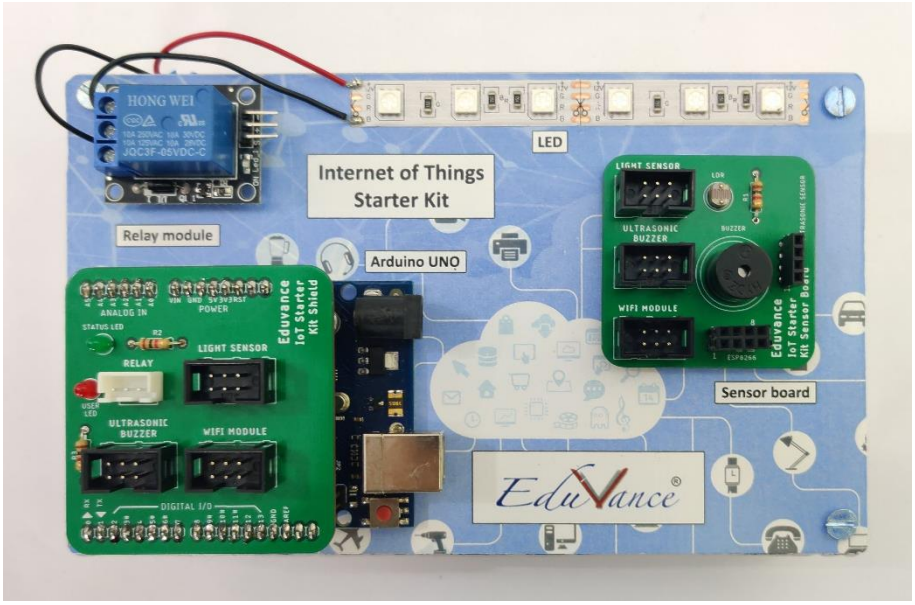
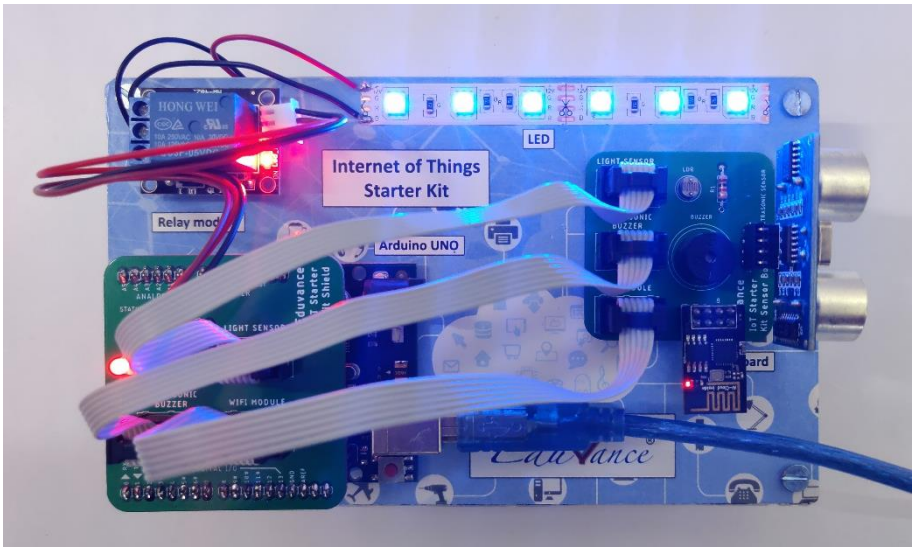
Introduction to IoT Starter Kit

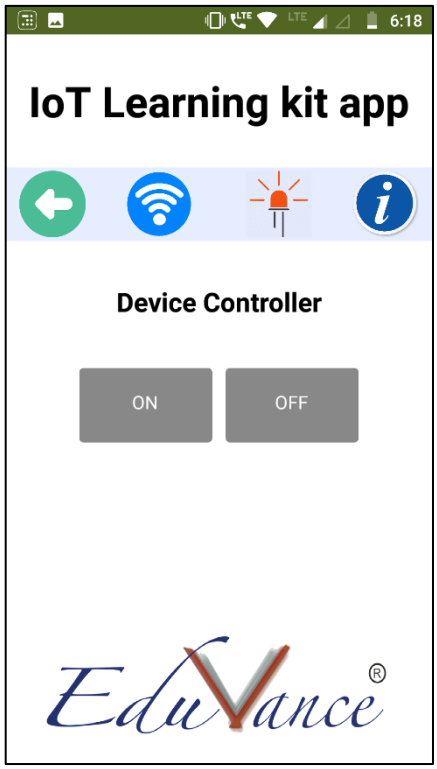
The IoT Starter Kit by Eduvance® introduces students to the exciting world of Internet of Things (IoT). The starter kit focuses on understanding the way Internet of Things work to such an extent that even a non-technical person can use this kit.

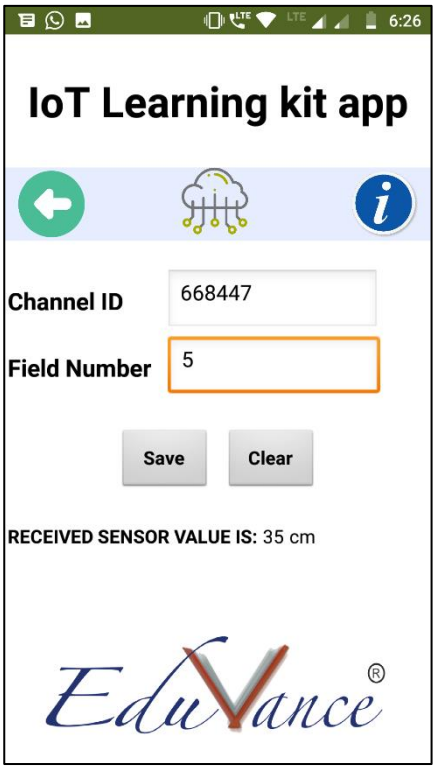
The starter kit contains various components which range from different types of sensors to measure various physical parameters, modules which help you connect to the cloud through internet, control different devices, visualize data and many more things. All this can be done by connecting simple cables from the board containing all the sensors to the board containing the main controlling device underneath it.





The starter kit also includes a basic Android application which will help you control electronics devices and download data from the cloud on your smartphone. The simple yet elegant user interface will help you understand the concepts in a very reliable way.





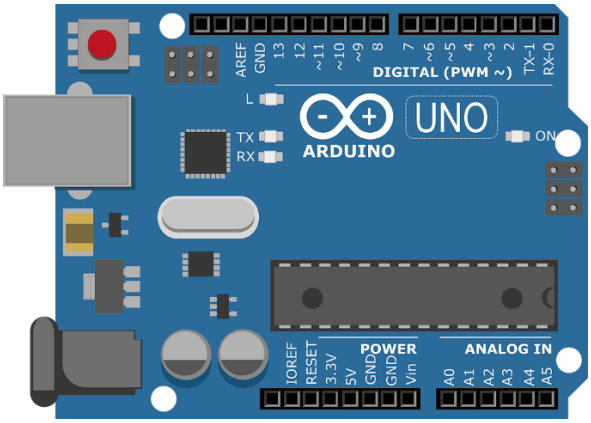
Using the IoT Starter Kit you will understand how to monitor sensor values from a remote location and access it from anywhere in the world with the help of IoT cloud. You will learn how to properly make use of specific sensors in your project and perform a particular task by analyzing and processing the output values from those sensors.

Using this kit, you can make exciting projects like Drawer Alarm System which will notify you whenever anyone tries to access your personal stuff, Automatic Night Lamp which will switch on the lights as soon as its dark and many more. You will be able to control lights and other devices using the Android application and also access data from the cloud directly into your Android smartphone.

Components in IoT Starter Kit

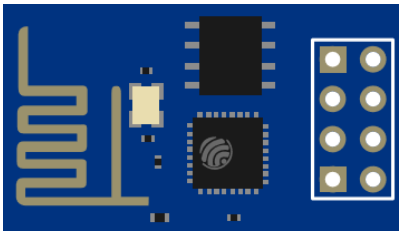
Arduino UNO

Arduino is an open-source programmable circuit board that can be integrated into a wide variety of simple and complex projects. This board contains a microcontroller which can be programmed to sense and control objects in the physical world. By responding to inputs from the sensors, the Arduino is able to interact with a large number of output devices such as LEDs and other modules. Hence, because of its flexibility and low cost, Arduino is the most popular choice to get started with Internet of Things.



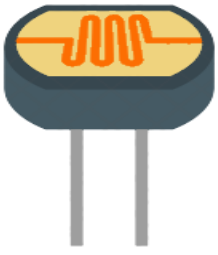
Wi-Fi Module

The Wi-Fi module ESP8266 is a highly integrated chip designed to provide full internet connectivity in a small package. The Wi-Fi module can connect to the internet wirelessly which in turn enables us to easily access the cloud services. We will use the Wi-Fi module to control devices wirelessly through the app.



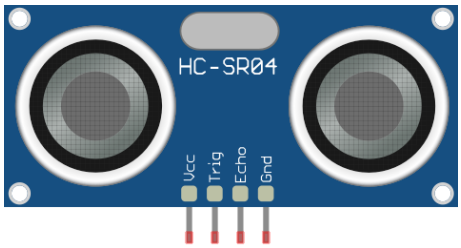
Light Sensor

A Light Sensor is a light sensitive device. This device depends on the light. When the amount of light falling on the light sensor increases, the sensor value increases and as the amount of light decreases the, sensor value decreases.



Distance sensor

Ultrasonic sensor works by emitting the sound waves at a very high frequency. The distance is measured by the time required for the sound wave to reflect back from the detected object. This is similar how radar works in a naval battle ship, but on a much smaller scale. Therefore, the output value of the sensor is proportional to the distance of the object.

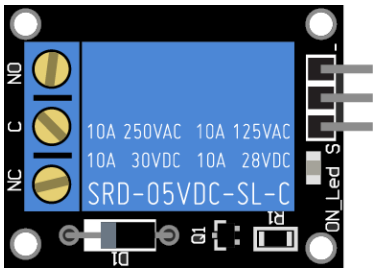


LED Strip

LED strip lights are new form of lighting which consists of many individual LED's. This LED strip is mainly used as output to see any changes due to change in sensor values.

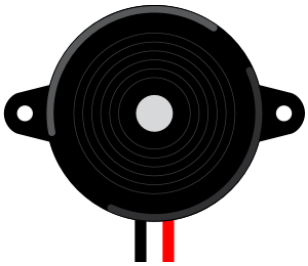
Relay Module

A relay is an electrically operated switch that can be turned on or off which can be controlled with low voltages, like the 5V provided by the Arduino pins to provide a much higher voltage. The LED strip works on a higher voltage as compared to the voltage provided by the Arduino UNO. Hence, a relay is used to operate it. By changing connections of the relay module, we can also connect and control a household light bulb instead of the LED Strip.



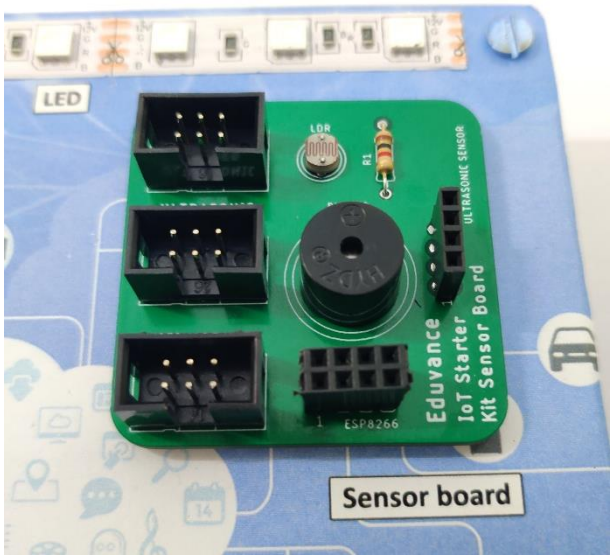
Buzzer

A buzzer is an audio signaling device. Similar to the LED strip, buzzer is mainly used as output device to detect any changes due to change in sensor values.



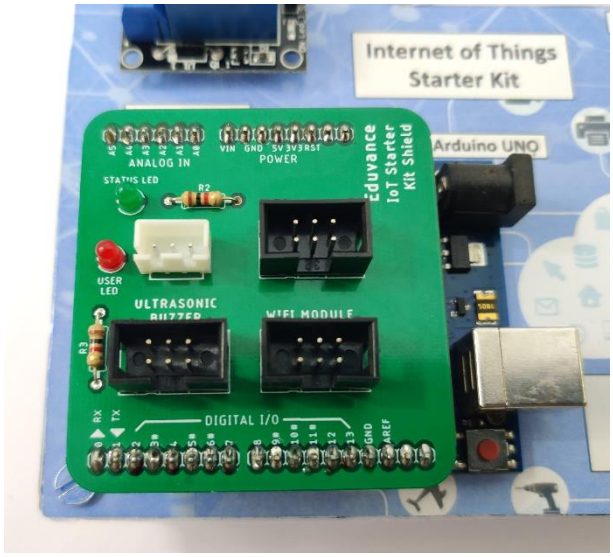
Starter Kit Sensor board

This board consists of Light Sensor, Distance sensor, Wi-Fi Module and connector blocks where all the pins of the sensor as well as the module are connected internally. Due to this, the need for using breadboard or soldering to design projects is entirely eliminated.



Starter Kit Arduino Shield

This Shield is connected on the Arduino board and contains all the connector blocks with proper naming for the purpose of connection. The Starter kit Arduino Shield is used to connect the Arduino UNO to the Starter Kit sensor board. It also has a power LED to indicate whether the Arduino which it is connected to is on or off.



Cables

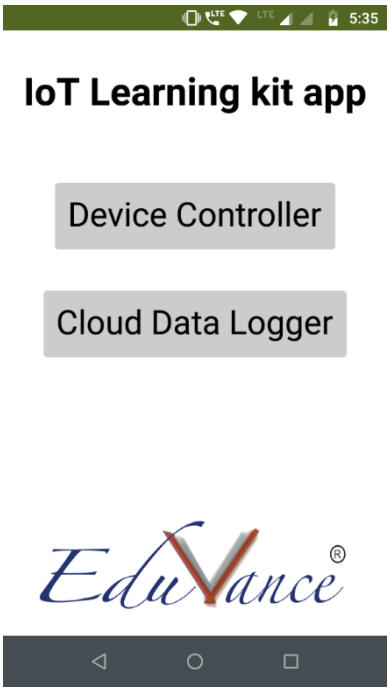
The kit contains group of wires known as ribbon cable which is mainly used to connect the Starter kit Sensor board and Starter Kit Arduino Shield so that we could interface and use the sensors and module by programming the Arduino UNO.

Battery

A small 9V battery is needed to power the LED strip through the relay. The battery is connected to the relay module through a connector.

IoT Starter Kit App

The IoT Starter Kit includes an Android app which serves the purpose of giving you access to control output devices connected with the kit through the Wi-Fi module, downloading the sensor data uploaded on the IoT Cloud and viewing the same on your phone by simply entering your IoT Cloud details.



Learning Outcomes

1. Arduino UNO is one of the most commonly used microcontroller-based board used by hobbyists and developers around the world. Hence, by using the IoT Starter kit you will gain basic knowledge and have a pretty good exposure along with hands-on experience with Arduino UNO which will help you in a long run to develop your own projects.

2. You will understand how a common IoT network works; right from connecting to the internet to connecting to the cloud and using services like data logging, data visualization and many more things.

3. You will learn how to operate the Wi-Fi module at a basic level. You will gain basic knowledge of the sensors and modules used in IoT Starter Kit. You will also learn how to use the sensor output for your own purpose like controlling other output modules.

4. You will learn how to get the output from the sensor, process it, upload it to the cloud, use the uploaded data to perform a particular action accordingly. You will also learn how to download the sensor data from the cloud.

5. Using the app included with the kit, you will learn how to control various devices using your Android smartphone and Wi-Fi module. Using the app, you will also learn how to read the sensor data from the cloud.

General Instructions

1. Make all the necessary connections by unplugging the Arduino USB cable from its power source since the Arduino USB cable serves both purposes of programming the Arduino as well as providing the necessary power to it.

2. When you want to connect wires from the Starter kit Sensor board to the Starter kit Arduino Shield while performing a project, connect the two connector blocks of the same name mentioned above the blocks on their respective boards. In case you connect it to some other connector block, you may end up damaging the kit.

3. Make the proper connections on the kit for the respective projects to be performed. Avoid unnecessary connections.

4. Always ensure that the Starter kit boards do not come in contact with any metal objects or liquids.

5. Connect the battery with correct electrical polarity. Precaution's should be taken not to short-circuit battery terminals to each other which can damage the product and generate heat that may result in burns.

6. If the battery exhausts, please purchase a new one because the battery included in the Starter kit is a non-rechargeable type Battery. Never attempt to recharge a non-rechargeable battery.

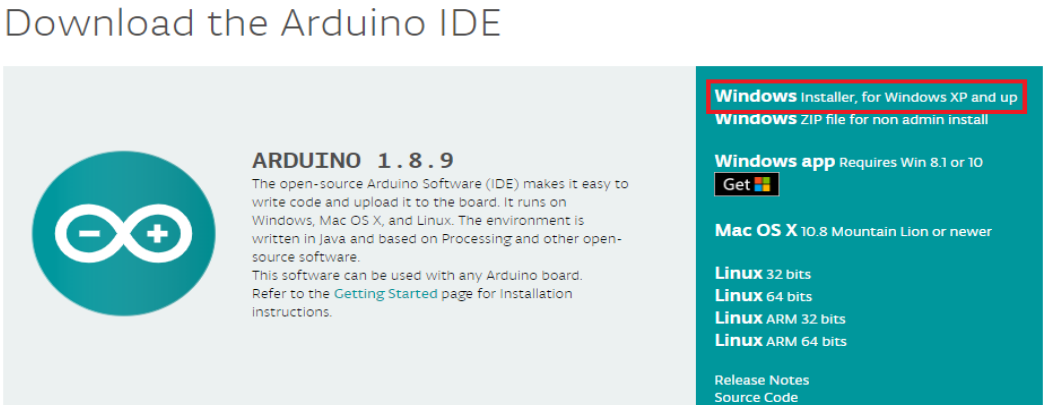
Prerequisites

Arduino IDE

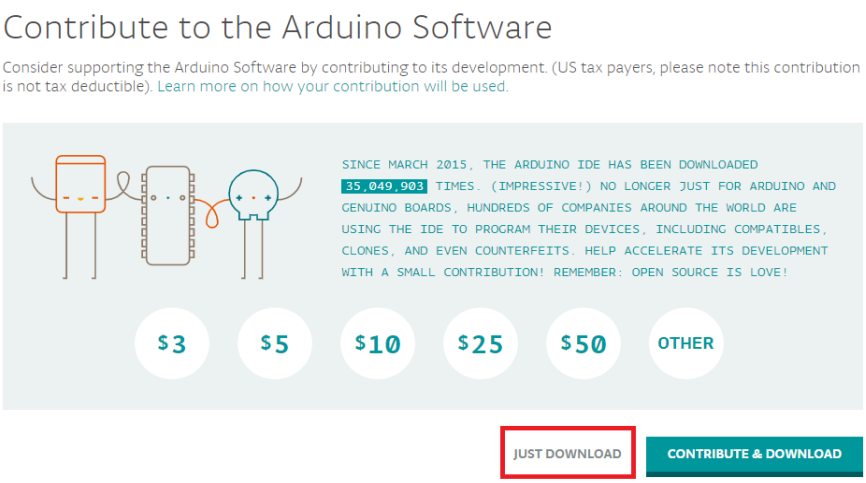
1. To get started with the IoT Starter Kit, the first and foremost thing you need to install Arduino Integrated Development Environment (Arduino IDE) as it is the software which helps us program the Arduino UNO board through your laptop or PC.

2. To install Arduino IDE on any machine, first open any web browser and type "**Arduino.cc**" in your search tab and press enter. Then click on **"Downloads".**

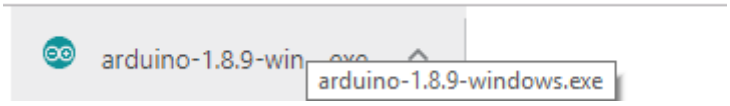
3. On the next page click on the installer type depending upon your system. In this manual, we'll show the process for Windows OS. So, click on "**Windows installer, for Windows XP and up**".



4. After that, click on "**Just Download**" and the file will start downloading.

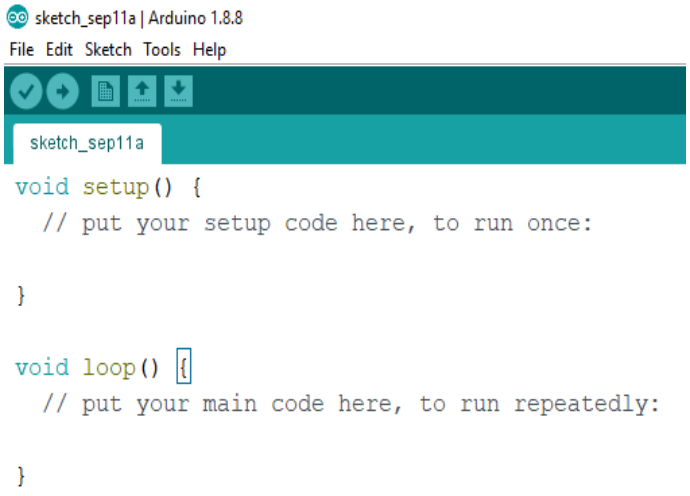


5. After downloading the file successfully, click on the **Arduino-x.x.x-windows.exe** file. An installation dialogue box will appear. Click on "**yes**" follow the steps indicated by the installation dialogue box. Once the software is installed, click on "**Finish**".

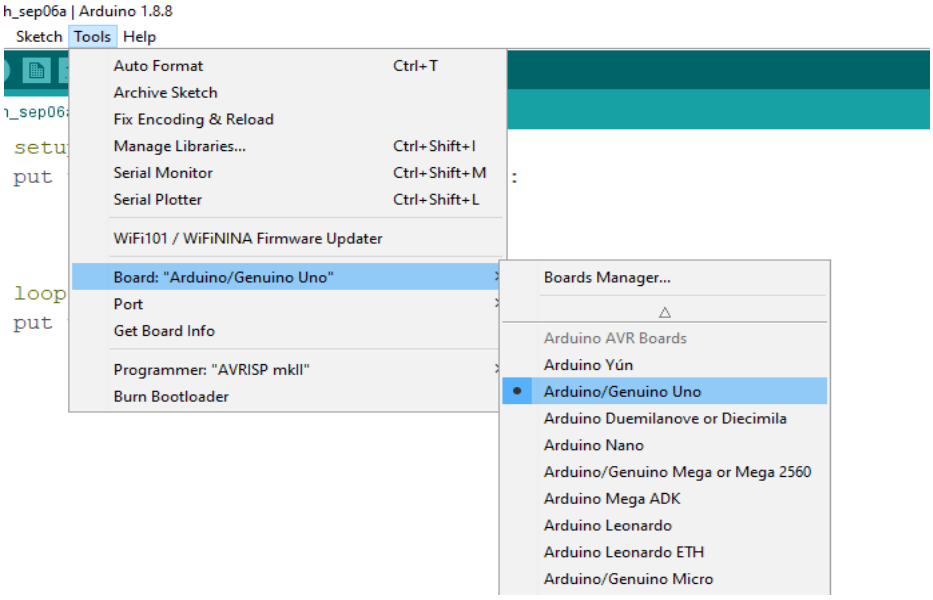


6. Double click on the **"Arduino"** shortcut on your desktop screen or simply search for **"Arduino"** and click **"Enter".** You will see a loading screen and after some time the Arduino IDE will open.

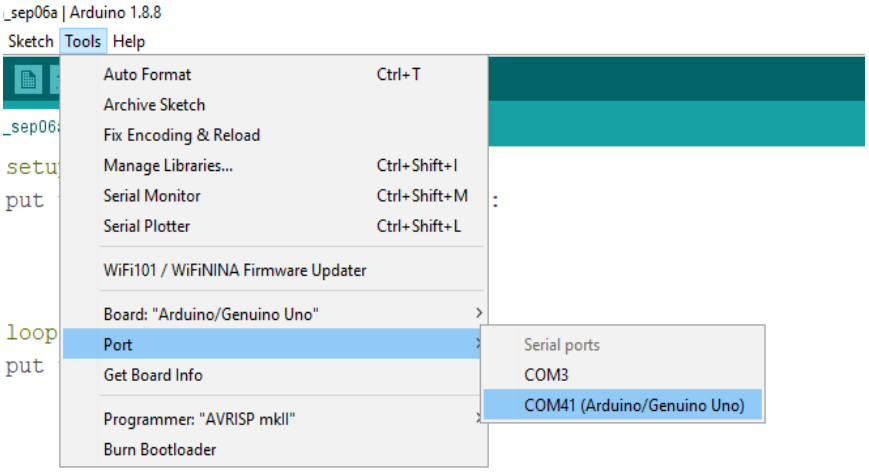
7. This is how the Arduino IDE looks.



8. First, we need to configure the Arduino in the Starter kit using the downloaded Arduino IDE. Connect one end of the USB cable included with the Starter kit to the Arduino UNO and another end to your laptop or PC. Then go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".



9. Then go to "**Tools**" → "**Port**" and check if your Arduino has been assigned a Port number in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**".

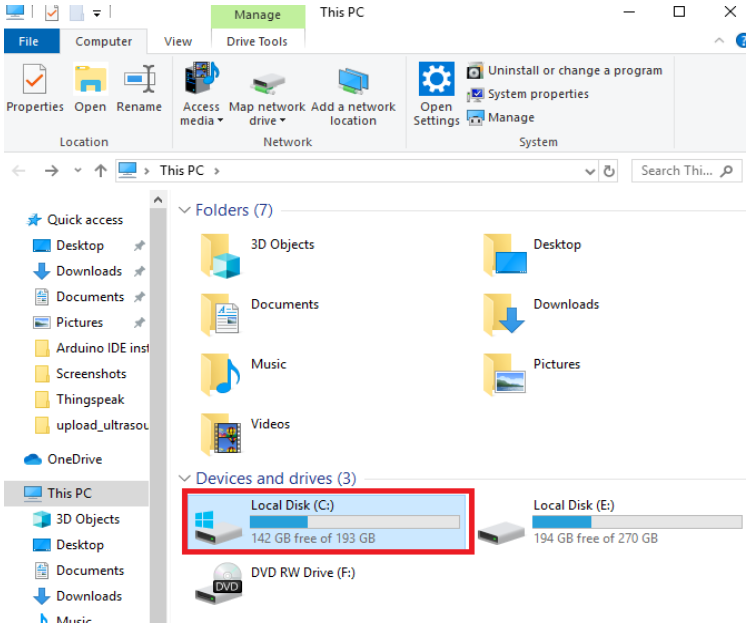


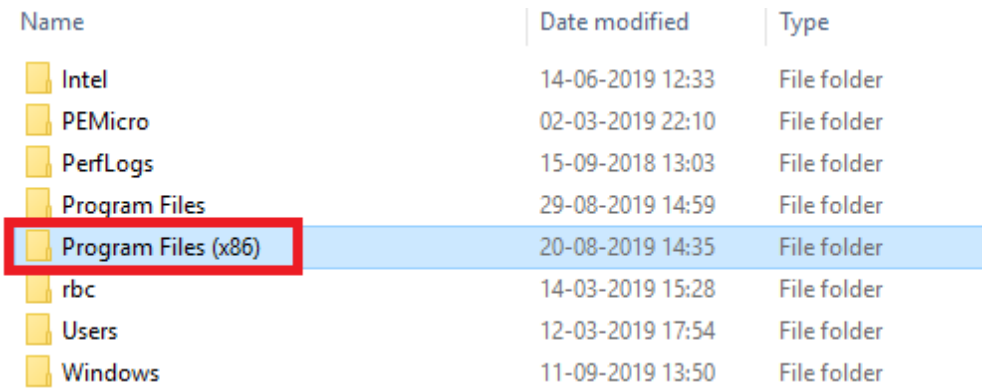
10. Now you are done installing the Arduino IDE.

Installing Libraries

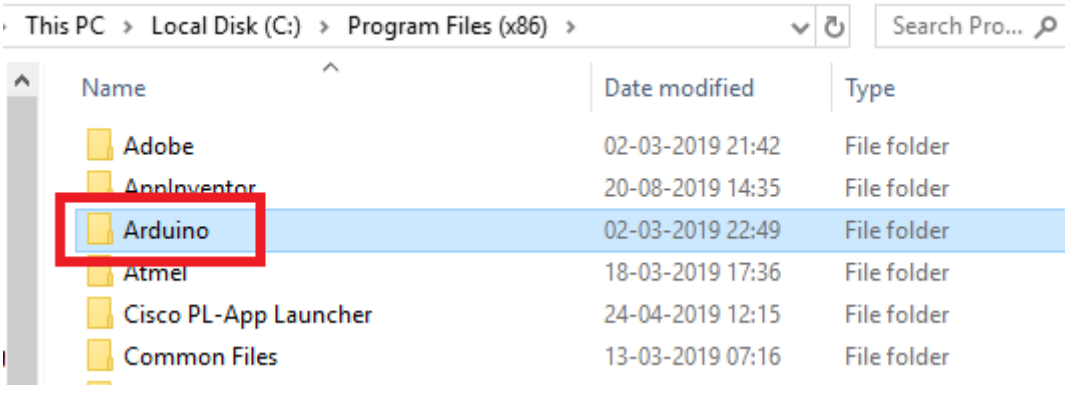
1. Download the iotlab library in the laptop or PC that you want to use it on (you can also transfer it using a pen drive), preferably paste it on Desktop.

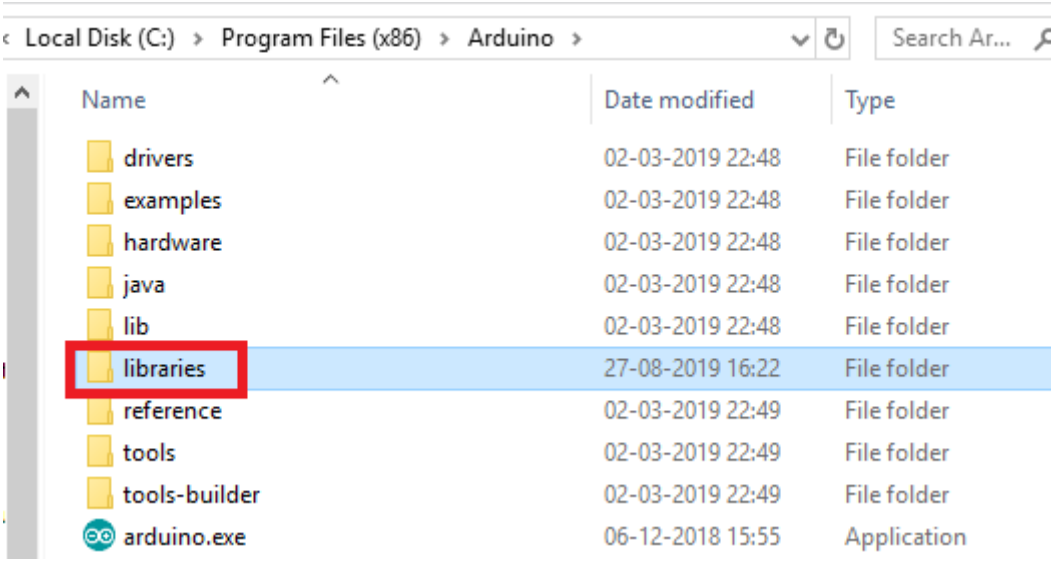
2. Now open **This PC → "Local disk C" → "Program Files (x86)"**



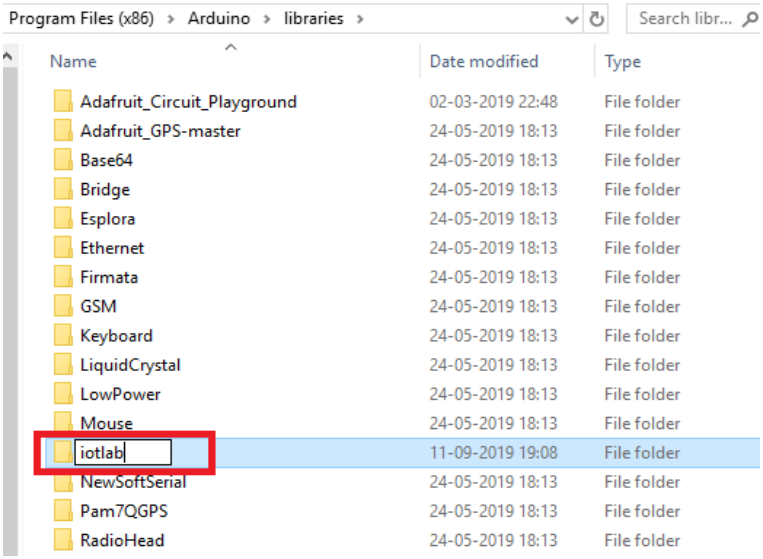


Then go to → "**Arduino**" → "**libraries**"



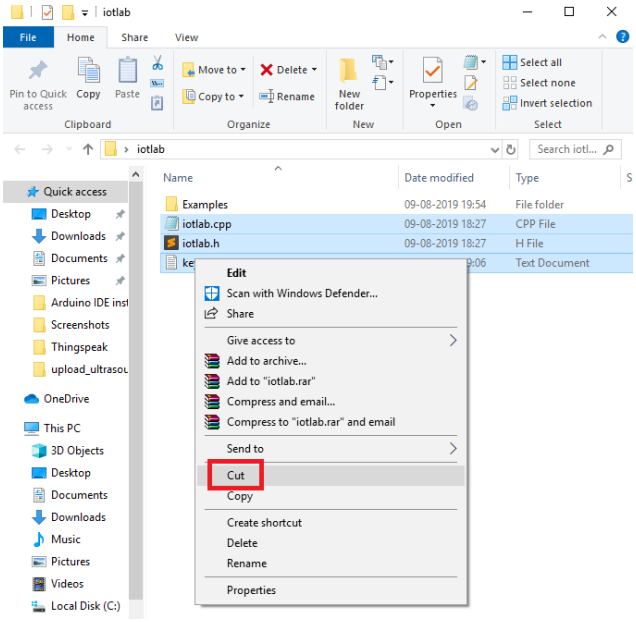


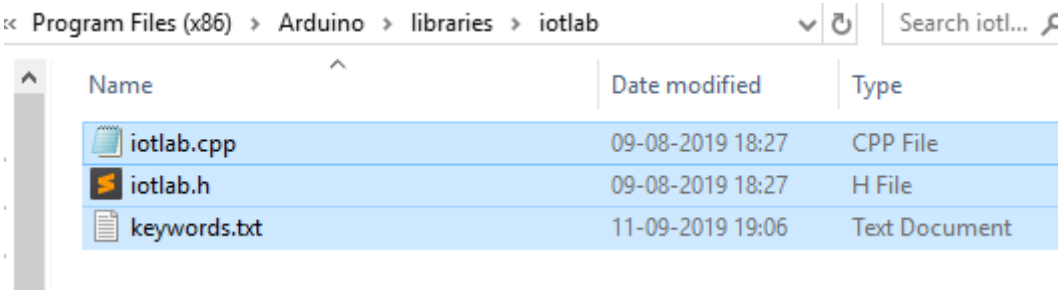
Create a new folder and name it as "**iotlab**" → Minimize this tab.



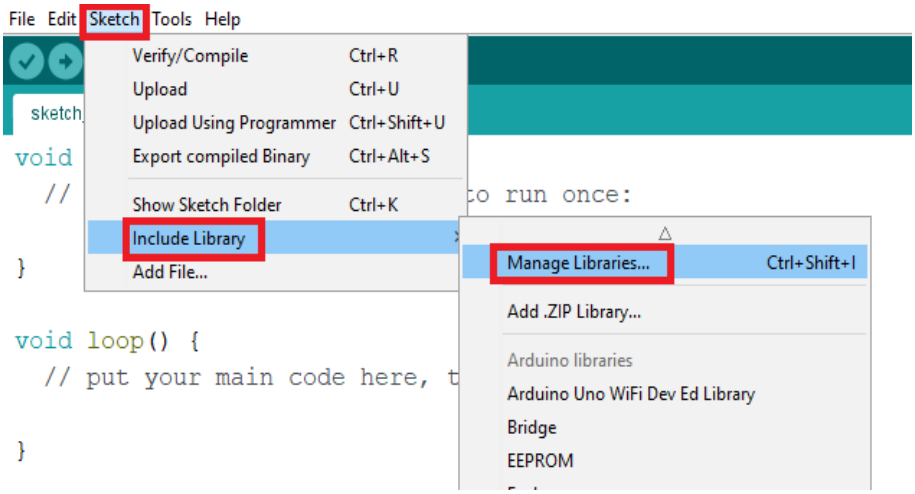
3. Open the downloaded iotlab library folder and select "**iotlab.cpp**", "**iotlab.h**" and "**keywords.txt**".

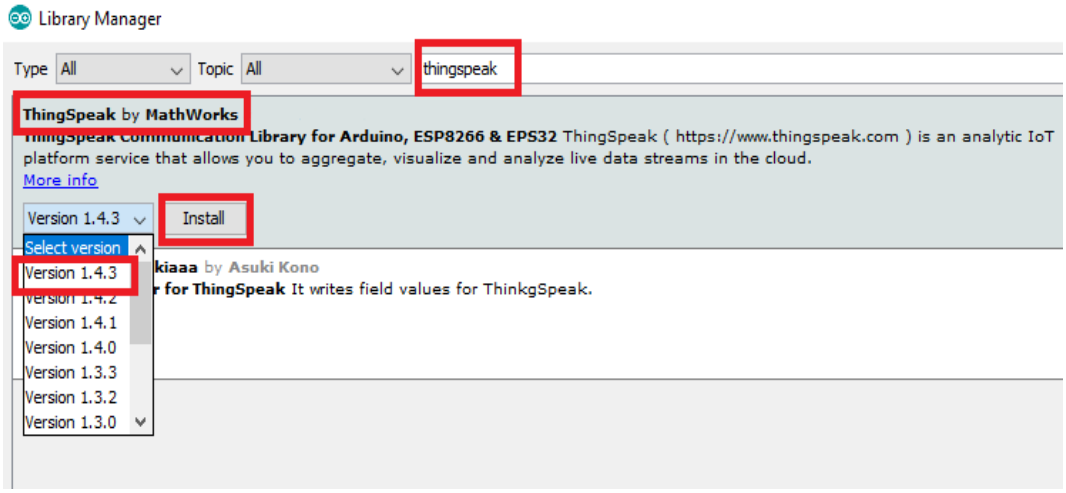
4. Copy the files mentioned in Step 3 and paste it in the folder created in Step 2.



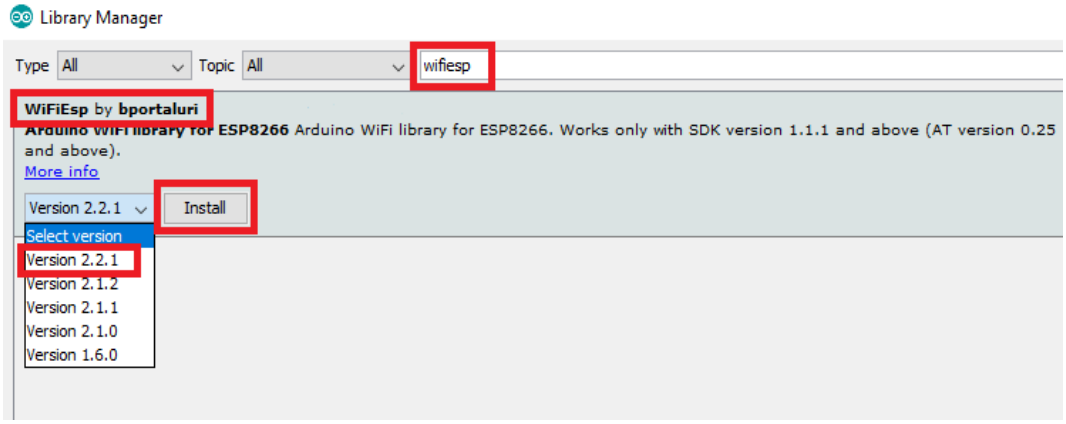


5. Open "**Arduino IDE**", go to "**Sketch**" → "**Include Library**" → "**Manage Libraries**" → Library Manager dialogue box will appear. Search for "**Thingspeak**" → locate "**Thingspeak by MathWorks**" and install the latest version → close the library manager.



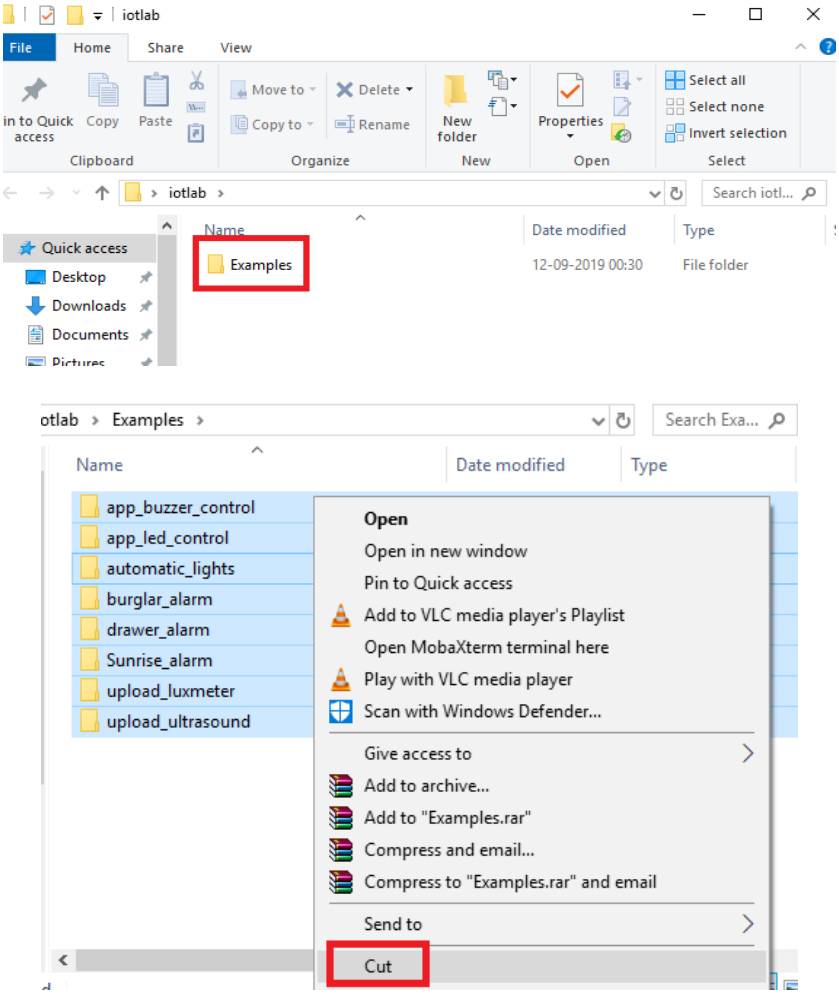


6. Go to “**Sketch**”→ “**Include Library**” → “**Manage Libraries**” → Library Manager dialogue box will appear, Search for “**wifiesp**” → locate “**WiFiEsp by bportaluri**” and install the latest version → close the library manager.

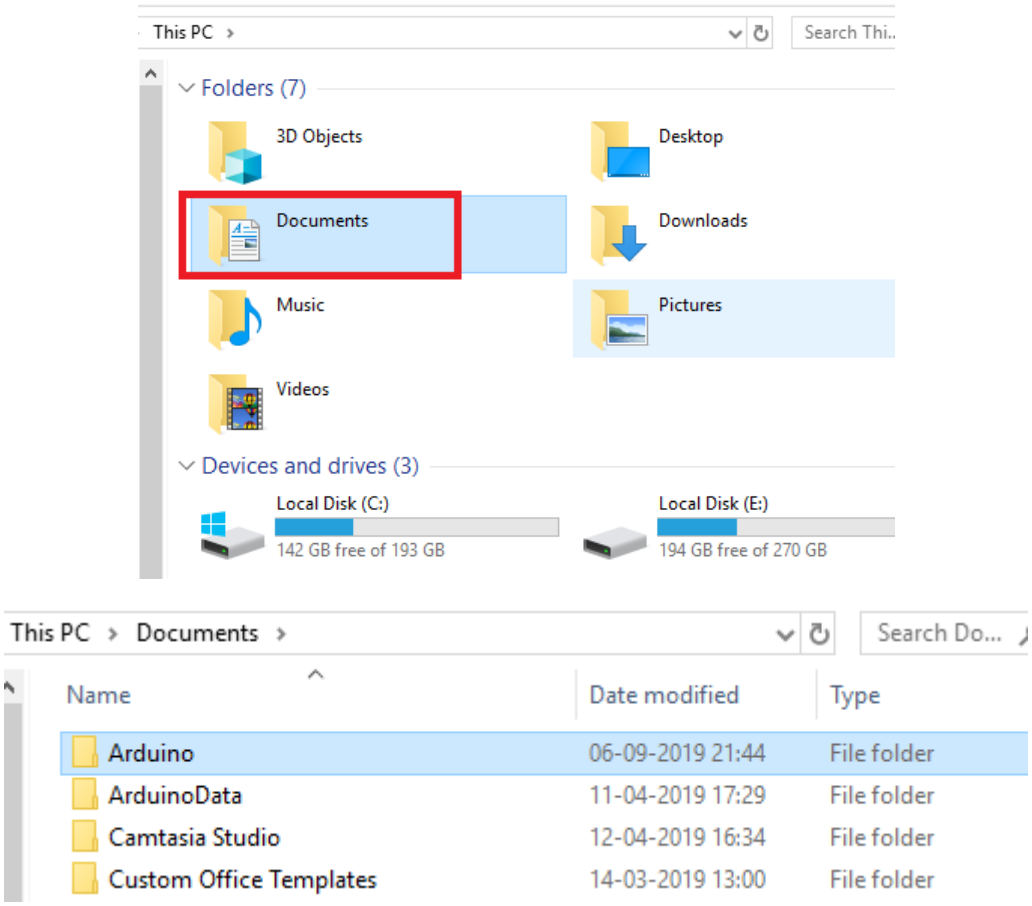


7. Now you are done with installing the library.

8. Open the “**iotlab**” folder copied on Desktop, then open “**Examples**” cut all the files present in this folder.



9. Open “**This PC**” → “**Documents**” → “**Arduino**” → paste all the files from Step 8.

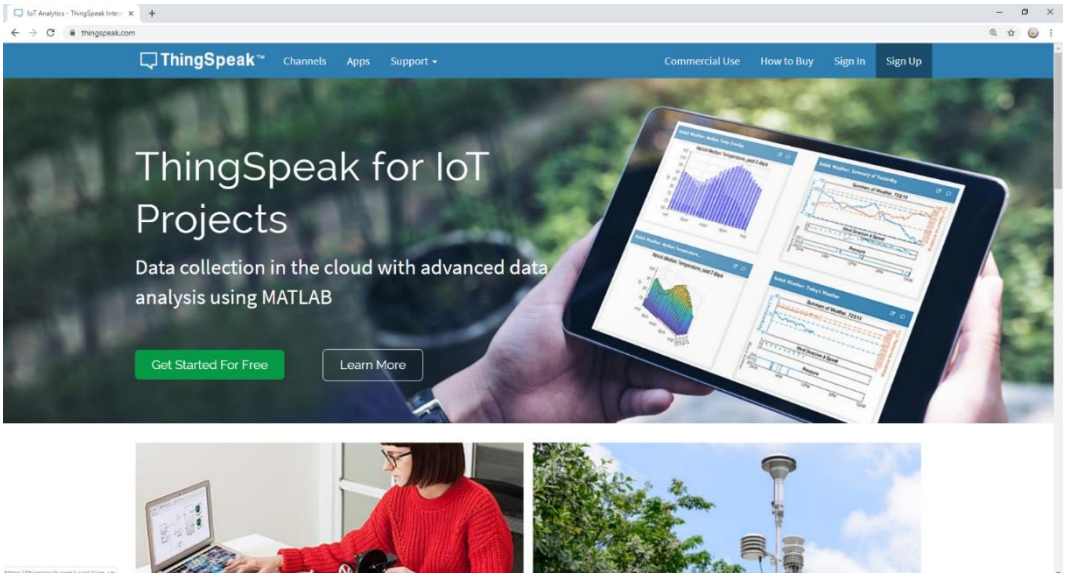


10. Now you can run all the Projects on your Arduino Starter kit.

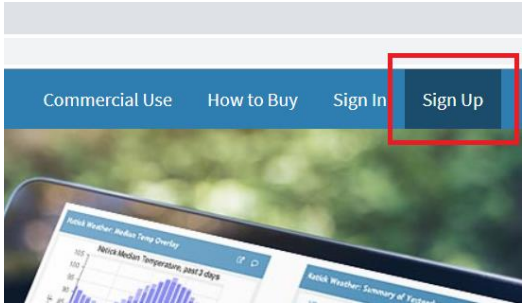
IoT Cloud Account - Thingspeak Account

Thingspeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet which enables the creation of sensor logging, analyze and visualize uploaded data. To successfully upload to and download data from the cloud you need to have a personal account on the Cloud platform in our case we are using Thingspeak cloud. Hence, to perform all the projects successfully you need to have an account on Thingspeak. So just follow these simple steps and create your own account.

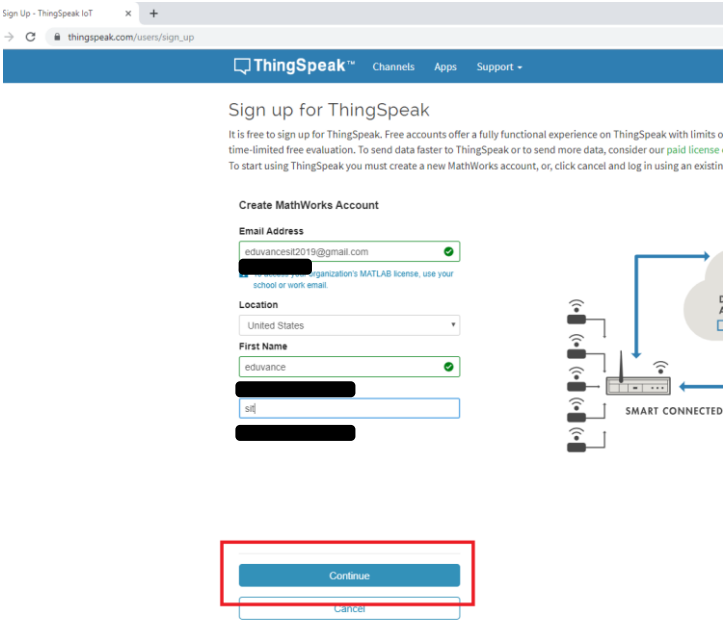
1. Open any web browser and type “**Thingspeak.com**” in your search tab and press enter.



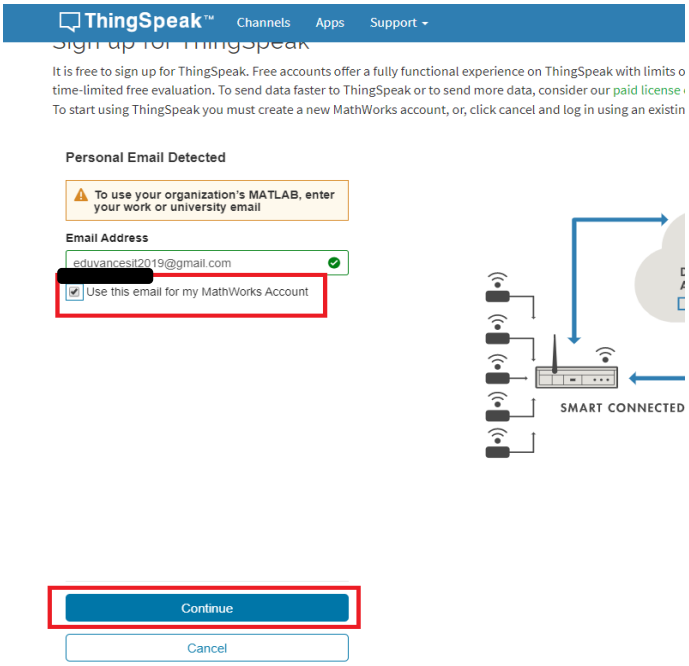
2. The Thingspeak webpage will appear Click on "**Sign Up**" button at the top-right side.



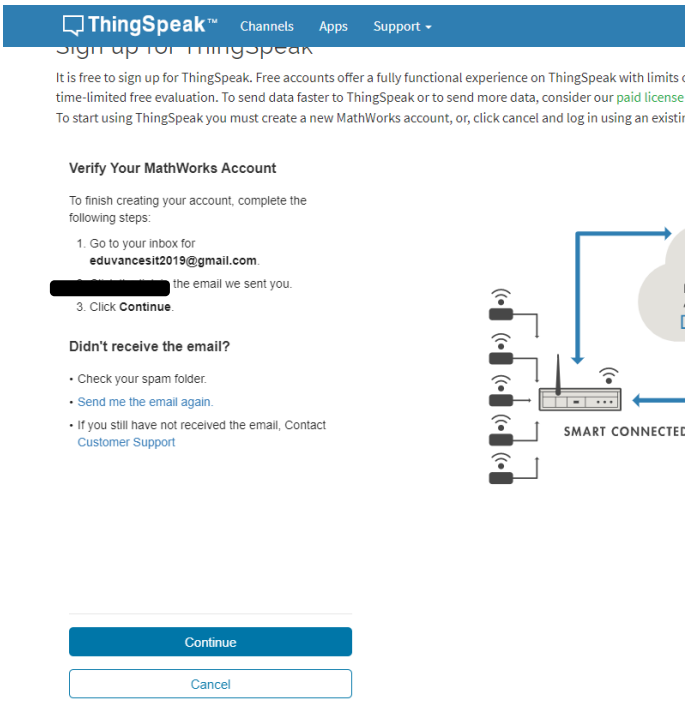
3. Now you will see a page where you need to fill your email address and other details, after filling everything click on "**Continue**".



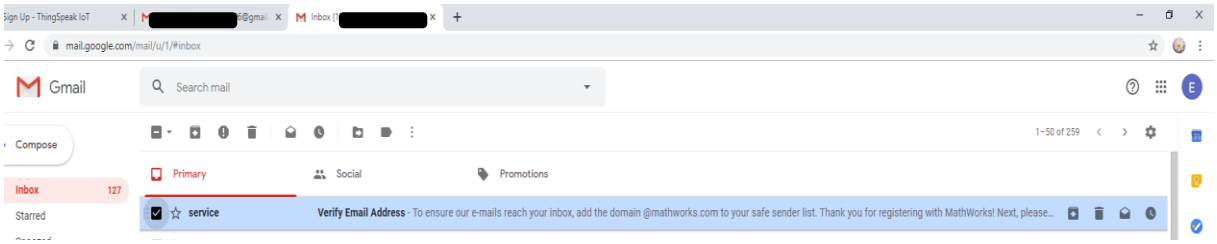
4. On the next page click on the checkbox which says "**Use this email for my MathWorks Account**", then Click on "**Continue**".

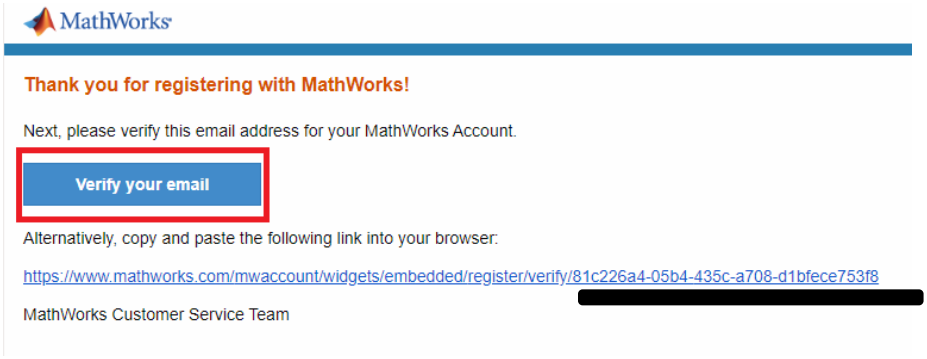


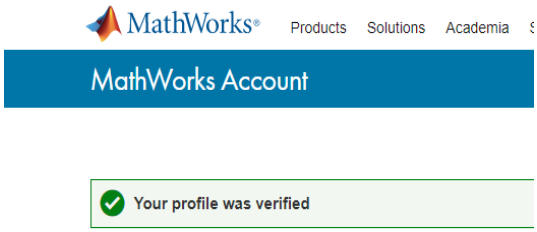
5. Now to complete your account registration you need to confirm registration from the email address which you entered during signing up.



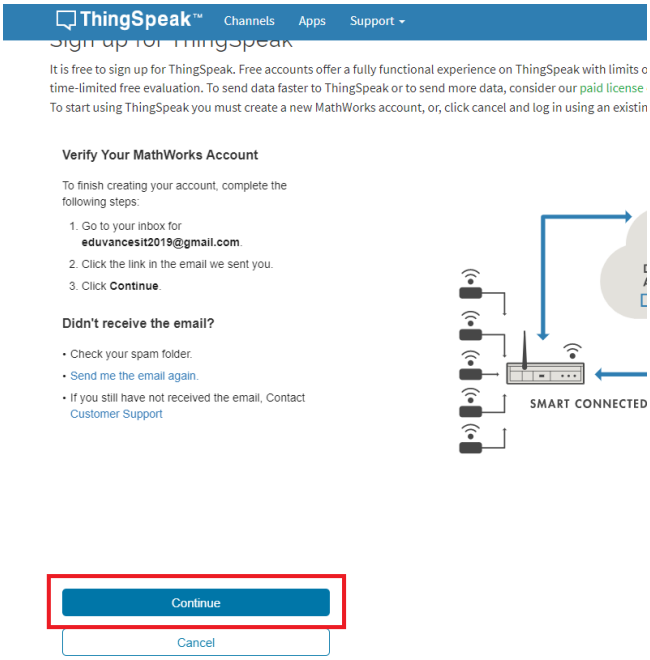
6. Open a new tab and sign in to your email address, in your inbox you will have an email from **“service@mathworks.com”** open that email and click on “**Verify**” your email button. A new tab will open telling that “**Your account was verified**”.



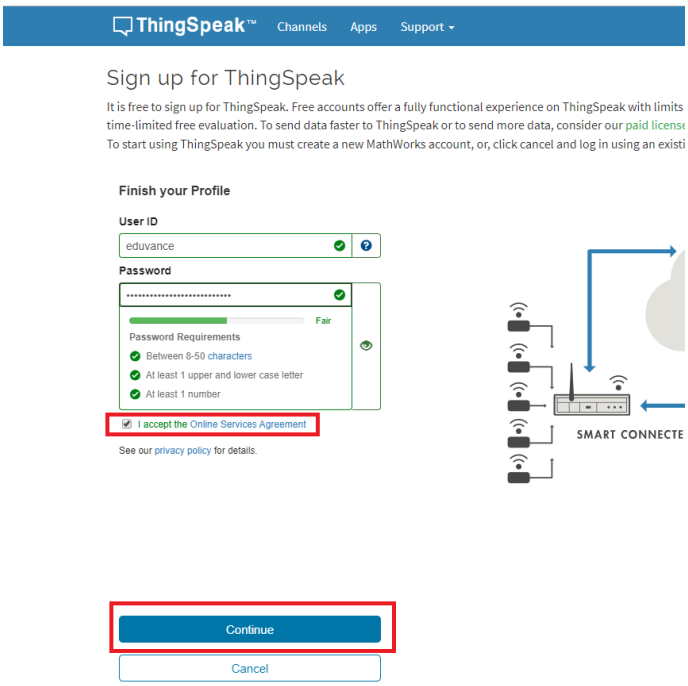




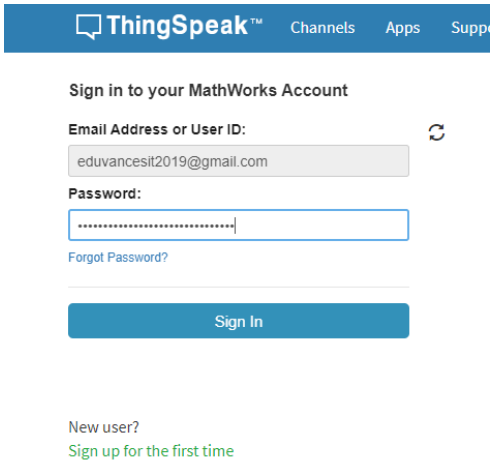
7. Go back to the Thingspeak tab and Click on “**Continue**”.



8. Set a valid User ID and Password and click on the "**I accept**" checkbox. Now click on "**Continue**".

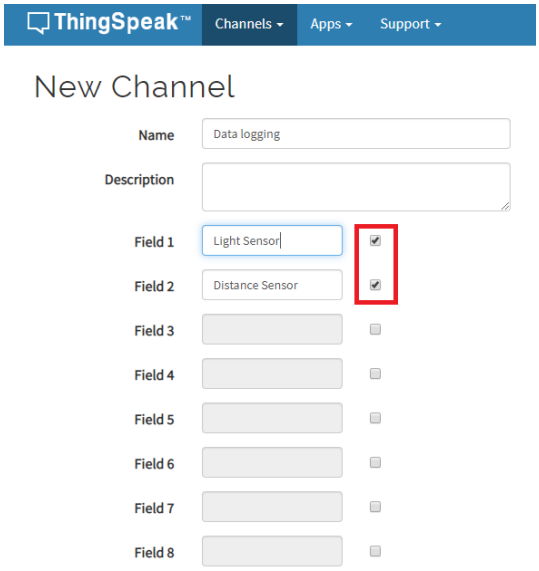


9. Now you will see a Sign-up successful page Click on "**OK**" and Sign in with the credentials that you have set during Sign-up.

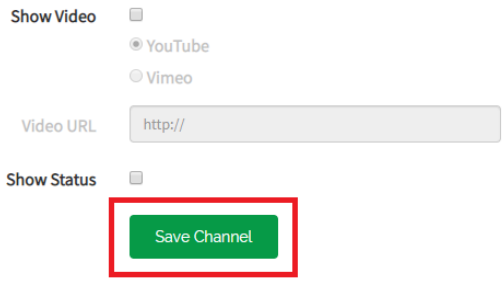


10. A "**Thingspeak Usage intent**" dialogue box will appear select "**Student Use**" and Click on "**OK**".

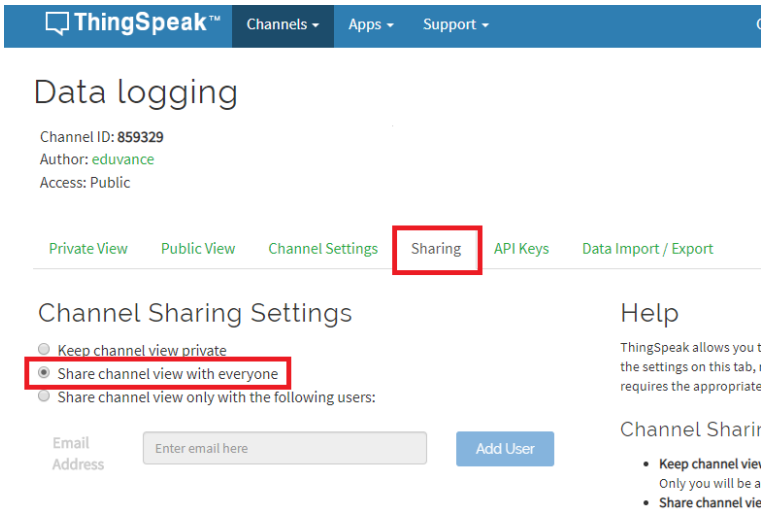
11. Now you need to create a channel where you can upload to and download sensor data from, click on "**New Channel**", give an appropriate name and click on the checkboxes of whatever number of fields you want to visualize on your channel. Field is mainly meant for graphical visualization of the uploaded values in our case lets select 2 fields to begin with and properly name them accordingly.



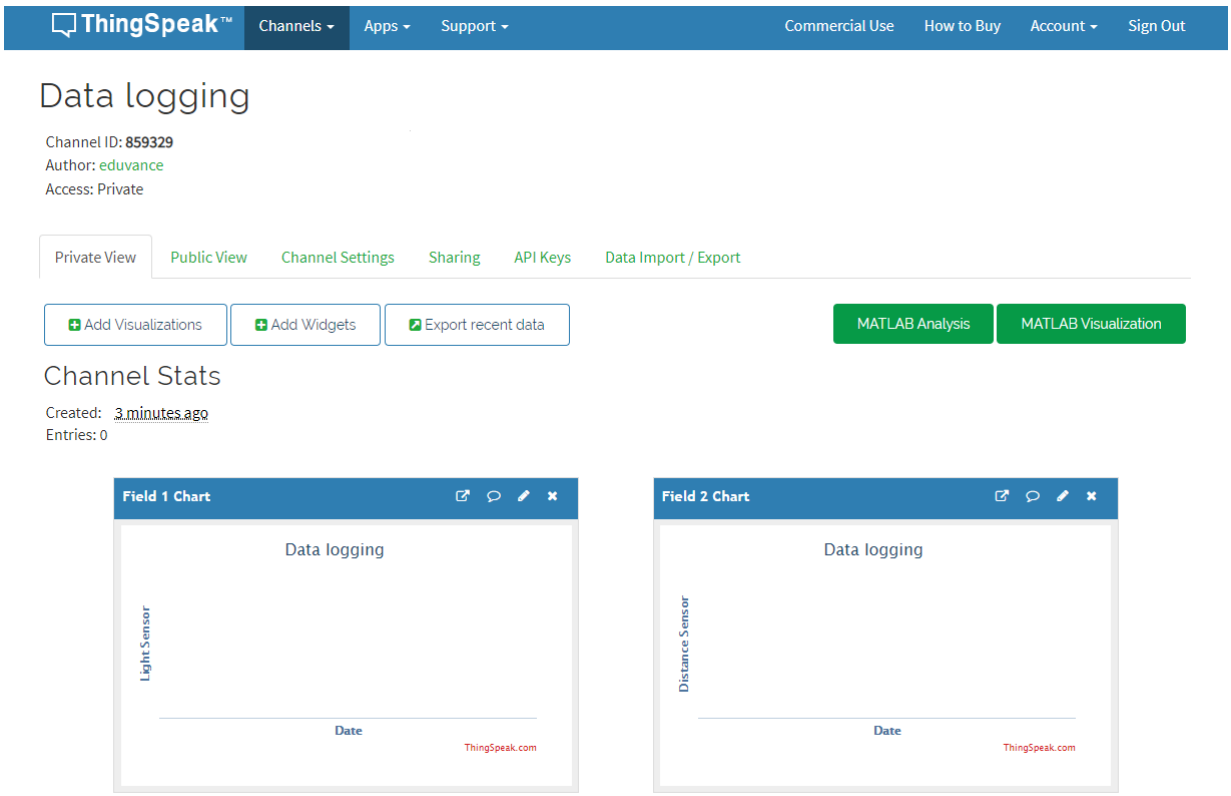
12. After finishing the above step scroll at the bottom of the page and click on "**Save Channel**".



13. Now you will see your own channel with a unique Channel ID, click on "**Sharing**" and Click on "**Share channel view with everyone**", this will be needed to successfully download values from the cloud. After finishing click on "**Private view**".



14. Now you have finished creating your own channel.



A Wi-Fi Router with active Internet Connection

You will need Internet to Upload data to the cloud and download data from the cloud, so you need to have a proper Wi-Fi router with active internet connection which will also be useful when you want to control other devices using IoT Starter kit app through your Android Smartphone.

IoT Starter Kit Android Application

As the App included with the IoT Starter Kit is an android application, so to use it and have a full-fledged uninterrupted IoT experience you need to have an Android Smartphone with Android Operating System version 2.3 ("Gingerbread") or higher. Install the APK from the link provided to you.

Projects

Project 1: Uploading the Light Sensor values to the Cloud

Summary

To understand what cloud is and how data is uploaded to the cloud, let's begin with a simple project by just uploading the values of the light sensor to Thingspeak cloud. The uploaded values can be observed as a graphical representation in "Field 1 Chart" of your channel. The amount of light falling on the light sensor is calculated in lumens per sq. meter and the device which measures light is known as a luxmeter. So, let's start by making our own little luxmeter.

Activity

1. We will need Light sensor and Wi-Fi module to perform this project, open your IoT Starter Kit Box, take 2 wires and connect the connector blocks named "**Light sensor**" and "**Wifi module**" on the Starter kit sensor board to Starter kit Arduino Shield.
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**upload_luxmeter.ino**" click "**open**".
3. Edit the program with your own Wi-Fi SSID, Password, Write API key from your Thingspeak account, Channel ID from your Thingspeak account. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "Board" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see "**Channel update successful**" message.
7. As soon as you see the above message log in to your Thingspeak account and check the "**Field 1 Chart**" of your Thingspeak Channel.
8. You can observe that the sensor value was updated with the corresponding graphical representation, keep checking the "**Field 1 Chart**" it will be updated every 15 seconds as Thingspeak only accepts the uploaded data after a 15 seconds interval.
9. Now that we are done with the first project let's move on to the next one.

Project 2: Uploading the Distance Sensor values to the Cloud

Summary

Let's continue to understand how data is uploaded to the cloud with another simple project. In this project we will upload the values of the Distance sensor to Thingspeak cloud. The uploaded values can be observed as a graphical representation in "Field 2 Chart" of your channel.

Activity

1. We will need distance sensor and Wi-Fi module to perform this project Open your IoT Starter Kit Box take 2 wires and connect the connector blocks named "**Distance sensor**" and "**Wifi module**" on the Starter kit sensor board to Starter kit Arduino Shield.
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**upload_ultrasound.ino**" click "**open**".
3. Edit the program with your Wi-Fi SSID, Password, Write API key from your Thingspeak account, Channel ID from your Thingspeak account. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see "**Channel update successful**" message.
7. As soon as you see the above message log in to your Thingspeak account and check the "**Field 2 Chart**" of your Thingspeak Channel.
8. You can observe that the sensor value was updated with the corresponding graphical representation, keep an eye on the "**Field 2 Chart**" it will be updated every 15 seconds as Thingspeak only accepts the uploaded data after a 15 seconds interval.

Project 3: Automatic night lamp

Summary

When the sun begins to set and darkness takes over, have you ever wondered how easy would it be if the lights in your room would switch ON automatically without you switching ON manually. In this project we will try to replicate the above scenario by using light sensor and LEDs included in the Starter kit. So, whenever the amount of light on the Light sensor decreases the LED will turn ON automatically.

NOTE: *If you want to use an actual light bulb make sure you make proper connections of the relay and only attempt it under adult supervision.*

Activity

1. We will need Light sensor, LED strip and Wi-Fi module to perform this project Open your IoT Starter Kit Box take 2 wires and connect the connector blocks named "**Light sensor**" and "**Wifi module**" on the Starter kit sensor board to Starter kit Arduino Shield.
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**automatic_light.ino**" click "**open**".
3. Edit the program with your Wi-Fi SSID, Password, Write API key from your Thingspeak account, Channel ID from your Thingspeak account. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see "**Channel update successful**" message which will update the sensor value on Thingspeak. After updating you will see a message "**Downloading from the cloud**" after this the Wi-Fi module will download the updated sensor value and tell you whether the LED will be ON or OFF.
7. If you want you can observe that the sensor value was updated with the corresponding graphical representation, keep an eye on the "**Field 1 Chart**" it will be updated every 15 seconds as Thingspeak only accepts the uploaded data after a 15 seconds interval.
8. If the updated value is less than the preset threshold which indicates that the amount of light incident on the light sensor is low, then the LED will be switched **ON** else LED will be **OFF**.

Project 4: Drawer and Closet Alarm

Summary

Sometimes, you don't want anyone to mess with your personal belongings. In this project we are going to make a system to protect your drawers and closet from external interference. We will use a distance sensor and a buzzer which will start beeping if someone opens your Drawer or Closet.

Activity

1. We will need distance sensor, buzzer and Wi-Fi module to perform this project Open your IoT Starter Kit Box take 2 wires and connect the connector blocks named "**Distance sensor**" and "**Wifi module**" on the Starter kit sensor board to Starter kit Arduino Shield.
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**drawer_alarm.ino**" click "**open**".
3. Edit the program with your Wi-Fi SSID, Password, Write API key from your Thingspeak account, Channel ID from your Thingspeak account. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see "**Channel update successful**" message which will update the sensor value on Thingspeak. After updating you will see a message "**Downloading from the cloud**" after this the Wi-Fi module will download the updated sensor value.
7. If you want you can observe that the sensor value was updated with the corresponding graphical representation, keep an eye on the "**Field 2 Chart**" it will be updated every 15 seconds as Thingspeak only accepts the uploaded data after a 15 seconds interval.
8. If the updated value is more than the threshold which you have set using "**iot.max_distance(10)**" function which indicates that the distance between the alarm system and the object in front of it then the buzzer will start beeping else buzzer will be OFF.

Project 5: Sunrise alarm

Summary

Ever wondered, how to make an alarm which will let us know when the pitch-black night is over and the warm sunrise starts to enlighten our day? The goal of this project is to make an alarm system based on amount of light on the light sensor. That means alarm will start in morning when there is sufficient sunlight.

Activity

1. We will need light sensor, buzzer and Wi-Fi module to perform this project Open your IoT Starter Kit Box take 3 wires and connect the connector blocks named "**Light sensor**","**Wifi module**" and "**Distance sensor**" on the Starter kit sensor board to Starter kit Arduino Shield as the buzzer is internally connected to the "**Distance sensor**" connector block .
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**sunrise_alarm.ino**" click "**open**".
3. Edit the program with your Wi-Fi SSID, Password, Write API key from your Thingspeak account, Channel ID from your Thingspeak account. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see "**Channel update successful**" message which will update the sensor value on Thingspeak. After updating you will see a message "**Downloading from the cloud**" after this the Wi-Fi module will download the updated sensor value.
7. If you want you can observe that the sensor value was updated with the corresponding graphical representation, keep an eye on the "**Field 1 Chart**" it will be updated every 15 seconds as Thingspeak only accepts the uploaded data after a 15 seconds interval.
8. If the updated value is less than the preset threshold which indicates that the amount of light incident on the light sensor is higher, then the buzzer will start **beeping** else buzzer will be **OFF**.

Project 6: Burglar alarm

Summary

Suppose a person enters your room or house and attempts to steal all your things without you even noticing it. After you get back it won't be a pleasant situation to handle. The objective of this project is to sound an alarm and blink LED lights as soon as the system detects a person entering your house or room while there's no one in there.

Activity

1. We will need distance sensor, buzzer, LED strip and Wi-Fi module to perform this project Open your IoT Starter Kit Box take 3 wires and connect the connector blocks named "**Light sensor**","**Wifi module**" and "**Distance sensor**" on the Starter kit sensor board to Starter kit Arduino Shield as the LED strip is internally connected to the "**Light sensor**" connector block .
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**burglar_alarm.ino**" click "open".
3. Edit the program with your Wi-Fi SSID, Password, Write API key from your Thingspeak account, Channel ID from your Thingspeak account. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see "**Channel update successful**" message which will update the sensor value on Thingspeak. After updating you will see a message "**Downloading from the cloud**" after this the Wi-Fi module will download the updated sensor value.
7. If you want you can observe that the sensor value was updated with the corresponding graphical representation, keep an eye on the "**Field 2 Chart**" it will be updated every 15 seconds as Thingspeak only accepts the uploaded data after a 15 seconds interval.
8. If the updated value is less than the threshold which you have set using "**iot.min_distance()**" function which indicates that the distance between the alarm system and the person in front of it, then the buzzer will start **beeping** and LED strip will start **blinking** else buzzer and LED strip will be **OFF**.

Project 7: Controlling LED using IoT Starter kit app

Summary

Suppose you arrived home completely exhausted and head towards your bedroom straight to sleep. Now in this tired situation, if you forget to turn off the lights it will be really painful to get up, walk towards the switch board and turn off the lights. Instead of this, it will be really handy if you could just switch off your lights through your Smartphone. This is where the need for IoT Starter kit App comes in. In this project you will learn how to control an LED strip using the Android app.

Activity

1. We will need **IoT Starter Kit android app, LED strip and Wi-Fi module** to perform this project Open your IoT Starter Kit Box take a wire and connect the connector blocks named "**Wifi module**" on the Starter kit sensor board to Starter kit Arduino Shield.
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**app_led_control.ino**" click "**open**".
3. Edit the program with your own Wi-Fi SSID, Password. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see the **IP address** of the that is assigned to our **Wi-Fi module** by your Wi-Fi router. **Make a note of this IP address as it will be used to control the LED strip through the Android App.**
7. Open the app you will now see the first screen appearing before you. Now click on "**Get Started**" button, this will direct you to another screen.
8. As soon as the new screen opens you will see 2 buttons, first will be named as "**Device Controller**" and the second will be "**Cloud Data Logger**".
9. For the purpose of controlling devices, click on "**Device Controller**" button you will be presented with 2 more buttons, one which has a "**Wi-Fi**" icon and other one with "**LED**" icon.

10. Firstly, click on the "**Wi-Fi**" icon to enter the IP address of your Wi-Fi Module which was obtained after programming the Wi-Fi Module through Arduino and Click on "**Save**", this will enable your phone to connect to the Wi-Fi Module.
11. Once you are done with the previous step click on the "**LED**" icon doing so will present you with "**ON**" and "**OFF**" buttons. Now you can turn your device ON and OFF using your Smartphone via the IOT App.
12. In case you want to connect to some other Wi-Fi Module which is in turn Connected to another device and start controlling it, Just Click on the "**Wi-Fi**" icon again, Press "**Clear**" doing so will disconnect your App from the Previous Wi-Fi Module and now you could enter the new IP Address for the device you want to control which is connected to another Wi-Fi module.

Project 8: Controlling Buzzer using IoT Starter kit app

Summary

This project is just a different version of the previous project which basically works the same way as the previous one does. It is included to demonstrate that how a simple change in the pin number while programming will enable you to control a whole new different device through the Android App.

Activity

1. We will need **IoT Starter Kit android app, buzzer and Wi-Fi module** to perform this project Open your IoT Starter Kit Box take 2 wires and connect the connector blocks named "**Distance sensor**" and "**Wifi module**" on the Starter kit sensor board to Starter kit Arduino Shield as the **buzzer** is internally connected to the "**Distance sensor**" connector block.
2. Open Arduino IDE go to "**Files**" → "**Open**" → a file selection dialogue box will appear, locate "**app_buzzer_control.ino**" click "**open**".
3. Edit the program with your own Wi-Fi SSID, Password. Connect your IoT Starter kits Arduino board to your laptop or desktop, then in your Arduino IDE go to "**Tools**" → "**Board**" and select "**Arduino/Genuino UNO**".
4. Then go to "**Tools**" → "**Port**" and select the correct port in this case in our case the assigned Port number is "**COM41(Arduino/Genuino UNO)**" in your case it may be different so select the port accordingly.
5. Now Click on "**upload**" button indicated by the "**right arrow icon**". After some time, you will see Done Uploading in the bottom of your screen.
6. Go to tools click on "**Serial Monitor**" or click on the "**magnifying glass icon**" at the top right-hand corner of your screen. A Serial monitor window will open and show some messages wait until you see the **IP address** of the that is assigned to our **Wi-Fi module** by your Wi-Fi router. **Make a note of this IP address as it will be used to control the LED strip through the Android App.**
7. Open the app you will now see the first screen appearing before you. Now click on "**Get Started**" button, this will direct you to another screen.
8. As soon as the new screen opens you will see 2 buttons, first will be named as "**Device Controller**" and the second will be "**Cloud Data Logger**".
9. For the purpose of controlling devices, click on "**Device Controller**" button you will be presented with 2 more buttons, one which has a "**Wi-Fi**" icon and other one with "**LED**" icon.

10. Firstly, click on the "**Wi-Fi**" icon to enter the IP address of your Wi-Fi Module which was obtained after programming the Wi-Fi Module through Arduino and Click on "**Save**", this will enable your phone to connect to the Wi-Fi Module.
11. Once you are done with the previous step click on the "**LED**" icon doing so will present you with "**ON**" and "**OFF**" buttons. Now you can turn your device ON and OFF using your Smartphone via the IOT App.
12. In case you want to connect to some other Wi-Fi Module which is in turn Connected to another device and start controlling it, Just Click on the "**Wi-Fi**" icon again, Press "**Clear**" doing so will disconnect your App from the Previous Wi-Fi Module and now you could enter the new IP Address for the device you want to control which is connected to another Wi-Fi module.

Project 9: Cloud data scanning using IoT Starter kit app

Summary

The main benefit of cloud in an IoT system is its ease of access, provided you have an active internet connection. You can easily access the data existing on the cloud from anywhere in the world. Now suppose you have set up certain sensors to monitor a particular area and you want to check the sensor values remotely. You can easily do this by using the cloud data logger function of the IoT starter kit Android app even if you are far away from the kit.

Activity

1. We will only need **IoT Starter Kit android app with an active internet connection** to perform this project.
2. Open the app you will now see the first screen appearing before you. Now click on "**Get Started**" button, this will direct you to another screen.
3. As soon as the new screen opens you will see 2 buttons, first will be named as "**Device Controller**" and the second will be "**Cloud Data Logger**".
4. When you Click on "**Cloud Data Logger**" button you will be presented with 1 more button with a "**cloud icon**".
5. Click on the "**Cloud**" icon and enter your Thingspeak Channel id and the Field number from which you want to download the sensor values from.
6. Once you have finished the previous step Click on "**Save**", Now you can see the most recent data being displayed on your screen.
7. In case you want to connect to some other Thingspeak Channel or change the field number then click the "**Clear**" button, this will stop downloading values from the previous channel and now you can enter the new Channel id and field number and download sensor values from your desired channel.'