**GANESH COLLEGE OF ENGINEERING**

**ENVIRONMENTAL MONITORING**

****

****

**M.POOJA**

**S.RASITHA**

**S.NARMADHA**

**ABSTRACT:**

**Environmental pollution is an international serious, acute, health and social problem that humans are responsible for. Rapid technology evolution could provide us solutions to predict and monitor environmental parameters and thus minimize or even eliminate environmental pollution. Using a wireless sensor network (WSN) and an Internet of Things (IoT) implementation, we can create an eco-friendly and controlled environment. This paper presents an embedded and standalone system that can monitor, store and analyse environmental data as well as calculate energy consumption in specific settings. Information is available via a web application with central management abilitie.**

**Environmental monitoring article:**

**We talked about the purpose and importance of monitoring processes that can have an ecological impact. This blog post explores the critical role of IoT and how industries seeking ways to improve their environmental stewardship can utilize the Internet of Things to create change.**

**Embedded communications modules:**

** These monitoring systems can be programmed to detect abnormalities or specific conditions, then trigger alerts via email or text, as well as automated processes. These can include anything from launching service tickets to shutting systems down to thwart a disaster. In other words, an environmental monitoring system using IoT acts as the eyes, ears, and mouthpiece for an application — watching, listening, and reporting on a vast range of processes — and even taking action to thwart damage**

**IoT-based environmental monitoring :**

* **Monitor the Environment: Environmental condition monitors across fields, industrial sites and water management systems require installed sensors as well as an information delivery system, such as Digi XBee wireless communication modules and sensor connectivity gateways. These connected devices gather and deliver critical information exactly where it is needed.**
* **Measure Data: To measure environmental impact, these systems must make it possible to evaluate key data points that can indicate everything from water and chemical leaks to critical equipment failures.**
* **Catalog Data: The massive amounts of data collected from environmental monitoring stations around the globe cannot be overstated. There are global databases that catalog an enormous range of environmental data, such as the Microsoft Planetary Computer. Industrial sites and other enterprises, similarly, must utilize cloud and data center storage to catalog the gathered data for accessibility by business applications**

**Water Quality Monitoring:**

****

**These advanced smart water monitoring systems using IoT technologies enable accurate measurements of contaminants, oxygen levels, additional factors, and pH levels. IoT technology allows the detection of harmful substances public it reaches homes and buildings. The innovative technology helps us to sustain our health and wellness.**

**Some examples include:**

* **Municipal water treatment monitoring**
* **Stormwater and groundwater monitoring**
* **Agricultural irrigation monitoring and control**

**Air Quality Monitoring:**

****

**Air quality monitoring, science and industry can create change. These critical metrics deliver the insights for municipalities to make decisions for urban planning, for industrial operations to mitigate their impact, and for entire auto makers to continually improve designs to reduce emissions. Even deploying IoT to manage traffic flow in cities can massively reduce vehicle emissions and support cleaner air**

**.Some real-world examples of air quality monitoring include:**

* **Carbon monoxide monitoring in homes and buildings**
* **Methane monitoring in agriculture and waste management**
* **Ambient air quality monitoring for pollutants, lead and toxic particulates**

**Energy Monitoring:**

****

**Leading energy providers today are rapidly integrating a wide range of IoT monitoring and mitigation techniques to reduce usage, as well as clean energy solutions to reduce energy consumption and promote sustainability. In the process, these techniques can also save money for everyone relying on the electric grid.**

**Energy monitoring supports numerous energy management goals:**

* **Reduction in the use of fossil fuels in homes and businesses**

**Stabilizing the power grid**

* **Preventing spikes in energy usage, and associated equipment failures and service disruption**

**Import standard python modules**

**Import time**

**PYTHON CODEING**

**# import Adafruit Blinka**

**Import board**

**Import busio**

**# import CircuitPython sensor libraries**

**Import adafruit\_sgp30**

**Import adafruit\_veml6070**

**From adafruit\_bme280 import basic as adafruit\_bme280**

**# import Adafruit IO REST client**

**From Adafruit\_IO import Client, Feed, RequestError**

**# SPDX-FileCopyrightText: 2018 Brent Rubell for Adafruit Industries**

**#**

**# SPDX-License-Identifier: MIT**

**#**

**# Adafruit IO Environmental Monitor for Feather or Raspberry Pi –**

**# an internet-enabled environmental monitor**

**# Import standard python modules**

**Import time**

**# import Adafruit Blinka**

**Import board**

**Import busio**

**# import CircuitPython sensor libraries**

**Import adafruit\_sgp30**

**Import adafruit\_veml6070**

**From adafruit\_bme280 import basic as adafruit\_bme280**

**# import Adafruit IO REST client**

**From Adafruit\_IO import Client, Feed, RequestError**

**# loop timeout, in seconds.**

**LOOP\_DELAY = 10**

**# Set to your Adafruit IO key.**

**# Remember, your key is a secret,**

**# so make sure not to publish it when you publish this code!**

**ADAFRUIT\_IO\_KEY = ‘YOUR\_AIO\_KEY’**

**# Set to your Adafruit IO username.**

**# (go to** [**https://accounts.adafruit.com**](https://accounts.adafruit.com) **to find your username)**

**ADAFRUIT\_IO\_USERNAME = ‘YOUR\_AIO\_USERNAME’**

**# Create an instance of the REST client**

**Aio = Client(ADAFRUIT\_IO\_USERNAME, ADAFRUIT\_IO\_KEY)**

**Try: # if we already have the feeds, assign them.**

**Tvoc\_feed = aio.feeds(‘tvoc’)**

**eCO2\_feed = aio.feeds(‘eco2’)**

**uv\_feed = aio.feeds(‘uv’)**

**temperature\_feed = aio.feeds(‘temperature’)**

**humidity\_feed = aio.feeds(‘humidity’)**

**pressure\_feed = aio.feeds(‘pressure’)**

**altitude\_feed = aio.feeds(‘altitude’)**

**except RequestError: # if we don’t, create and assign them.**

**Tvoc\_feed = aio.create\_feed(Feed(name=’tvoc’))**

**eCO2\_feed = aio.create\_feed(Feed(name=’eco2’))**

**uv\_feed = aio.create\_feed(Feed(name=’uv’))**

**temperature\_feed = aio.create\_feed(Feed(name=’temperature’))**

**humidity\_feed = aio.create\_feed(Feed(name=’humidity’))**

**pressure\_feed = aio.create\_feed(Feed(name=’pressure’))**

**altitude\_feed = aio.create\_feed(Feed(name=’altitude’))**

**# Create busio I2C**

**I2c = busio.I2C(board.SCL, board.SDA)**

**# Create VEML6070 object.**

**Uv = adafruit\_veml6070.VEML6070(i2c)**

**# Create BME280 object.**

**Bme280 = adafruit\_bme280.Adafruit\_BME280\_I2C(i2c)**

**Bme280.sea\_level\_pressure = 1013.25**

**# Create SGP30 object using I2C.**

**Sgp30 = adafruit\_sgp30.Adafruit\_SGP30(i2c)**

**Sgp30.iaq\_init()**

**Sgp30.set\_iaq\_baseline(0x8973, 0x8aae)**

**# Sample VEML6070**

**Def sample\_VEML():**

**For \_ in range(10):**

**Uv\_raw = uv.uv\_raw**

**Return uv\_raw**

**While True:**

**Print(‘Reading sensors…’)**

**# Read SGP30.**

**eCO2\_data = sgp30.eCO2**

**tvoc\_data = sgp30.TVOC**

**# Read VEML6070.**

**Uv\_data = sample\_VEML()**

**# Read BME280.**

**Temp\_data = bme280.temperature**

**# convert temperature (C->F)**

**Temp\_data = int(temp\_data) \* 1.8 + 32**

**Humid\_data = bme280.humidity**

**Pressure\_data = bme280.pressure**

**Alt\_data = bme280.altitude**

**Print(‘sending data to adafruit io…’)**

**# Send SGP30 Data to Adafruit IO.**

**Print(‘eCO2:’, eCO2\_data)**

**Aio.send(eCO2\_feed.key, eCO2\_data)**

**Print(‘tvoc:’, tvoc\_data)**

**Aio.send(tvoc\_feed.key, tvoc\_data)**

**Time.sleep(2)**

**# Send VEML6070 Data to Adafruit IO.**

**Print(‘UV Level: ‘, uv\_data)**

**Aio.send(uv\_feed.key, uv\_data)**

**Time.sleep(2)**

**# Send BME280 Data to Adafruit IO.**

**Print(‘Temperature: %0.1f C’ % temp\_data)**

**Aio.send(temperature\_feed.key, temp\_data)**

**Print(“Humidity: %0.1f %%” % humid\_data)**

**Aio.send(humidity\_feed.key, int(humid\_data))**

**Time.sleep(2)**

**Print(“Pressure: %0.1f hPa” % pressure\_data)**

**Aio.send(pressure\_feed.key, int(pressure\_data))**

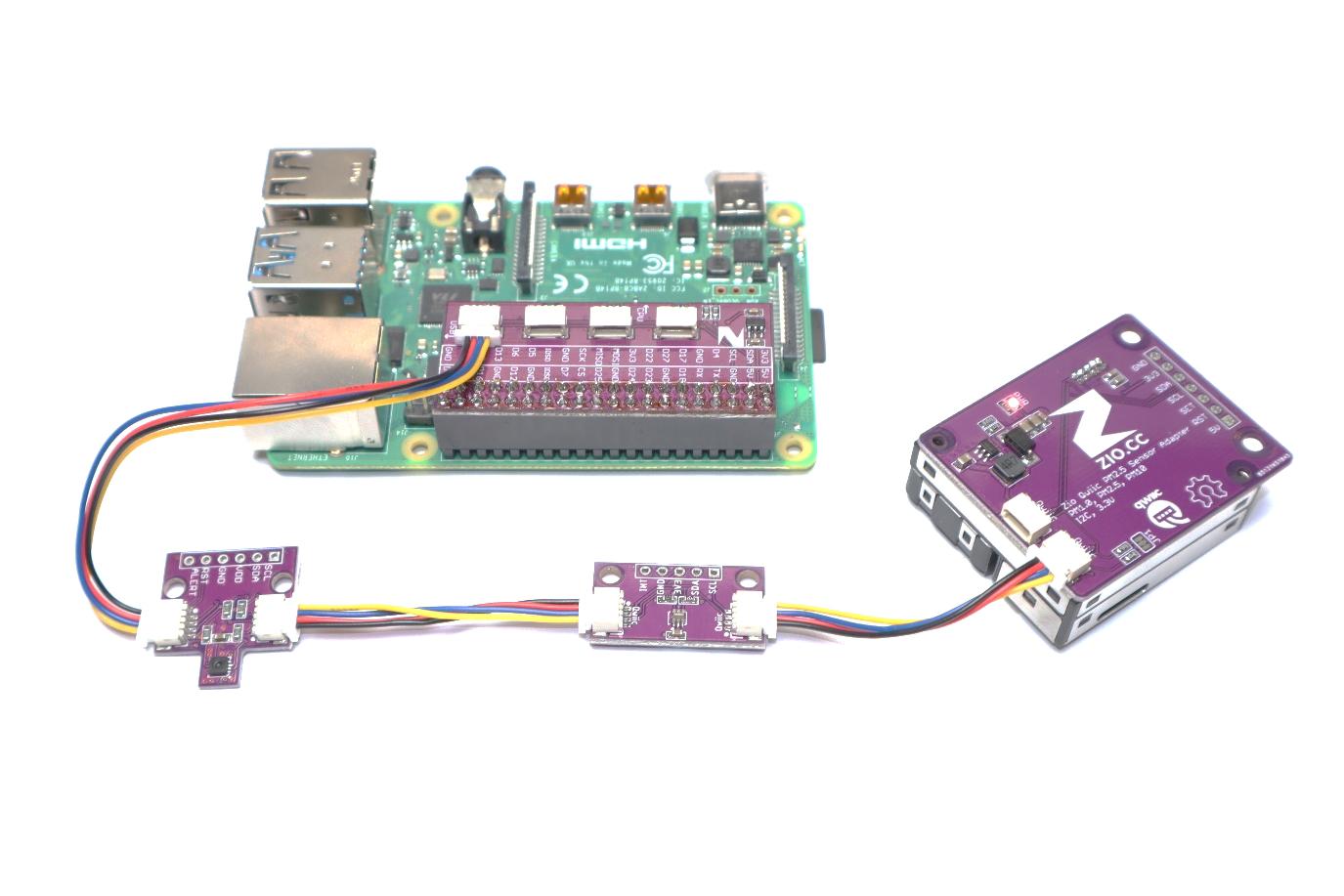
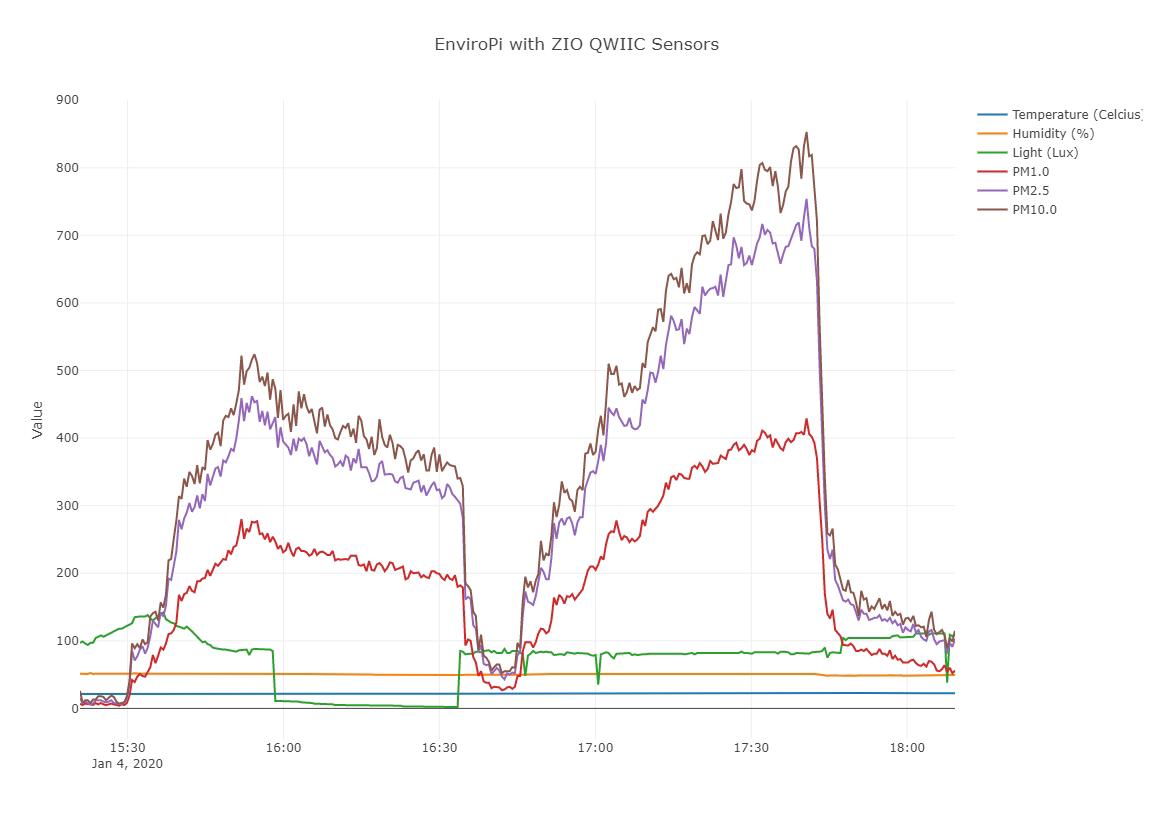
**Print(“Altitude = %0.2f meters” % alt\_data)**

**Aio.send(altitude\_feed.key, int(alt\_data))**

**# avoid timeout from adafruit io**

**Time.sleep(LOOP\_DELAY \* 60)**

**OUTPUT:**

****

****