London-Stockholm Cities Partnerships Programme Measuring Air Quality Using Enviro+ and Raspberry Pi

Zian Huang[1], Zarin Haque[2]
[1]Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK
[2]Department of Electronic & Electrical Engineering, University College London, London WC1E 7JE, UK

Air quality data is imperative for environmental research. This project aimed to test the feasibility of Raspberry Pi module that detects particulate matter concentration, humidity and temperature of the surroundings. The three main components include a Raspberry Pi 3B, Enviro+ and a Particulate Matter Sensor. The module had been tested indoors long term to check its performance of data logging and uploading it to an online developers' platform called Luftdaten. The module was effective, and establishing a future feasibility of a practical solution that can be placed within the city.

***Index Terms*—air quality, particulate matter, Raspberry Pi, IoT.**

I. Introduction

The Cities Partnerships Programme was launched in 2018. It funds and promotes the work that UCL academics carry out with partners in global cities, with the aim of developing sustainable, multidisciplinary research and teaching partnerships [1]. Supported by this scheme, a student run electrical engineering project was introduced to foster academic and cultural exchange between undergraduate students from UCL and Stockholm University.

The aim of the project was to deploy an air quality sensor that uploads real-time measurements to an open data platform named Luftdaten. Due to limited time the London team had on the project being only 1 month, the specific objective was set to be creating a laboratory prototype with compact footprint that can fully function under a controlled physical and network environment.

Consider the electronics required, the aim was achieved by using a Raspberry Pi Zero computer for its compact hardware size. Raspberry Pi Zero provides computing resources and enables wireless network connection for processing data and uploading it to the cloud [2]. To obtain environmental measurement data, a sensor module called Enviro+ was connect to the Raspberry Pi to measure temperature, pressure and humidity [3]. Particulate matter pollution level was another important parameter in understanding local air quality. An extra PM sensing device was connected to the Enviro+ board to measure this quantity [4].

On the software level, the recommended Linux operating system Raspberry Pi OS was chosen for its stability and compatibility [5]. There were supported Python scripts provided by Pimoroni, the manufacturer of Enviro+, to support the Raspberry Pi environment [6]. The data platform chosen, Luftdaten, which was later renamed to *Sensor.Community*, is an open environmental data network based in Stuttgart, Germany [7]. An open data community was preferred to maximise the transferability and accessibility of the data collected, fulfilling the philosophy of the scheme.

II. Methodology

The early stage of setting up operating system and driver software was performed on a Raspberry Pi 3 to provide a smoother and more responsive developing experience. This was possible due to the focus on compatibility the manufacturer had with its different Raspberry Pi models [8]. There was high confidence that the same outcome would be obtained when identical scripts and commands are run on the 2 devices with the same Raspberry Pi OS installed. When the OS image was successfully installed with scripts tested on the Raspberry Pi 3, the measured data was fed to Luftdaten and visual representation of data set was automatically generated.

The above testing process was then repeated on the targeted Raspberry Pi Zero. A 2×20 pin header was required to be first soldered on the Raspberry Pi Zero board to enable module connection. The specific objective of this project ended at this stage as a laboratory prototype has been completed. Further extension to the aim was suggested that a hard case for the sensing station could be made using 3D printing. This will provide water resistance to the electronics and thus the sensor could be bought to an open remote area to obtain a less biased measurement.

A. Equipment and Setup

The software used in the project is shown in Table I. The *.img.xz* OS image file downloaded from the official website was burnt to an SD card which was then inserted into the Raspberry Pi [8]. The image burning could be achieved by using tools such as *dd* or *balenaEtcher*.

The electronics equipment used is presented in Table II. A labelled diagram is shown in Figure 8 showing the setup of the developing environment. A comparison highlighting the compactness of Raspberry Pi Zero can also been seen from the figure. The Enviro+ sensor was connected directly to the Raspberry Pi 3 using the pre-soldered header, and the PM sensor was attached to the Enviro+ via a 8-pin connector. The set up was compact and could be replicated on a Raspberry Pi Zero.

At this stage, the setup was complete and further operations were be moved to the Linux user space.

TABLE I: Software list

Name	Details/Version
Raspberry Pi OS (32-bit)	Linux Kernel 5.15, Debian 11
Enviro+ software	github.com/pimoroni/enviroplus-python
Luftdaten required package	smbus2 0.4.2

TABLE II: Equipment list

Name	Details/Descriptions
environmental data module	Pimoroni Enviro+
particulate matter sensor	Plantower PMS5003
deployment computing unit	Raspberry Pi Zero W
development computing unit	Raspberry Pi 3
storage for OS image	micro SD card
USB Power supply	5A with min. 1A
external display	monitor and HDMI cable
PM sensor connection	8-female-pin cable
soldering kit	heat iron and zinc solder
pin header	2×20 in dimension

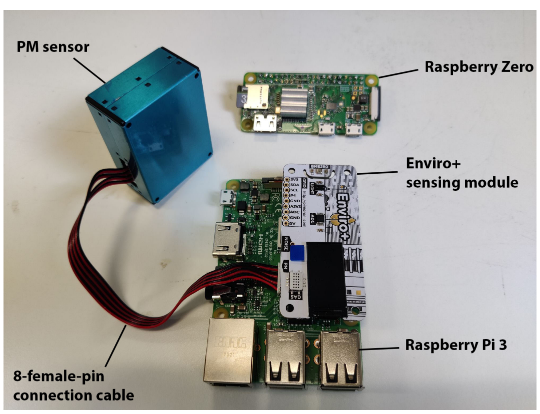

Fig. 1: labelled diagram of development and testing environment setup with a comparison in size between Raspberry Pi Zero and Raspberry Pi 3

B. Procedure

The Pi was first booted up and connected to external display with keyboard and mouse attached. Alternatively, this step could be replaced using a head-less connection through SSH connection to an external computer. Terminal could be accessed through Putty and VNC viewer providing a GUI. With the user being able to access the machine with giving input and receiving output, the Pi was connected to Wi-Fi for installing scripts and software. Remote control to an individual Pi was possible at the deployment and operational stage with SSH enabled.

There is a rigorous community of developers involved in the air quality monitoring, and so, there are pre-existing tutorials that have acted as guides on how to build the station [9]. The code to collect and upload data is also included by Pimoroni, the manufacturer of the Enviro+ sensor [10]. The scripts below were run to install the Enviro+ software and the Luftdaten script. A reboot was required to apply the changes.

```
git clone \
https://github.com/pimoroni/enviroplus-python
cd enviroplus-python
sudo ./install.sh
sudo pip install smbus2
sudo reboot
```

The Python code to capture and upload data was available in pre-existing library created by Pimoroni. Once the required libraries had been installed, the modules were connected and the program was run by executing the command

```
python examples/luftdaten.py
```

To upload the data to the platform, an account was registered on Luftdaten with the sensor assigned to the account [11]. Each developer account could register multiple sensors, but every sensor on the database was unique through the board ID. This was recorded as the unique serial number of the main processor board. Registration required the precise location of the sensor, including the height above the ground as well as the proximity to a road or garden. Once the sensor had been registered, the *luftdaten.py* program was run on the module so that data was uploaded to the platform. The sensor was placed indoors, next to a garden window away from the road in a flat in Waterloo. The sensor placement is shown in Figure 2. Two variations were used for this project - one with the window open, and one with the window closed. Then, for a detailed analysis, the system was run for an entire day (24 hours) with the window open.

Data from the sensors would be uploaded onto the platform approximately every three minutes, and stored as a *.csv* file on the Sensor Community archive with a unique sensor identifier. This was then downloaded and analysed.

III. Results

Refer to the data acquired, the granularity of the data is every three minutes giving a precise data logging ecosystem. With four features being captured (temperature, humidity, 2 types of particulate matter) it depicts a clear image of the air quality. On the Luftdaten platform, there was a fine dust map which located sensors on the map.

IV. Discussion

A. Comments on Results

The quality of the data is precise and consistent, producing valid graphs that accurately represent the temperature, humidity and particulate matter measurement. By operating the module for 24 hours, the inferences could be drawn clearly that the operation can continue long term. However, for correct readings, the module has to be placed away from direct sunlight - this is depicted in the spike from 12.00-13.00 in Fig

Fig. 2: sensor placement with window open and the surrounding urban residential environment

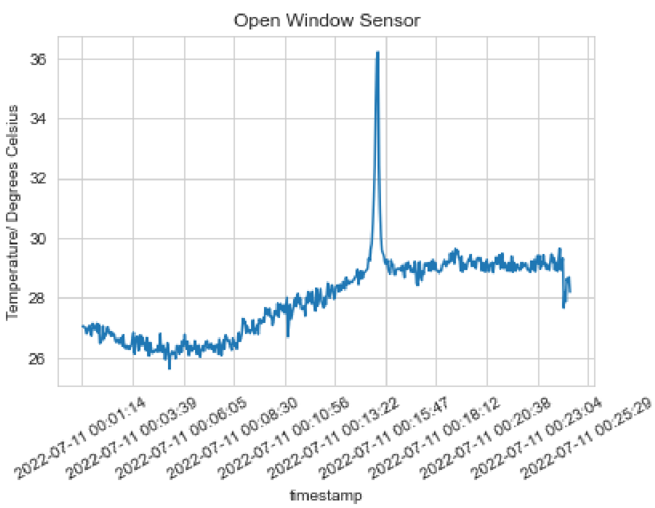

Fig. 3: temperature change with window open

3, where placement in direct sunlight caused overshoot of the temperature reading. Subsequently, the blinds were drawn to prevent further misreadings. Overall, the experiment enables us to design a hardware shell for the module that will be used when it is independently deployed in the city.

B. Conclusions

The final module was compact, and was cost-effective in terms of energy and financial footprints, especially when compared to the quality of readings that it captured. The developer platform was accessible and precise, increasing integrity of the data. The aim of the project was to test a deployable prototype that could be used throughout London to monitor air quality. At this stage, it is surely plausible that units can be placed throughout the city to increase data availability for further research.

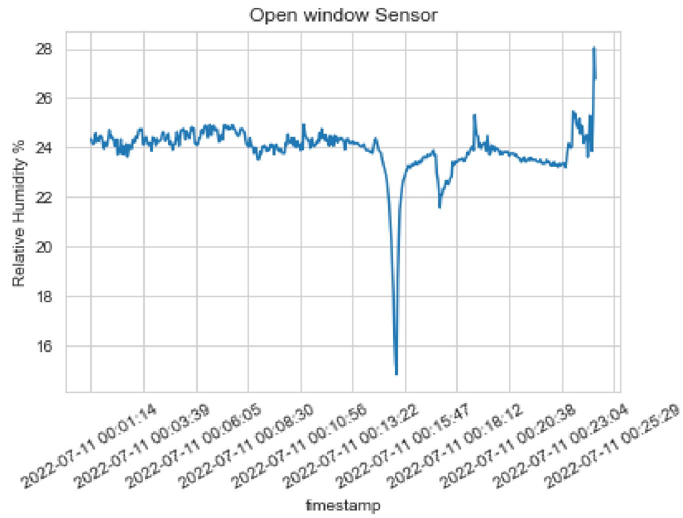

Fig. 4: relative humidity change with window open

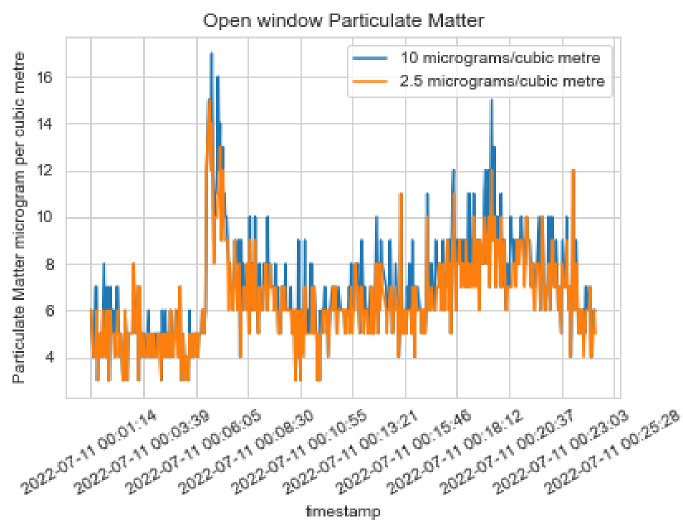

Fig. 5: particulate matter concentration change with window open

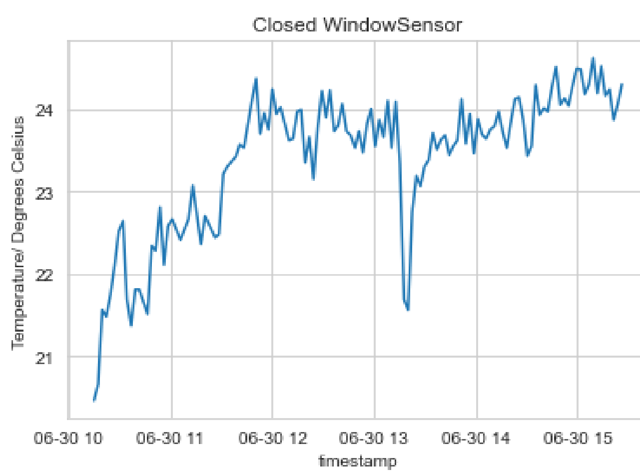

Fig. 6: temperature change in the afternoon 30 June 2022 with window closed

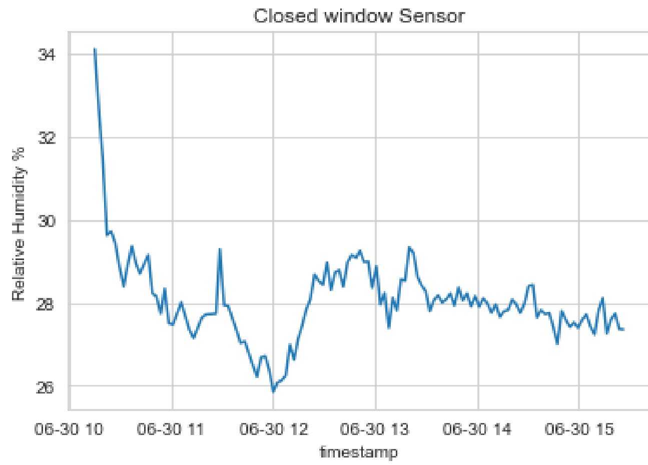

Fig. 7: relative humidity change in the afternoon 30 June 2022 with window closed

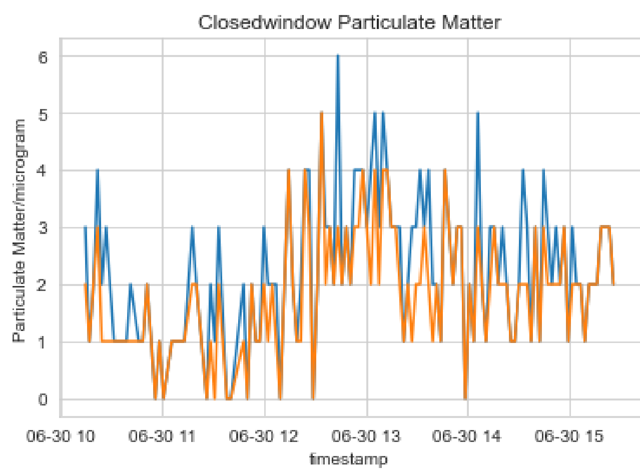

Fig. 8: particulate matter concentration change in the afternoon 30 June 2022 with window closed

Acknowledgment

The authors would like to thank Professor Paul Bartlett for giving us the opportunity to work on this project. We would also like to thank the team in the laboratories of the Physics and Astronomy Department and the Electronic and Electrical Engineering Department for their support and guidance throughout the project.

References

[1] UCL Global, "Cities Partnerships Programme, UCL Global, UCL University College London," ucl.ac.uk. https://www.ucl.ac.uk/global/cities-partnerships-programme (accessed: June. 15, 2021).

[2] The Raspberry Pi Foundation, "Buy a Raspberry Pi Zero W – Raspberry Pi," raspberrypi.com. https://www.raspberrypi.com/products/raspberry-pi-zero-w/ (accessed: June. 15, 2021).

[3] PIMORONI, "Enviro for Raspberry Pi, Monitor your world!, Pimoroni," pimoroni.com. https://shop.pimoroni.com/products/enviro (accessed: June. 15, 2021).

[4] PIMORONI, "PMS5003 Particulate Matter Sensor with Cable, Pimoroni," pimoroni.com. https://shop.pimoroni.com/products/pms5003-particulate-matter-sensor-with-cable (accessed: June. 15, 2021).

[5] The Raspberry Pi Foundation, "Raspberry Pi Documentation, Raspberry Pi OS," raspberrypi.com. https://www.raspberrypi.com/documentation/computers/os.html (accessed: June. 15, 2021).

[6] GitHub, "GitHub, pimoroni/enviroplus-python: Python library for the Enviro+ environmental monitoring board," github.com. https://github.com/pimoroni/enviroplus-python (accessed: June. 15, 2021).

[7] Sensor.Community, "devices.sensor.community, user configuration," sensor.community. https://devices.sensor.community/ (accessed: June. 15, 2021).

[8] The Raspberry Pi Foundation, "Operating system images, Raspberry Pi," raspberrypi.com. https://www.raspberrypi.com/software/operating-systems/ (accessed: June. 15, 2021).

[9] PIMORONI, "An outdoor air quality station with Enviro+ and Luftdaten," pimoroni.com. https://learn.pimoroni.com/article/enviro-plus-and-luftdaten-air-quality-station#installing-the-enviro-software (accessed: June. 15, 2021).

[10] GitHub, "enviroplus-python/luftdaten.py at master pimoroni/enviroplus-python GitHub," github.com. https://github.com/pimoroni/enviroplus-python/blob/master/examples/luftdaten.py (accessed: June. 15, 2021).

[11] Sensor.Community, "devices.sensor.community, user configuration," sensor.community. https://devices.sensor.community/register (accessed: June. 15, 2021).