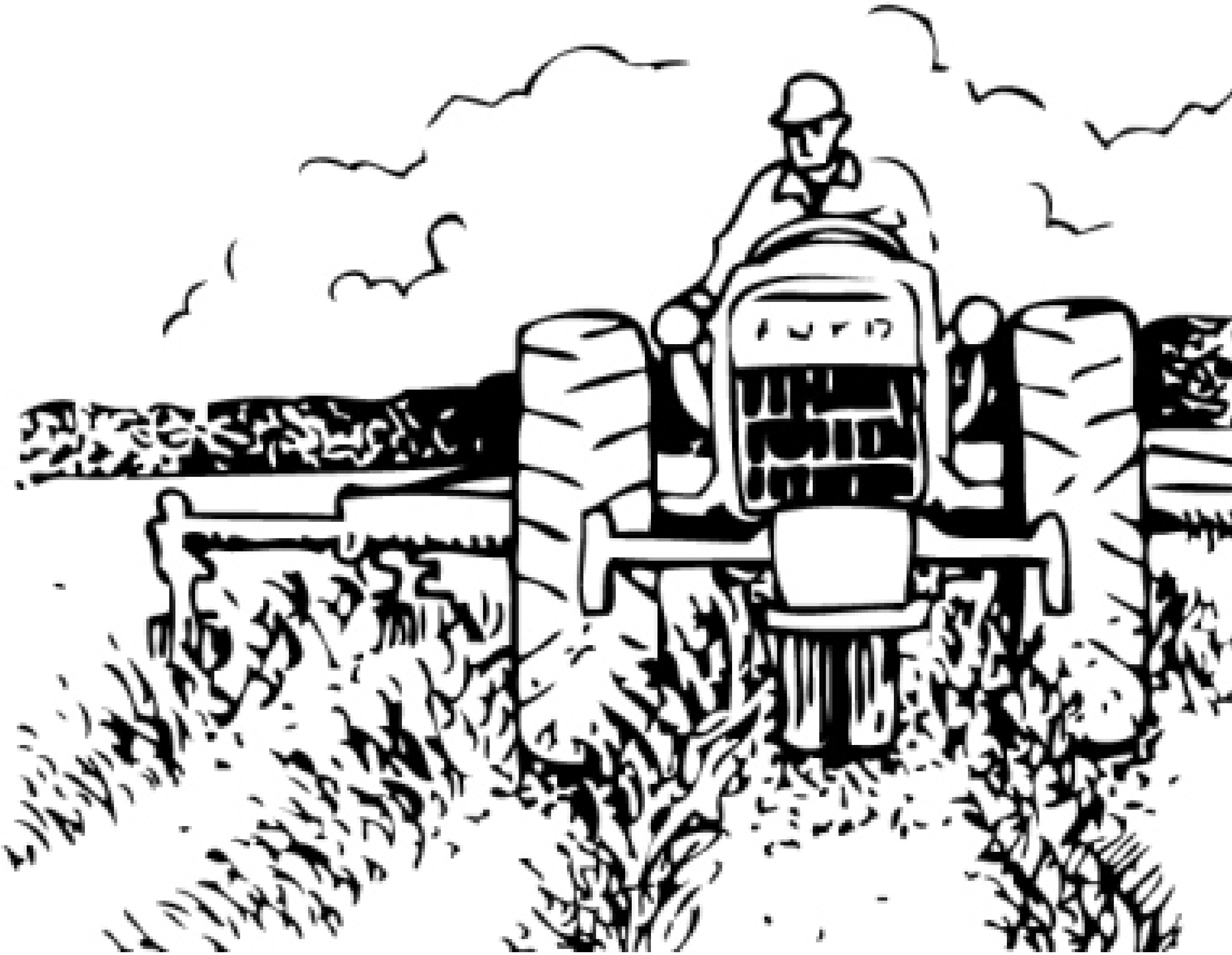
INFIIOT TECHNOLOGIES
PVT. LTD.

User Manual

InfiSense

A Practical Approach to IoT



InfiIoT Technologies

⌂ 302, Shubham Park,
Anand-Tirth nagar,
Kondhwa,
Pune - 411048

🌐 info@infiiot.com
✉ www.infiiot.com

Preface:

This Manual is for the users of Infi Sense and Infi Cloud Platform, with in-depth details of products including reference designs and examples.

For further details and queries contact:

Infiniti Systems,

Bandal Dhankude Plaza,

Kothrud, Pune-411038,

Maharashtra, India.

Ph: +91-9405435681

info@infinitisystems.co.in

www.infinitisystems.co.in

Scan for company details:

Table of Contents

Type chapter title (level 1) **1**

Type chapter title (level 2) 2

Type chapter title (level 3) 3

Type chapter title (level 1) **4**

Type chapter title (level 2) 5

Type chapter title (level 3) 6

1 Intro to IoT:

Internet of Things (IoT) is a system of interconnected objects, usually called smart devices, through the Internet. The object can be a heart monitor, a remote or an automobile with built-in sensors. That is objects that have been assigned an IP address and have the capability to collect and transfer data over a network. The objects interact with the external environment with the help of embedded technology, which helps them in taking decisions. Since these devices can now represent themselves digitally.

In other words,
The globally ruling technology acting as a single key to shrinking this whole universe to a tiny globally connected village, whereas IoT comprises of just two words which precisely depicts its definition.

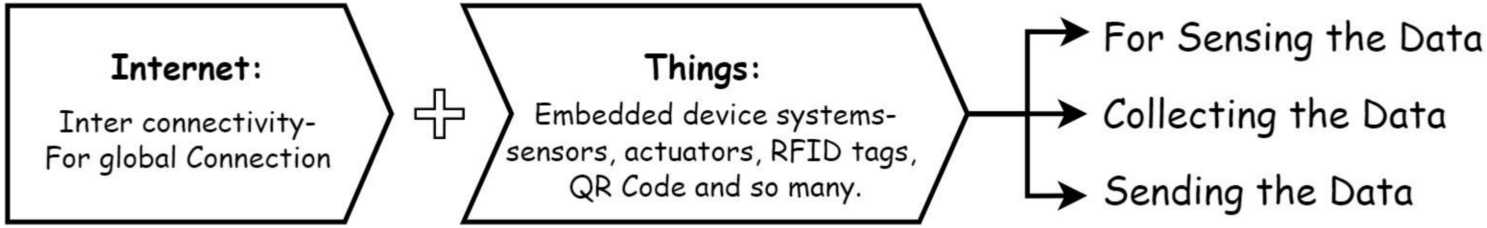


Figure 1: Introduction to IOT

1.1 Main Components of IOT:

- **Low-power embedded systems –**
  Less battery consumption, high performance are the inverse factors play a significant role during the design of electronic systems.
- **Cloud computing –**
  Data collected through IoT devices is massive and this data has to be stored on a reliable storage server. This is where cloud computing comes into play. The data is processed and learned, giving more room for us to discover where things like electrical faults/errors are within the system.

- **Availability of big data –**
  We know that IoT relies heavily on sensors, especially real-time. As these electronic devices spread throughout every field, their usage is going to trigger a massive flux of big data.
- **Networking connection –**
  In order to communicate, internet connectivity is a must where each physical object is represented by an IP address. However, there are only a limited number of addresses available according to the IP naming. Due to the growing number of devices, this naming system will not be feasible anymore. Therefore, researchers are looking for another alternative naming system to represent each physical object.

1.2 Characteristics of IOT:

- Massively scalable and efficient.
- IP-based addressing will no longer be suitable in the upcoming future.
- An abundance of physical objects is present that does not use IP, so IoT is made possible.
- Devices typically consume less power. When not in use, they should be automatically programmed to sleep.
- A device that is connected to another device right now may not be connected in another instant of time.
- Intermittent connectivity – IoT devices aren't always connected. In order to save bandwidth and battery consumption, devices will be powered off periodically when not in use. Otherwise, connections might turn unreliable and thus prove to be inefficient.

1.3 Advantages and Disadvantage of IOT:

Advantages		***Disadvantages***
- Communication	- Automation	- Lagging of standard Compatibility
- Remote Control	- Continuous Monitoring -	- More opportunities for future
- Better Decision	- More Information	- Loss of privacy or security
- Time Saving	- Money Saving	- More dependent on technology

Table 1: Advantages and Disadvantages of IOT

1.4 Applications of IOT:

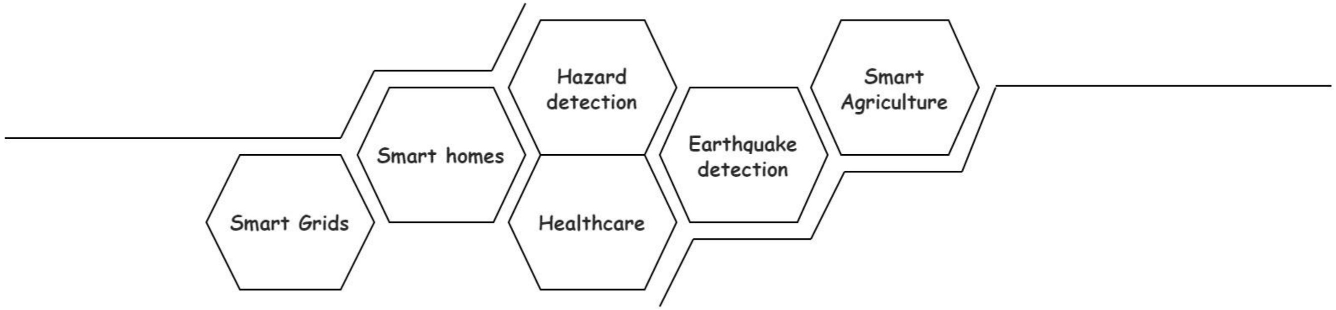


Figure 2: Applications of IOT

1.5 IOT in Agriculture:

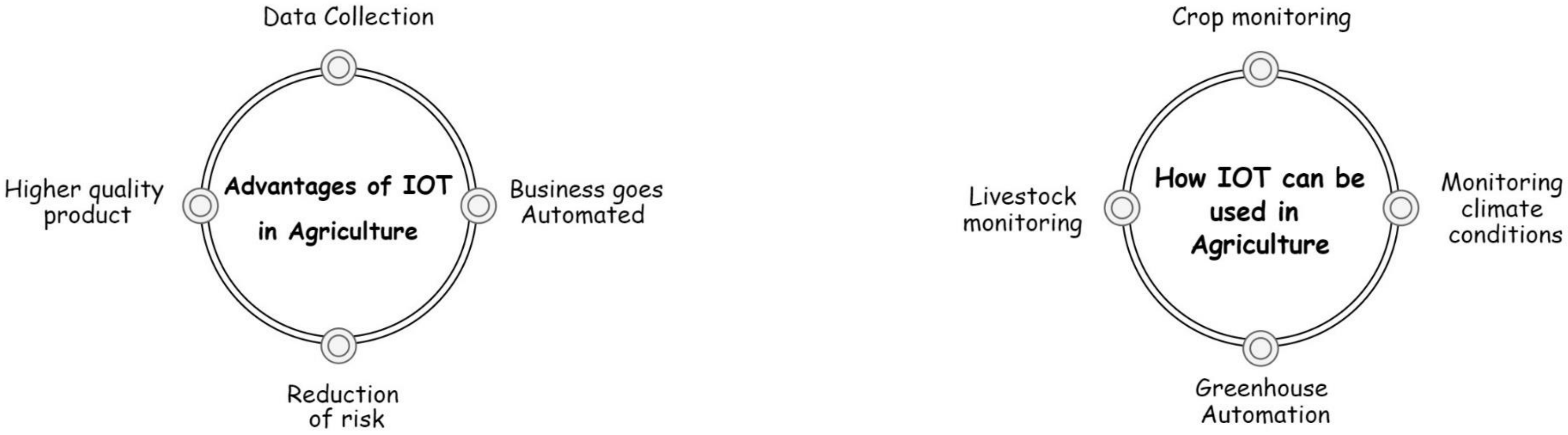


Figure 3: Applications of IOT in Agriculture

2 Introduction to Infi Sense Kit:

Infi Sense is development and evaluation Kit that allows user to evaluate the Infiniti System's sensors and Infi IOT cloud platform.

The purpose of the Infi Sense Agriculture Kit is to enable you to build your complete IoT solution from sensors and actuators, through a gateway up to different clouds and mobile apps just in a matter of a couple of hours and use your solutions for proof-of-concept for your projects.

Infi Sense Kit is to teach in-depth interfacing of various important sensors and actuators with Controller, connecting to network using various protocols, plotting and analysing of the collected data thus using IOT for monitoring and control things.

2.1 Features of Infi Sense:

- Quick and Easy Installation and Setup
- Configure and control via built-in API's
- Easy Plug-and-Play Phoenix connection for sensors and actuators
- Comprehensive working and principle of all sensors
- Easy to use API's with elaborative example codes
- On-the-fly Infi IOT Cloud Platform Integration
- Interfacing of many important sensors and actuators for agricultural IOT
- Portable Kit- test real field parameters
- Easy programmable/replaceable microcontroller
- On-board indication Led and Button for users to play with.
- On-board I2C based LCD module for real time display of important Data
- Provision for Deep Sleep modes for real on-field testing of your project.
- Includes on-board precision 16-bit Analog-to-Digital Converters.

2.2 Block Diagram:

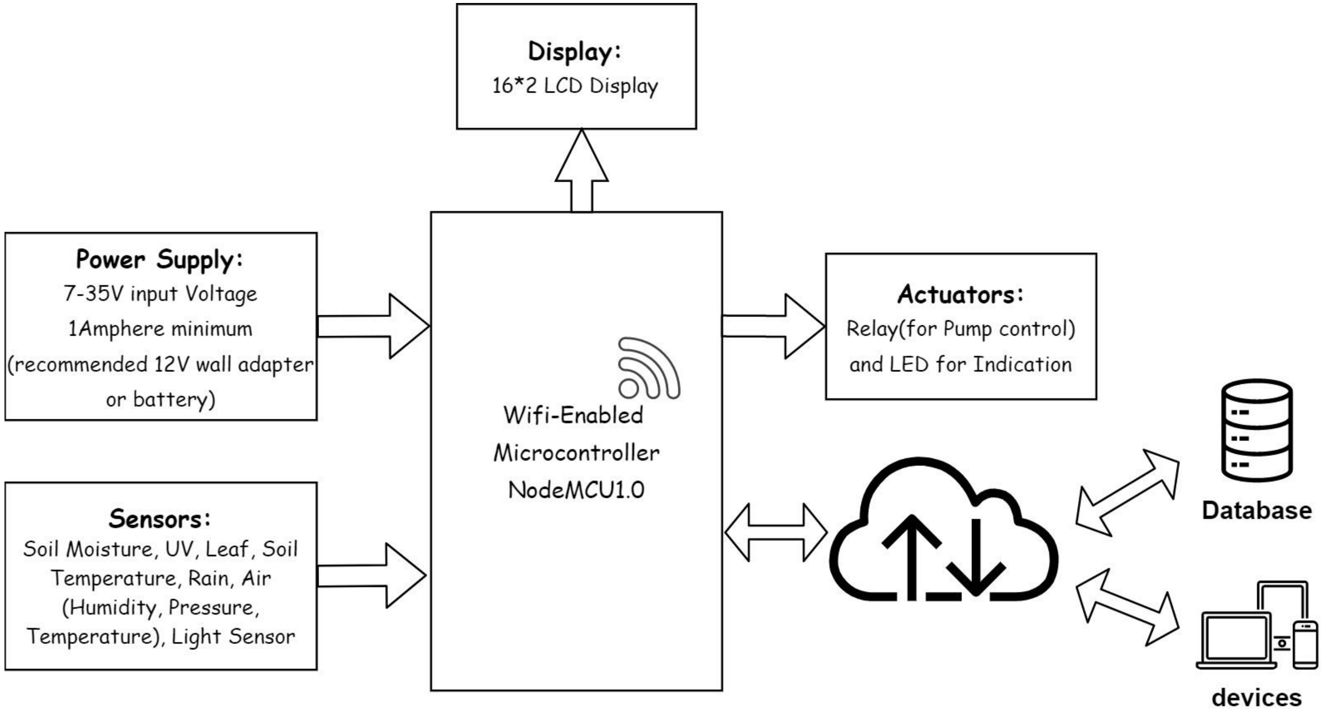


Figure 4: Block Diagram of Infi Sense

2.3 Kit Includes:

- **Shipped with Infi Sense Box-** Infi Sense Kit, Subscription to Infi IOT Cloud Service, Capacitive Soil Moisture Sensor, UV Sensor(ML8511), Light Sensor(BH1750), Capacitive Leaf Moisture Sensor(Infi Leaf), Soil Temperature Probe(DS18B20), Resistive Rain Sensor, Air Sensor(BME280), 16*2 LCD Panel, Water Submersible Pump with small tank, Flower Pot, Infi Sense kit enclosure, Stand, 12v wall adapter.
- **Not Shipped with Infi Sense Kit-** USB Micro-B Cable, Wi-Fi Hotspot with active internet connection, 12v battery(optional).

2.4 Specifications of Infi Sense:

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage of Microcontroller: 3.3V
- Input Voltage: **7-12V** (for USB Micro-B) and **7-35V** (for DC Power Jack)
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 Mhz
- Wi-Fi: IEEE 802.11 b/g/n:
    - Integrated TR switch, balun, LNA, power amplifier and matching network
    - WEP or WPA/WPA2 authentication, or open networks
- On-board 16-bit precision Analog to Digital Converter module (ADS1115)
- **Sleep Jumper** to allow auto wakeup of module when using Deep Sleep API
- Digital pin D3 and D8 extension on **J2** header and **J8** Male header**(pin1)**
- Hardware UART extended on **J8** Male header
- Indicator Led connected to **D6**
- Tactile Switch (Push Button) connected to **D4**
- Switches & Indicators on NodeMCU:
    - RST – Reset the ESP8266 chip
    - Flash – Download new programs
    - Blue LED – User Programmable (Connected to **D0** pin)

2.5 Individual Module Specifications and Features:

- **Capacitive Soil Moisture Sensor:**
  - Operating Voltage: 3.3-5.5 VDC
  - Output Voltage: 0-3 VDC (Analog)
  - Operating Current: 5mA
  - Interface: PH2.0-3P
  - Excellent Service life due to Corrosion resistant material
  - Non-Corrosive probes
- **UV Sensor (ML8511):**
  - Operating Voltage: 2.7-3.3 VDC
  - Operating Current: 300µA (active), 0.1µA (Standby)
  - Output type: Analog
  - Photodiode sensitive to UV-A and UV-B
  - Embedded operational Amplifier
- **Leaf Sensor (Infi Leaf):**
  - Operating Voltage: 3.3-5.5 VDC
  - Output Voltage: 0-3 VDC (Analog)
  - Operating Current: 5mA
  - Interface: PH2.0-3P
  - Mimics an actual Leaf
  - Leaf Wetness Sensing using capacitive grid sensor
  - Unmatched accuracy in an easy to use sensor
  - Can sense milligram levels of water condensing on leaf surface

- **Battery Level Sensor:**
  - Mimics battery level voltage for field-based projects using variable Potentiometer
  - Can be Connected to input voltage for real time analysis of input voltage
- **Temperature Probe (DS18B20):**
  - Input Voltage 3.0V to 5.5V power/data
  - Usable temperature range: -55 to 125°C (-67°F to +257°F)
  - Operating Current: 1.5mA
  - Interface: 1-Wire interface- requires only one digital pin for communication
  - Unique 64-bit ID burned into chip
  - Multiple sensors can share one pin
  - ±0.5°C Accuracy from -10°C to +85°C
  - Query time is less than 750ms
  - Stainless steel tube 6mm diameter by 30mm long
- **Rain Sensor:**
  - Input Voltage: 3.3-5VDC
  - Output type: Analog Voltage
  - Operating Current: 10mA
  - Anti-Oxidation and Anti-Conductivity
  - Rain sensor board resistance when Wet-100KΩ and when Dry- 2MΩ
- **Air Sensor (BME280):**
  - Supply Voltage: 3.3V
  - Interface: I2C
  - Operating Current: 3.6µA on full usage and 0.1µA in sleep mode
  - Air Temperature Sensor: -40°-85°C
  - Air Humidity Sensor: ±3% relative humidity
  - Air Pressure Sensor: 300-1100hPa

- **Light Sensor (BH1750):**
  - Supply Voltage: 2.4-3.6VDC
  - Operating Current: 0.12mA
  - Interface: I2C
  - No need of calculations-Direct Digital values is given as output
  - On-board ADC to convert analog light intensity to digital LUX values
  - Consists of 50Hz/60Hz noise rejection function
- **16*2 LCD Module with I2C extension module:**
  - Operating Voltage: 4.7-5.3VDC
  - Current Consumption: 10mA
  - 8-bit, 4-bit modes
  - Contrast and Backlight Control
- **Analog to Digital Converter (ADS1115):**
  - Operating Voltage: 2.0-5.5VDC
  - Operating Current: 150µA
  - Interface: I2C
  - 4 ADC (Analog to Digital) channels
  - 16-bit precision ADC
- **5V Relay:**
  - Maximum Switching Voltage: 30VDC or 25VAC @10W
  - Switching Voltage: 5VDC
  - Control higher loads with small trigger voltage
  - Provides Isolation to microcontroller from higher loads
- **Water Submersible Pump:**
  - Operating Voltage: 9VDC
  - Operating Current:10mA approx.
  - Water submersible pump

2.6 Component and Connection Diagram:

2.7 Pinout and Mappings:

Component pin	Connection
NodeMCU-1.0	
D0	Sleep Jumper / On-Controller Blue LED
D1	SCL (I2C)
D2	SDA (I2C)
D3	J2 male header
D4	Switch (sw1)
D5	Temperature Probe (DS18B20)
D6	LED
D7	Relay
D8	J5 male header (pin 1)
TX, RX	J5 male header (pin 2 and pin 3)
ADS1115 (Analog-to-digital Module)	
Channel 0	Soil Moisture Sensor
Channel 1	UV Sensor (ML8511)
Channel 2	Leaf Sensor (Infi Leaf)
Channel 3	Battery Voltage

Table 2: Pinout and Mapping

2.8 How can the Gathered Data be used? What is the Significance of Data being Collected?

Parameter	Significance
Soil Moisture	• The yield of crop is more often determined by the amount of water available rather than the deficiency of other food nutrients • Soil water acts as nutrient itself and help regulate soil temperature. • Soil water helps in chemical and biological activities of soil • Soil Water is also essential for photosynthesis
UV Light	• UV light has been shown to drive increases in the plant production of active substances in medicinal plants, including antioxidant benefits of numerous plants or THC levels in cannabis. • UV light can also help maintain a healthy growing environment by reducing mold, mildew, and certain plant pests - all of which need alternatives to chemicals due to increasing fungicidal resistance. • As the price of UV LEDs continues to decline, the ability to cost effectively incorporate targeted UV into the growing process with the right wavelengths, the right dosage, and at the right time in the life cycle of specific plant species will improve.
Leaf Sensor	• Relationships between leaf wetness and plant diseases have been studied for centuries. • Leaf wetness refers to the presence of free water on the canopy, and is caused by intercepted precipitation, dew, or guttation. • Regardless of how the duration of leaf wetness is determined, its values have been used in many disease warning systems. • leaf wetness duration (LWD) and air temperatures combined are used to predict infection of several hosts by three different phytopathogenic fungi.

Table 3:Significance of Parameters (part1)

Parameter	Significance
Battery Voltage	• Monitoring Battery Voltage is very important for application which includes IOT devices located at harsh environmental conditions such as underwater, underground, high altitudes • Many times, it is important to know battery voltage for tracking health of battery to optimize battery for longer life
Soil Temperature	• Soil temperature directly affects plant growth. In other words, nearly every crop slows down its growth when soil temperatures are below 9°C and above 50°C • Germination of various seeds requires different soil temperature ranges. For example, maize starts to germinate at soil temperatures from 7 to 10°C • Most organisms within soil thrive at temperatures between 25-35°C • Nitrification requires a soil temperature of 32°C
Rain Sensor	• Precipitation, especially rain, has a dramatic effect on agriculture. All plants need at least some water to survive, therefore rain (being the most effective means of watering) is important to agriculture. • Drought can kill crops and increase erosion, while overly wet weather can cause harmful fungus growth. • Well irrigated plants suffer less from pests and diseases due to strong immune systems. • Heat stress is reduced and plants produce higher yield and better-quality produce. • Plants must have water in order to carry out photosynthesis • While a regular rain pattern is usually vital to healthy plants, too much or too little rainfall can be harmful, even devastating to crops.

Table 4:Significance of Parameters (part2)

Parameter	Significance
Air Parameters	• **Air Temperature-** o High temperature, even for short period, affects crop growth especially in temperate crops like wheat. High air temperature reduces the growth of shoots and in turn reduces root growth o The major impact of warmer temperatures is during the reproductive stage of development and in all cases grain yield in maize was significantly reduced by as much as 80−90% from a normal temperature regime • **Air Humidity-** o When conditions are too humid, it may promote the growth of mold and bacteria that cause plants to die and crops to fail, as well as conditions like root or crown rot o Humid conditions also invite the presence of pests, such as fungus gnats, whose larva feed on plant roots and thrive in moist soil • **Air Pressure-** o Air pressure in itself will alter the rate of gas exchange from the pores in the plant's leaves (stoma, lenticels, etc) which includes co2 and water. o If plant have differing air pressure, it likely means different elevation. Different elevation maintains the ideas of different gas exchange
Light Sensor	• Process of photosynthesis uses light energy which are used to combine CO2 with water to make sugars and O2. Higher the light intensity, more rapid this process • In places high light intensities, leaves are smaller and the internodal distance (distance between where leaves and branches emerge) will be shortened.

Table 5:Significance of Parameter (part3)

3 Infi IOT Cloud Platform:

Infi Iot is an Advance IoT development platform for connecting devices to the cloud and to remotely monitor and manage these devices.

Infi Iot provides features such as dashboards, automated data integration, Upload your past data, device alerts, notifications and the most amazing Analytics.

Get started with your project ideas connect your devices, control from platform, analyse your data and make future predictions with our unique IoT cloud platform

3.1 What's an IoT cloud platform?

To bring physical objects online and make them communicate, cooperate and act intelligently without human intervention, the Internet of Things relies on IoT platforms to enable provisioning, management, and automation of smart objects within a given IoT infrastructure. Generally speaking, each IoT environment is a mashup of technologies by various vendors that form a complex and inherently diverse ecosystem which, without a common base for their integration, would stay fragmented, 'dumb', and, ultimately, unable to function. Therefore, it can be said that an IoT platform provides a 'meeting point' for all the connected devices and serves to collect and handle the data they deliver over the network.

At the other end of the Internet of Things cloud solution, there is cloud computing. Breathing new life into IT services, cloud computing is the latest technology buzz that has moved consumer and business applications to the web, thus enabling enterprises to optimise their IT performance and reduce costs that would be otherwise bloated by the need of creating and maintaining on-site IT architecture for storing data and running applications. Cloud-based solutions are not only more cost-effective in the long run; they also provide better security, corporate data mobility, increased co-worker collaboration, more advanced disaster recovery solutions, to name only a few benefits. What is more, cloud computing offers more flexibility which helps to shift the company's focus from IT hosting-related issues towards aspects directly affecting its business bottom-line.

3.2 Features of Infi IoT Cloud Platform:

- Super Easy Integration
- Customizable data formats (Easy JSON Format)
- **No point count** restrictions (some clouds restrict amount of data you upload)
- **Future prediction** feature allows the trend of your data
- **Multi-Threaded Interface** prevents slow or non-responsive behavior
- Support AES **Secure Payloads**
- Supports many popular **IoT Protocols**
- High security User Authentication
- **No setup** needed at cloud side for sending first data
- **Export** your data as **csv** to local machine
- Interactive Widget for authentic personal experience
- Beautiful UI/UX for better visuals
- Consists of **Line Charts-** Single axis double axis, scatter plot, Bar chart, Pie Chart, **Labels, Tables, Indicators-** On/Off Indication, Gauge, Thermometer, **Controls-** Switch, Slider

3.3 Benefits of IoT Cloud Platforms:

Figure 5: Benefits of IoT Cloud Platforms

3.4 Design Flow of Infi Iot:

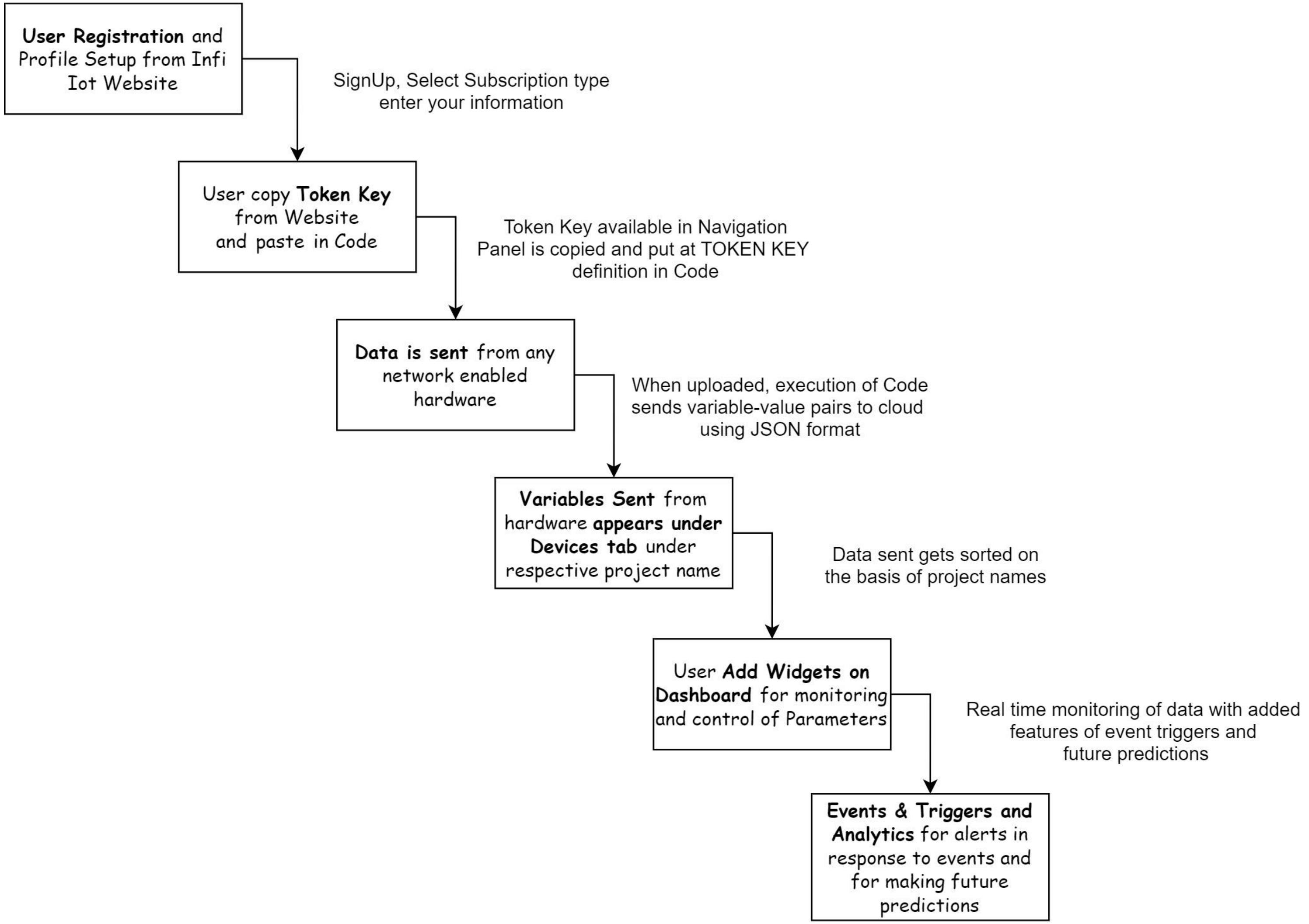

User Registration and Profile Setup from Infi Iot Website
SignUp, Select Subscription type enter your information
User copy Token Key from Website and paste in Code
Token Key available in Navigation Panel is copied and put at TOKEN KEY definition in Code
Data is sent from any network enabled hardware
When uploaded, execution of Code sends variable-value pairs to cloud using JSON format
Variables Sent from hardware appears under Devices tab under respective project name
Data sent gets sorted on the basis of project names
User Add Widgets on Dashboard for monitoring and control of Parameters
Real time monitoring of data with added features of event triggers and future predictions
Events & Triggers and Analytics for alerts in response to events and for making future predictions

Figure 6: Design Flow of Infi IoT Cloud Platform

4 Setting up things…

4.1 Tools needed:

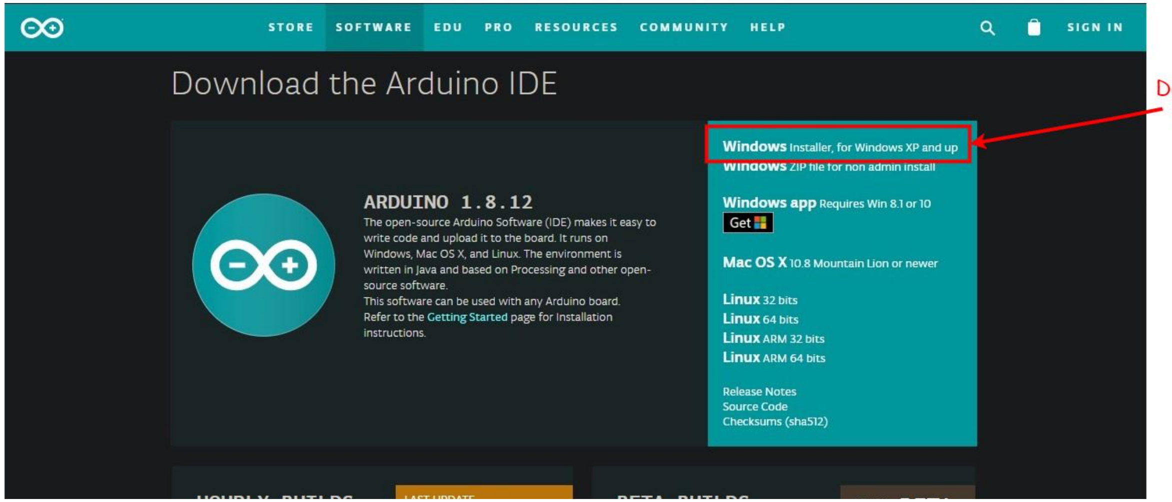
- **PC** – Windows PC with minimum 1 USB port, Java Virtual Machine 8u32 or later installed. OS: Windows 7, 8 or 10.
- **Arduino Ide** – Version 1.8 and above
- **ESP8266** Library
- **Infi Sense and Infi IoT Library** – Version 1.0 and above
- **Internet** Connection –
- **Web Browser – Latest** edition Browsers (Edge, IE, Chrome, Firefox, Safari) with Adobe Flash Player plugins

4.2 Installing Arduino Ide:

Important Resources: watch this: https://www.youtube.com/watch?v=TbHsOgtCMDc or read: https://www.arduino.cc/en/guide/windows

Step 1:

Get the latest version from https://www.arduino.cc/en/Main/Software. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually.



Download the Arduino IDE

ARDUINO 1.8.12

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for Installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10

Mac OS X 10.8 Mountain Lion or newer

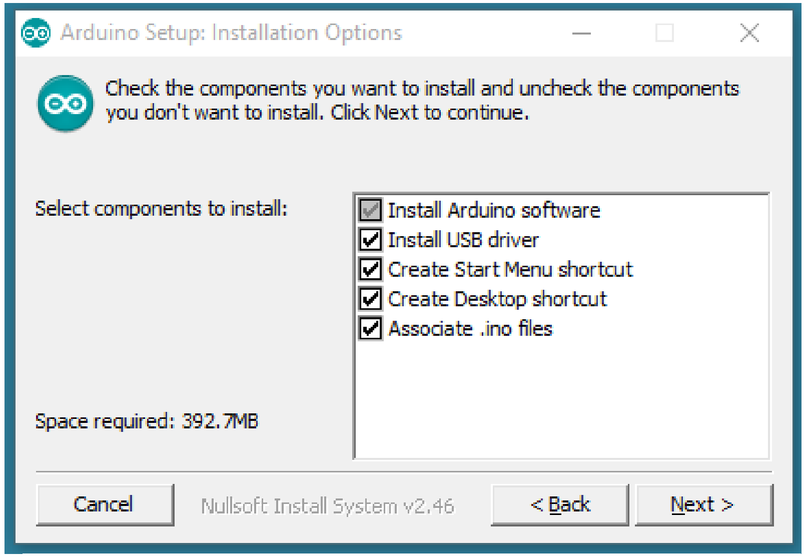
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits

Release Notes
Source Code
Checksums (sha512)

Download Installer for Arduino Ide

Figure 7:Arduino installation step 1

Step 2:



Arduino Setup: Installation Options

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

Select components to install:

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

Space required: 392.7MB

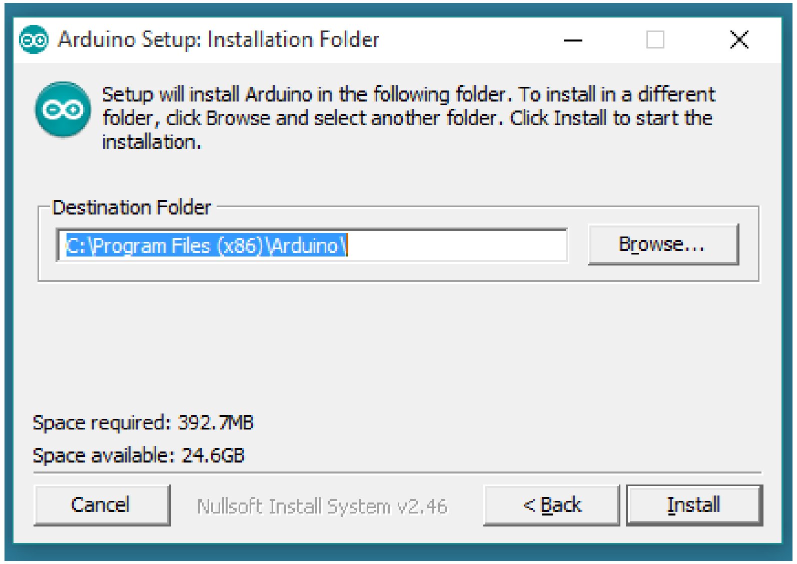
Figure 8: Arduino Installation step 2

When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.

Choose all components to install

Step 3:

Choose the installation directory (we suggest to keep the default one)



Arduino Setup: Installation Folder

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

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

Space required: 392.7MB

Space available: 24.6GB

Figure 9: Arduino Installation step 3

Step 4:

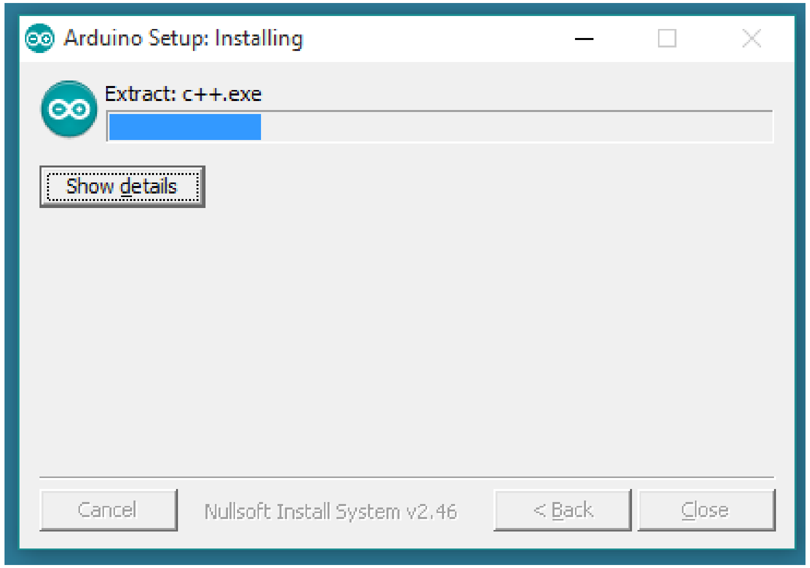
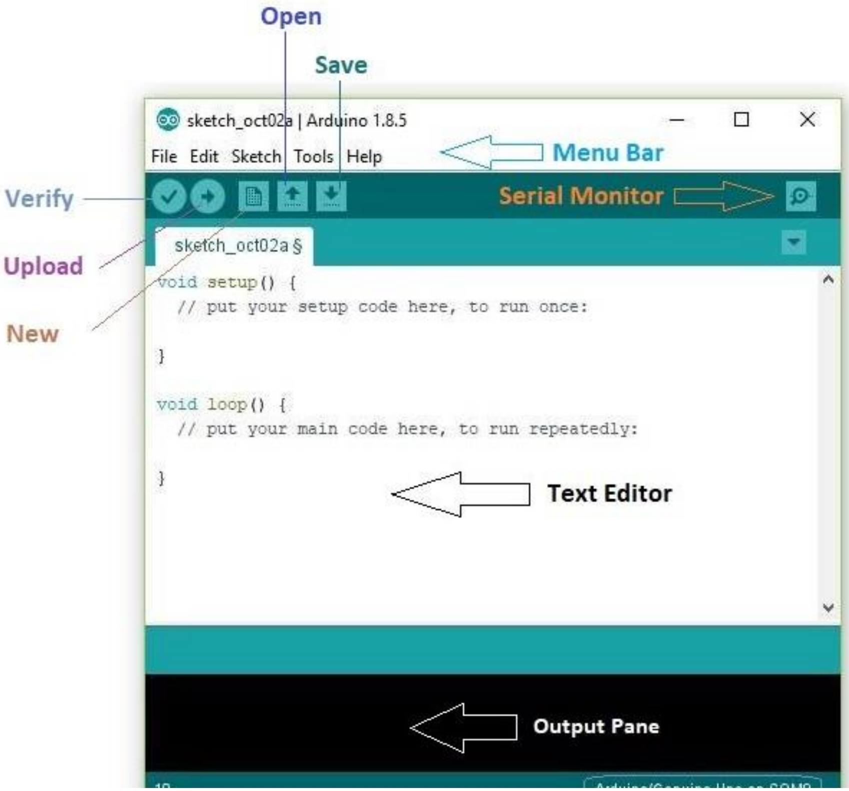


Figure 10: Arduino Installation step 4

The process will extract and install all the required files to execute properly the Arduino Software (IDE)

Step 5:



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

}

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

}
```

Figure 11: Arduino Installation step 5

This is how Arduino welcome page will look like:

Important areas (text Editor and Output Pane) and

Important Options (Verify, Upload, New, Open, Save, Serial Monitor)

are shown in the picture

4.3 *Installing ESP8266 board on Arduino IDE:*

Important Resources: watch this: https://www.youtube.com/watch?v=OC9wYhv6juM&feature=emb_logo or

Read this : https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/

Step 1:

Launch Arduino Ide

Go to File option on Menu bar

Select Preferences Option

Dialog Box will open with title

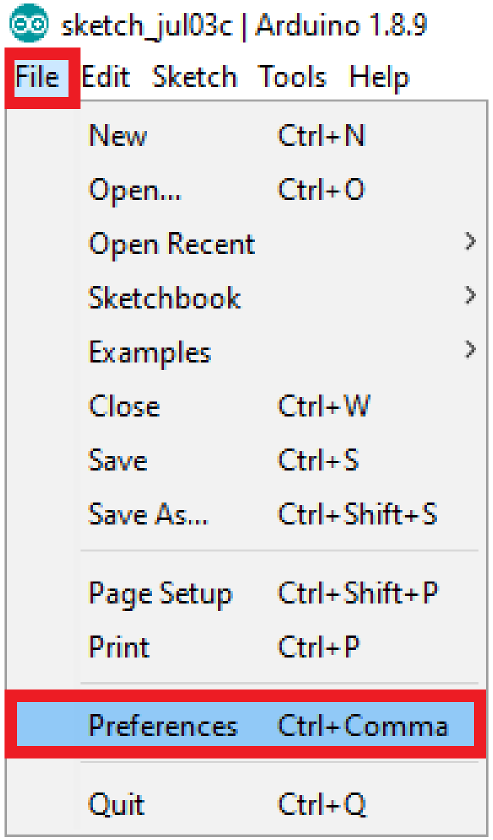
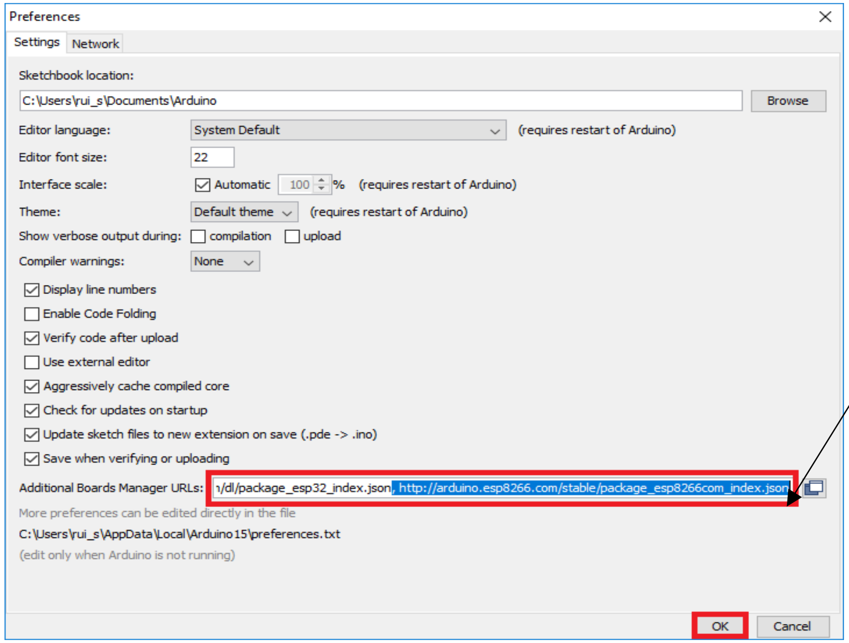


Figure 11: ESP8266 installation step1

Step 2:



Preferences

Settings | Network

Sketchbook location: C:\Users\rui_s\Documents\Arduino [Browse]

Editor language: System Default (requires restart of Arduino)

Editor font size: 22

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

☑ Display line numbers

□ Enable Code Folding

☑ Verify code after upload

□ Use external editor

☑ Aggressively cache compiled core

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

☑ Save when verifying or uploading

Additional Boards Manager URLs: ı/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json

More preferences can be edited directly in the file

C:\Users\rui_s\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

[OK] [Cancel]

Figure 13: ESP8266 installation step2

Enter:

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

into the “Additional Boards Manager URLs” field as shown in the figure below.

Note: if you already have the ESP32 boards URL, you can separate the URLs with a comma

Then, click the “OK” button:

Step 3:

Launch Arduino Ide

Go to File option on Menu bar

Select Preferences Option

Dialog Box will open with title

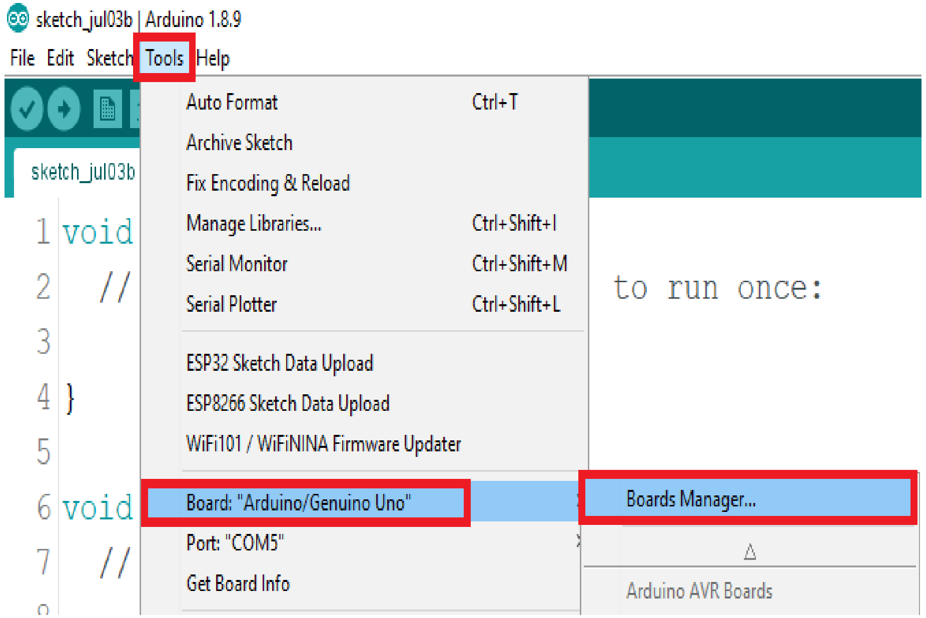
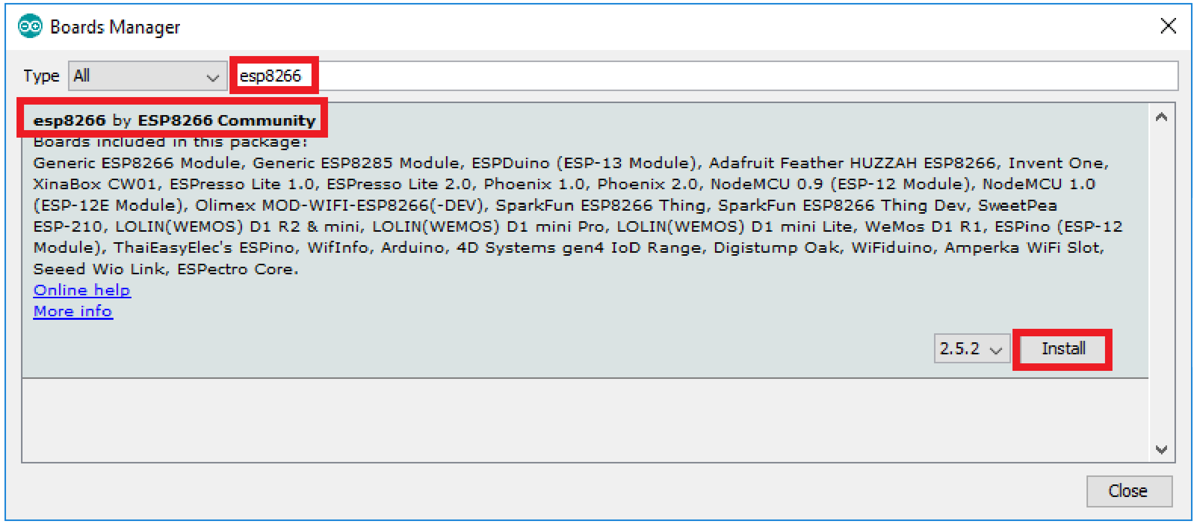

sketch_jul03b | Arduino 1.8.9
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
ESP32 Sketch Data Upload
ESP8266 Sketch Data Upload
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino/Genuino Uno"
Port: "COM5"
Get Board Info
Boards Manager...
Arduino AVR Boards
sketch_jul03b
1 void
2 //
3
4 }
5
6 void
7 //
to run once:

Figure 14: ESP8266 installation step 3

Step 4:



Boards Manager

Type All esp8266

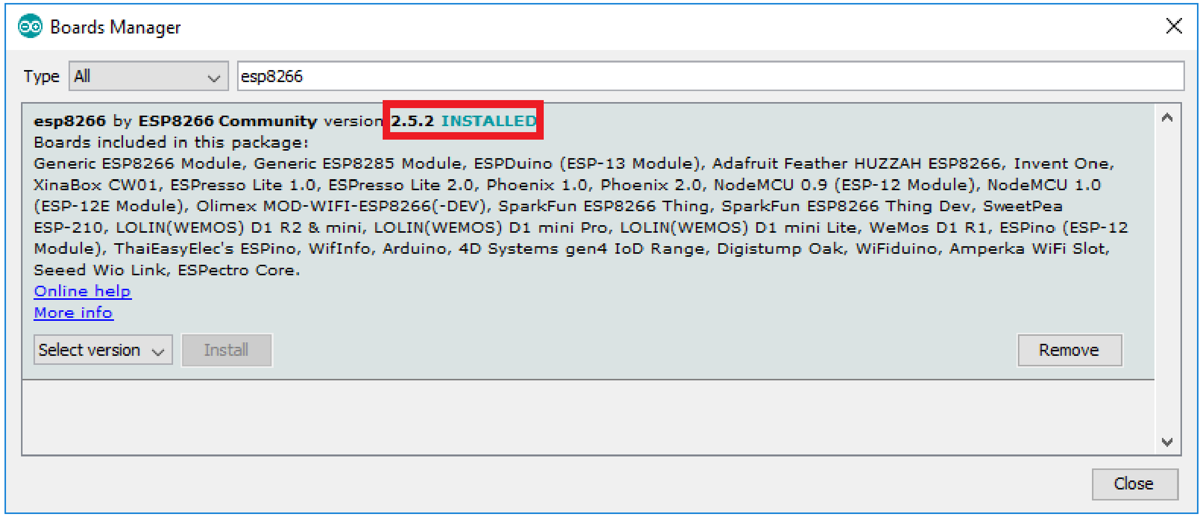
esp8266 by ESP8266 Community
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E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

Search for **ESP8266** and press install button for the “**ESP8266 by ESP8266 Community**”

Figure 15: ESP8266 installation step 4

Step 5:

That’s it. It should be installed after a few seconds.



Boards Manager

Type All esp8266

esp8266 by ESP8266 Community version 2.5.2 INSTALLED
Boards included in this package:
Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core.
Online help
More info

Figure 16: ESP8266 installation step 5

Step 6:

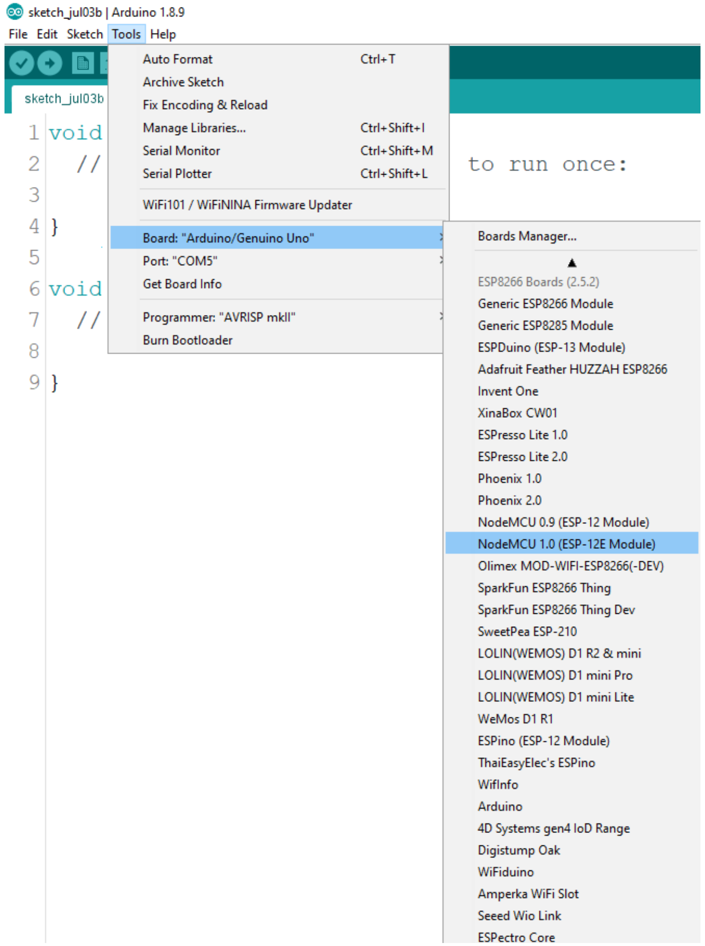
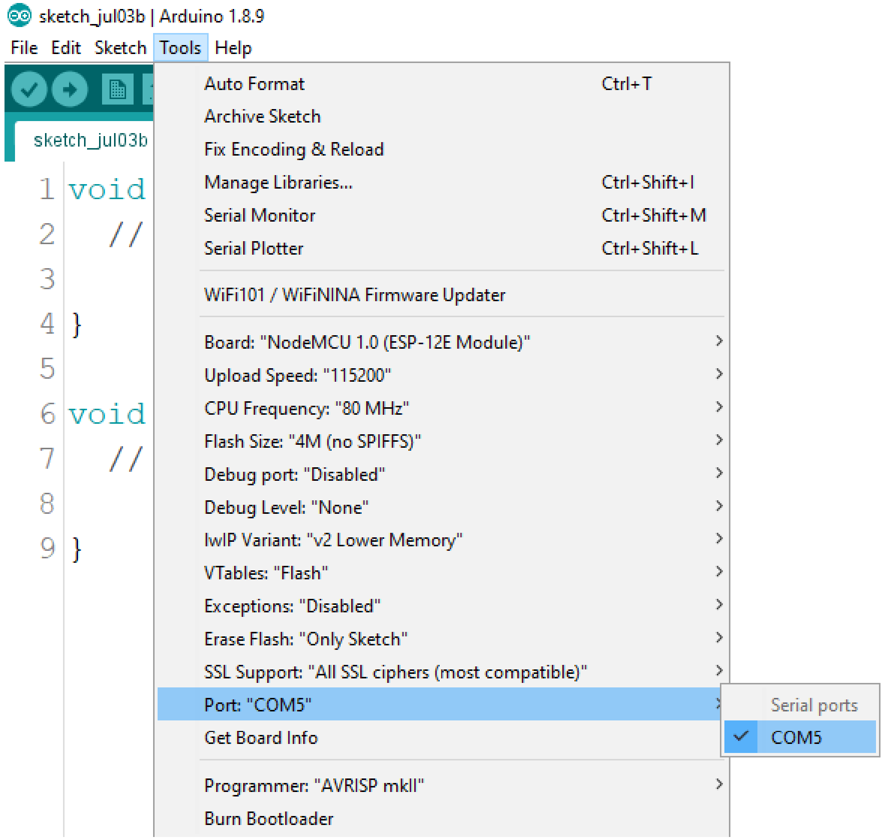


Figure 17: ESP8266 installation step 6

Verify Installation by Going to **Tools-> Boards->NodeMCU1.0(ESP-12) Module**

Note: If you don't see board try restarting Arduino IDE

Step 7:


sketch_jul03b | Arduino 1.8.9
File Edit Sketch Tools Help
sketch_jul03b
void
//
}
void
//
}
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Get Board Info
Burn Bootloader
Serial ports
✓ COM5

Board: "NodeMCU 1.0 (ESP-12E Module)"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4M (no SPIFFS)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Disabled"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
Port: "COM5"
Programmer: "AVRISP mkII"

Figure 18: ESP8266 installation step 7

Connect NodeMCU and look for available COM port

You can verify COM port device manager

4.4 Installing Infi Sense and Infi IoT Library:

Step 1:

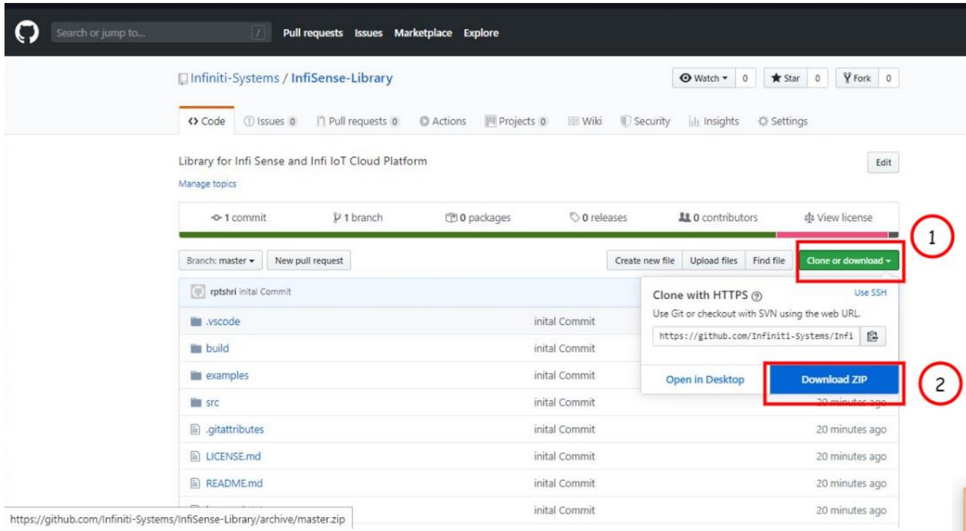

Search or jump to...
Pull requests Issues Marketplace Explore
Infiniti-Systems / InfiSense-Library
Watch 0 Star 0 Fork 0
Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings
Library for Infi Sense and Infi IoT Cloud Platform
Manage topics
Edit
1 commit 1 branch 0 packages 0 releases 0 contributors View license
Branch: master New pull request
Create new file Upload files Find file Clone or download
rptshri inital Commit
.vscode inital Commit
build inital Commit
examples inital Commit
src inital Commit
.gitattributes inital Commit 20 minutes ago
LICENSE.md inital Commit 20 minutes ago
README.md inital Commit 20 minutes ago
inital Commit 20 minutes ago
Clone with HTTPS
Use SSH
Use Git or checkout with SVN using the web URL.
https://github.com/Infiniti-Systems/Infi
Open in Desktop
Download ZIP
1
2
https://github.com/Infiniti-Systems/InfiSense-Library/archive/master.zip

Figure 12: Library install step 1

Open the given link in your browser:

https://github.com/Infiniti-Systems/InfiSense-Library

Click on “Clone or download” (1)

Click on “Download Zip” (2)

Step 2:

Open Arduino IDE

Go to Sketch->**Include Library->Add .Zip Library...**

In the Dialog box Select the Zip downloaded in previous step

Wait for the message “Library added to your libraries”

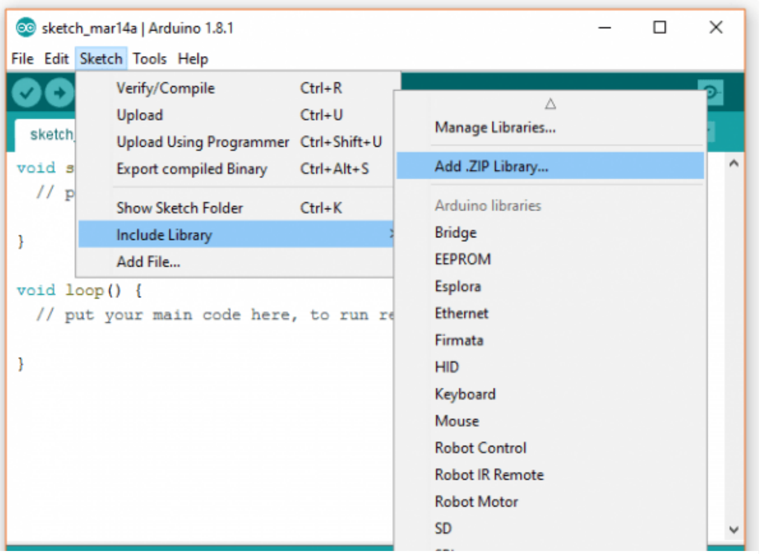


Figure 20: Library install step 2

Step 3:

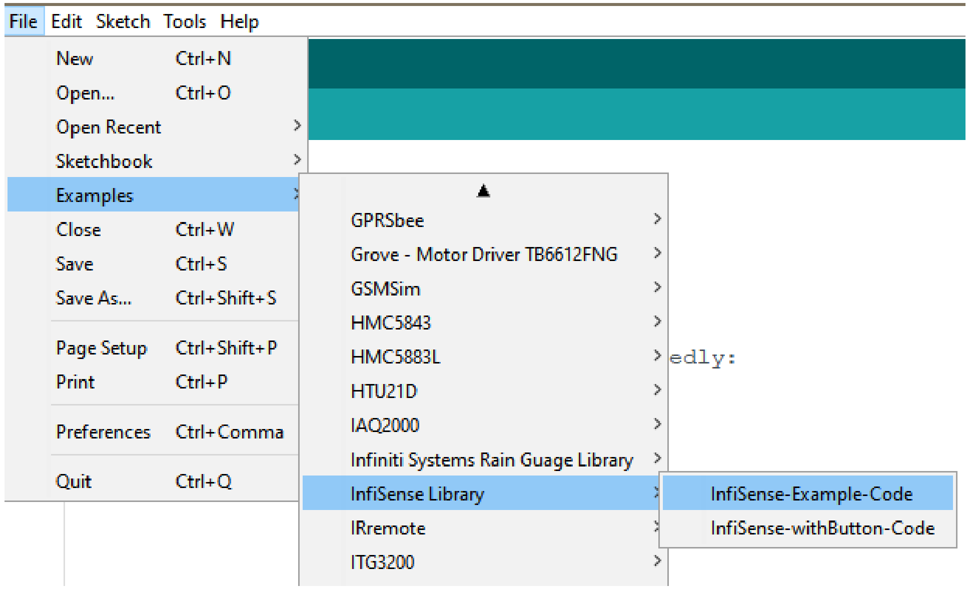


Figure 21: Library install step 3

Once library is added

Go to **File->Examples->InfiSense Library->InfiSense-Example-Code**

It will launch the InfiSense example code

NOTE: For latest updates of all the installation procedures and Library API refer **README.md** file on link:

https://github.com/Infiniti-Systems/InfiSense-Library

4.5 Getting Started (Connecting Infi Sense Hardware and Running First Example Code):

Step 1:

Connect NodeMCU using USB Micro-B cable

Launch Arduino Ide

Select Board as shown in Figure 17: ESP8266 installation step 6

Or Go to **Tools->Boards->NodeMCU1.0 (ESP-12E) module**

Step 2:

Connect NodeMCU using USB Micro-B cable

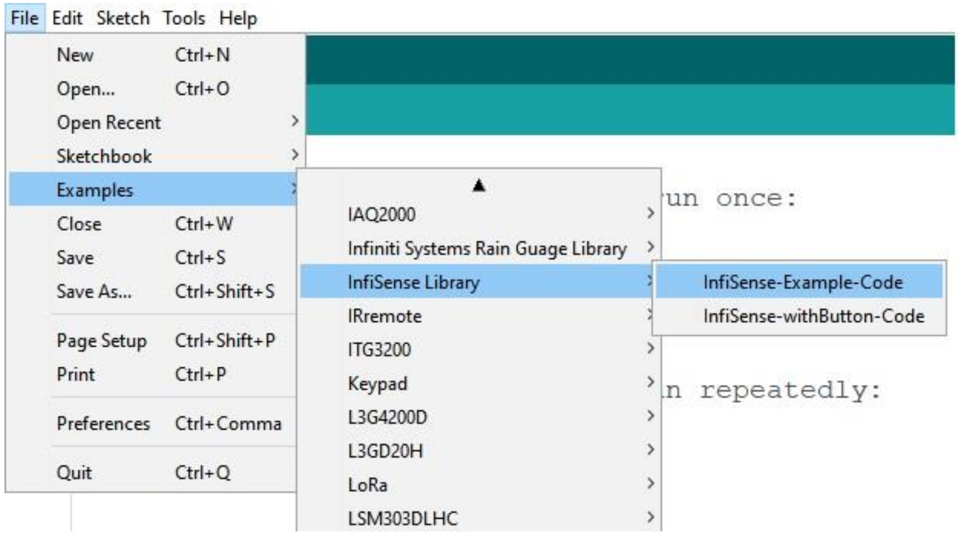
Select Port as shown in Figure 18: ESP8266 installation step 7

Or Go to **Tools->Port-> "**Select your COM Port number**"**

Step 3:

Now Go to:

File->Examples->InfiSense Library->InfiSense-Example-Code



Step 4:

Now we need to provide the Code with:

a. Token Key

For Token we need to Sign-in/Sign-up for account on
Infi Iot cloud: http://www.infiiot.com

b. Wi-Fi SSID and Password
Create a Wi-Fi Hotspot and fill the credentials into code

c. Project Name
Any name of your choice

Step 4:

Copy the Token Key from Infi IoT Cloud from the navigation bar and paste it inside the example code at definition of Token

Step 5:

Now we connect all the sensors by matching the label on the sensor wire probe with the labels on the hardware

Once done with connecting all the sensor we will now upload the code into micro-controller

Step 6:

For Uploading the Code Go to:

Sketch->Upload or click on the **round → button** below the menu bar

Wait for program upload, Once Done you can see Done Uploading message in the Output Pane

Step 7:

Now we open Serial Monitor to have a glimpse of collected data and its delivery to the cloud

Go to: **Tools->Serial Monitor** or click the **magnifying glass button** under menu bar

Now Set the baud rate at bottom right corner dropdown to 9600

Step 8:

Now we move to Infi Iot Cloud to see delivered data online and analyze it

First, we sign-in to the account

Then, Move to the Device section from the navigation bar

Under Devices, select your Project name and click “View”

You can see all the variables on the panel, select any variable by clicking on “View Table” or “View Chart”

Make sure that latest readings are being updated on to the cloud (there can be some delay between data send and update)

Step 9:

Now we **create a Widget on dashboard:**

Go to dashboard Option in Navigation Panel

Click on “**Add Widget**” Button on the top right side

Select “**Line Chart**” Widget option (or you can select any different)

Select the **variable** you want to plot by selecting the Project name from Project dropdown and then Select the variable(s)

Provide other information such as **Title, axis labels, max/mins, unit,** etc.

The Widget gets added and data is updated on dashboard regularly

Step 10:

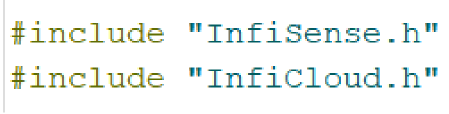
Now we move on to making **Future Predictions:**

Click on “**Prediction**” Section from navigation panel

There are three Sections in Predictions Section

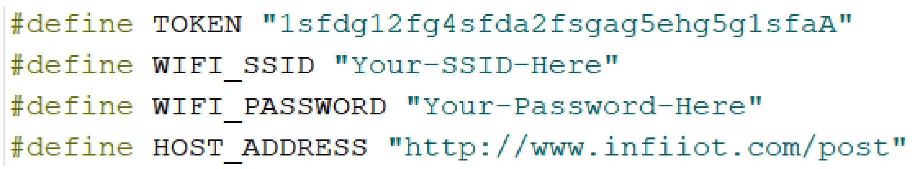
1. **Include data**
   -Select variable or upload CSV file and plot it
   -Select the number of points to predict
   -Select the Algorithm (ARIMA works best for all data types)
2. **Train the Model**
   -Training may take some time (please: read Note**)**
3. **See the Predictions**
   -Finally, we can see future prediction of the data trend

4.6 Example Code Snippets with Explanation:



```
#include "InfiSense.h"
#include "InfiCloud.h"
```

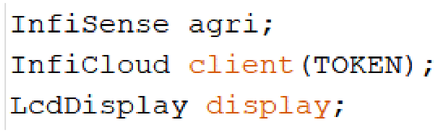
First, we Include InfiSense And InfiIoT library to the code



```
#define TOKEN "1sfdg12fg4sfda2fsgag5ehg5g1sfaA"
#define WIFI_SSID "Your-SSID-Here"
#define WIFI_PASSWORD "Your-Password-Here"
#define HOST_ADDRESS "http://www.infiiot.com/post"
```

#define is a useful C++ component that allows the programmer to give a name to a constant value before the program is compiled.

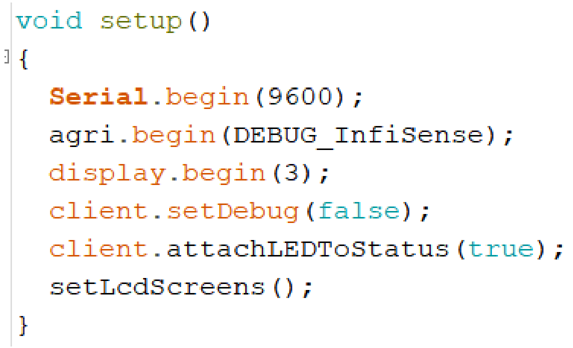
Here we define token, Wi-Fi Credentials and host address.



```
InfiSense agri;
InfiCloud client(TOKEN);
LcdDisplay display;
```

Then we make objects (agri, client and display) for the included Library classes

Note that we need to pass TOKEN while creating object of InfiCloud class



```
void setup()
{
  Serial.begin(9600);
  agri.begin(DEBUG_InfiSense);
  display.begin(3);
  client.setDebug(false);
  client.attachLEDToStatus(true);
  setLcdScreens();
}
```

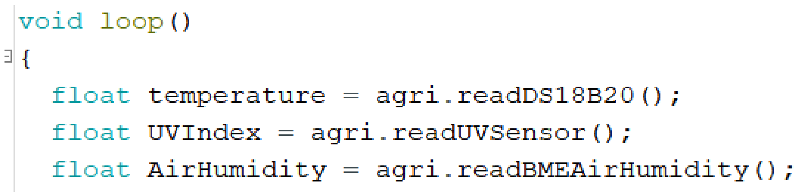
Code inside **void setup ()** only runs at the beginning of the code

We define the baud rate to 9600 using Serial.begin(9600) command

.begin() function is used to initialize the objects which we just created

We also set the debug values, true- if we want to see the detailed working of code

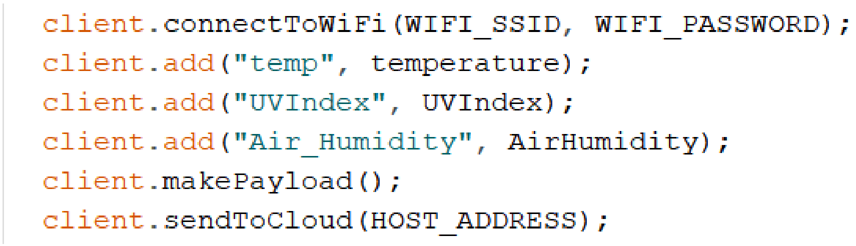
False- if we don’t want details on the Serial Monitor



```
void loop()
{
  float temperature = agri.readDS18B20();
  float UVIndex = agri.readUVSensor();
  float AirHumidity = agri.readBMEAirHumidity();
```

void loop () is the code which runs continuously

we then make variables with float data type and call the functions (readDS18B20(), readUVSensor(), readBMEAirHumidity(), etc.) which return the values from the respective sensors



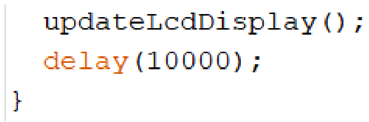
```
  client.connectToWiFi(WIFI_SSID, WIFI_PASSWORD);
  client.add("temp", temperature);
  client.add("UVIndex", UVIndex);
  client.add("Air_Humidity", AirHumidity);
  client.makePayload();
  client.sendToCloud(HOST_ADDRESS);
```

We first connect to Wi-Fi using the function **connectToWiFi(char SSID, char Password)**

Then we add the variables to data string to be sent to cloud using function **add(char variable_id, float value)**

Then we call **makePayload()** to make the Data String

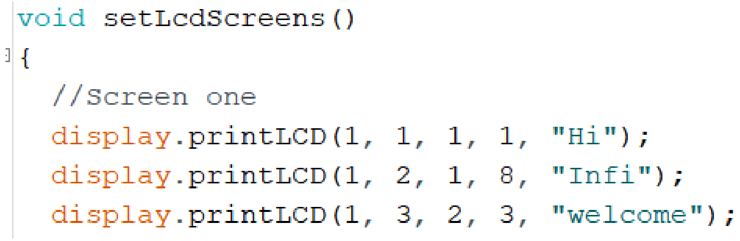
Call **sendToCloud(HOST_Address)** to send data to the cloud



```
  updateLcdDisplay();
  delay(10000);
}
```

Update the LCD by calling **updateLcdDisplay()** function

Then we give delay of 10000ms i.e. 10sec, to make sure data gets sent every 10seconds



```
void setLcdScreens()
{
  //Screen one
  display.printLCD(1, 1, 1, 1, "Hi");
  display.printLCD(1, 2, 1, 8, "Infi");
  display.printLCD(1, 3, 2, 3, "welcome");
```

In the definition of setLcdScreens() function:

We tell the code what to print on the LCD Display by using the function

printLCD(int screen_num, int instance_num, int row_num, int column_num, char* data)

This function run only once because it is called in setup() function

5. Library API reference guide:

5.1 InfiSense Library:

`.begin(Boolean DEBUG)`

Initialize Infi Sense library, sets the DEBUG value, prints welcome message on the serial monitor, **Returns** nothing.

`.readSoilSensor()`

Initializes ADS(adc) channel 0 for reading soil sensor, take 30 readings with moving average, map sensor values from 0 to 65535 into 0 to 100, **Returns** Soil Moisture in percentage.

`.readML8511() or .readUVSensor()`

Initializes ADS(adc) channel 1 for reading UV sensor, take raw readings and convert them, **Returns** UVindex in mW/cm^2.

`.readRainSensor()`

Initializes ADC 0 channel of microcontroller to read Rain Sensor, Convert raw values into Rain percentage and **Return** Rain percentage.

`.readBMEAirHumidity()`

Checks for valid BME280 Sensor, if invalid return 9999, if valid reads air humidity from BME Sensor and **Return** Air Humidity values

`.readBMEAirTemp()`

Checks for valid BME280 Sensor, if invalid return 9999, if valid reads air temperature from BME Sensor and **Return** Air Temperature values in °C

`.readBMEAirPressure()`

Checks for valid BME280 Sensor, if invalid return 9999, if valid reads air pressure from BME Sensor and **Return** Air Pressure in hPa

.readBMEAltitude()

Checks for valid BME280 Sensor, if invalid return 9999, if valid reads altitude from BME Sensor and Return Altitude values in meters.

.readBH1750() or .readLUXSensor()

Initialize I2C communication with sensor, reads LUX intensity values from Sensor and **Return** LUX Intensity in lx.

.readBatteryVoltage()

Initializes ADS 3 channel of microcontroller to read Rain Sensor, Convert raw values into Rain percentage and **Return** Rain percentage.

.triggerRelayOn() or .triggerPumpOn()

Tuns on the Pump relay connected to GPIO13 (D7), **Returns** nothing.

Note: For Operation of pump 9V-12V must be available at the main power input

.triggerRelayOff() or .triggerPumpOff()

Tuns off the Pump relay connected to GPIO13 (D7), **Returns** nothing.

.turnLedOn() or .triggerLedOn()

Tuns on the LED connected to connected to GPIO12 (D6), **Returns** nothing.

.turnLedOff() or .triggerLedOff()

Tuns off the LED connected to connected to GPIO12 (D6), **Returns** nothing.

.attachButton()

Attach button to GPIO2(D4), **Return** nothing.

.readButton()

Reads button on GPIO2(D4), **Return** true if pressed or false if not pressed.

5.2 LcdDisplay Library(included):

```
LcdDisplay object_name
```

Initializes object of class LcdDisplay, **Return** nothing

```
.begin(int Number_of_Screens)
```

Initializes the Lcd, turns on backlight, **Return** nothing.

```
.printLCD(int Screen_Num, int instance_num, int row_num, int Column_num, char* Data)
```

Gets the Data from user based on which Screen to print, instance number is the total number of data instances on screen, row number and column number.

e.g:

5.3 InfiCloud Library:

```
InfiIot object_name(char *TOKEN_KEY, char *PROJECT_NAME)
```

Initializes object of class InfiIot, Saves the token Key and the Project Name, **Return** nothing

```
.setDebug(bool DEBUG)
```

Sets Debug to true or false, **Return** nothing

```
.add(char *variable_id, float value)
```

This function adds the variable value with the variable id(name) into the data Payload, **Return** nothing

```
.makePayload()
```

This function makes data payload including token, project name and data to be sent and **Returns** the Payload String

```
.connectToWiFi(char* WIFI_SSID, Char* WIFI_PASSWORD)
```

Scans for Wi-Fi hotspot using the SSID and Password and connect if found, **Return** local IP Address of ESP Controller.

```
.sendToCloud()
```

Sends the data to Cloud and **Return** http Code response from cloud.

```
.attachLedToStatus()
```

Attach the LED to status response from cloud, shorter On duration of LED blink if data sent successfully and longer on duration of LED if data not sent successfully

6. Troubleshooting:

7. Glossary of terms:

- **Actuators**
- **AES Secure Payloads**
- **Analog-to-Digital Converters**
- **Anti-Conductivity**
- **Anti-Oxidation**
- **API**
- **Big Data**
- **dashboards**
- **Deep Sleep**
- **Embedded Systems**
- **Embedded Operational Amplifier**
- **Gateway**
- **JSON Format**
- **male header**
- **Non-Corrosive**
- **Interface**
- **IoT Protocols**
- **IP address**
- **PH2.0-3P**
- **Photosynthesis**
- **Phoenix connection**
- **Proof-of-Concept**
- **Submersible Pump**
- **Token Key**

- **Query Time**

Sales and Design Support

For any queries related to products and services feel free to visit Infiniti Systems Website: www.infinitisystems.co.in

Products

Infi IoT	infinitisystems.co.in/InfiIoT
Infi Drive	infinitisystems.co.in/infidrive
Infi Sense	infinitisystems.co.in/infisense
Infi Mesh	infinitisystems.co.in/infimesh
Industry Grade 3D Printers	infinitisystems.co.in/printer
Remote Industry Power Analysis	infinitisystems.co.in/poweranalysis
Infi Weather	infinitisystems.co.in/infiweather
Infi Gauge	infinitisystems.co.in/infigauge
Infi Leaf	infinitisystems.co.in/infileaf
FARU Logger	infinitisystems.co.in/farudatalogger
Infi Soil/ECE	infinitisystems.co.in/infisoil

Design Solutions:

Hardware Design Embedded Software Product Design

Web Solutions:

Website Design Backend Design Digital Marketing

TO THE EXTENT PERMITTED BY APPLICABLE LAW, INFINITI SYSTEMS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Infiniti Systems reserves the right to make changes to this document without further notice. Infiniti Systems does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Infiniti Systems products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, lifesupport devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Infiniti Systems is not liable, in whole or in part, and you shall and hereby do release Infiniti Systems from any claim, damage, or other liability arising from or related to all Unintended Uses of Infiniti Systems products. You shall indemnify and hold Infiniti Systems harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Infiniti Systems products.

Infiniti Systems, the Infiniti Systems logo, trademarks or registered trademarks of Infiniti Systems in the India and other countries. For a more complete list of Infiniti Systems trademarks, visit Infiniti Systems.com. Other names and brands may be claimed as property of their respective owners.

NOTES: