Air Quality Index (AQI) Monitor Design

[AQI Standards 2](#_Toc52615205)

[Table of AQI Ranges and Color Codes 2](#_Toc52615206)

[Table of AQI Ranges and µg/m3 Measurements 2](#_Toc52615207)

[Measurement Normalization to EPA Nephelometer readings 3](#_Toc52615208)

[Time Averaging Measurements 4](#_Toc52615209)

[AQI Monitor Design 6](#_Toc52615210)

[Sensors 6](#_Toc52615211)

[Plantower Air Quality Sensor 6](#_Toc52615212)

[Clue Relative Humidity Sensor 6](#_Toc52615213)

[Clue Temperature Sensor 6](#_Toc52615214)

[Clue Elevation Sensor 6](#_Toc52615215)

[Data Logging and Sharing 6](#_Toc52615216)

[Clue On-Board Data Logging 6](#_Toc52615217)

[Data Aggregation and History 6](#_Toc52615218)

[Bluetooth Link Data Upload 6](#_Toc52615219)

[Bluetooth Link GPS Location and Time Data Download 6](#_Toc52615220)

[Display 6](#_Toc52615221)

[Current AQI 6](#_Toc52615222)

[AQI Trend 6](#_Toc52615223)

[Detailed Current Raw Data 6](#_Toc52615224)

[Raw Data Trend 6](#_Toc52615225)

[Status 6](#_Toc52615226)

[User Interaction 6](#_Toc52615227)

[Cycle Display 6](#_Toc52615228)

[Choose Display 6](#_Toc52615229)

[Reset Data Aggregation 6](#_Toc52615230)

[Wake Up 6](#_Toc52615231)

[Power Management 6](#_Toc52615232)

[Battery Operation 6](#_Toc52615233)

[Line Power Operation 6](#_Toc52615234)

[Sensor Sleep 6](#_Toc52615235)

[Sensor Wake 6](#_Toc52615236)

[Display Sleep 6](#_Toc52615237)

[Display Wake 6](#_Toc52615238)

[CPU Sleep 6](#_Toc52615239)

[CPU Wake 6](#_Toc52615240)

[Physical 6](#_Toc52615241)

[AQI Sensor Placement 6](#_Toc52615242)

[User Interface Access 6](#_Toc52615243)

[Weather Protection 6](#_Toc52615244)

[Mounting Options 6](#_Toc52615245)

[Physical Design Drawings 6](#_Toc52615246)

# AQI Standards

## Table of AQI Ranges and Color Codes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **AQI Range** | **AQI Category** | **AQI Color** | **Hex** | **R** | **G** | **B** |
| 0-50 | Good | Green | 00E400 | 0 | 228 | 0 |
| 51-100 | Moderate | Yellow | FFFF00 | 255 | 255 | 0 |
| 101-150 | Unhealthy for Sensitive Groups | Orange | FF7E00 | 255 | 126 | 0 |
| 151-200 | Unhealthy | Red | FF0000 | 255 | 0 | 0 |
| 201-300 | Very Unhealthy | Purple | 8F3F97 | 143 | 63 | 151 |
| 301-500 | Hazardous | Maroon | 7E0023 | 126 | 0 | 35 |

## Table of AQI Ranges and µg/m3 Measurements

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **AQI Range** | **AQI Category** | **PM2.5 24hr µg/m3** | **AQI Factor** | **PM10.0 24 hr µg/m3** | **AQI Factor** |
| 0-50 | Good | 0-12 | 4.17 | 0-54 | 0.93 |
| 51-100 | Moderate | 12.1-35.4 | 2.10 | 55-154 | 0.49 |
| 101-150 | Unhealthy for Sensitive Groups | 35.5-55.4 | 2.46 | 155-254 | 0.49 |
| 151-200 | Unhealthy | 55.5-150.4 | 1.94 | 255-354 | 0.49 |
| 201-300 | Very Unhealthy | 150.5-250.4 | 1.01 | 355-424 | 1.43 |
| 301-500 | Hazardous | 250.5-500.4 | 1.26 | 425-604 | 1.11 |

The AQI ratings in the above table apply only to PM2.5 and PM10.0 measurements. There are no equivalent AQI ratings for smaller particle masses, but the Plantower device measures counts of particles at 0.3 µm, 0.5 µm, 1.0 µm, 2.5 µm, 5.0 µm and 10.0 µm and masses of particles per cubic meter at 1.0 µm, 2.5 µm and 10.0 µm. The latter mass measurements come in “standard” or laboratory adjusted measurements and “environment” or atmospheric measurements. These make different assumptions about how to convert the raw data and experimenters have determined that for typical use the environmental measures should be used. All of the raw data is available if you are creating your own measures that use different assumptions.

The sensors on the PM2.5 and the Clue do not give us the other AQI pollutant measures for ozone, CO, SO2 or NO2. For a complete AQI monitoring system, additional sensors, available from Sparkfun, can be integrated into a more complex device.

The guidance from AQI Technical Assistance Document published by the EPA (Sept 2018) provides information on how to calculate AQI across all of the pollutant measures, if they are available. The AQI is the highest value calculated for each pollutant by identifying the highest concentration among all of the monitors and truncate as follows:

* Ozone (ppm) – truncate to 3 decimal places
* PM2.5 (µg/m3) – truncate to 1 decimal place
* PM10 (µg/m3) – truncate to integer
* CO (ppm) – truncate to 1 decimal place
* SO2 (ppb) – truncate to integer
* NO2 (ppb) – truncate to integer

Using the breakpoints table, find the upper and lower values that contain the measurement and then use this equation (linear extrapolation within each segment):

Where

* Ip is the index for pollutant p
* Cp is the truncated pollutant measurement
* BPHi is the upper breakpoint greater than the measurement
* BPLo is the lower breakpoint less than the measurement
* IHi is the AQI upper breakpoint for that same row in the table
* ILo is the AQI lower breakpoint for that same row in the table

## Measurement Normalization to EPA Nephelometer readings

The EPA uses sensors deployed by local governments called nephelometers that capture a sample of air, place it in a controlled environment of temperature and humidity, and then user a laser sensing device similar to that of the Plantower to estimate the mass per cubic meter for different particle sizes. The introduction of citizen science sensors from Purple Air, which uses the Plantower sensor and which directly measures particle counts without the environment controls, typically reads higher than the EPA nephelometer reported AQI. Purple Air has incorporated the results of one study into their software, from LRPA, that uses the results of that study to adjust the readings they report so they begin to match EPA reports better. A second study by a group in Salt Lake City AQ&U resulted in a different linear adjustment. This chart compares the unadjusted, LRPA and AQ&U.

These adjustments do not take into account the differences that are created by atmospheric temperature and relative humidity when measuring directly as compared to the controlled conditions in the nephelometers.

This [paper](http://lar.wsu.edu/nw-airquest/docs/20200610_meeting/NWAQ_20200611_1030_Hadley.pdf) gives the adjustment equations for the two studies:

* LRAPA: (PM2.5 = 0.5PA-0.66)
* AQ&U (PM2.5 = 0.77PA +2.6)

This paper also reports that there are preliminary results that indicate better fits to EPA AQI nephelometer reporting when using sigmoidal fits from the Plantower data, but there are no studies available to adjust for relative humidity (most important) or temperature.

## Time Averaging Measurements

The standard for reporting on AQI Category from particulate measurements is a 24-hour average. That’s appropriate for general weather reporting to give a sense of the overall pollutant load in the air, but for a personal sensor shorter time averaging seems appropriate. The guidance from the EPA for shorter times is to use 12 hours for medium term averages and 3 hours for “real-time” reporting. A personal AQI monitor might also provide 1 hour and 5-minute averaging options.

In all cases, the time averaging should be available for display at any time so the monitor should be accumulating the data so that it can report the full time averaged statistic as well as providing historical trends for the day, week or month, if the sensor is kept running continuously.

PurpleAir uses a ring display to show multiple time averages. Each ring represents an average for a time range:

Center of the circle = Real time   
1st ring = Short-term average (5 minutes?)  
2nd ring = 30-minute average  
3rd ring = 1-hour average  
4th ring = 6-hour average  
5th ring = 24-hour average  
6th ring = 1-week average

Aggregation of data are recomputed as sensor measurements are acquired and used to update the list of readings for a particular aggregation period. Measurement frequency is settable in increments of 10 seconds for each sensor. The current period aggregate is also input to the aggregate history record with a set history period for aggregates.

Current aggregate structure

|  |  |  |
| --- | --- | --- |
| **Aggregate Period** | **Measurement Frequency** | **Count of Aggregate Members** |
| One minute | 10 seconds | 6 |
| 30 minutes | 1 minute aggregate | 30 |
| One hour | 1 minute aggregate | 60 |
| 8 hours | 1 hour aggregate | 8 |
| 12 hours | 1 hour aggregate | 12 |
| 24 hours | 1 hour aggregate | 24 |
| 3 days | 8 hour aggregate | 24 |
| 1 week | 24 hour aggregate | 7 |
| 2 weeks | 24 hour aggregate | 14 |

Aggregate History Structure

|  |  |  |
| --- | --- | --- |
| **Aggregate period** | **History period** | **Count of History Members** |
| 1 minute | 1 day | 1440 |
| 30 minutes | 3 days | 144 |
| 1 hour | 1 week | 168 |
| 8 hours | 4 weeks | 84 |
| 12 hours | 8 weeks | 112 |
| 24 hours | 13 weeks | 91 |
| 3 days | 1 year | 18 |
| 1 week | 3 years | 156 |

# AQI Monitor Design and Requirements

## Sensors

### Plantower Air Quality Sensor

### Clue Relative Humidity Sensor

### Clue Temperature Sensor

### Clue Elevation Sensor

## Data Logging and Sharing

### Clue On-Board Data Logging

### Data Aggregation and History

### Bluetooth Link Data Upload

### Bluetooth Link GPS Location and Time Data Download

## Display

### Current AQI

### AQI Trend

### Detailed Current Raw Data

### Raw Data Trend

### Status

## User Interaction

### Cycle Display

### Choose Display

### Reset Data Aggregation

### Wake Up

## Power Management

### Battery Operation

### Line Power Operation

### Sensor Sleep

### Sensor Wake

### Display Sleep

### Display Wake

### CPU Sleep

### CPU Wake

## Physical

### AQI Sensor Placement

### User Interface Access

### Weather Protection

### Mounting Options

### Physical Design Drawings

# AQI Monitor Software Design

## Main

Set up PM2.5 classes.

Set up BusDevice classes for each of the I2C devices:

* PM2.5
* Relative humidity
* Temperature
* Elevation (air pressure)

Set up display classes for Clue display.

Set up buttons for user interface.

Set up power management for line power or battery power.

Set up time keeping for tracking trend data and set starting timer for Sleep mode at 60s

Set up data logging.

Set up Bluetooth link for logging upload and time and location download.

Create opening splash display.

Set up starting data display.

Loop Forever Once per Minute

* Start up the PM2.5 and set timer for 30s
* Read Relative Humidity and log data and update aggregations
* Read Temperature and log data and update aggregations
* Read Elevation/air pressure and log data and update aggregations
* Wait for 30s timer to expire
* Read PM2.5 data set and log data and update aggregations
* Convert PM2.5 and PM10.0 data to AQI and log data and update aggregations
* Convert PM2.5 AQI to LRAPA and AQ&U data and log data and update aggregations
* Read location data over Bluetooth link
* Send new log data over Bluetooth link
* Update display with new data if Wake is TRUE and if Sleep timer is not expired

Interrupt the loop for any user interface key press

* Reset the Sleep mode timer to 60s
* Wake if Sleep is TRUE
* Cycle display type: Current Data, Trend Data, Status, Sleep Display
* Reset data aggregations and restart trend data time keeping

## CircuitPython BusDevice Classes for I2C Devices

The *I2CDevice* and *SPIDevice* helper classes make managing transaction state

on a bus easy. For example, they manage locking the bus to prevent other

concurrent access. For SPI devices, it manages the chip select and protocol

changes such as mode. For I2C, it manages the device address.

Example code: examples/busdevice\_read\_register\_i2c\_simpletest.py

import busio

import board

from adafruit\_bus\_device.i2c\_device import I2CDevice

DEVICE\_ADDRESS = 0x68 # device address of DS3231 board

A\_DEVICE\_REGISTER = 0x0E # device id register on the DS3231 board

# The follow is for I2C communications

comm\_port = busio.I2C(board.SCL, board.SDA)

device = I2CDevice(comm\_port, DEVICE\_ADDRESS)

with device as bus\_device:

bus\_device.write(bytes([A\_DEVICE\_REGISTER]))

result = bytearray(1)

bus\_device.readinto(result)

print("".join("{:02x}".format(x) for x in result))

## CircuitPython PM25 Classes

CircuitPython library for PM2.5 Air Quality Sensors.

Example code: examples/pm25\_simpletest.py

"""

Example sketch to connect to PM2.5 sensor with either I2C or UART.

"""

# pylint: disable=unused-import

import time

import board

import busio

from digitalio import DigitalInOut, Direction, Pull

import adafruit\_pm25

reset\_pin = None

# If you have a GPIO, its not a bad idea to connect it to the RESET pin

# reset\_pin = DigitalInOut(board.G0)

# reset\_pin.direction = Direction.OUTPUT

# reset\_pin.value = False

# Create library object, use 'slow' 100KHz frequency!

i2c = busio.I2C(board.SCL, board.SDA, frequency=100000)

# Connect to a PM2.5 sensor over I2C

pm25 = adafruit\_pm25.PM25\_I2C(i2c, reset\_pin)

print("Found PM2.5 sensor, reading data...")

while True:

time.sleep(1)

try:

aqdata = pm25.read()

# print(aqdata)

except RuntimeError:

print("Unable to read from sensor, retrying...")

continue

print()

print("Concentration Units (standard)")

print("---------------------------------------")

print(

"PM 1.0: %d\tPM2.5: %d\tPM10: %d"

% (aqdata["pm10 standard"], aqdata["pm25 standard"], aqdata["pm100 standard"])

)

print("Concentration Units (environmental)")

print("---------------------------------------")

print(

"PM 1.0: %d\tPM2.5: %d\tPM10: %d"

% (aqdata["pm10 env"], aqdata["pm25 env"], aqdata["pm100 env"])

)

print("---------------------------------------")

)

print("---------------------------------------")

print("Particles > 0.3um / 0.1L air:", aqdata["particles 03um"])

print("Particles > 0.5um / 0.1L air:", aqdata["particles 05um"])

print("Particles > 1.0um / 0.1L air:", aqdata["particles 10um"])

print("Particles > 2.5um / 0.1L air:", aqdata["particles 25um"])

print("Particles > 5.0um / 0.1L air:", aqdata["particles 50um"])

print("Particles > 10 um / 0.1L air:", aqdata["particles 100um"])

print("---------------------------------------")

## AQI Classes