

ENVIRONMENTAL MONITORING

(phase-5)

Documentation:

Air pollution, water pollution, and radiation pollution are significant environmental factors that need to be addressed. Proper monitoring is crucial with the goal that by preserving a healthy society, the planet can achieve sustainable development. With advancements in the internet of things (IoT) and the improvement of modern sensors, environmental monitoring has evolved into a smart environment monitoring (SEM) system in recent years. This article aims to have a critical overview of significant contributions and SEM research, which include monitoring the quality of air, water pollution, radiation pollution, and agricultural systems. The review is divided based on the objectives of applying SEM methods, analyzing each objective about the sensors used, machine learning, and classification methods. Moreover, the authors have thoroughly examined how advancements in sensor technology, the Internet of Things, and machine learning methods have made environmental monitoring into a truly smart monitoring system.

ENVIRONMENTAL MONITORING :

Currently, society overlooks specific challenges in environmental monitoring since the aim is to collect and investigate environmental data to avoid undefined potential hazards. Concurrently, the primary causes of environmental infection are rising communities, urban populations, electricity, transportation, and rural improvements. Natural disasters, such as landslides, earthquakes, ; Article no.AJRCOS.68800 monitoring systems used for various purposes. ted on water monitoring quality, air monitoring quality, and smart agriculture monitoring systems to address This paper is organized as follows.

Section II explains background theory that is related to the es the related work that are linked to environment monitoring. Section IV focuses on discussion and analysis and finally, S (IOT) The Internet of Things or IoT means the trillions connected to the Internet and the worldwide storage and exchange of data. effective computer based wireless network, anything from a pill to an aircraft can now be transformed into a part of the IoT. Through attaching sensors to all these different things, artificial intelligence can be applied to otherwise dumb devices so they can time data without needing a human. The Internet of things makes our society more d fuses the digital and Fig. 1. Concepts of Internet of Things (IoT) ITORING Currently, society overlooks specific challenges in environmental monitoring since the aim is to collect and investigate environmental data to avoid undefined potential hazards. Concurrently, the primary causes of environmental 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 59 hurricanes, water surges, and tsunamis, are also causes of environmental aspects that amplify attacks [13]. Furthermore, global warming, seawater acidification, and biodiversity loss may have a far-reaching effect on the atmosphere. Moreover, air, water, and noise pollution are thought to be the most extreme environmental complexities. Surprisingly, the more association between air, water, and noise infection and human well-being is acknowledged, 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] << 8) + data[index + 1]).value

def getUshort(data, index):
    # return two bytes from data as an unsigned 16-bit value
    return (data[index] << 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 <<< 8) + lsb


# Read pressure

bus.write_byte_data(addr, REG_MEAS, CRV_PRES + (OVERSAMPLE <<< 6))
time.sleep(0.04)
(msb, lsb, xsb) = bus.read_i2c_block_data(addr, REG_MSB, 3)
UP = ((msb <<< 16) + (lsb <<< 8) + xsb) >>> (8 - OVERSAMPLE)


# Refine temperature

```

```

X1 = ((UT - AC6) * AC5) >> 15
X2 = (MC << 11) / (X1 + MD)
B5 = X1 + X2
temperature = int(B5 + 8) >> 4
temperature = temperature / 10.0

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

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

P = (B7 * 2) / B4




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

```

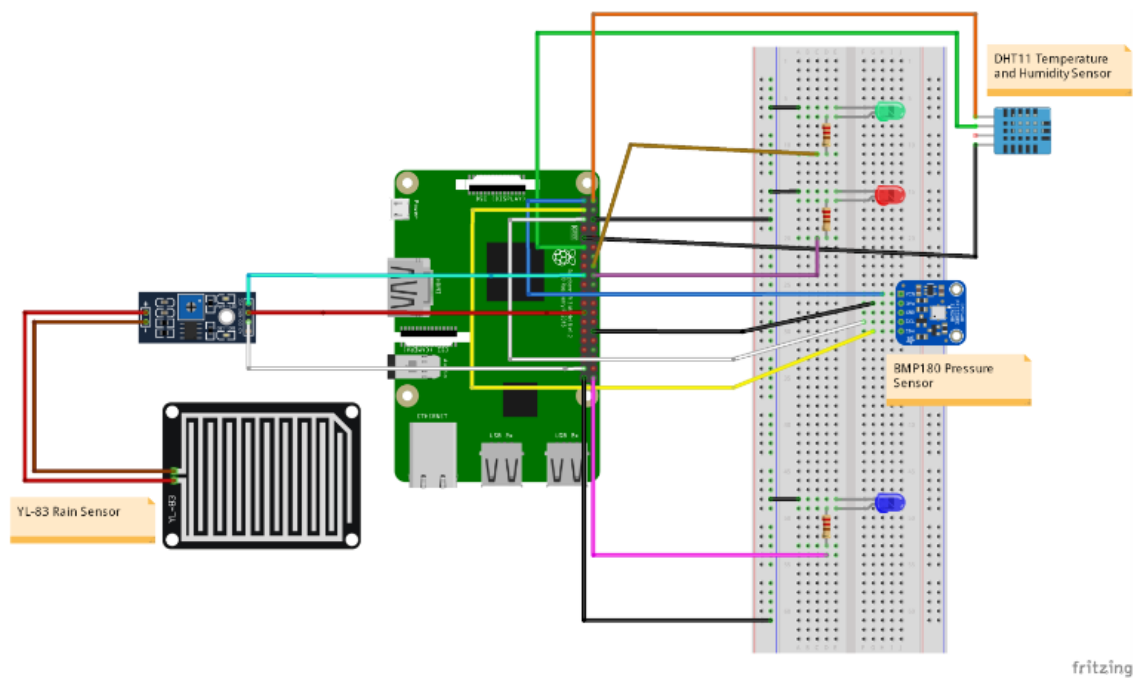
```
altitude = round(altitude,2)
```

```
return (temperature,pressure,altitude)
```

LIBRARY FILES:

-  Smbus
-  Ctypes
-  C_short

SIMULATION:



OUTPUT:

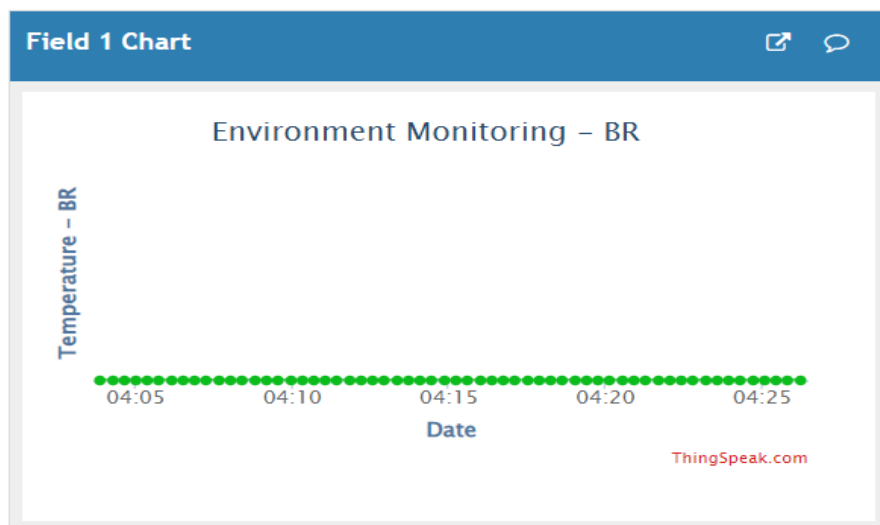
```
pi@raspberrypi: ~/Documents/Project_WRS
File Edit Tabs Help
Average Pressure:101034.1729430108
.....
WEATHER MONITORING SYSTEM - PROJECT 1
.....
Temperature: 32.68 C
.....
Pressure: 1011.43 hPa
.....
No Rain
.....
0"227"
WEATHER MONITORING SYSTEM - ACTUATION:
.....
Average Temperature: 32.91505376344687
It's a VERY hot day
.....
Average Pressure:101034.1729430108
.....
WEATHER MONITORING SYSTEM - PROJECT 1
.....
Temperature: 32.68 C
.....
Pressure: 1011.53 hPa
.....
No Rain
.....
0"228"
WEATHER MONITORING SYSTEM - ACTUATION:
.....
Average Temperature: 32.91505376344687
It's a VERY hot day
.....
Average Pressure:101034.1729430108
.....
```

Environmental Monitoring System:

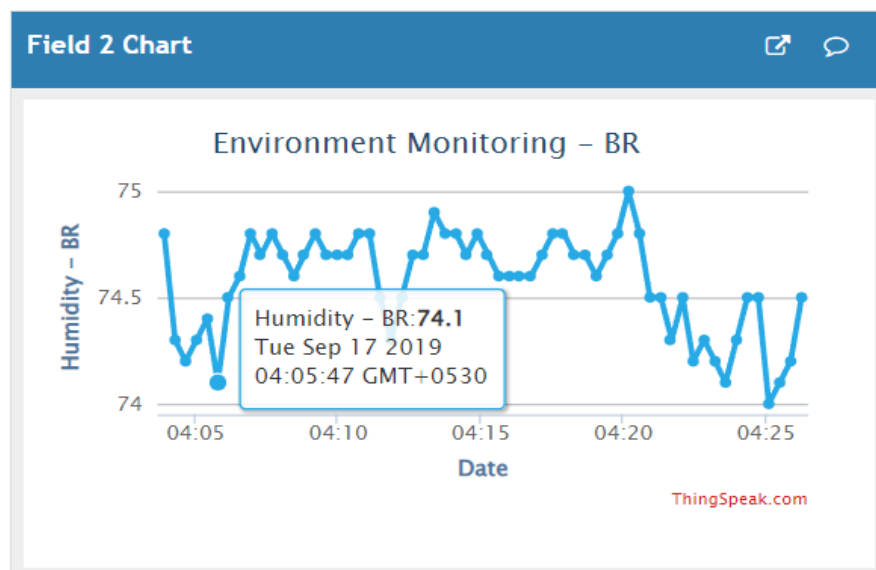


Channel to monitor environmental conditions of my apartment. Channel is supplied data via an ESP8266 Wifi module which has a BME280 sensor connected to it. These are powered from a USB wall socket outlet for minimal setup/installation.

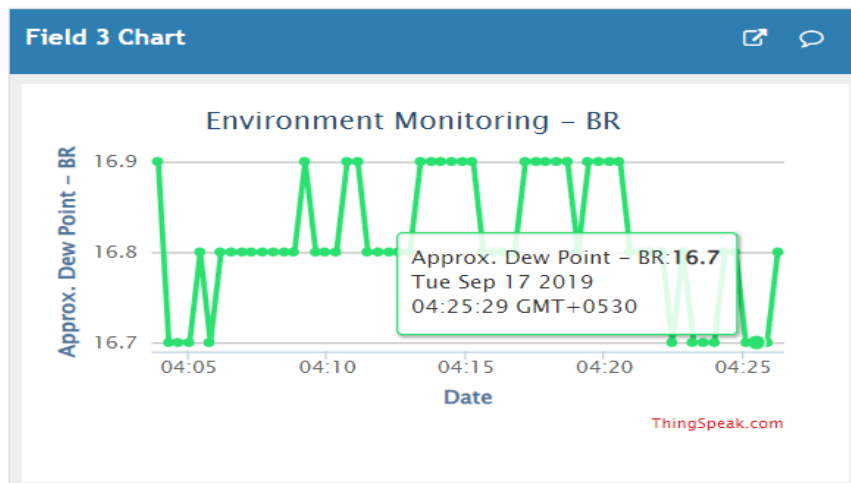
Temperature:



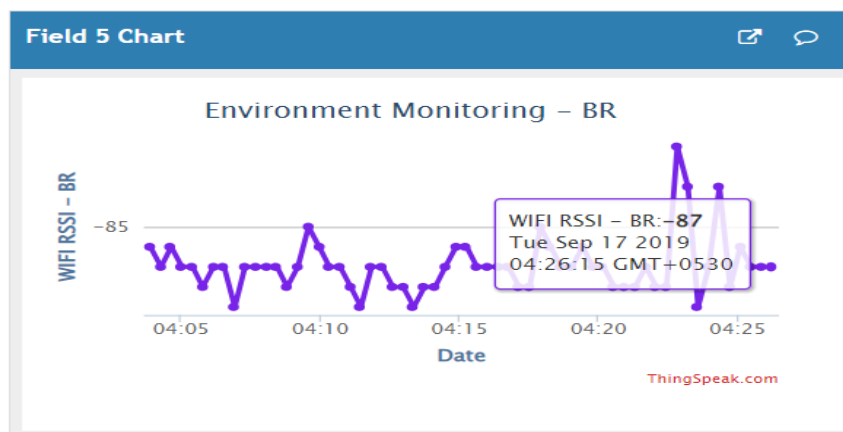
Humidity:



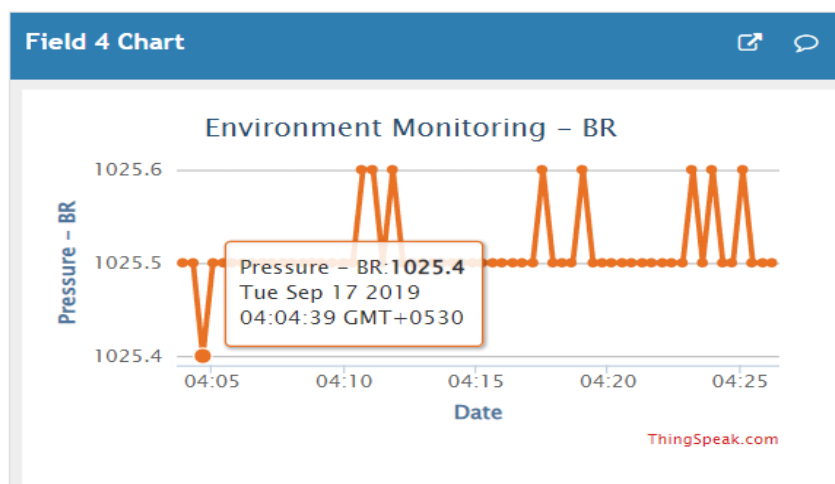
Approximate Dew Point:



WiFi :



Pressure:



PROJ PROJECT-ID: _224686_TEAM_1

PROJECT NAME: ENVIRONMENTAL MONITORING

NAME:GOKUL.M

COLLEGE CODE:4204

REGISTER NO.:420421106015

FINAL SUBMISSION

(Project Submission Part-5)