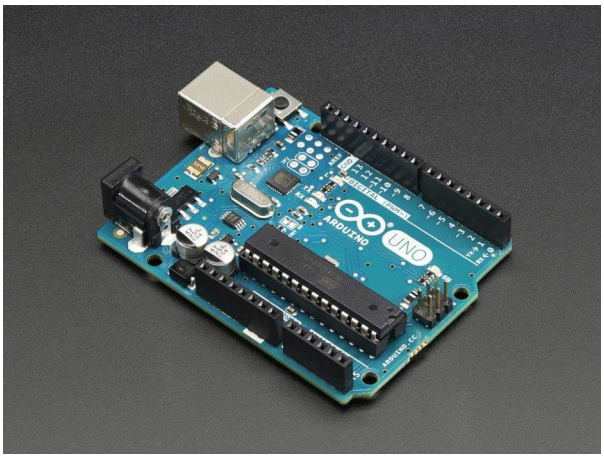
Assignment 01

Aim : Study of Raspberry-Pi, Beagle board, Arduino and other micro controller. (History & Elevation).

Theory:

1. **Arduino**

   Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.



Arduino

2. **Raspberry Pi**

   The **Raspberry Pi** is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals (such as keyboards, mice and cases). However, some accessories have been included in several official and unofficial bundles.

Raspberry Pi

3. **Beagleboard**

The **BeagleBoard** is a low-power open-source single-board computer produced by Texas Instruments in association with Digi-Key and Newark element14. The BeagleBoard was also designed with open source software development in mind, and as a way of demonstrating the Texas Instrument's OMAP3530 system-on-a-chip. The board was developed by a small team of engineers as an educational board that could be used in colleges around the world to teach open source hardware and software capabilities. It is also sold to the public under the Creative Commons share-alike license. The board was designed using Cadence OrCAD for schematics and Cadence Allegro for PCB manufacturing; no simulation software was used.

BeagleBoard

Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	$29.95	$35	$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog

Comparison of Embedded board

Conclusion: Thus we have studied Raspberry-Pi, Beagle board, Arduino and other microcontroller.

Assignment 02

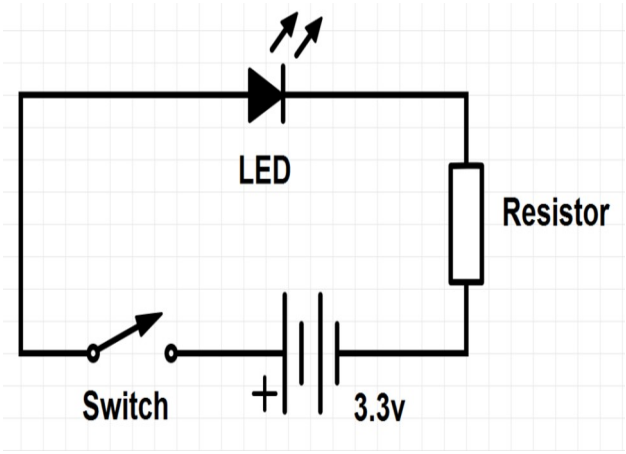
Aim: Study of Connectivity and configuration of Raspberry Pi/ Arduino board circuit with basic peripherals, LEDS. Understanding GPIO and its use in program.

Requirement: Raspberry Pi/ Arduino UNO, basic peripherals like LEDS.

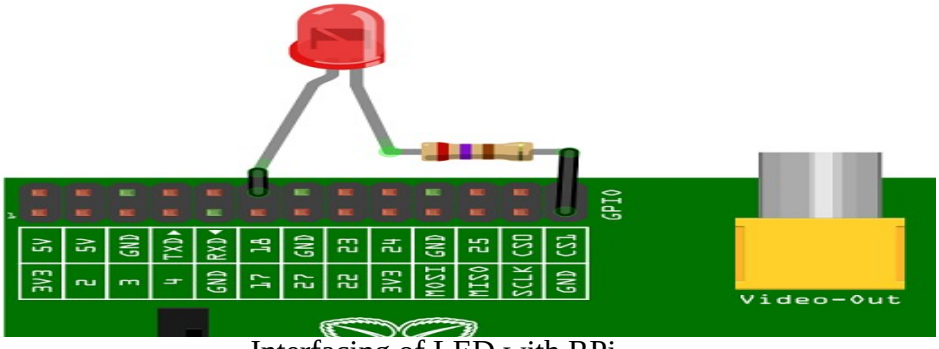
Theory:

1. LED

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current is passed through it. Light is produced when the particles that carry the current (known as electrons and holes) combine together within the semiconductor material. Since light is generated within the solid semiconductor material, LEDs are described as solid- state devices. The term solid-state lighting, which also encompasses organic LEDs (OLEDs),distinguishes this lighting technology from other sources that use heated filaments (incandescent and tungsten halogen lamps) or gas discharge (fluorescent lamps)



LED connecting circuit



Interfacing of LED with RPi

Program for blinking LED

```
#led.py for Raspberry pi where led is connected to pin 3

import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3,GPIO.OUT)
try:
        while True:
                GPIO.output(3,True)
                time.sleep(2)
                GPIO.output(3,False)
                time.sleep(2)
except KeyboardInterrupt:
        GPIO.cleanup()
```

Conclusion: Thus we have studied, connectivity and configuration of Raspberry-Pi with basic peripherals, LEDS and understanding GPIO and its use in program.

Assignment 03

Aim: Understanding the connectivity of Raspberry-Pi /Beagle board circuit with temperature sensor. Write an application to read the environment temperature. If temperature crosses a threshold value, the application indicated user using LEDSs.

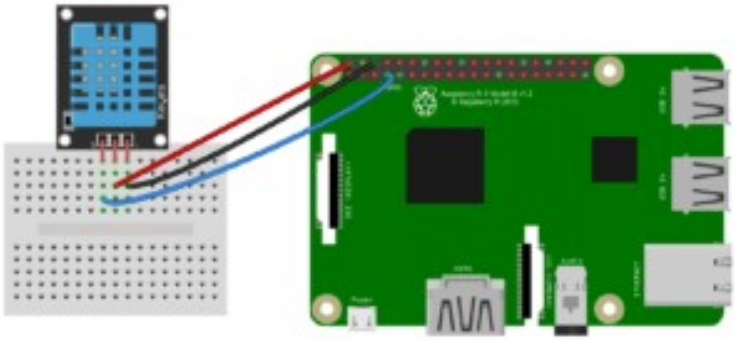
Requirement: LED, Beagle Board/Raspberry pi, DHT11 temperature and humidity sensor

Theory:

THREE PIN DHT11 WITH SSH OUTPUT

DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output. DHT11 can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results. DHT11 is a low cost humidity and temperature sensor which provides high reliability and long term stability.

DHT11 Temperature sensor



Connection of DHT11 to raspberry pi

As shown in fig. DHT11 sensor has three pins: GND, DATA and VCC
Steps:

1. Connect GND and VCC to ground and VCC of raspberry pi
2. Connect Data pin of temp sensor to GPIO 4 pin.
3. Connect LED to 21 pin of raspberry to indicate when temperature crosses a threshold value
4. Run Python code temp.py

Before running the python script perform the following steps:

git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo apt-get update
sudo apt-get install build-essential python-dev
sudo python setup.py install

Program for sensing humidity and temperature

```
#temp.py

import Adafruit_DHT
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(21,GPIO.OUT) # LED on 21 pin
while True:
humidity, temp = Adafruit_DHT.read_retry(11, 17) # data pin of DHT11 sensor on GPIO 11
print ("Humidity = {} %; Temperature = {} C".format(humidity, temp))
if(temp>25):
#threshold value 25
print "value is greate than 25"
GPIO.output(21,True)
time.sleep(2)
GPIO.output(21,False)
else:
print "temperature if below 25"
GPIO.output(21,False)
time.sleep(2)
```

Conclusion: Thus we have studied, connectivity of Raspberry-Pi with temperature sensor.

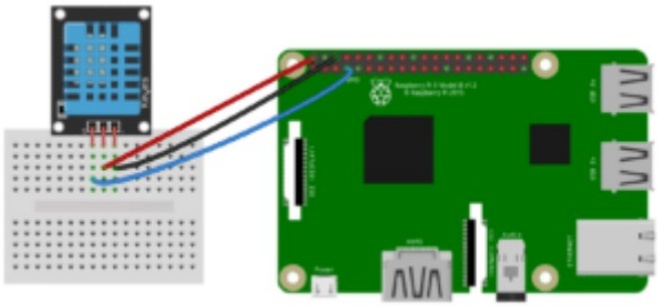
Assignment No. 04

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

Requirement: LED, Beagle Board/Raspberry pi, DHT11 temperature and humidity sensor, Cloud Platform ThingSpeak

Theory:
DHT11 is a Humidity and Temperature Sensor, which generates calibrated digital output. DHT11 can be interface with any microcontroller like Arduino, Raspberry Pi, etc. and get instantaneous results. DHT11 is a low cost humidity and temperature sensor which provides high reliability and long term stability.

DHT11 Temperature sensor



Connection of DHT11 to raspberry pi

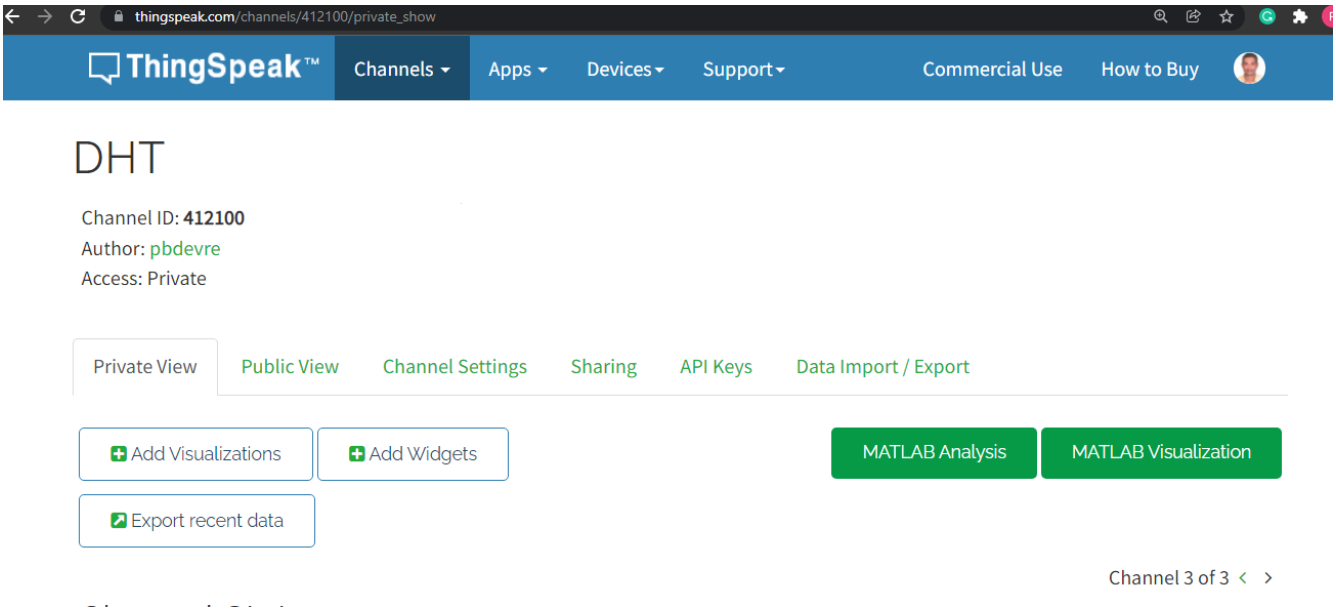
As shown in fig. DHT11 sensor has three pins: GND, DATA and VCC
Steps:
1. Connect GND and VCC to ground and VCC of raspberry pi
2. Connect Data pin of temp sensor to GPIO pin.
3. Connect LED to one of the GPIO pin of raspberry to indicate when temperature crosses a threshold value
4. Run Python code temp.py Before running the python script perform the following steps:
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT
sudo apt-get update sudo apt-get install build-essential python-dev
sudo python setup.py install

Thingspeak Cloud Platform for IoT

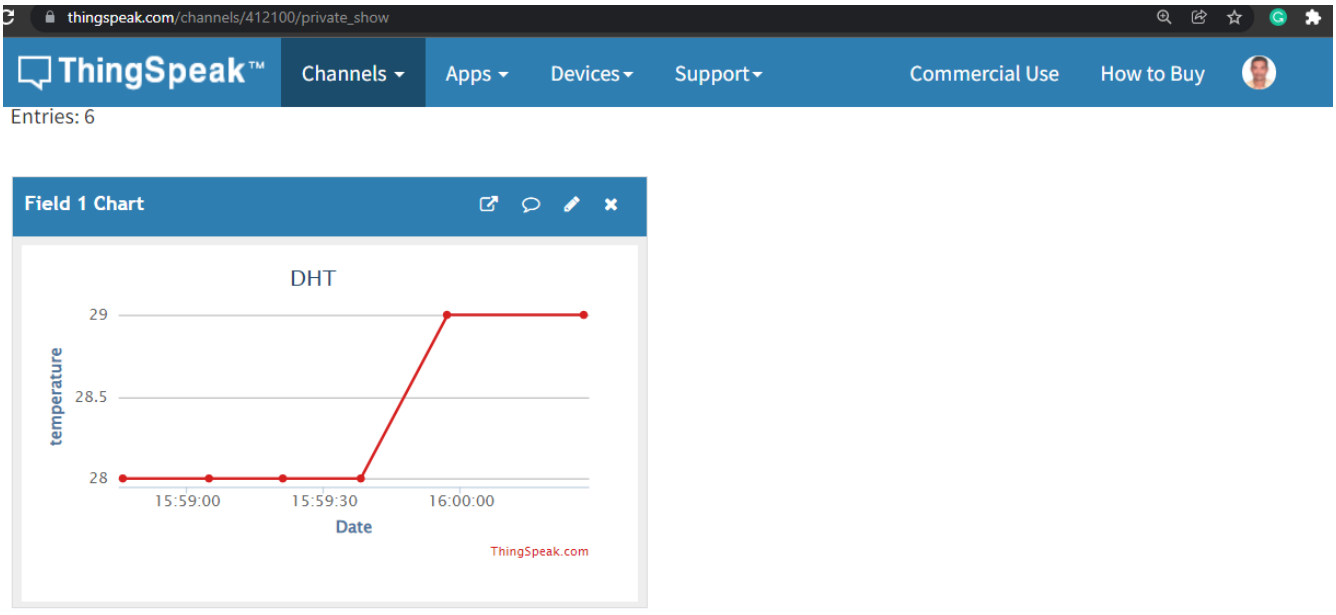
ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.

1. SignUp and Login

2. Create Channle for storing data over cloud



Channel in ThingSpeak Account



Data Analysis in Thingspeak Account

Program

```
#!/usr/bin/python
import datetime
from datetime import datetime
import RPi.GPIO as GPIO
import time
import decimal
import httplib, urllib
import time
import subprocess
import sys
import Adafruit_DHT

sleep = 2 # how many seconds to sleep between posts to the channel
key = 'GNRSGS61ZETE14L3'  # Thingspeak channel to update
timer=time.time()

while True:
                humidity, temperature = Adafruit_DHT.read_retry(11,17)
                print temperature, humidity
                params = urllib.urlencode({'field1':temperature,'field2':humidity,'key':key})
                headers = {"Content-typZZe": "application/x-www-form-urlencoded","Accept":
"text/plain"}
                conn = httplib.HTTPConnection("api.thingspeak.com:80")
                try:
            conn.request("POST", "/update", params, headers)
            response = conn.getresponse()
            print response.status, response.reason
            data = response.read()
            conn.close()
        except:
            print "connection failed"
```

Conclusion: We have studied connectivity of Raspberry-Pi with DHT11 sensor and an application to get the temprature and huidity for analysis and store in on cloud environment.