**ASSIGNMENT NUMBER: D2**

* **Title**

Create a small dashboard application to be deployed on cloud.

* **Problem Statement**

Create a small dashboard application to be deployed on cloud. Different

publisher devices can publish their information and interested application

can subscribe.

* **Objective**

- To develop comprehensive approach towards building small low cost embedded IoT system.

- To understand different sensory inputs.

* **Outcomes**

I will be able to create a small dashboard application deployed on cloud.

* **Software & Hardware Requirements**

- Cloud (ThingSpeak)

- Client server model

- Controller/processor

- Python

- PC with the configuration as Latest Version of 64 bit Operating Systems

- Open Source Fedora-GHz. 8 G.B. RAM

- 500 G.B. HDD

- 15"Color Monitor

- Keyboard

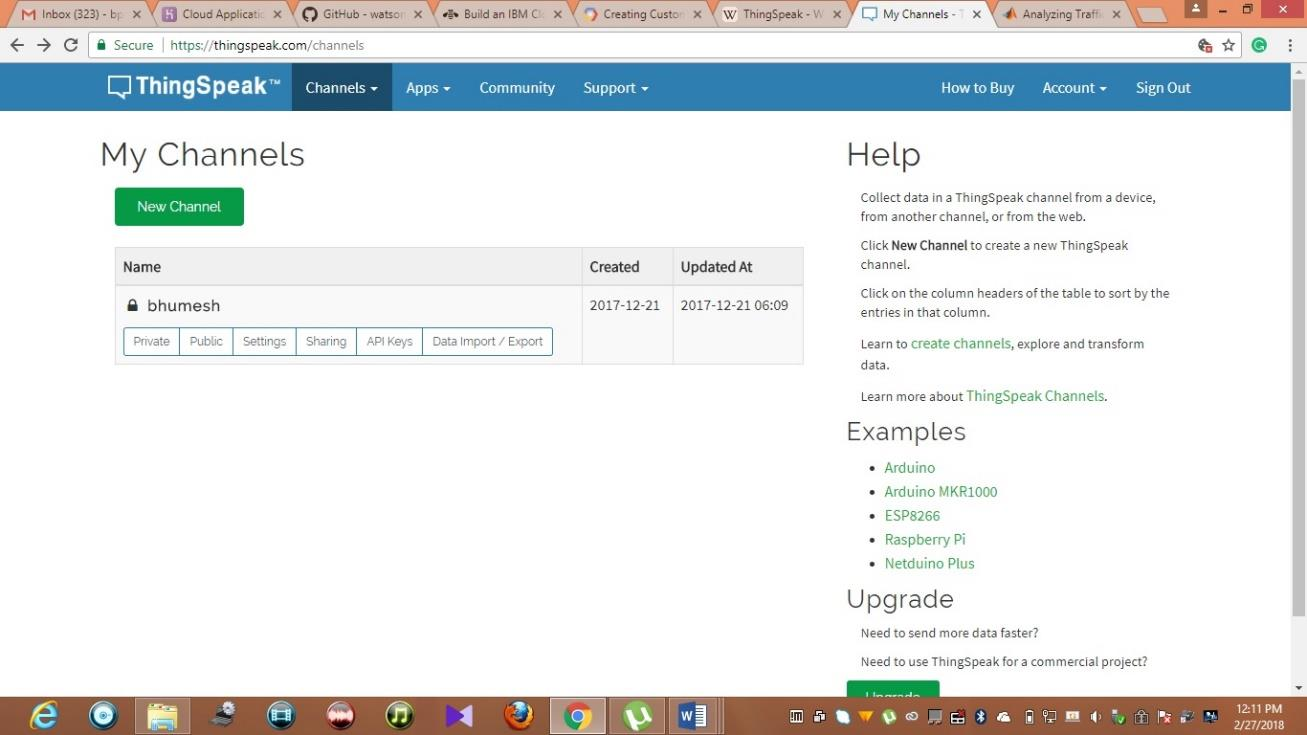
- Mouse

* **Theory**

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 or via a Local Area Network.ThingSpeak enables the creation of sensor logging applications, location tracking applications,and a social network of things with status updates. ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications. ThingSpeak has integrated support from the umerical computing software MATLAB from MathWorks, allowing ThingSpeak users to analyze and visualize uploaded data using Matlab without requiring the purchase of a Matlab license from Mathworks.

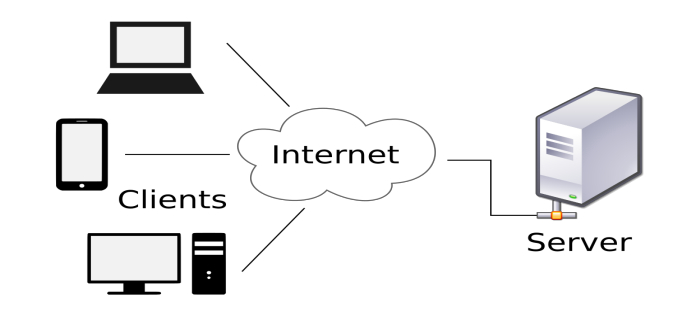
For accessing Thingspeak

Need to create account on https://thingspeak.com/login



**Client Server Model**

The client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests. Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.



**SETTING UP AN APACHE WEB SERVER ON A RASPBERRY PI**

Apache is a popular web server application you can install on the Raspberry Pi to allow it to serve web pages.On its own, Apache can serve HTML files over HTTP, and with additional modules can serve dynamic web pages using scripting languages such as PHP.

I**NSTALL APACHE**

First install the apache2 package by typing the following command in to the Terminal:

sudo apt-get install apache2 -y

**TEST THE WEB SERVER**

By default, Apache puts a test HTML file in the web folder. This default web page is served

when you browse to http://localhost/ on the Pi itself, or http://192.168.1.10 (whatever the Pi's IP address is) from another computer on the network. To find the Pi's IP address, type hostname I at the command line (or read more about finding your IP address).

**CHANGING THE DEFAULT WEB PAGE**

This default web page is just a HTML file on the filesystem. It is located

at /var/www/html/index.html.

Navigate to this directory in the Terminal and have a look at what's inside:

cd /var/www/html

ls -al

This will show you:

total 12

drwxr-xr-x 2 root root 4096 Jan 8 01:29 .

drwxr-xr-x 12 root root 4096 Jan 8 01:28 ..

-rw-r--r-- 1 root root 177 Jan 8 01:29 index.html

This shows that there is one file in /var/www/html/ called index.html. The . refers to the directory

itself /var/www/html and the .. refers to the parent directory /www/.

**ADDITIONAL - INSTALL PHP**

To allow your Apache server to process PHP files, you'll need to install PHP5 and the PHP5

module for Apache. Type the following command to install these:

sudo apt-get install php5 libapache2-mod-php5 -y

Now remove the index.html file:

sudo rm index.html

and create the file index.php:

sudo leafpad index.php

Note: Leafpad is a graphical editor. Alternatively, use nano if you're restricted to the command

line

-------------------------now open terminal and enter this:-------------------

$cd /var/www

$sudo nano rahul.php

this will open a empty black screen window where we have to write these instructions

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>LED Control</title>

</head>

<body>

**WEB PAGE ON PHP BASED GPIO Control:**

<form method="get" action="gpio.php">

<input type="submit" value="ON" name="on">

<input type="submit" value="OFF" name="off">

</form>

<?php

$setmode17 = shell\_exec("/usr/local/bin/gpio -g mode 17 out");

if(isset($\_GET['on'])){

$gpio\_on = shell\_exec("/usr/local/bin/gpio -g write 17 1");

echo "LED is on";

}

else if(isset($\_GET['off'])){

$gpio\_off = shell\_exec("/usr/local/bin/gpio -g write 17 0");

echo "LED is off";

}

?>

</body>

</html>

-------------------------------------------------------------------------------------------------------------

now press "cnt+O" to save and then "cnt+X" to exit

once go to /var/www library there we can find rahul.php open it and cross check it

now check ip address of our pi by giving ifconfig in terminal window

enter that ip in your mobile brouser as 192.168.2.26/rahul.php

bingo here it is-

**Mail code**

import RPi.GPIO as GPIO

from subprocess import call

import time

import os

import glob

import smtplib

import base64

from email.mime.image import MIMEImage

from email.mime.multipart import MIMEMultipart

import subprocess

gmail\_user = "checking999mail@gmail.com"

gmail\_pwd = "mail999checking"

FROM = 'checking999mail@gmail.com'

TO = ['hyd.embedded@pantechmail.com'] #must be a list

i=1

while (i):

i=i-1

subprocess.Popen( "fswebcam -r 1280x720 /home/pi/Downloads/pan.jpg", shell=True )

time.sleep(1)

msg = MIMEMultipart()

time.sleep(1)

msg['Subject'] ="testing msg send from python"

time.sleep(1)

fp = open("/home/pi/Downloads/pan.jpg", 'rb')

time.sleep(1)

img = MIMEImage(fp.read())

time.sleep(1)

fp.close()

time.sleep(1)

msg.attach(img)

time.sleep(1)

try:

server = smtplib.SMTP("smtp.gmail.com", 587) #or port 465 doesn't seem to work!

print "smtp.gmail"

server.ehlo()

print "ehlo"

server.starttls()

print "starttls"

server.login(gmail\_user, gmail\_pwd)

print "reading mail & password"

server.sendmail(FROM, TO, msg.as\_string())

print "from"

server.close()

print 'successfully sent the mail'

except:

print "failed to send mail"

sudo apt-get install apache2

sudo apt-get install php5 libapache2-mod-php5

sudo apt-get install git-core

git clone git://git.drogon.net/wiringPi

cd wiringPi

./build

* **Conclusion**

Thus we have successfully created small dashboard application deployed on cloud.