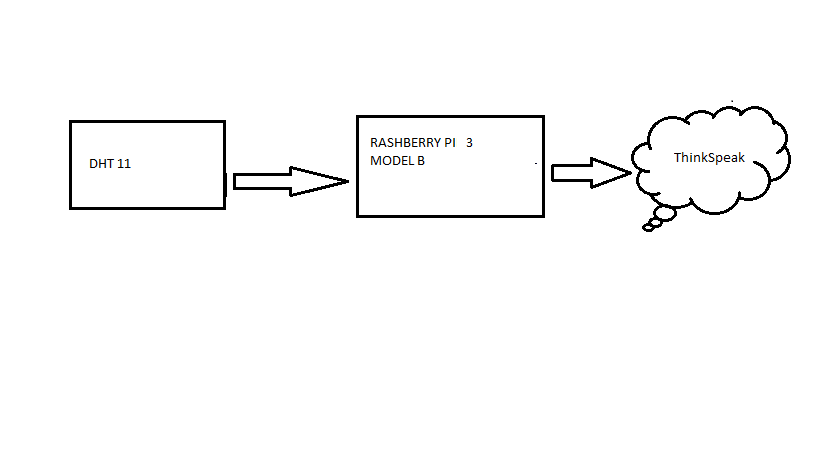
**GROUP 9**

**Exp No.11(A)**

**OBJECTIVE : Publishing data in Thingspeak using DHT11 Humidity Sensor using Rashberry Pi 3 Model B**

**BLOCK DIAGRAM:**

****

**Fig 1**

**EXPLANTION:**

Here we firstly connect the DHT11 sensor in a bread-board then to Rashberry Pi 3 Model B (in any GPIO pin) and and code a program to get a desired output in the serial-monitor and upload it to the ThingSpeak IoT cloud server.

**APPARATUS:**

* DHT 11 sensor
* Resistor
* Jumper Wires
* Rashberry Pie 3 Model B board
* PC(with RPI IDE )
* Micro USB cable

**CODE:**

import sys

import urllib.request

from time import sleep

import Adafruit\_DHT as dht

# Enter Your API key here

myAPI = 'XDJ4ULCJP8EA8A1N'

# URL where we will send the data, Don't change it

baseURL = 'https://api.thingspeak.com/update?api\_key=%s' % myAPI

def DHT11\_data():

# Reading from DHT11 and storing the temperature and humidity

humi, temp = dht.read\_retry(dht.DHT11, 23)

return humi, temp

while True:

try:

humi, temp = DHT11\_data()

# If Reading is valid

if isinstance(humi, float) and isinstance(temp, float):

# Formatting to two decimal places

humi = '%.2f' % humi

temp = '%.2f' % temp

print(humi)

print(temp)

# Sending the data to thingspeak

conn = urllib.request.urlopen(baseURL + '&field1=%s&field2=%s'

%(temp, humi))

conn.read()

# Closing the connection

conn.close()

else:

print ('Error')

# DHT11 requires 2 seconds to give a reading, so make sure to add delay

of above 2 seconds.

sleep(20)

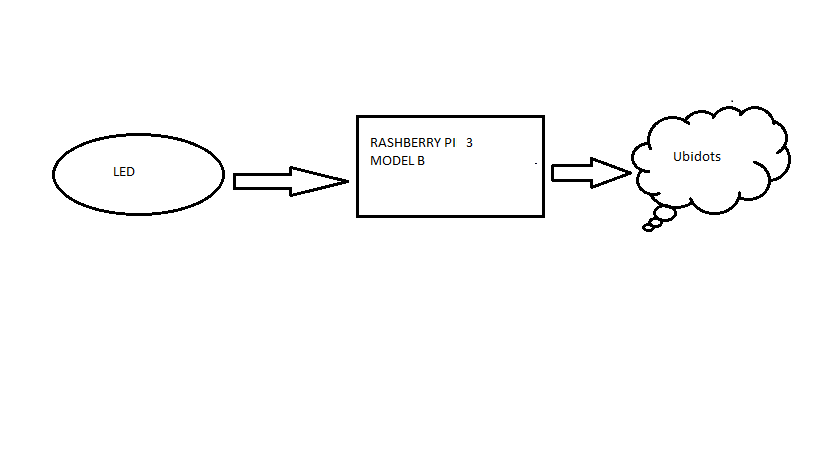
except:

break}

**Exp No.11(B)**

**OBJECTIVE : LED On/Off using Ubidots ( in Rashberry Pi)**

**BLOCK DIAGRAM:**

****

**Fig 2**

**EXPLANATION:**

Here we firstly connect the LED in a bread-board then to Rashberry Pi 3 Model B (in any GPIO pin) and and code a program to get a desired output in the serial-monitor and upload it to the Ubidots IoT cloud server.

**APPARATUS:**

* LED
* Resistor
* Jumper Wires
* Rashberry Pie 3 Model B board
* PC(with RPI IDE )
* Micro USB cable

**CODE:**

import time

import requests

import RPi.GPIO as GPIO

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

GPIO.setup(21,GPIO.OUT)

DEVICE = "rpi\_light"

VARIABLE1 = "light\_on\_off"

TOKEN = "BBFF-0wnE9y4GWddlgbFB6sEFGVl42MskEL"

try:

while True:

light1 = requests.get("http://things.ubidots.com/api/v1.6/devices/"+DEVICE+"/"+VARIABLE1+"/lv?token="+TOKEN

print(light1.content) # Returns the content of the response, in bytes

print(light1.json()) # Returns a JSON object of the result

data = light1.json()

print(type(data))

if data==1.0:

GPIO.output(21,GPIO.HIGH)

#GPIO.output(26,GPIO.LOW)

print("On")

#time.sleep(3)

if data==0.0:

GPIO.output(21,GPIO.LOW)

#GPIO.output(26,GPIO.HIGH)

print("Off")

#time.sleep(3)

except KeyboardInterrupt:

pass

finally:

GPIO.cleanup()

light1 = ''