ENVIRONMETAL MONITORING

(phase-5)

Documentation:

Air pollution, water pollution, and radiation pollutionare significantenvironmental factors that need to be addressed.Proper monitoring is crucial with the goal that by preserving ahealthy society, the planet can achieve sustainable development.With advancements in the internet of things (IoT) and theimprovement of modern sensors, environmental monitoring hasevolved into a smart environment monitoring (SEM) system in recentyears. This article aims to have a critical overview of significantcontributions and SEM research, which include monitoring thequality of air , water pollution, radiation pollution, and agriculturalsystems. The review is divided based on the objectives of applyingSEM methods, analyzing each objective about the sensors used,machinelearning, and classification methods. Moreover, the authorshave thoroughly examined how advancements in sensor technology,the Internet of Things, and machine learning methods have madeenvironmental monitoring into a truly smart monitoring system.

ENVIRONMENTAL MONITORING :

Currently, society overlooks specific challenges inenvironmental monitoring since the aim is to collect and investigateenvironmental data to avoid undefined potential hazards. Concurrently, theprimary causes of environmental infection are rising communities, urbanpopulations, electricity, transportation, and rural improvements. Naturaldisasters, suchas landslides, earthquakes, ; Article no.AJRCOS.68800monitoring systems used for various purposes. ted on water monitoringquality, air monitoring quality, and smart agriculture monitoring systems toaddress This paper is organized as follows. Section II explains backgroundtheory that is related to the es the related work that are linked to environmentmonitoring. Section IV focuses on discussion and analysis and finally, S (IOT)The Internet of Things or IoT means the trillions connected to the Internet andthe worldwide storage and exchange of data. effective computer basedwireless network, anything from a pill to an aircraft can now be transformedinto a part of the IoT. Through attaching sensors to all these different things,artificial intelligence can be applied to otherwise dumb devices so they cantime data without needing a human. The Internet of things makes our societymore d fuses the digital and Fig. 1. Concepts of Internet of Things (IoT)ITORING Currently, society overlooks specific challenges in environmentalmonitoring since the aim is to collect and investigate environmental data toavoid undefined potential hazards. Concurrently, the primary causes ofenvironmental infection are ties, urban populations, electricity, transportation,and rural improvements. Natural disasters, such as landslides, earthquakes,Haji and Sallow; AJRCOS, 9(1): 57-70, 2021; Article no.AJRCOS.68800 59hurricanes, water surges, and tsunamis, are also causes of environmentalaspects that amplify attacks [13]. Furthermore, global warming, seawateracidification, and biodiversity loss may have a far-reaching effect on theatmosphere. Moreover, air, water, and noise pollution are thought to be themost extreme environmental complexities. Surprisingly, the more associationbetween air, water, and noise infection and human well-being isacknowledged, the more risk is mitigated .

SIMULATION:

PROGRAMMING CODE(Python):

```
# make sure to install python-smbus using below command

# sudo apt-get install python-smbus

import smbus

import time

from ctypes import c_short

DEVICE = 0x77 # Default device I2C address

#bus = smbus.SMBus(0) # Rev 1 Pi uses 0
```

```
bus = smbus.SMBus(1) # Rev 2 Pi uses 1

def convertToString(data):

# Simple function to convert binary data into

# a string

return str((data[1] + (256 * data[0])) / 1.2)

def getShort(data, index):

# return two bytes from data as a signed 16-bit value

return c_short((data[index] &lt;&lt; 8) + data[index + 1]).value

def getUshort(data, index):

# return two bytes from data as an unsigned 16-bit value

return (data[index] &lt;&lt; 8) + data[index + 1]

def readBmp180Id(addr=DEVICE):

# Chip ID Register Address

REG_ID = 0xD0

(chip_id, chip_version) = bus.read_i2c_block_data(addr, REG_ID, 2)

return (chip_id, chip_version)

def readBmp180(addr=0x77):

# Register Addresses

REG_CALIB = 0Xaa

REG_MEAS = 0xF4

REG_MSB = 0xF6

REG_LSB = 0xF7

# Control Register Address

CRV_TEMP = 0x2E

CRV_PRES = 0x34

# Oversample setting

OVERSAMPLE = 3 # 0 - 3

# Read calibration data
```

```
# Read calibration data from EEPROM
cal = bus.read_i2c_block_data(addr, REG_CALIB, 22)
# Convert byte data to word values
AC1 = getShort(cal, 0)
AC2 = getShort(cal, 2)
AC3 = getShort(cal, 4)
AC4 = getUshort(cal, 6)
AC5 = getUshort(cal, 8)
AC6 = getUshort(cal, 10)
B1 = getShort(cal, 12)
B2 = getShort(cal, 14)
MB = getShort(cal, 16)
MC = getShort(cal, 18)
MD = getShort(cal, 20)

# Read temperature
bus.write_byte_data(addr, REG_MEAS, CRV_TEMP)
time.sleep(0.005)
(msb, lsb) = bus.read_i2c_block_data(addr, REG_MSB, 2)
UT = (msb&lt;&lt; 8) + lsb

# Read pressure
bus.write_byte_data(addr, REG_MEAS, CRV_PRES + (OVERSAMPLE &lt;&lt; 6))
time.sleep(0.04)
(msb, lsb, xsb) = bus.read_i2c_block_data(addr, REG_MSB, 3)
UP = ((msb&lt;&lt; 16) + (lsb&lt;&lt; 8) + xsb) &gt;&gt; (8 - OVERSAMPLE)

# Refine temperature
```

```
X1 = ((UT - AC6) * AC5) &gt;&gt; 15
X2 = (MC &lt;&lt; 11) / (X1 + MD)
B5 = X1 + X2
temperature = int(B5 + 8) &gt;&gt; 4
temperature = temperature / 10.0

# Refine pressure
B6 = B5 - 4000
B62 = int(B6 * B6) &gt;&gt; 12
X1 = (B2 * B62) &gt;&gt; 11
X2 = int(AC2 * B6) &gt;&gt; 11
X3 = X1 + X2
B3 = (((AC1 * 4 + X3) &lt;&lt; OVERSAMPLE) + 2) &gt;&gt; 2

X1 = int(AC3 * B6) &gt;&gt; 13
X2 = (B1 * B62) &gt;&gt; 16
X3 = ((X1 + X2) + 2) &gt;&gt; 2
B4 = (AC4 * (X3 + 32768)) &gt;&gt; 15
B7 = (UP - B3) * (50000 &gt;&gt; OVERSAMPLE)

P = (B7 * 2) / B4

X1 = (int(P) &gt;&gt; 8) * (int(P) &gt;&gt; 8)
X1 = (X1 * 3038) &gt;&gt; 16
X2 = int(-7357 * P) &gt;&gt; 16
pressure = int(P + ((X1 + X2 + 3791) &gt;&gt; 4))
#pressure = float(pressure / 100.0)
altitude = 44330.0 * (1.0 - pow(pressure / 101325.0, (1.0/5.255)))
```

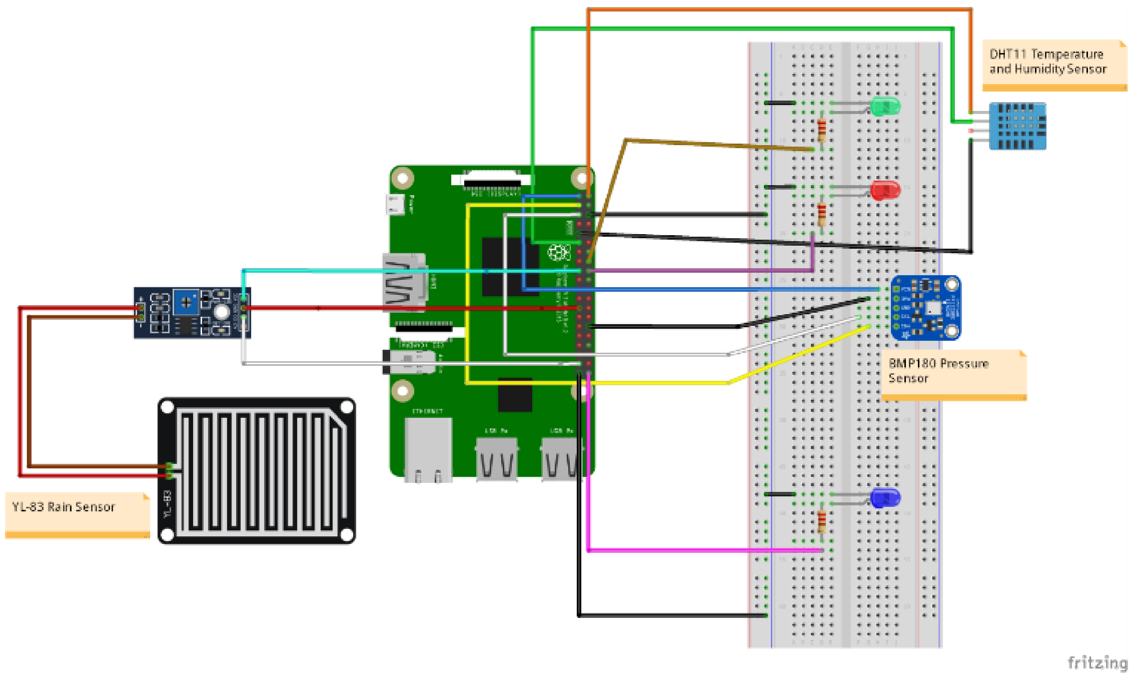
altitude = round(altitude,2)

return (temperature,pressure,altitude)

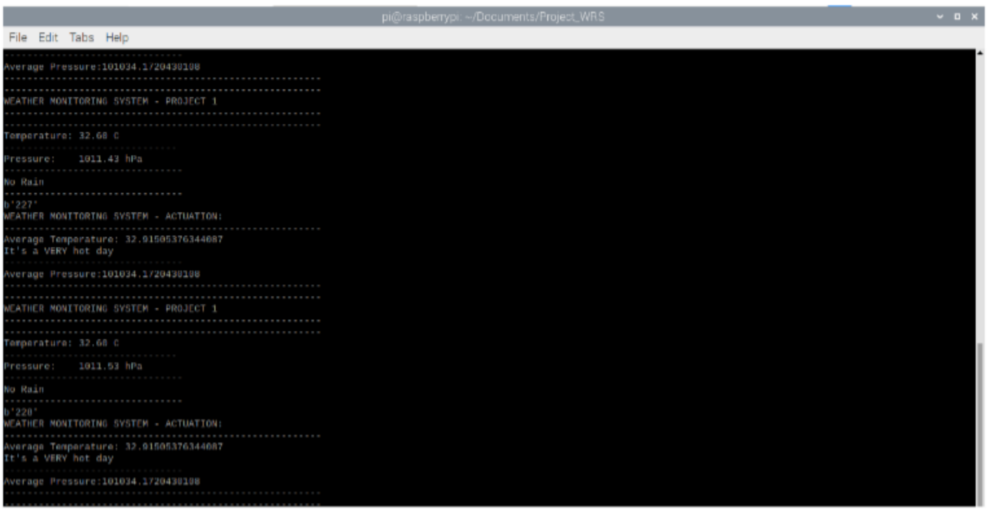
LIBRARY FILES:

- Smbus
- Ctypes
- C_short

SIMULATION:



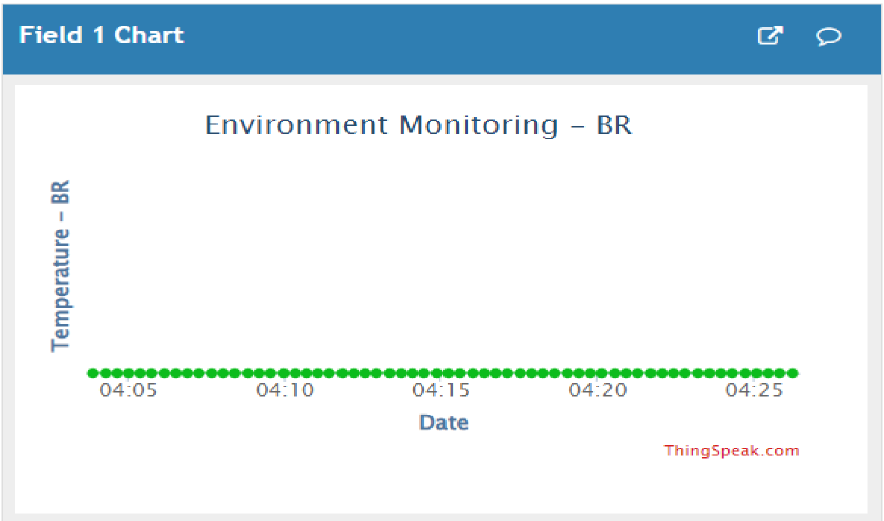
OUTPUT:



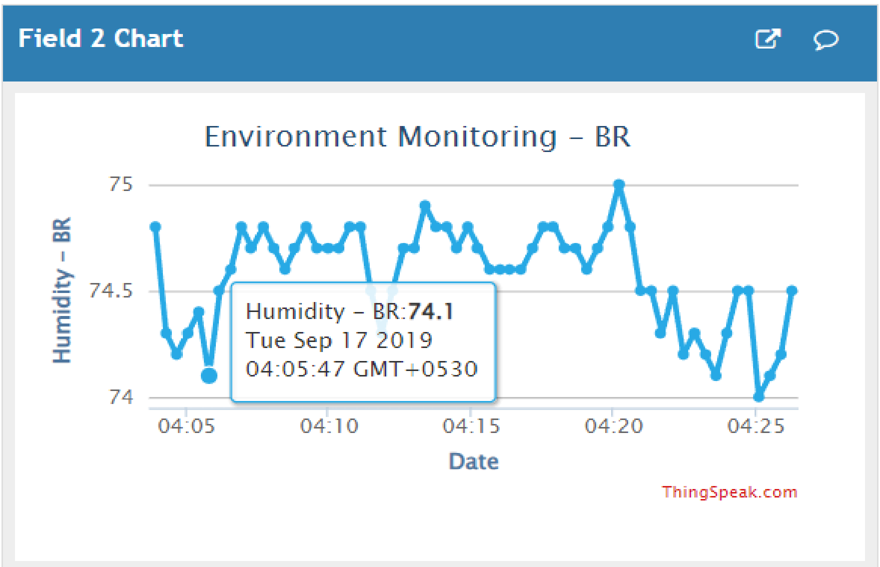
Environmental Monitoring System:

Channel to monitor environmental conditions of my apartment. Channel is supplied data via an ESP8266 Wifi module which has a BME280 sensor connected to it. These are powered from a USB wall socket outlet for minimal setup/installation.

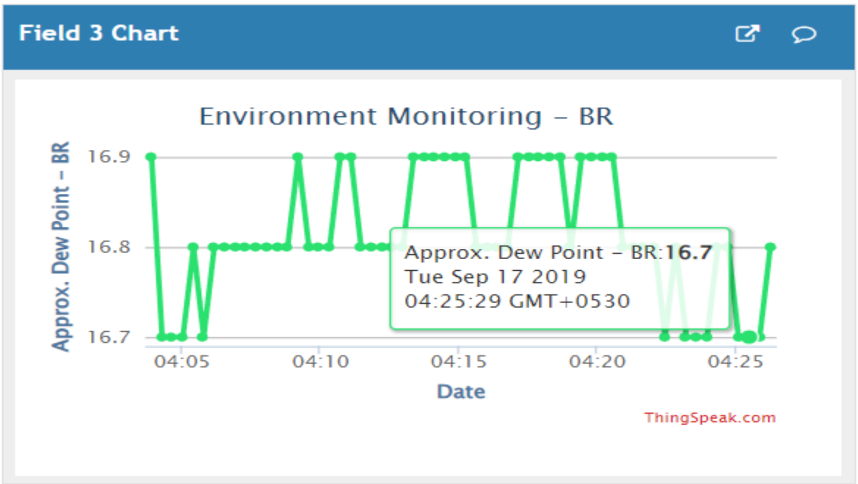
Temperature:



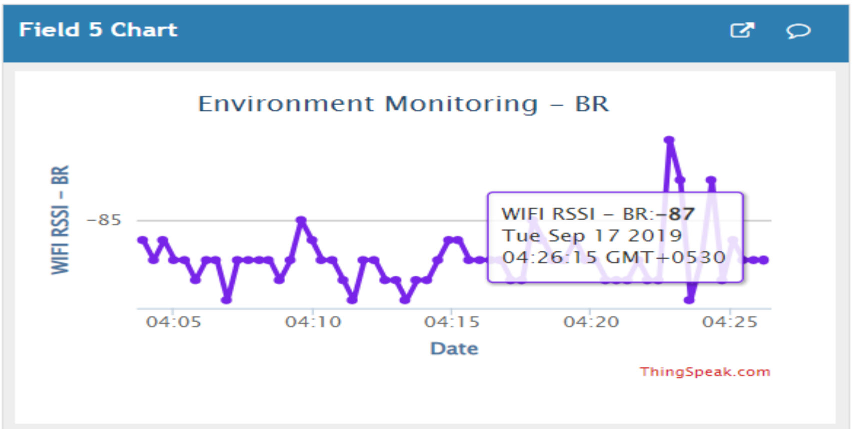
Humidity:



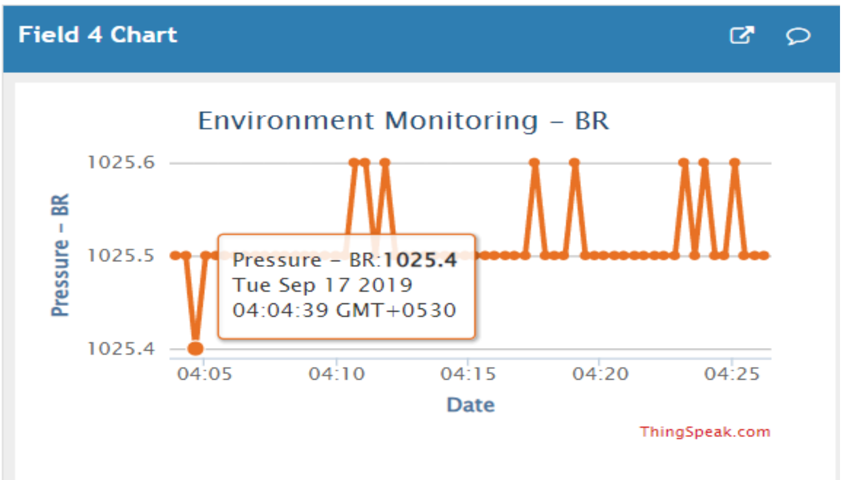
Approximate Dew Point:



WiFi :



Pressure:



PROJPROJECT-ID:_224686_TEAM_1

PROJECT NAME: ENVIRONMENTAL MONITERING

NAME:KAMALESH.K

COLLEGE CODE:4204

REGISTER NO.:420421106023

FINAL SUBMISSION

(Project Submission Part-5)