**Assignment No. 12**

**Title:** Write a server application to be deployed on Raspberry-Pi /Beagle board. Write client applications to get services from the server application.. .

**Aim:** To understand the interface between web server and Raspberry Pi.

Demonstration of creating own web server in order to store information generated through sensors.

**Hardware Requirement**:

Raspberry Pi board with an SD card

**Software Requirement:** Raspbian O.S, Python.

**Theory:**

**IoT Platform**

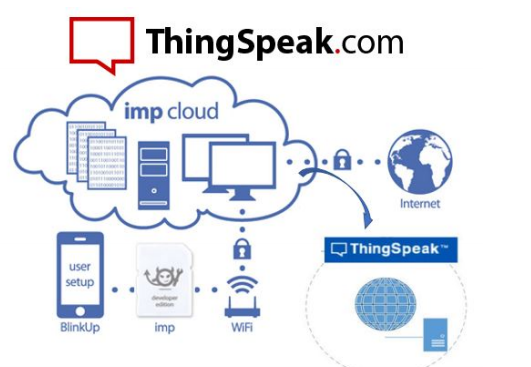
* The IoT platforms are suites of components those help to setup and manage the internet connected devices.
* A person can remotely collect data, monitor and manage all internet connected devices from a single system.
* There are a bunch of IoT platforms available online but building an IoT solution for a

company is all depend on IoT platform host and support quality.

**IOT Cloud Platforms**

* Kaa IoT Platform
* SiteWhere: Open Platform for the Internet of Things
* ThingSpeak: An open IoT platform with MATLAB analytics
* DeviceHive: IoT Made Easy
* Zetta: API-First Internet of Things Platform
* DSA: Open Source Platform & “Toolkit” for Internet Of Things Devices
* Thingsboard.io Open-source IoT Platform
* Thinger.io: The Opensource Platform for Internet of things
* WSo2- Open source platform for Internet of Things and mobile projects

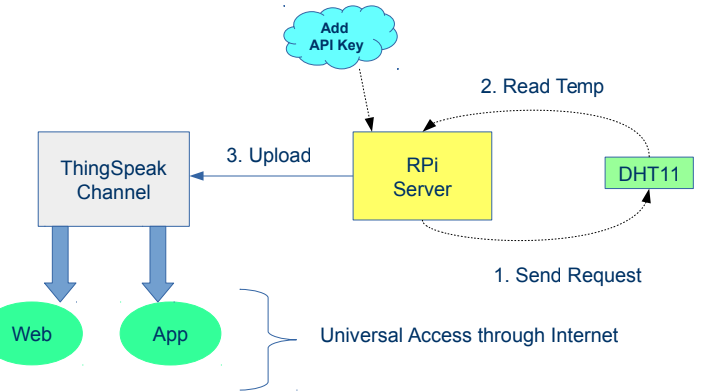
**ThingSpeak**

****

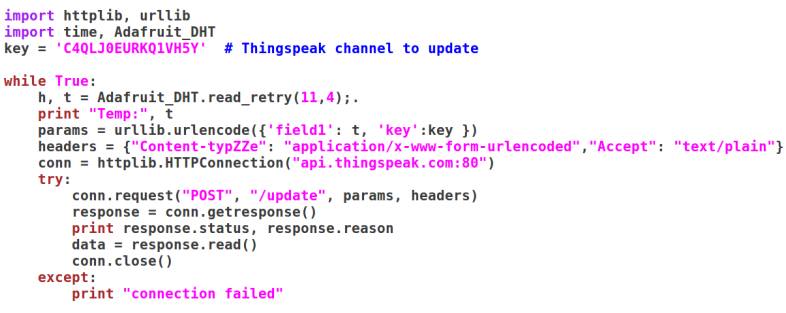
**Thing Speak Features**

* Collect data in private channels
* Share data with public channels
* RESTful and MQTT APIs
* MATLAB analytics and visualizations
* Alerts
* Event scheduling
* App integrations
* Worldwide community

**Design IoT App**

****

**Upload the temperature**

****

**Conclusion:** We have successfully executed the program of displaying temperature on web

page remotely using Raspberry Pi.

**Code :(temaperature.py)**

import os

# import os module

import glob

# import glob module

import time

# import time module

os.system('modprobe w1-gpio')

# load one wire communication device kernel modules

os.system('modprobe w1-therm')

base\_dir = '/sys/bus/w1/devices/'

# point to the address

device\_folder = glob.glob(base\_dir + '28\*')[0]

# find device with address starting from 28\*

device\_file = device\_folder + '/w1\_slave'

# store the details

def read\_temp\_raw():

f = open(device\_file, 'r')

lines = f.readlines()

# read the device details

f.close()

return lines

def read\_temp():

lines = read\_temp\_raw()

while lines[0].strip()[-3:] != 'YES':

# ignore first line

time.sleep(0.2)

lines = read\_temp\_raw()

equals\_pos = lines[1].find('t=')

# find temperature in the details

if equals\_pos != -1:

temp\_string = lines[1][equals\_pos+2:]

temp\_c = float(temp\_string) / 1000.0

# convert to Celsius

return temp\_c

while True:

print(read\_temp())

# Print temperature

time.sleep(1)