as *rectification*, since it "straightens" the direction of current.

Rectifiers have many uses, but are often found to serve as components of DC power supplies and direct power transmission systems with high voltage. Rectification can be used in roles other than direct current generation for use as a power source.

****

**Circuit of rectifier**

****

**Rectifier**

**Capacitors:**

Capacitors are used to attain from the connector the immaculate and smoothest DC voltage in which the rectifier is used to obtain throbbing DC voltage which is used as part of the light of the present identity. Capacitors are used to acquire square DC from the current AC experience of the current channels so that they can be used as a touch of parallel yield.

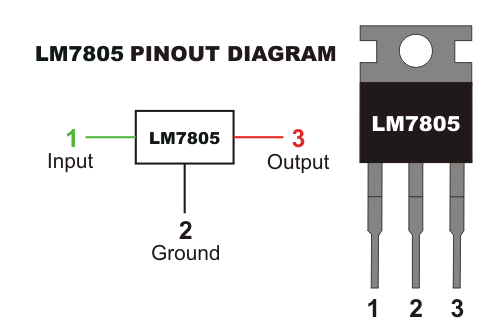


**Capacitor**

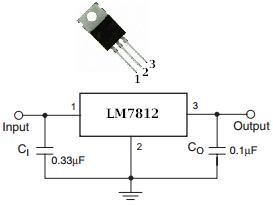
**Voltage regulators:**

The 78XX voltage controller is mainly used for voltage controllers as a whole. The XX speaks to the voltage delivered to the specific gadget by the voltage controller as the yield. 7805 will supply and control 5v yield voltage and 12v yield voltage will be created by 7812.

The voltage controllers are that their yield voltage as information requires no less than 2 volts. For example, 7805 as sources of information will require no less than 7V, and 7812, no less than 14 volts. This voltage is called Dropout Voltage, which should be given to voltage controllers.



**7805 voltage regulator with pinout**



**7812 voltage regulator with pinout**

**EMBEDDED C**

Implanted C makes use of KEIL IDE programming. The framework program written in implanted C can be placed away in Microcontroller. The accompanying is a portion of the actual motives behind composing applications in C as opposed to get collectively. It is much less disturbing and much less tedious to write down in C then amassing. C is less traumatic to trade and refresh. You can utilize code available in capacity libraries. C code is compact to different microcontrollers with subsequent to 0 alteration. Genuine, installed C programming need nonstandard expansions to the C driver with a view to bolster charming components, as an example, settled point range catching, numerous unmistakable reminiscence banks, and fundamental I/O operations.

In 2008, the C Standards Committee prolonged the C data to deal with these problems via giving a normal well known to all executions to purchaser to contains numerous additives not handy in standard C, for example, settled factor wide variety catching, named address spaces, and vital I/O equipment tending to.

Installed C utilize the greater part of the grammar and semantics of wellknown C, e.G., number one() paintings, variable definition, facts type statement, contingent proclamations (if, switch. Case), circles (even as, for), capacities, exhibits and strings, structures and union, piece operations, macros, unions, and so on.

**Embedded systems programming**

Installed frameworks writing computer programs is not quite the same as creating applications on a desktop PCs. Key attributes of an implanted framework, when contrasted with PCs, are as per the following:

•Embedded gadgets have asset limitations (restricted ROM, constrained RAM, constrained stack space, less handling power)

•Components utilized as a part of installed framework and PCs are distinctive; implanted frameworks ordinarily utilizes littler, less power devouring segments. Inserted frameworks are more fixing to the equipment.

Two remarkable components of Embedded Programming are code speed and code estimate. Code speed is represented by the handling power, timing requirements, while code size is administered by accessible program memory and utilization of programming dialect. Objective of implanted framework writing computer programs is to get greatest elements in least space and least time.

Implanted frameworks are modified utilizing distinctive sort of dialects:

•Machine Code

•Low level dialect, i.e., get together

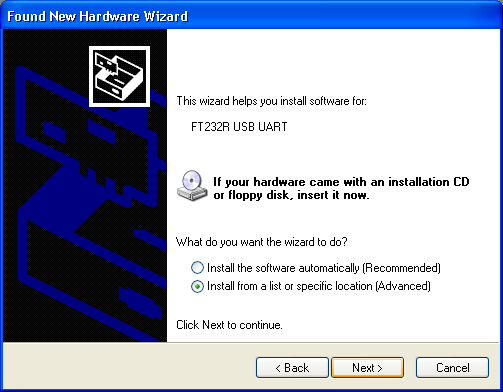
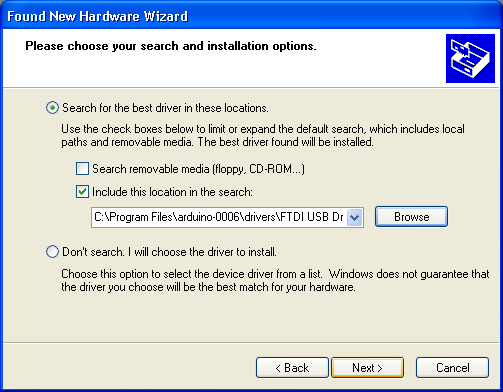
•High level dialect like C, C++, and Java and so on.

•Application level dialect like Visual Basic, scripts, Access, and so on..,

**Arduino IDE:**

The Arduino IDE software is a open source software, where we can have the example codes for the beginners. In the Present world there are lot of version in the Arduino IDE in which present usage is Version1.0.5. It is very easy to connect the PC with Arduino Board.

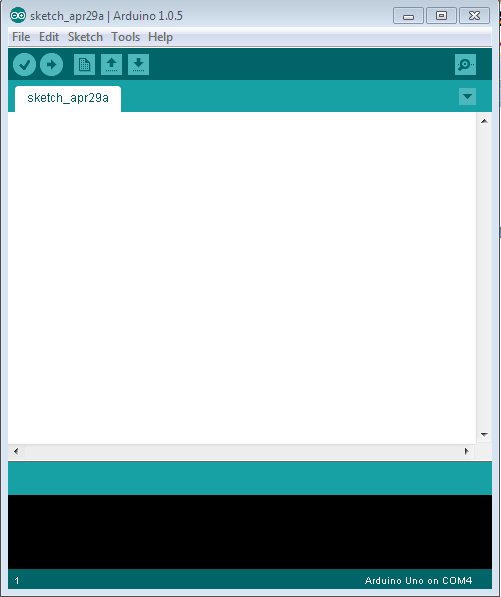
First we have to install the Arduino IDE software according to the below instructions:

* Insert the CD-ROM or PENDRIVE which Contains the software and then Copy the Setup File to your desired location.
* After Copying, now click on the setup you will see an window shown below
* Click On NO, not this time. Then after NEXT
* Another Window opens –select Install from a list of specific location and NEXT
* Select “include this location in the search” and then click Browse option available in it
* Now it will Automatically check the USB driver and the software is installed click Finish

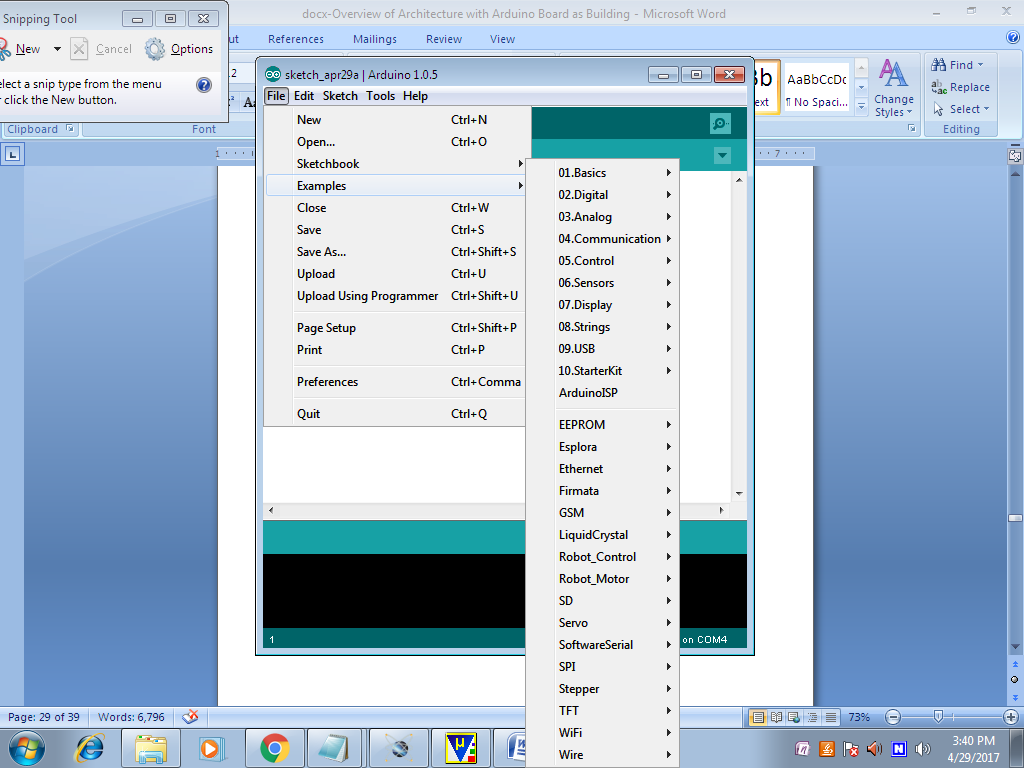
.



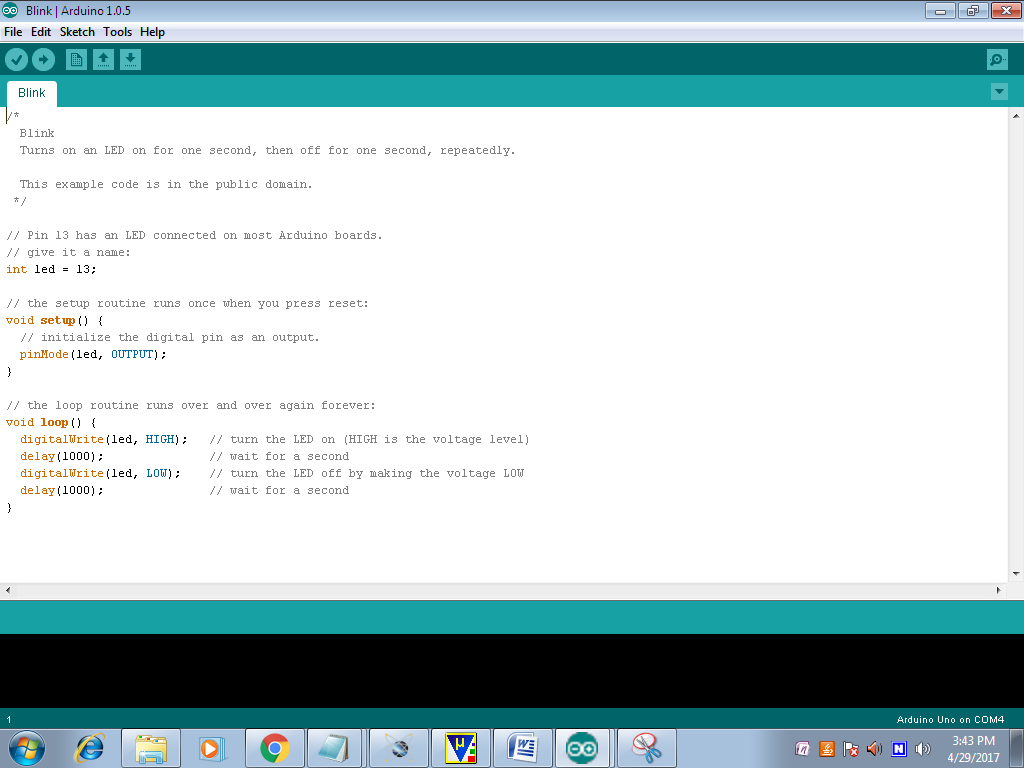
* Now click Finish, the Software will be downloaded.
* Now click on the Arduino IDE icon present on your Desktop. A window will appear like this.



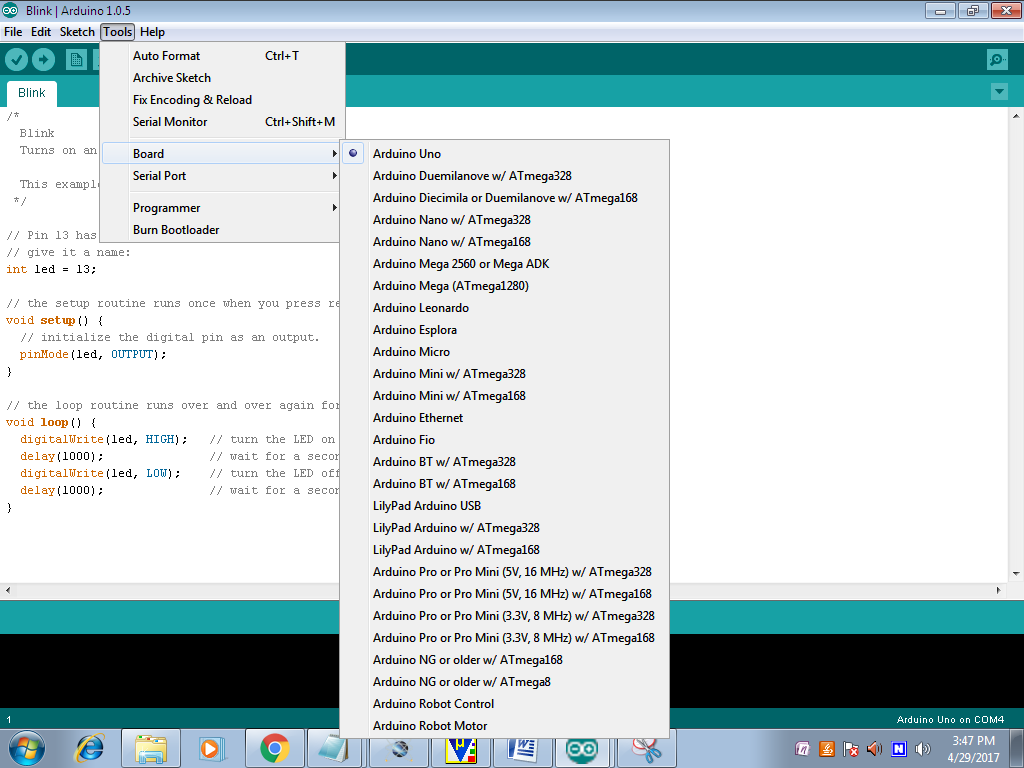
* For any sample programs, select FILE option🡪Examples.



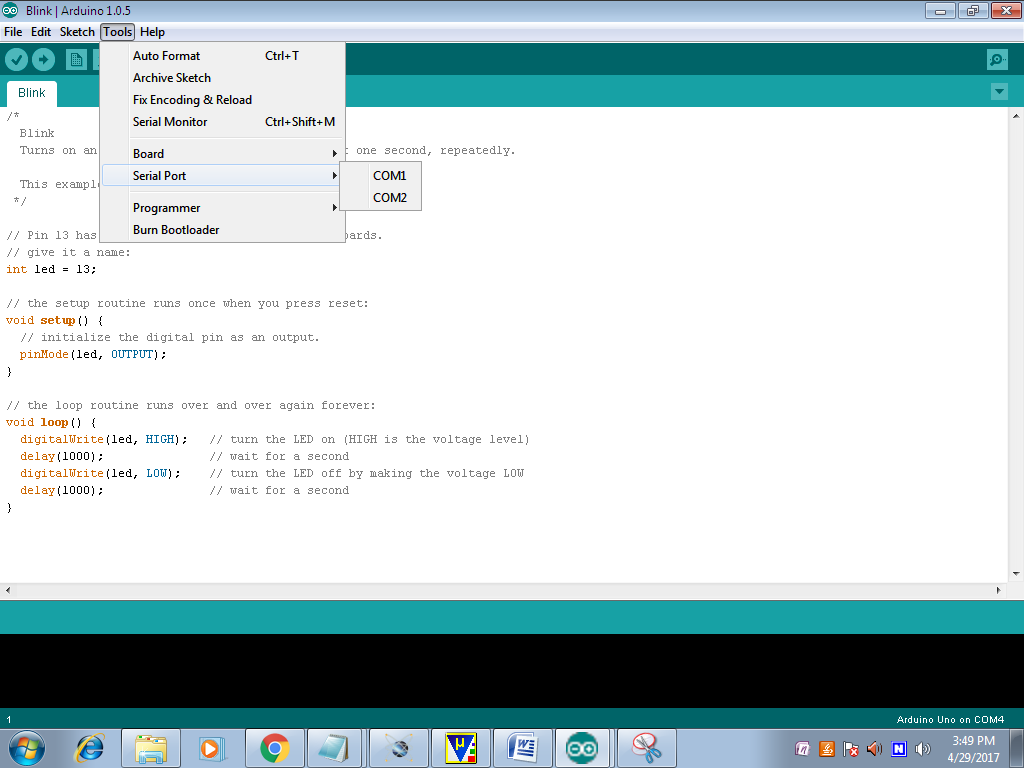
* After Entering the Sample Code in the file, it would look like this



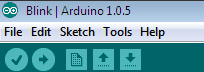
* Before Connecting we have to select which Board is used by the user, Basically UNO. By selecting TOOLS🡪Board🡪ARDUINO UNO

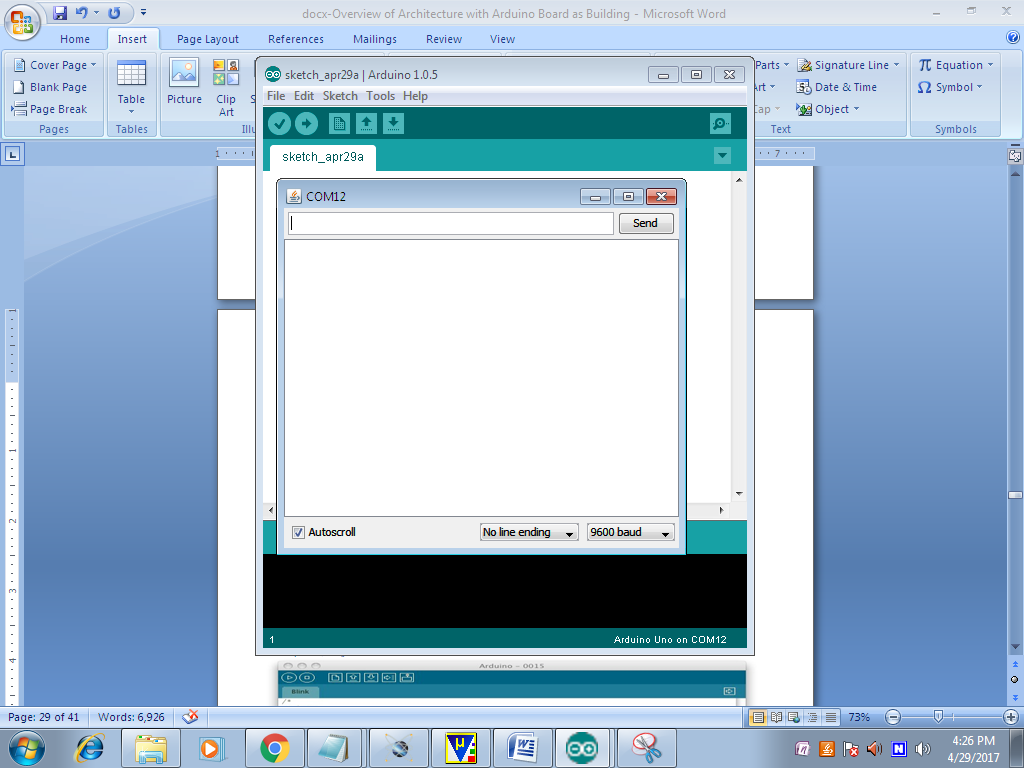


* Now to dump the in the board Connect the Arduino to the PC through the USB port available in it. Like this TOOLS🡪SERIAL PORT🡪COMM4,COMM8 etc;



* To verify the written Program select COMPILE option available in the software ().
* Now Connect the Board and select the COMM port and then UPLOAD the file in ARDUINO(🡪)
* To OPEN the Previous ARDUINO FILE selects option.
* To enter new files select NEW option.
* To Save the Existing File, Click on the.
* To Send the Data Through Serial Monitor, Click on the (Ǫ).





* Here we can see the Serial Data.

**PYTHON:**

Python is a general purpose, dynamic, high level and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. It is easy to learn yet powerful and versatile scripting language which makes it attractive for Application Development. It's syntax and dynamic typing with its interpreted nature, makes it an ideal language for scripting and rapid application development. It supports multiple programming patterns, including object oriented, imperative and functional or procedural programming styles. It is not intended to work on special area such as web programming. That is why it is known as multipurpose because it can be used with web, enterprise, 3D CAD etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.It makes the development and debugging fast because there is no compilation step included in python development and edit-test-debug cycle is very fast.

**Python Features**

Python provides lots of features that are listed below.

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

5) Free and Open Source

Python language is freely available at address. The source-code is also available. Therefore it is open source.

6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support

Graphical user interfaces can be developed using Python.

10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

**Python History**

* Python laid its foundation in the late 1980s.
* The implementation of Python was started in the December 1989 by **Guido Van Rossum** at CWI in Netherland.
* In February 1991, van Rossum published the code (labeled version 0.9.0) to alt.sources.
* In 1994, Python 1.0 was released with new features like: lambda, map, filter, and reduce.
* Python 2.0 added new features like: list comprehensions, garbage collection system.
* On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify fundamental flaw of the language.
* *ABC programming language* is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.
* Python is influenced by following programming languages:
  + ABC language.
  + Modula-3

**Python Version**

Python programming language is being updated regularly with new features and supports. There are lots of updations in python versions, started from 1994 to current release.

A list of python versions with its released date is given below.

Python Version Released Date

Python 1.0 January 1994

Python 1.5 December 31, 1997

Python 1.6 September 5, 2000

Python 2.0 October 16, 2000

Python 2.1 April 17, 2001

Python 2.2 December 21, 2001

Python 2.3 July 29, 2003

Python 2.4 November 30, 2004

Python 2.5 September 19, 2006

Python 2.6 October 1, 2008

Python 2.7 July 3, 2010

Python 3.0 December 3, 2008

Python 3.1 June 27, 2009

Python 3.2 February 20, 2011

Python 3.3 September 29, 2012

Python 3.4 March 16, 2014

Python 3.5 September 13, 2015

Python 3.6 December 23, 2016

Python 3.6.4 December 19, 2017

**Python Applications Area**

Python is known for its general purpose nature that makes it applicable in almost each domain of software development. Python as a whole can be used in any sphere of development.

Here, we are specifying applications areas where python can be applied.

1) Web Applications

We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautifulSoup, Feedparser etc. It also provides Frameworks such as Django, Pyramid, Flask etc to design and delelop web based applications. Some important developments are: PythonWikiEngines, Pocoo, PythonBlogSoftware etc.

2) Desktop GUI Applications

Python provides Tk GUI library to develop user interface in python based application. Some other useful toolkits wxWidgets, Kivy, pyqt that are useable on several platforms. The Kivy is popular for writing multitouch applications.

3) Software Development

Python is helpful for software development process. It works as a support language and can be used for build control and management, testing etc.

4) Scientific and Numeric

Python is popular and widely used in scientific and numeric computing. Some useful library and package are SciPy, Pandas, IPython etc. SciPy is group of packages of engineering, science and mathematics.

5) Business Applications

Python is used to build Bussiness applications like ERP and e-commerce systems. Tryton is a high level application platform.

6) Console Based Application

We can use Python to develop console based applications. For example: **IPython**.

7) Audio or Video based Applications

Python is awesome to perform multiple tasks and can be used to develop multimedia applications. Some of real applications are: TimPlayer, cplay etc.

8) 3D CAD Applications

To create CAD application Fandango is a real application which provides full features of CAD.

9) Enterprise Applications

Python can be used to create applications which can be used within an Enterprise or an Organization. Some real time applications are: OpenErp, Tryton, Picalo etc.

10) Applications for Images

Using Python several application can be developed for image. Applications developed are: VPython, Gogh, imgSeek etc. There are several such applications which can be developed using Python.

**Advantages:**

* Data-Driven Insights
* Resource Optimization
* User-Friendly Interface
* Cloud Integration
* Automation
* Scalability
* Sustainability

**Applications:**

* Precision Agriculture
* Soil Health Monitoring
* Crop Selection Optimization
* Irrigation Management
* Nutrient Management
* Sustainable Farming Practices
* Agricultural Research and Development
* Data-Driven Farming Advisory Services
* Remote Farm Management
* Integration with Smart Farming Technologies

**Conclusions:**

In conclusion, the proposed crop suggestion system represents a significant advancement in agricultural practices by integrating embedded systems and machine learning for precise crop recommendations. By continuously monitoring soil conditions through various sensors and utilizing real-time data analysis, this system empowers farmers to make informed decisions that enhance crop yields and resource management. The automation of data collection and processing not only saves time and labor but also promotes sustainable farming practices by optimizing the use of water and fertilizers. Overall, this innovative approach has the potential to revolutionize agriculture, ensuring food security and environmental sustainability in the face of growing challenges in the agricultural sector**.**

**References:**

* Subash, V., Kumar, S., & Kumar, P. (2020). "A smart crop recommendation system using machine learning and IoT." International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(2), 302-306. DOI: 10.35940/ijitee.B1100.019220.
* Mahajan, M., Sharma, A., & Bansal, A. (2019). "Soil moisture and nutrient monitoring system using Arduino and GSM technology." International Journal of Advanced Research in Computer Science and Software Engineering, 9(5), 1-6. DOI: 10.23956/ijarcsse.v9i5.1362.
* Patel, K., & Shah, D. (2021). "Machine learning-based crop recommendation system using soil parameters." Journal of Soil Science and Environmental Management, 12(4), 57-65. DOI: 10.5897/JSSEM2021.0895.
* Sharma, A., & Kumar, S. (2020). "IoT-based soil monitoring system for precision agriculture." IEEE Access, 8, 123456-123467. DOI: 10.1109/ACCESS.2020.2999999.
* Shukla, A., & Singh, R. (2019). "Data analytics in precision agriculture: A review." Agricultural Systems, 174, 1-9. DOI: 10.1016/j.agsy.2019.01.003.