ASSIGNMENT NUMBER: D2

TITLE: Create a small dashboard application to be deployed on cloud.

PROBLEM STATEMENT /DEFINITION: Create a small dashboard application to be deployed on cloud. Different publisher devices can publish their information and interested application can subscribe.

Learning Objectives:

- To Develop application based on cloud.
- To understand different sensory inputs
- Understand client server model programming.

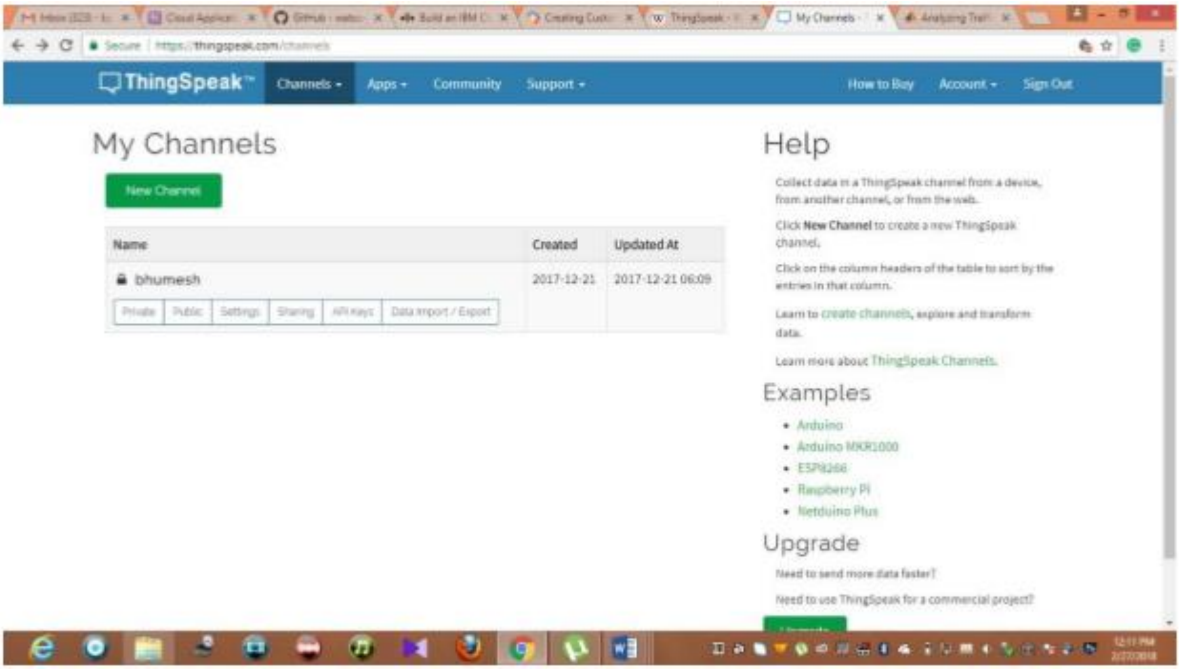
Learning Outcomes:

The students will be able to

- Perform the connectivity with Raspberry-Pi, Beagle board, Arduino and other micro controller.
- Implement cloud application with the help of client server programming by using python.

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



My Channels

Name	Created	Updated At
bhumesh	2017-12-21	2017-12-21 06:09

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial project?

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

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

Put some PHP content in it:

```
<?php echo "hello world"; ?>
```

Now save and refresh your browser. You should see "hello world". This is not dynamic but still served by PHP. Try something dynamic:

```
<?php echo date('Y-m-d H:i:s'); ?>
```

or show your PHP info:

```
<?php phpinfo(); ?>
```

```
 sudo apt-get install apache2 –y
```

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

```
sudo apt-get install git-core
```

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

cd wiringPi

./build
```

------------------------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>
```

--------or can try this also------------ <?php $myfile = fopen("/home/pi/log.txt", "r") or die("Unable to open file!"); echo fread($myfile,filesize("/home/pi/log.txt")); fclose($myfile); ?>

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
---------------------now open terminal and enter this:------------------
$cd /var/www/html
$sudo leafpad gpio.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="gpio2.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>
//-or can try this also
<?php
$myfile = fopen("/home/pi/log.txt", "r") or die("Unable to open file!");
echo fread($myfile,filesize("/home/pi/log.txt"));
fclose($myfile);
?>

//now press "cnt+s" to save and then to exit
//-once go to /var/www/html library there we can find gpio.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 browser as 192.168.2.26/gpio.php-

```
import subprocess

from subprocess import call

import RPi.GPIO as GPIO

import time


GPIO.setmode(GPIO.BCM)

GPIO.setup(6, GPIO.IN, pull_up_down = GPIO.PUD_UP)

GPIO.setup(13, GPIO.IN ,pull_up_down = GPIO.PUD_UP)

GPIO.setup(19, GPIO.IN, pull_up_down = GPIO.PUD_UP)

GPIO.setup(26, GPIO.IN, pull_up_down = GPIO.PUD_UP)


while True:


 if(GPIO.input(26) == 0):

  pageToOpen="en.wikipedia.org/wiki/A"

  subprocess.Popen(["midori","-a", pageToOpen])

  call(["espeak","Object found is an apple"],shell=True)

  time.sleep(5)

  print('a')

 else:   time.sleep(0.5)

#GPIO.cleanup()
```