

IoT- Environmental Monitoring

NAME : Ayswariya.S

ID:AUT962921104703

EMAIL:ayswariya.sathiyamoorthy@gmail.com

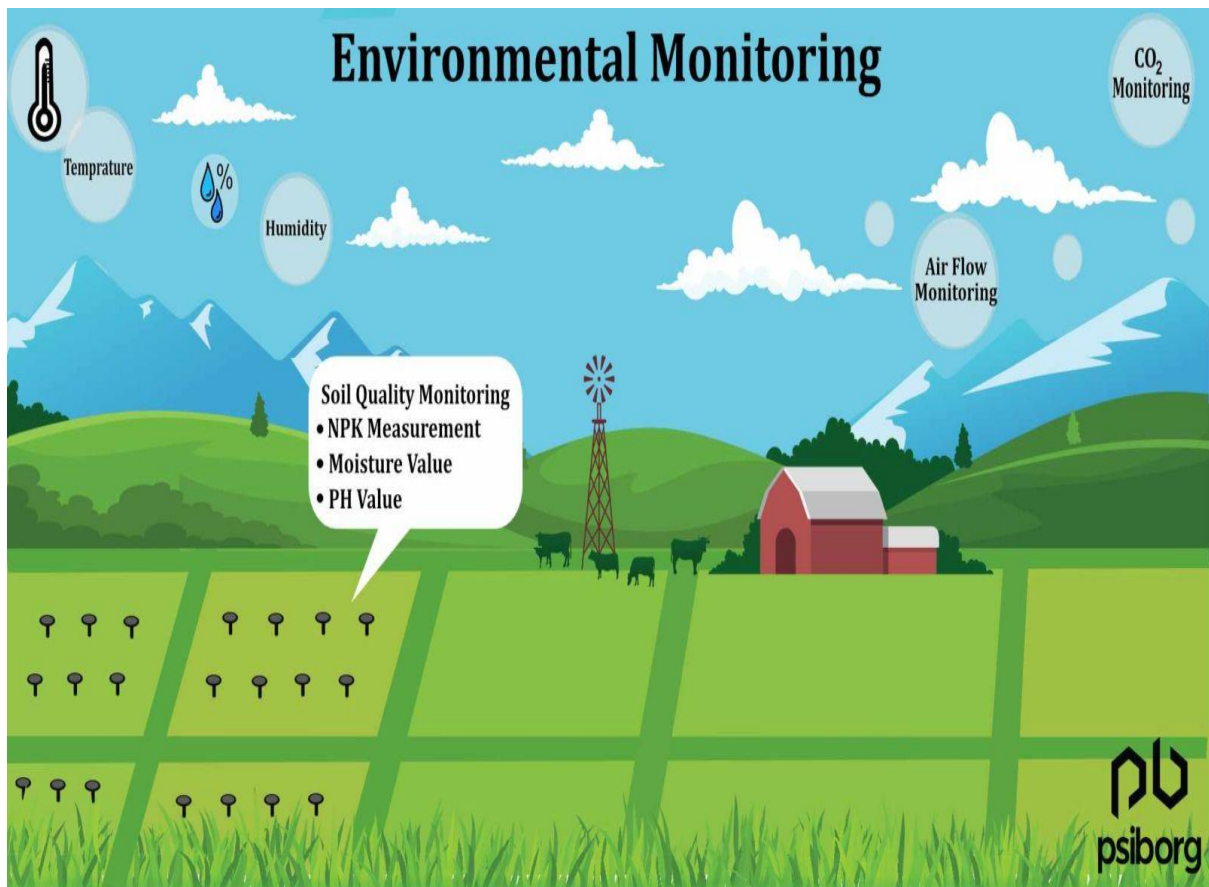


TABLE OF CONTENTS



INTRODUCTION.....	3
OVERALL DESIGN.....	3
SOFTWARE DESIGN.....	3
SYSTEM TESTING.....	4
PYTHON SCRIPT.....	4
CONCLUSION.....	11

DEVELOPMENT PART 1

Start building the IOT enabled environmental monitoring in park system:

1.Introduction:

With the continuous expansion of the total industrial volume, the integration of enterprises has gradually formed a relatively concentrated industrial layout. Industrial centralization has brought advantages in capital, technology and industrial chain, and also put forward higher requirements for environmental protection and safety. However, in recent years, major environmental pollution incidents have occurred frequently, and the life safety of construction workers has not been guaranteed. The root cause is that the supervisory department lacks effective real-time monitoring methods, so that it cannot grasp the pollution situation in time. Traditional human inspections and on-site assessments by professionals have high implementation costs and insufficient coverage of time and space.

2. Overall Design:

In order to meet the requirements of remote access, scalable capacity, and low power consumption, an NB-IoT-based networking system was designed, which is mainly composed of a monitor terminal, an operator network, and an independently built network server. The monitor terminal is mainly composed of sensors and STM32L151 control chip and BC95. The NB-IoT module transmits the signal to the operator's base station, which is forwarded by the operator to the server, and the data sent by the server is also transmitted to the monitor terminal through the operator. The NB-IoT networking structured in this way realizes multi-point monitoring data and can be transmitted to the server in time. After being obtained, the data is saved in the database and further analysed then displayed on the web page. The overall structure of the system.

3. Software design

The overall software flow mainly includes three steps: system initialization, measuring and processing data, and sending data. When the system is powered on, the system is initialized first, which includes the system working frequency initialization, delay function initialization, BC95 initialization, I2C initialization and so on. After the sensor measures the data, packing the acquired data through a custom communication protocol data format, and then sends the packed data to the server through the NB-IoT module. After the transmission is completed, it will enter the stop mode, and then give a 30-second transmission interval, so that the system is in the low-power mode as much as possible, and only when the module transmits data will it enter normal mode.

4. System testing

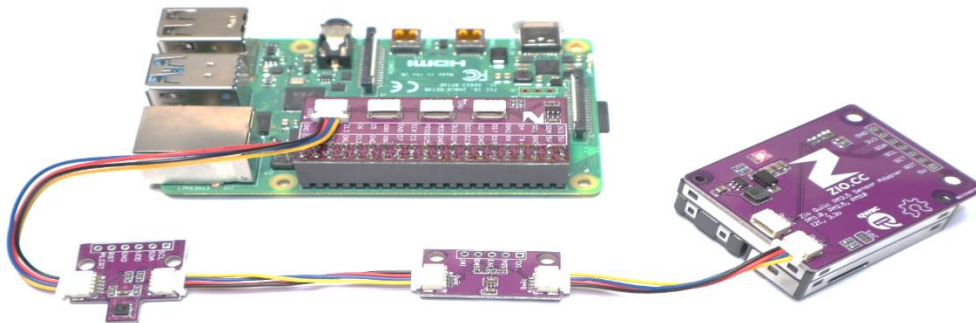
Since this chapter mainly introduces the hardware and software implementation of the system, the specific implementation of the server on the network will not be described in detail. The basic process is that when the server receives UDP data, the data is parsed according to the set format and stored in the MySQL database. The original data received by the server and the processed data in the database. the server receives the data sent by an IP. The beginning of the data is the flag bit and the version number. The following data are separated by '##', which are the values of device ID, smoke, carbon monoxide, carbon dioxide, TVOC, temperature and humidity. In addition, you can see that the corresponding data is stored in the database.

PYTHON SCRIPT

Step 1 Setup the Raspberry Pi

Step 2 Connect

Connect the ZIO SHT31 Temperature and Humidity sensor, the ZIO TSL2561 Light sensor and the ZIO Particulate Matter Sensor to the Raspberry Pi via the ZIO QWIIC HAT.



Step 3 enable the I2C bus

Either from the Raspberry Pi's desktop or via SSH, Open the terminal and type in:

```
sudo raspi-config and press enter.
```

```
raspberrypi - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help
Linux raspberrypi 4.19.75-v7l+ #1270 SMP Tue Sep 24 18:51:41 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 10 05:00:01 2019 from 192.168.1.246
pi@raspberrypi:~$ sudo raspi-config
```

Using your keyboards arrow keys, select '5 Interfacing Options' and press enter.

```
raspberrypi - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help
Raspberry Pi 4 Model B Rev 1.1

Raspberry Pi Software Configuration Tool

1 Change User Password Change password for the current user
2 Network Options Configure network settings
3 Boot Options Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options Configure connections to peripherals
6 Overclock Configure overclocking for your Pi
7 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool

<Select> <Finish>
```

Scroll down to 'P5 I2C' and press enter.

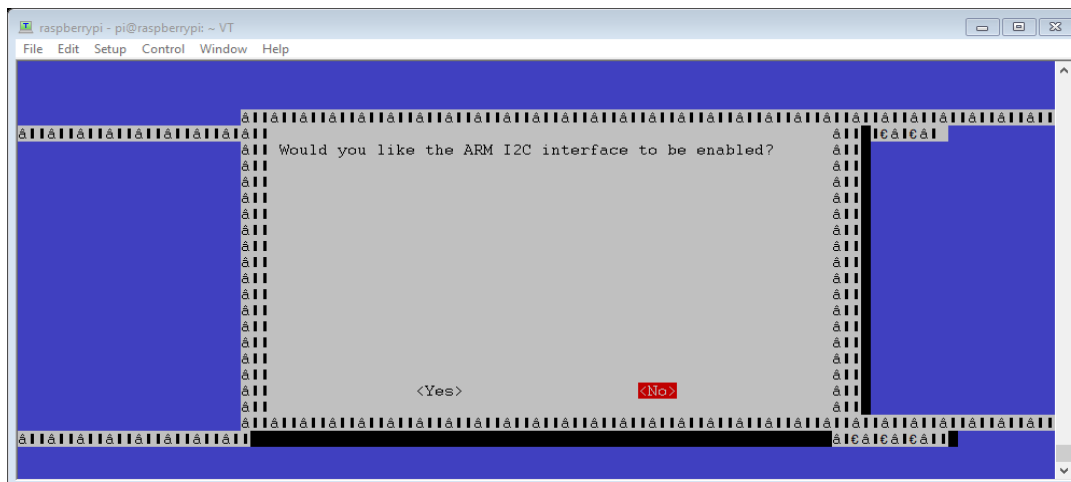
```
raspberrypi - pi@raspberrypi: ~ VT
File Edit Setup Control Window Help

Raspberry Pi Software Configuration Tool

P1 Camera Enable/Disable connection to the Raspberry Pi Camera
P2 SSH Enable/Disable remote command line access to your Pi using SSH
P3 VNC Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI Enable/Disable automatic loading of SPI kernel module
P5 I2C Enable/Disable automatic loading of I2C kernel module
P6 Serial Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

<Select> <Back>
```

Then select '<Yes>' and press enter to enable the I2C interface for the ZIO QWIIC ecosystem.



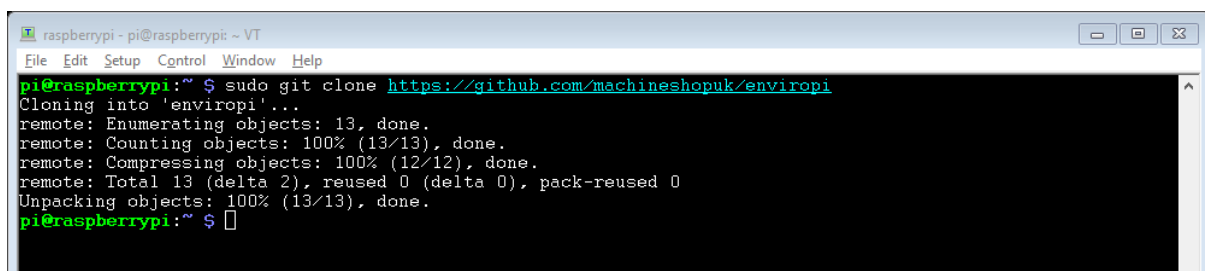
Select '<Ok>' and '<Finish>' to return to the terminal.

Step 4 Download python script

Whilst still in the terminal type:

```
sudo git clone
```

```
https://github.com/machineshopuk/enviropi
```



Step 4 Install Plotly

We need to install the Plotly library to utilise the functions to communicate with the plotly servers. Simply use this command to install it. This will also install any necessary support files.

```
sudo pip install plotly==3.10
```

```
raspberrypi - pi@raspberrypi: ~
File Edit Setup Control Window Help
pi@raspberrypi:~ $ sudo pip install plotly==3.10
```

```
raspberrypi - pi@raspberrypi: ~
File Edit Setup Control Window Help
5.0.>=2.4->nbformat>=4.2->plotly==3.10) (19.3.0)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in /usr/local/lib/python2.7/dist-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (1.3.0)
Requirement already satisfied: pyrsistent>=0.14.0 in /usr/local/lib/python2.7/dist-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (0.15.6)
Requirement already satisfied: funtools32; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (3.2.3.post2)
Requirement already satisfied: configparser>=3.5; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from importlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (4.0.2)
Requirement already satisfied: contextlib2; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from importlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (0.6.0.post1)
Requirement already satisfied: pathlib2; python_version < "3" in /usr/local/lib/python2.7/dist-packages (from importlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (2.3.5)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python2.7/dist-packages (from importlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (0.6.0)
Requirement already satisfied: scandir; python_version < "3.5" in /usr/local/lib/python2.7/dist-packages (from pathlib2; python_version < "3"->importlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (1.10.0)
Requirement already satisfied: more-itertools in /usr/local/lib/python2.7/dist-packages (from zipp>=0.5->importlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.2->plotly==3.10) (5.0.0)
Installing collected packages: plotly
Successfully installed plotly-3.10.0
pi@raspberrypi:~ $
```

Step 5 Generate Plotly API details

Create a Plotly account by clicking on Signup at <https://chart-studio.plot.ly> and entering your desired login details.

Once you have your account setup, hover over your username and click on settings.

Click on API Keys and click Regenerate Key, it will ask for your account password. This will create a long string of lower and uppercase letters and numbers. Copy this into a text file, we are going to need this in a minute.

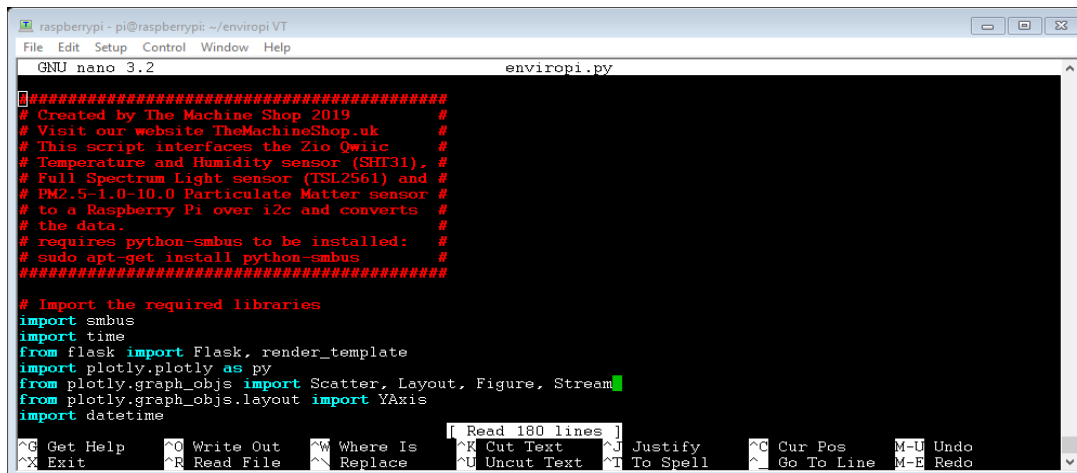
We are also going to need several streaming tokens, one for each data set. In this tutorial we have Temperature, Humidity, Light, PM1.0, PM2.5 and PM10.0, so we will need 6 streaming tokens. Click Add A New Token 6 times to generate all of the key and note them down with you API Key you generated above as we will need to enter them into our python script.

Step 6 Edit Python script

Before we run the script, we need to enter the Plotly API details we collected in Step 5.

At the command prompt, enter the enviropi folder and open the python script.

```
cd enviropi
sudo nano enviropi.py
```

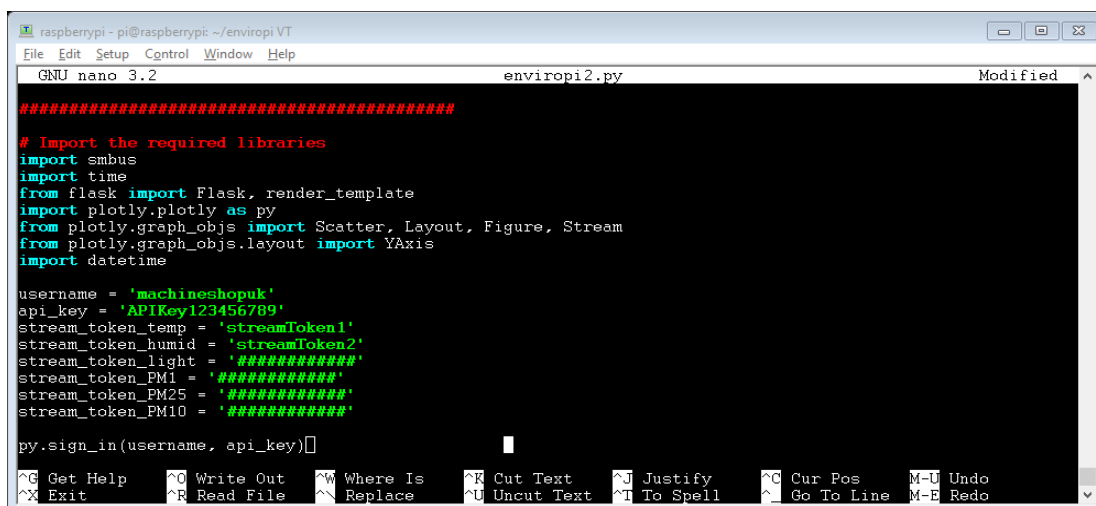


```
#####
# Created by The Machine Shop 2019
# Visit our website TheMachineShop.uk
# This script interfaces the Zio Qwiic
# Temperature and Humidity sensor (SHT31),
# Full Spectrum Light sensor (TSL2561) and
# PM2.5-1.0-10.0 Particulate Matter sensor
# to a Raspberry Pi over i2c and converts
# the data.
# requires python-smbus to be installed:
# sudo apt-get install python-smbus
#####

# Import the required libraries
import smbus
import time
from flask import Flask, render_template
import plotly.plotly as py
from plotly.graph_objs import Scatter, Layout, Figure, Stream
from plotly.graph_objs.layout import YAxis
import datetime

^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos    M-U Undo
^X Exit          ^R Read File    ^M Replace     ^U Uncut Text  ^T To Spell   ^_ Go To Line  M-E Redo
```

Scroll down to the API settings section of the code and enter your API username and keys from step 5 in between the 'quotes' and replacing the ### symbols.



```
#####

# Import the required libraries
import smbus
import time
from flask import Flask, render_template
import plotly.plotly as py
from plotly.graph_objs import Scatter, Layout, Figure, Stream
from plotly.graph_objs.layout import YAxis
import datetime

username = 'machineshopuk'
api_key = 'APIKey123456789'
stream_token_temp = 'streamToken1'
stream_token_humid = 'streamToken2'
stream_token_light = '#####'
stream_token_PM1 = '#####'
stream_token_PM25 = '#####'
stream_token_PM10 = '#####'

py.sign_in(username, api_key)

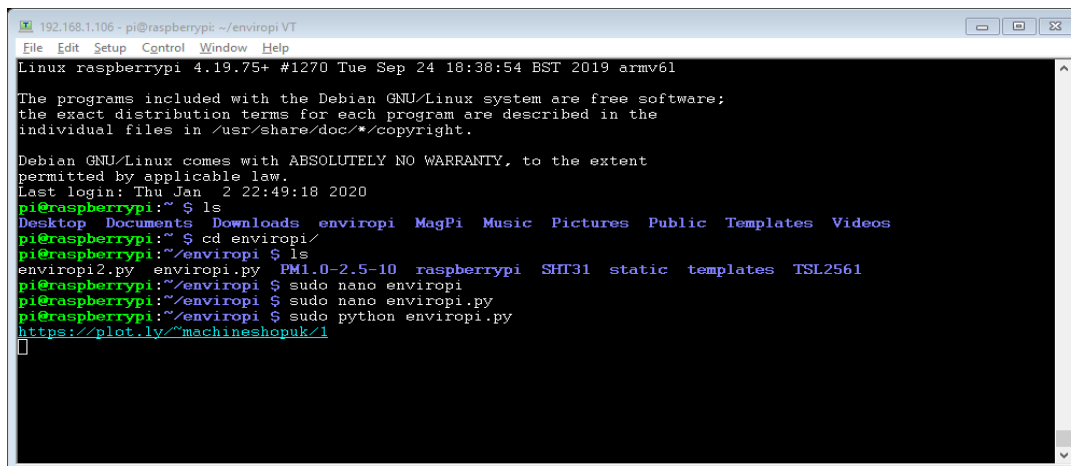
^G Get Help      ^O Write Out    ^W Where Is    ^K Cut Text    ^J Justify    ^C Cur Pos    M-U Undo
^X Exit          ^R Read File    ^M Replace     ^U Uncut Text  ^T To Spell   ^_ Go To Line  M-E Redo
```

Now save the script and exit from nano by pressing CTRL + X and when prompted 'Save modified buffer?' press Y

Step 7 Run the Script

Then run the script by typing:

```
sudo python enviropi.py
```



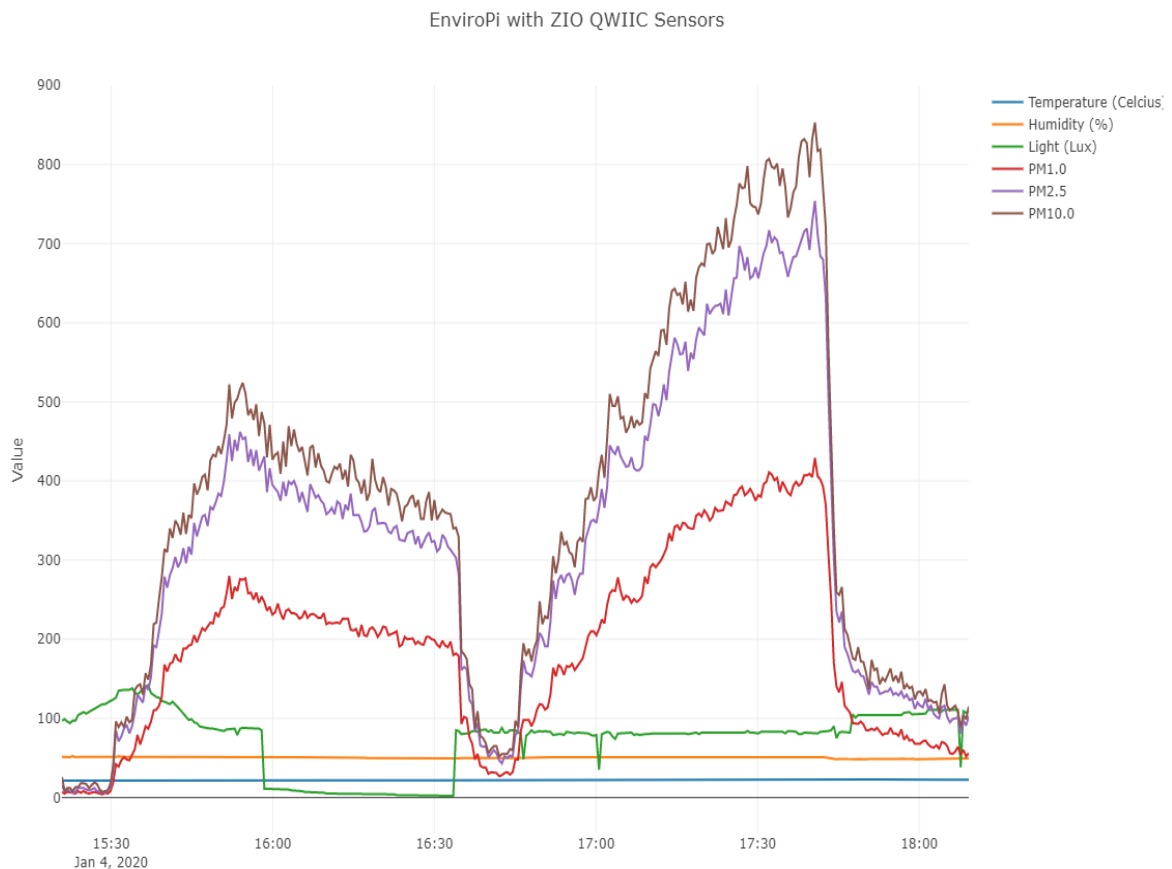
```
192.168.1.106 - pi@raspberrypi: ~/enviropi VT
File Edit Setup Control Window Help
Linux raspberrypi 4.19.75+ #1270 Tue Sep 24 18:38:54 BST 2019 armv6l
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Jan  2 22:49:18 2020
pi@raspberrypi:~$ ls
Desktop Documents Downloads enviropi MagPi Music Pictures Public Templates Videos
pi@raspberrypi:~$ cd enviropi/
pi@raspberrypi:~/enviropi$ ls
enviropi2.py enviropi.py PM1.0-2.5-10 raspberrypi SHT31 static templates TSL2561
pi@raspberrypi:~/enviropi$ sudo nano enviropi
pi@raspberrypi:~/enviropi$ sudo nano enviropi.py
pi@raspberrypi:~/enviropi$ sudo python enviropi.py
https://plot.ly/~machineshopuk/1
```

If you have entered all of your details correctly, then it should output a link to view the data on the Plotly website. It will start off with a graph with just one data point then after 30 seconds, another data point will be submitted and it will draw a line between the two points. As the graph fills up, it will automatically scale to show all of the data. This graph is configured to show a maximum of 10,000 data points with an interval of 30 seconds giving a total duration of 3.47 days' worth of data.

The graph below is an example of the output it will achieve. We had it setup in the office monitoring the environment. The blue and orange flat lines are the temperature and humidity which no surprise, didn't change much. The green line is the light intensity. This shows that the light was switched off at about 15:55 and switched back on again at 16:36. The red, purple and brown lines are from the ZIO Particulate Matter sensor. We had the soldering irons switched on whilst we were monitoring the environment and we can see that particulates in the air dramatically increased at 15:30 up until 15:55 when the soldering stopped and the person left the office, switching off the light as they left.

9

The particulates in the air slowly decreased so I'm guessing the office door was shut and at 16:36 when they came back they left the door open which quickly brought in fresh air. It looks like they started soldering again at 16:45 until 17:45 and then they opened the door but left the light on.



This data is extremally valuable and has given us an insight into the environment we work in. We can see that the temperature and humidity is nice and stable and that perhaps the light levels need to be slightly increased to ensure people can see what they are doing. The alarming data is from the particulate matter sensor and obviously shows that our extraction system needs to be improved. We

could even use the data from the sensor to automatically

10

activate the extraction system if the particulates in the air get too high. This would ensure that everyone in the office is working in clean air just in case someone who is soldering has forgotten to turn the extractor on. I know I've done that in the past far too many times!

5. Conclusion

Real-time and comprehensive environmental monitoring of industrial parks is indispensable for pollution prevention and safety monitoring, and traditional monitoring systems often have problems of high prices, poor deployment flexibility, and incomplete coverage. In view of the above shortcomings, this paper proposes a set of NB-IoT-based industrial park environment monitoring system. The system mainly includes the hardware side of the acquisition node and the software side of the cloud server. The test results show that the basic monitoring function of the monitoring system is complete and reliable, and it also has the advantages of low power consumption, wide coverage, and convenient interaction. The use of the system to achieve data visualization is more reliable and scientific.

