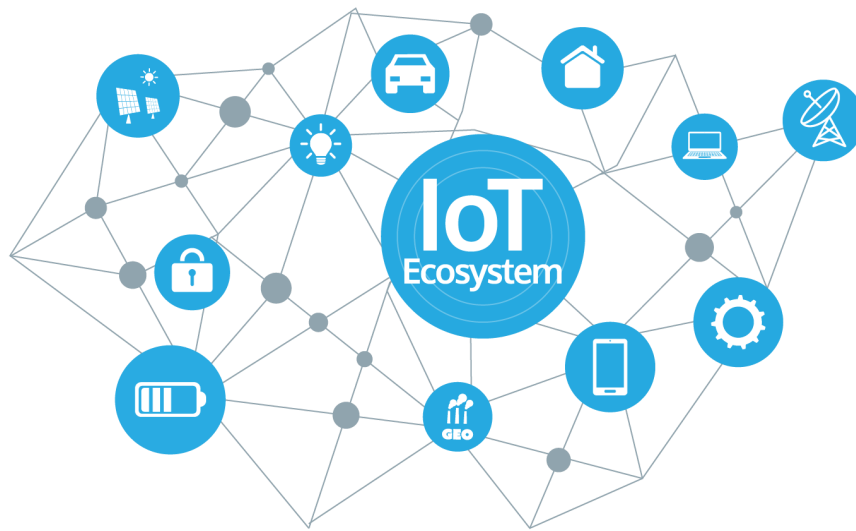


A Guide to the Internet of Things



By:

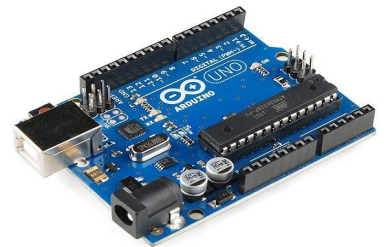
[Agniswar Paul](#)

Member, Next Tech Lab

For getting started with the **Internet of Things**, firstly you need to understand the basics of electronics, what are microcontrollers, how it works and how you can program it.

1. Start with **Arduino** first, Arduino is a microcontroller which runs on Embedded C and also very easy to use and make basic IoT Projects.

- Start watching Arduino tutorial videos on the Youtube by [Jeremy Blum](#) , After watching these videos try to program Arduino by yourself.
- Refer to the official website of [Arduino](#) , download the Arduino IDE
- In that website go to “Resources” then click on the “Tutorials” also go through the “Getting started” tab.
- For understanding all the functions, again go to the “Resources” and then click on “References”.
- [Download](#) the Arduino Uno Pinout Diagram.
- Understanding the [Serial Communication](#).
- Programming Arduino using [USB-TTL converter](#).



2. Once you are done with Arduino, start working with WiFi modules like **ESP-01**, **ESP-8266**, **ESP-32**.

- Introduction to the [ESP-01](#)
- Introduction to the [ESP_8266](#)
- [ATCommand Mode](#) (using it as a WiFi module for arduino boards)
- [Stand-Alone Mode](#) (Using it as a microcontroller)
- Interfacing with the [Cloud](#).
- Getting Started with the [ESP-32](#).

3. **NodeMCU**. It's a development board with ESP-8266 chip.

- Introduction to [NodeMCU](#).
- [Tutorials](#) on NodeMCU.
- [Download](#) the Pinout diagram of NodeMCU.
- [LED control](#) using Blynk app and NodeMCU.



4. Intro to **Raspberry Pi**

[Raspberry Pi](#) is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

First of all setup your Raspberry Pi, here are the steps:

- Download software of choice from the this [link](#)(raspbian-full verison is highly recommended)
- Format the SD card with SD card formatter [here](#).
- Download [win32 disk imager](#), and flash the .iso file(downloaded previously) to the SD card.
- For mac and linux users you can use [etcher](#) for flashing the SD card.
- Insert the SD card into the slot(CAREFULLY!!!) and plug the devices and the rpi to the monitor and power source.
- First thing is to connect the pi to the wifi.
- Steps 1-5 is required only for the first time. For later usage connect to the rpi using SSH (highly recommended, this is where your command line skills will come to use). For windows use [putty](#).



Once you are done with this, [download](#) the pinout diagram of the raspberry pi 3 and then get started. Go to the official [website](#) of the raspberry pi, then click on “projects” and then click on the “Getting Started”.

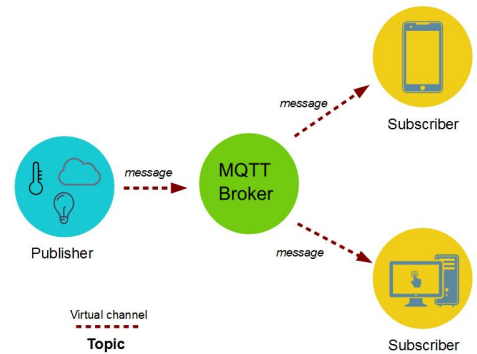
5. Refer to the [Instructables](#), for understanding and making projects using Arduino, NodeMCU, Raspberry Pi etc.

6. Visit the website [IFTTT](#)(If This Then That). It is a free web-based service to create chains of simple conditional statements, called applets. An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest.

7. Go to the [Adafruit](#), it is a system that makes data useful. Focusing on ease of use, and allowing simple data connections with little programming required. IO includes client libraries that wrap our REST and MQTT APIs. IO is built on Ruby on Rails, and Node.js.

8. Learn the **IoT Protocols**:

- [HTTP](#) Protocol.
- [MQTT](#) Protocol.
- [Other](#) Protocols.
- You can also refer to Youtube to get some more insights about these protocols.



9. Visit the website [Thingspeak](#). ThingSpeak is an open-source Internet of Things application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network.

10. Understanding the I2C Communication. The Inter-integrated Circuit (I2C) Protocol is a protocol intended to allow multiple "slave" digital integrated circuits ("chips") to communicate with one or more "master" chips.

That's all Folks!! I hope now you are pretty much comfortable with the basics of IoT, and you are all set to build cool IoT Hacks. Happy Hacking!!
