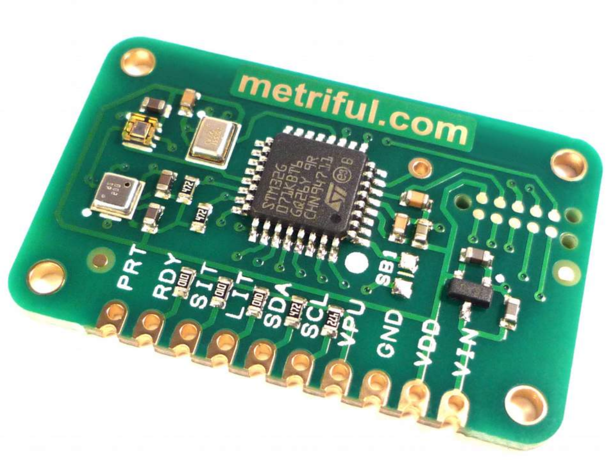
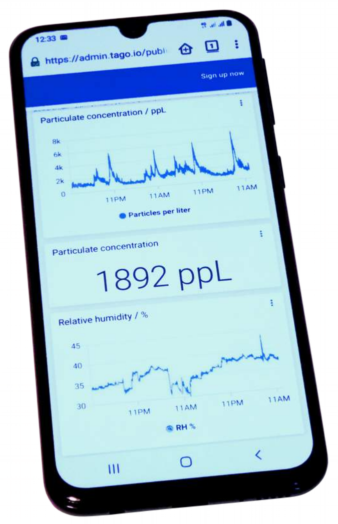
Metriful MS430

User guide

Indoor environment monitor with I2C compatible interface





DESCRIPTION

The Metriful MS430 is a low power, high accuracy, smart sensor cluster for indoor environment monitoring. It is operated via a simple I2C-compatible interface and measures eighteen variables including air quality, light and sound levels. The MS430 also offers an active interrupt feature to reduce burden on the host system.

In this document:

- How to set up working demo systems with Arduino, Raspberry Pi, ESP8266 and ESP32
- Summary of the software code examples
- Explanations of the environment data measured
- Application ideas

CONTENTS

1. FURTHER INFORMATION....3
2. HANDLING PRECAUTIONS....3
3. DEMONSTRATION SYSTEMS....3
4. HARDWARE SETUP....4
  Raspberry Pi....5
  Arduino Nano 33 IoT, MKR WiFi 1010, Uno, Nano, ESP8266, ESP32....5
5. PARTICLE SENSORS....6
6. SOFTWARE CODE EXAMPLES....7
  Simple read sound....8
  Simple read T H (Temperature and Humidity)....8
  Cycle read-out....8
  Interrupts....9
  On demand read-out....9
  Particle sensor toggle....9
  Home Assistant (not for Arduino Uno or Nano)....9
  IFTTT (not for Arduino Uno or Nano)....9
  IoT cloud logging (not for Arduino Uno or Nano)....9
  Web server and graph web server (not for Arduino Uno or Nano)....10
  Log data to file (Raspberry Pi only)....10
7. IOT CLOUD HOSTING SETUP....10
  Tago.io cloud....11
  Thingspeak.com cloud....12
8. PARTICLE SENSOR POWER CONTROL....13
9. INDOOR ENVIRONMENT DATA EXPLAINED....15
  SOUND....15
    Sound Pressure Level (A-weighted)....15
    Frequency band Sound Pressure Level....17
    Peak sound amplitude....17
    Ideas for further investigation of sound....18
  LIGHT....18
    Illuminance....18
    White light level....19
    Ideas for further investigation of light....19
  AIR....20
    Temperature....20
    Pressure....21
    Humidity....21
    Gas (volatile organic compound) sensor resistance....22
    Ideas for further investigation of air measurements....22
  AIR QUALITY....23
    Air Quality Index....23
    Equivalent breath VOC concentration....23
    Estimated carbon dioxide (CO_2) concentration....24
    Ideas for further investigation of air quality....24
  PARTICULATES....24
    Particle sensor duty cycle....25
    Estimated particle concentration....25
    Ideas for further investigation of particulates....25
10. DISCLAIMER....27

1. FURTHER INFORMATION

To achieve the fastest development time, use this guide together with the following resources which can be found at www.github.com/metriful/sensor

- Device datasheet – this describes the electrical characteristics and pin connections of the MS430 circuit board, and covers the low-level details of the communications interface.

- Demo code, examples and instructions to help you get started immediately.

2. HANDLING PRECAUTIONS

The MS430 can be damaged by static electricity discharges. Minimize this risk by observing the following precautions:

- Handle the board by the edges
- Avoid touching any metal part of the device or circuit it connects to
- Store in the provided antistatic bag when not connected in a circuit
- Keep away from metal objects which could cause shorted connections

3. DEMONSTRATION SYSTEMS

The MS430 must be used together with a host system, which transmits or displays the measured environment data. The minimum required features of the host are:

- I2C compatible two wire interface
- At least one digital input (three digital inputs if light and sound interrupts are used)
- System voltage of 3.3 V or 5 V

These requirements are met by a large number of microcontrollers, development boards and embedded computers. This guide contains specific wiring details, setup instructions and example software for using the MS430 with the following demonstration host systems:

With network access:

- Raspberry Pi
- Arduino Nano 33 IoT
- Arduino MKR WiFi 1010
- ESP8266
- ESP32

Without network access:

- Arduino Uno
- Arduino Nano

To run one of the demonstration systems, follow these steps:

1. Wire up the hardware according to Section 4
2. Download the software examples from www.github.com/metriful/sensor
3. Follow the GitHub Readme section for your host system for first-time software setup
4. Choose a software example – described in Section 6
5. Load the software to your host system and run it

The examples are a convenient starting point for developing new applications.

4. HARDWARE SETUP

Connect the MS430 to a host with wires only: no extra electronic components are needed. The user should initially wire all connections without a power source connected. Double-check the wiring is correct before applying power.

Table 1 describes the function of each terminal in the 10-pin header on the MS430 board. The following sections explain how to connect these to the different host systems.

Table 1 - MS430 connector pin description.

Connector label	Description
VIN	Power input if no 3.3 V supply is available. Leave unconnected if VDD is used.
VDD	Power input for 3.3 V supply. Leave unconnected if VIN is used.
GND	0 V ground
VPU	Pull-up voltage input. Apply the host system voltage.
SCL	Two-wire interface (I2C compatible) clock input
SDA	Two-wire interface (I2C compatible) data input/output
LIT	Light interrupt signal output (**optional**)
SIT	Sound interrupt signal output **(optional)**
RDY	READY signal output
PRT	PPD42 particle sensor signal input (**optional**)

Optional pins SIT, LIT and PRT may be left unconnected – this will disable the functions offered by these but still allows the other functions of the MS430 to operate. Further detail on electrical characteristics and pin functions is given in the device datasheet, available at www.github.com/metriful/sensor.

Raspberry Pi

The connections listed here are for the 40-pin header found on all current Raspberry Pi boards (from 2014 onward).

- Connect the MS430 to the Raspberry Pi according to Table 2.
- Leave MS430 pin VIN unconnected.
- If a PPD42 or SDS011 particle sensor is used:
  - Raspberry Pi pin 2 (5 V) is connected to PPD42 pin 3 or SDS011 pin “5V”
  - Raspberry Pi pin 9 (ground) is connected to PPD42 pin 1 or SDS011 pin “GND”
  - MS430 pin PRT is connected to PPD42 pin 4 or SDS011 pin “25um”

Table 2 - Raspberry Pi pin connections.

MS430 pin label	Raspberry Pi pin number	Raspberry Pi pin name
VIN	-	-
VDD	1	3V3 power
GND	6	Ground
VPU	17	3V3 power
SCL	5	GPIO 3 (SCL)
SDA	3	GPIO 2 (SDA)
LIT	7	GPIO 4
SIT	8	GPIO 14
RDY	11	GPIO 17

Arduino Nano 33 IoT, MKR WiFi 1010, Uno, Nano, ESP8266, ESP32

- Connect the MS430 to the host according to Table 3, leaving either VDD or VIN unconnected
- If a PPD42 or SDS011 particle sensor is used:
  - Host 5 V output is connected to PPD42 pin 3 or SDS011 pin “5V”
  - Host pin GND is connected to PPD42 pin 1 or SDS011 pin “GND”
  - MS430 pin PRT is connected to PPD42 pin 4 or SDS011 pin “25um”

- To obtain 5V output on the Nano 33 IoT: the solder bridge labeled “VUSB” on the underside of the Arduino must be soldered closed, then use the VUSB pin.
- 5V pins on some boards may instead be labeled Vin or VU
- To obtain a third 5V output on the Uno: use pin number 2 on the 6-pin ICSP header
- With all hosts, VPU can be supplied from any host digital output pin set to a high voltage state. This can be useful for hosts with limited power output pins.

Table 3 - Pin connections for six host systems

MS430 pin label	Arduino Nano 33 IoT pin	Arduino MKR WiFi 1010 pin	Arduino Uno pin	Arduino Nano pin	ESP8266 pin	ESP32 pin
VIN	-	-	5V	5V	-	-
VDD	3.3V	VCC	-	-	3V3	3V3
GND	GND	GND	GND	GND	GND	GND
VPU	3.3V	VCC	IOREF	5V	3V3	3V3
SCL	A5	SCL	SCL	A5	D2	D22
SDA	A4	SDA	SDA	A4	D1	D21
LIT	A1	D4	D4	D4	D3	D18
SIT	A2	D5	D7	D7	D5	D19
RDY	D11	D0	D2	D2	D6	D23

5. PARTICLE SENSORS

The MS430 provides an input connection (labeled PRT) and data analysis algorithm for an optional air particulate matter sensor (not included with the board). Two sensors are supported: the Shinyei PPD42 and the Nova SDS011, both of which can be purchased from many suppliers worldwide.

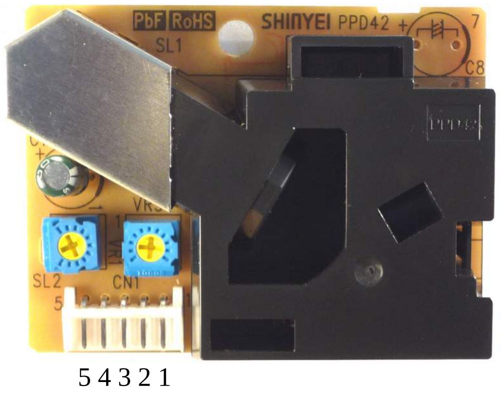


Figure 1 - PPD42 particle sensor with connector pin numbering shown

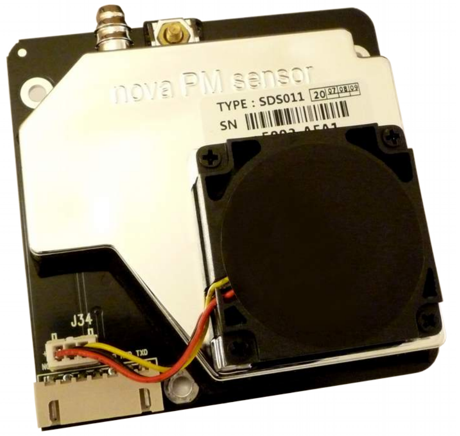


Figure 2 - SDS011 particle sensor

PPD42

The PPD42 uses a resistor heater and optical system to detect airborne particles of approximately 1µm diameter and larger.

Figure 1 shows the PPD42 and its 5-pin connector. Note that the connector is numbered 1 to 5 from **right to left**. Pin 1 (0 V ground) and pin 3 (5 V) provide power to the sensor, which must always be 5 V irrespective of the host system voltage. Pin 4 (signal output) connects to PRT on the MS430 board. PPD42 pins 2 and 5 are left unconnected.

For correct operation, mount the PPD42 vertically (connector pins pointing downwards) and cover the large central hole with opaque tape.

The PPD42 is sensitive to small power supply fluctuations and interference. For best results, power it with a dedicated, regulated 5 V supply. Using a 5 V output from the host system may cause increased signal noise, especially when the host is powered from a computer USB port.

SDS011

The SDS011, shown in Figure 2, uses a powered fan and laser system to detect particles of 0.3 – 10 µm in size. It has a labeled 7 pin connector; connect pins GND and 5V to the power supply and connect the MS430 PRT pin to either of the outputs labeled "25um" (for particles 0.3 – 10 µm) or "1um" (for particles 0.3 – 2.5 µm). The "25um" output is recommended due to its larger detection rate.

6. SOFTWARE CODE EXAMPLES

A set of host software code examples are provided in the GitHub repository at www.github.com/metriful/sensor. The repository readme file also lists the software and IDE setup steps needed for the demo host systems before first use.

The code examples are simple procedural programs which show various ways of using the MS430. This includes basic control and read-out, as well as saving data to files and to IoT cloud storage. Detailed code comments explain each part of the programs, with further details of operation given in the datasheet.

Some of the examples run on all featured host systems, while others are specific to only some of the hosts. All examples assume the standard hardware setups described in Section 4.

Arduino examples

All examples in the Arduino repository folder are designed for use on the Arduino Nano 33 IoT, MKR WiFi 1010, ESP8266 and ESP32. Examples not requiring a network connection also run on Arduino Uno and Nano. The Arduino IDE is recommended for programming all of these hosts.

Arduino Library

The **Metriful_Sensor** repository folder provides a library of reusable functions for common tasks, such as sending commands and displaying data. Some functions are provided in two versions: with or without using the float data type (floating point number). Floats are simple to use in programming but require extra memory and processing resources when run on the Arduino. This means that they should be avoided in some demanding situations.

The file **sensor_constants.h** lists numerical values taken from the datasheet (command codes, etc.) for convenient use in programs.

Data extraction on Arduino

The measured environment data are read from the MS430 as a plain sequence of bytes. This sequence must be split up and rearranged to recover the meaningful values. The simplest way of doing this is with a type cast of a whole category of data (sound, light, air quality, etc.) at once. This is demonstrated many times in the examples.

Raspberry Pi examples

The Raspberry Pi examples use the Python 3 language which comes already installed on the Raspbian operating system. Run them from the command prompt in a terminal window.

As the Raspberry Pi is a more powerful computer than the Arduino, a higher-level approach can be taken. The file **sensor_functions.py** contains convenient functions for common tasks, such as sending commands and displaying data. Environment data values are extracted individually from the received data bytes and floating point numbers are used. File **sensor_constants.py** lists numerical values taken from the datasheet (command codes, etc.) for convenient use in programs.

Code example summaries

Unless stated otherwise, all of the examples are provided for all demo host systems.

Simple read sound

Measures and displays only the sound data, demonstrating different reading methods.

Simple read T H (Temperature and Humidity)

Measures and displays the humidity and temperature, demonstrating the decoding of the signed temperature value. The data are also read out and displayed a second time, using a simple “data category read” of all Air data.

Cycle read-out

Continually measures and displays all environment data in a repeating cycle. The user can choose from a cycle time period of 3, 100, or 300 seconds. Data can be displayed with text explanation, or as columns of numbers for copy-pasting into spreadsheets etc.

Interrupts

Light and sound interrupts are configured on the MS430 and the program then waits indefinitely. When an interrupt occurs: a message is displayed, the interrupt is cleared, and the program returns to waiting.

On demand read-out

Similar to the cycle read-out example but using “on demand” measurements in standby mode. This allows measurements at intervals other than 3, 100, 300 seconds, but air quality data are not available. The on-demand read process takes 0.5 seconds and the user-selected delay adds to this, so choose e.g. 4.5 second delay to get a reading every 5 seconds. Note that most measurements except light and sound are less accurate if measured with delays less than 2 seconds. This is due to increased self-heating and reduced integration time.

Particle sensor toggle

For optional advanced use: this is the same as the cycle read-out example but also generates a power control signal for the external particle sensor. This can be used to turn the particle sensor on/off, reducing its fan noise or extending its lifetime. An external transistor circuit is required – see Section 8.

Home Assistant (not for Arduino Uno or Nano)

The host periodically sends environment data to an installation of Home Assistant on your network. You can display the data on the Home Assistant dashboard and also use it to trigger automations. More information is given in the GitHub Readme.

IFTTT (not for Arduino Uno or Nano)

Use this example together with a free account on IFTTT. The host device monitors some of the environment data and sends an alert to IFTTT when the values go outside your chosen "safe" ranges. IFTTT will then trigger an email which is sent to your email address. More setup information is given in the GitHub Readme.

IoT cloud logging (not for Arduino Uno or Nano)

A selection of environment data values are measured and logged to an internet cloud storage account every 100 seconds. This example requires an internet connection: on the Raspberry Pi, this can be wired or wireless, while for the other hosts it must be a WiFi network.

A cloud data hosting account is needed for this example. There is a choice of using either the Tago.io or Thingspeak.com cloud – both of these offer a free account for low data rates (compatible with this demo). See Section 7 for cloud account setup instructions.

The program sends data via HTTP POST requests, a widely used internet communication method. This can be adapted to work with other cloud services if needed.

Web server and graph web server (not for Arduino Uno or Nano)

All environment data values are measured and displayed on a web page generated by the host, which acts as a simple web server. The page can be accessed with web browsers on other devices on your network - navigate to the host's IP address to see the page. Arduino/ESP hosts can optionally act as an access point, generating their own WiFi network. There are two versions:

- **Web server**: a text-only page which shows the latest data. An internet connection is not required (the network can be local).
- **Graph web server**: the host stores a data history and creates a set of graphs using the Plotly javascript library (internet required). If there is no internet access, the browser may be able to use a cached copy if it previously accessed the page with internet access. Otherwise, the graphs will not load and you will only see text data. A button on the web page allows you to download the stored data as a CSV (comma separated value) text file, which can be opened with many spreadsheet applications.

Log data to file (Raspberry Pi only)

All environment data values are measured and saved as columns of numbers in a text file (one row of data every three seconds). This type of file can be imported into various graph and spreadsheet applications. To prevent very large file sizes, a new file is started every time it reaches a preset size limit.

7. IOT CLOUD HOSTING SETUP

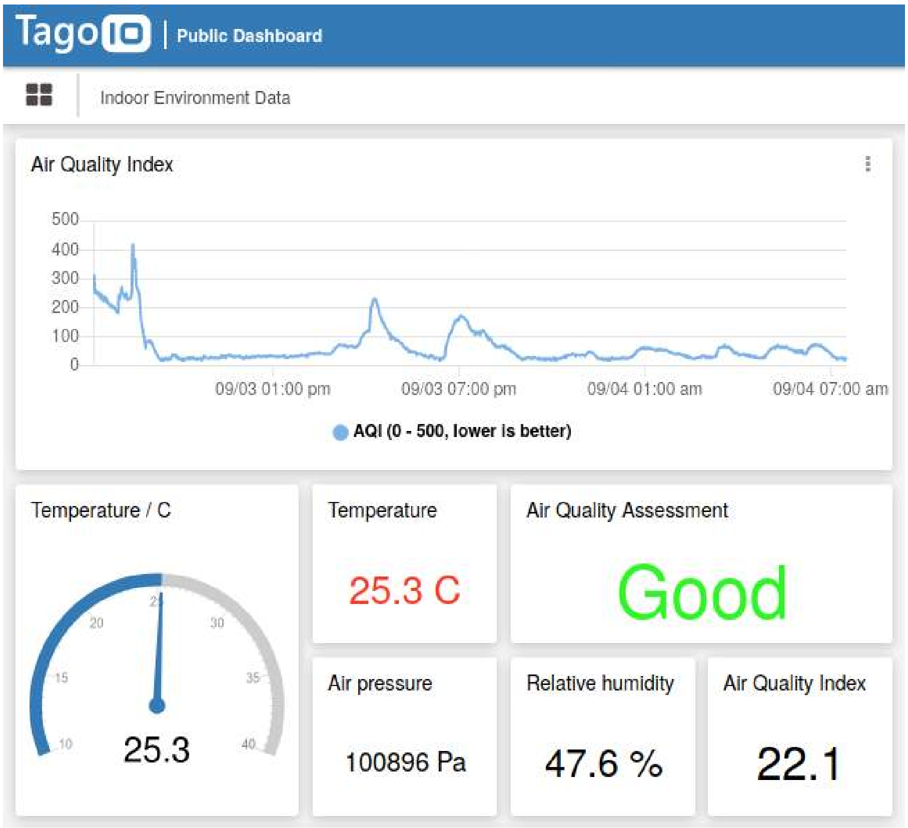
The Internet of Things (IoT) is the name given to devices connected to the internet which exchange data without direct human intervention. IoT has many applications, including home automation and environment monitoring.

One of the simplest uses of IoT is to periodically log sensor data in cloud storage (computer memory accessed over the internet). The data can then be shared and viewed from any internet connected device, at any time.

IoT cloud storage hosting is available from many providers around the world. Some offer free accounts (with storage or access limits) for non-commercial purposes. The IoT cloud logging code example shows how to send data to two such providers, Tago.io and Thingspeak.com. The following sections give a brief overview of how to set up free accounts with these providers: for further information see the relevant provider website.

Tago.io cloud

Figure 3 shows an example website screenshot of MS430 data logged with the Tago cloud.



Temperature / C 25.3

Temperature 25.3 C

Air Quality Assessment Good

Air pressure 100896 Pa

Relative humidity 47.6 %

Air Quality Index 22.1

Figure 3 - Tago.io screenshot with MS430 data

The steps required to set up Tago for the IoT cloud logging code example are:

1. Register for a free account at Tago.io
2. In the "Devices" menu, click Add Device.
3. Select "Custom HTTPS" (HTTP) as the device type.
4. Choose a device name (e.g. Indoor Environment Monitor) and click "Create device"
5. On the "General Information" tab of the new device, in the section "Token & Serial Number", click the eye icon to reveal the token (a long sequence of letters, numbers and hyphens). Copy this token.
6. Paste the token into the IoT cloud logging example code as the variable TAGO_DEVICE_TOKEN_STRING and set the variable "useTagoCloud" as true.
7. Run the IoT cloud logging example code on the host for a few minutes to ensure at least one set of data has been logged.
8. Verify that data are being stored in the Tago cloud by viewing the bucket's Variables tab. This should show the following short names for the environment data variables:
   temperature

pressure
humidity
aqi (air quality index, AQI)
aqi_string (a text interpretation of the AQI)
bvoc (equivalent breath VOC concentration)
spl (the A-weighted Sound Pressure Level)
peak_amp (the peak sound amplitude)
illuminance
particulates (air particle concentration)

9. Create a Tago dashboard for viewing the data: click the “+” to add a new dashboard, choose its name and click “save”.
10. Add widgets of various types to the dashboard (e.g. “Line” for a simple graph, “Card” to display text or numbers).
11. Configure each widget: the minimum configuration is to choose the Variable which it displays. It is also possible to edit graph and axis titles, coloring, calculate formulas, etc.
12. Set the graph displayed time period, e.g. for 24 hours: on the “Data Range & Format” tab of the chart settings, input 864 as the “maximum number of points to be displayed”. This assumes that the data are logged every 100 seconds (there are 86,400 seconds in 24 hours).
13. Create a public link for sharing your dashboard: click the three dots next to your dashboard name on the left-hand list. Choose “share” then copy the link displayed under the “share public” tab.

Thingspeak.com cloud

Figure 4 shows an example website screenshot of MS430 data logged with the Thingspeak cloud.

The steps required to set up Thingspeak for the IoT cloud logging code example are:

1. Register for a free account at Thingspeak.com
2. On the “My Channels” page, click the “New Channel” button
3. Choose a channel name (e.g. Indoor Environment Data)
4. Enable all eight fields (check boxes)
5. Put the following labels in the field name boxes (the order is important):
   Field 1: Temperature / C
   Field 2: Pressure / Pa
   Field 3: Humidity / %
   Field 4: Air Quality Index
   Field 5: Breath VOC / ppm
   Field 6: Sound Level / dBA SPL
   Field 7: Illuminance / lx
   Field 8: Particle Concentration
6. Click “Save Channel”. The channel will show eight (initially empty) graphs.
7. To set graph time periods to 24 hours: on each graph click the pencil icon, delete 60 from the Results box and put 1 in the Days box, then click “Save”. This changes the graph period from the last 60 values to the last 1 day. This must be done separately for both the private and public view if the channel is shared publicly.

8. The channel can be made public, if desired, from the Thingspeak “Sharing” tab.
9. Go to the “API Keys” tab and copy the Write API Key (a sequence of letters and numbers).
10. Paste the API key into the IoT cloud logging example code as the variable THINGSPEAK_API_KEY_STRING and set the variable “useTagoCloud” as false.

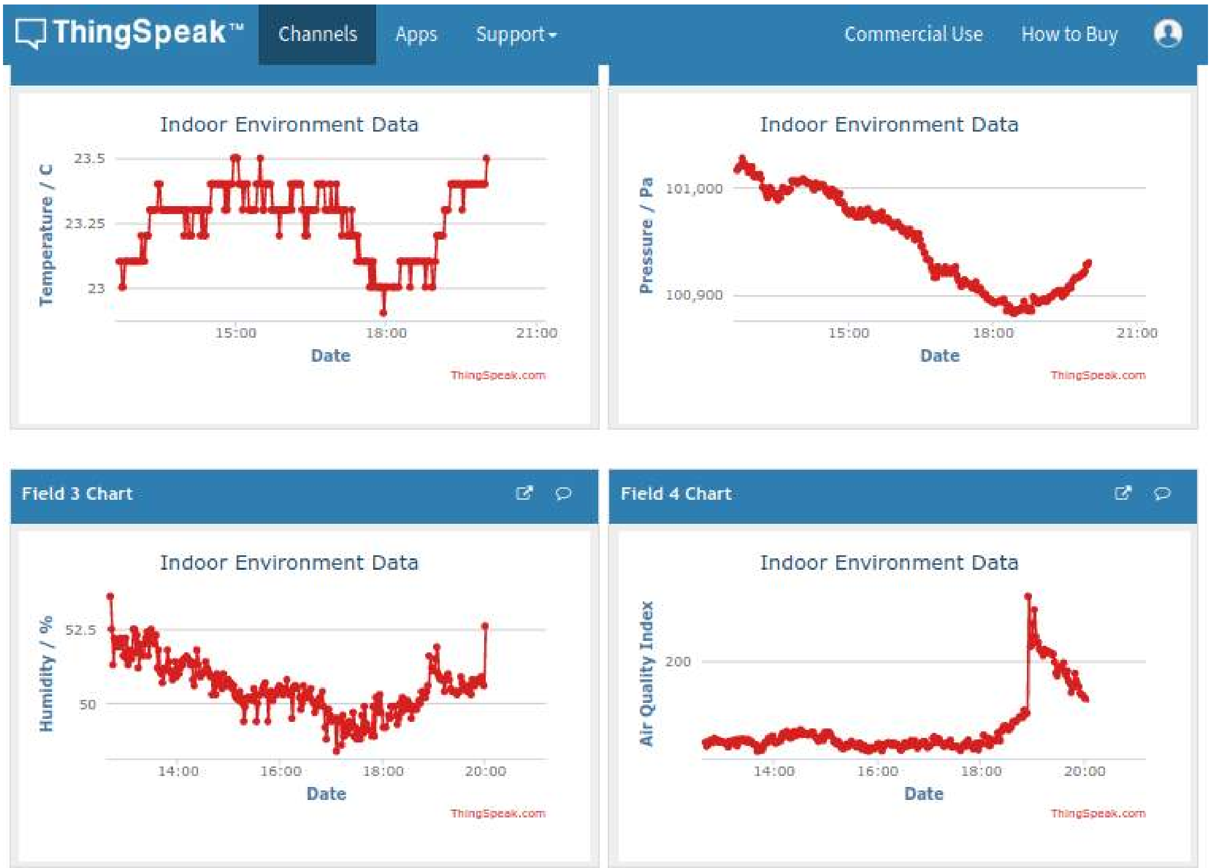


Figure 4 - Thingspeak.com screenshot with MS430 data

8. PARTICLE SENSOR POWER CONTROL

Any particle sensor used with the MS430 is normally continuously powered by an external 5 V supply. Some users may wish to periodically disable the particle sensor to reduce sound noise, reduce power, or increase the sensor lifetime. This section briefly describes one way this can be done.

The code example **particle_sensor_toggle** shows how the host can output a digital signal synchronized to the MS430 data cycle. This signal can be used to gate power to the particle sensor using an external circuit, such as shown in Figure 5 (for hosts using 3.3 V) or Figure 6 (for hosts using 5 V). Input “Host” is the digital signal and output “PS_5V” is connected to the particle sensor 5 V power input.

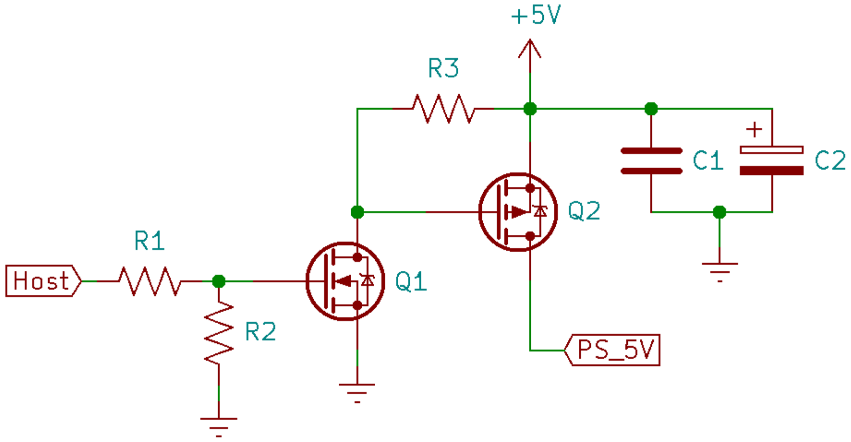


Figure 5 - Particle sensor power control circuit for 3.3 V hosts

The recommended component values are:

- R1, R4: 1 kΩ
- R2, R3, R5: 100 kΩ
- Q1: N-channel MOSFET, $V_{GS(th)}$ < 3.3 V (“logic level”) e.g. 2N7000
- Q2, Q3: P-channel MOSFET, $R_{DS(on)}$ < 1 Ω, $V_{GS(th)}$ < 3.3 V (“logic level”) e.g. SPP15P10PL H
- C1, C3: (optional, use if interference occurs) 100 nF
- C2, C4: (optional, use if interference occurs) ≥ 100 µF

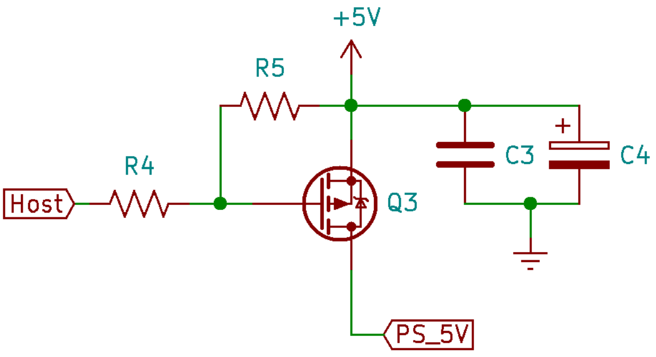


Figure 6 - Particle sensor power control circuit for 5 V hosts

9. INDOOR ENVIRONMENT DATA EXPLAINED

Each of the following sections describes the environment data quantities measured by the MS430. Table 14 (page 26) gives the numerical range and resolution of each measured variable.

SOUND

Sound travels as pressure fluctuations in the air, in the form of waves of measurable frequency and amplitude. Noise is often defined as unwanted or unpleasant sound.

Exposure to noise at home, at work, or during sleep has been associated with many adverse health and emotional outcomes. These include distraction, annoyance, cardiovascular disease, tinnitus and hearing loss. The World Health Organization states that traffic noise alone is harmful to the health of almost every third person in Europe, and that one in five Europeans is regularly exposed to sound levels at night that could significantly damage health.

The MS430 uses a high-performance digital microphone to detect frequencies between 50 Hz and 8000 Hz, covering the dominant range of human hearing.

Sound Pressure Level (A-weighted)

The Sound Pressure Level (SPL) is a popular measurement system for noise. Sound amplitudes measured by a microphone are averaged over all frequencies to produce a single SPL number, expressed on a logarithmic scale in decibel units. SPL measurements are best for ongoing constant noise, while peak amplitude measurements are best for brief, sudden sounds.

When calculating SPL, some frequencies can be emphasized relative to others – this is known as the *weighting*. The most common method is “A-weighting”, an internationally recognized standard which accounts for the variation in how the human ear hears different sound frequencies. For example, people’s perception of loudness tends to peak at around 3 kHz and drops at low and high frequencies. Noise around 3 kHz is therefore given a greater weighting when calculating the SPL – this weighting is shown in Figure 7.

The A-weighted Sound Pressure Level is a useful and very commonly used measure of environment noise and sound “loudness”. Table 4 and Table 5 give some example sound sources and typical SPL values. Note that the sound level (perceived or measured) generally depends on:

- What is creating the sound
- Distance from the source to the meter or ear
- Direction, angle or alignment
- Other factors such as nearby objects, air conditions, etc.

Table 4 - Typical A-weighted Sound Pressure Level values. All values are approximate and will vary depending on exact source type and positioning.

Sound source or description	Distance from source / m	A-weighted Sound Pressure Level / dBA	Typical sound amplitude / mPa
Quiet breathing	1	25	0.50
Quiet room in house	Ambient	30	0.9
Conversation	1	40 - 60	3 - 30
Hair dryer	1	70	90
Inside a car at 70 mph	Ambient	75	200
Expressway traffic	25	80	300
Lawnmower	1	90	900
Jackhammer	1	100	3,000
Night club with music	Ambient	110	9,000
Jet engine at takeoff	100	120	30,000

Table 5 - A-weighted Sound Pressure Level approximate guidelines from the World Health Organization and US government agencies NIOSH and OSHA.

Sound source or description	A-weighted Sound Pressure Level at ear / dBA
Continuous noise limit for unimpaired sleep	30
Distraction from tasks	40
Noise induced annoyance/stress	45 – 55
Long-term exposure causing adverse cardiovascular effects	65 – 70
Recommended maximum for long-term exposure (24 hours)	70
Recommended maximum for 8 hour exposure	85
Recommended maximum for 15 minute exposure	100

WARNING: these values are not safety advice and should not be treated as such. The sensor product is not suitable for safety or health-critical measurements.

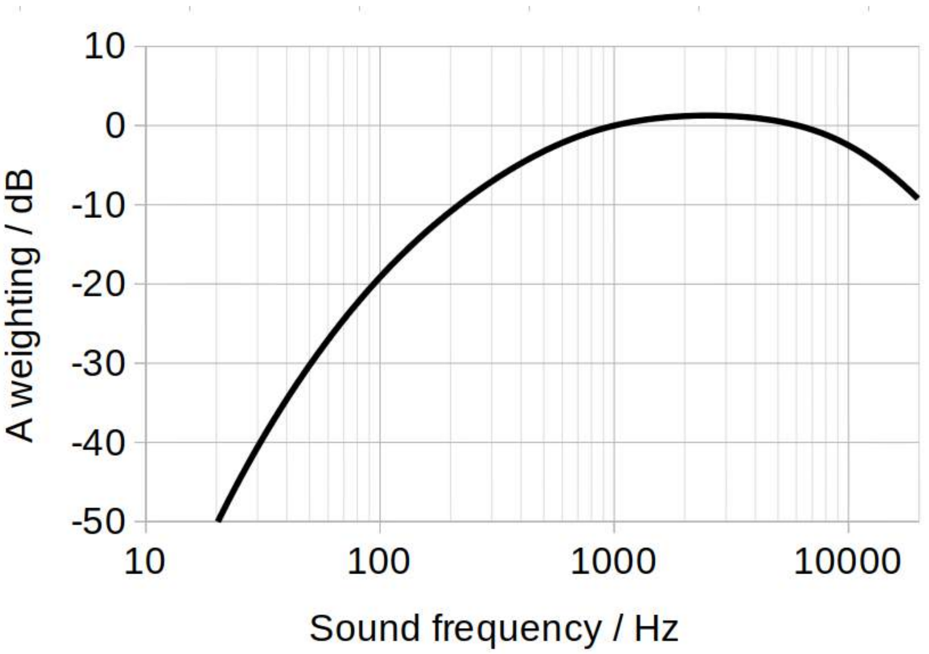


Figure 7 - The standard A-weighting curve.

Frequency band Sound Pressure Level

The MS430 also provides the unweighted SPL for several ranges of sound frequency (known as frequency bands). This reveals what pitches are present in the sound, for example the treble and bass notes in music. The band frequency ranges are listed in Table 6.

Table 6 - Frequency bands for unweighted Sound Pressure Level measurement.

Band	Midpoint frequency / Hz	Lower frequency / Hz	Upper frequency / Hz
1	125	88	177
2	250	177	354
3	500	354	707
4	1000	707	1414
5	2000	1414	2828
6	4000	2828	5657

Peak sound amplitude

The peak sound amplitude is a measurement of the largest pressure fluctuation to occur since the last time the value was read. The MS430 continually monitors the sound amplitude and internally updates this peak value (rather than making a one-off measurement). This means that sudden, impulsive noises are not missed. The peak value automatically resets after being read.

The sound interrupt system (explained further in the device datasheet) uses the peak amplitude value as a trigger for a digital output signal. This can be used to respond rapidly to changes in sound level, without the need for software processing.

Ideas for further investigation of sound

- Road/aircraft traffic noise – what levels are reached and which times of the day are worst?
- Use of sound interrupts for control of appliances e.g. turn on light with specific sequence of hand claps.
- Is your home music system set at an appropriate volume?
- Are areas for work or study maintained at a low enough sound level?
- Test the effectiveness of sound-proofing methods for reducing noise in the home from external sources.
- Use the frequency band SPL values to display a frequency spectrum for live visualization of music.

LIGHT

Light comprises photons of different wavelengths which are perceived as colors by the human eye and brain, while perceived brightness depends on the quantity of photons.

We need different minimum levels of light for vision during various indoor activities such as movement, reading/writing and detailed manual work. At the other extreme, too much light at night can adversely affect sleep quality.

Illuminance

Illuminance is a measure of the total brightness of light illuminating a surface, as would be perceived by a person. The illuminance calculation emphasizes some wavelengths more than others to represent differences in the brightness perception of human vision. Our vision responds to light between about 400 nm and 700 nm, with strongest response around 550 nm (green light). This is represented by the illuminance response curve shown in Figure 8.

Table 7 lists some examples of natural illuminance levels, while Table 8 lists recommended values for various indoor areas and activities.

The MS430 light interrupt system (see datasheet for further details) uses the measured illuminance value as a trigger for a digital output signal. This can be used to respond rapidly to changes in light level, without the need for software processing.

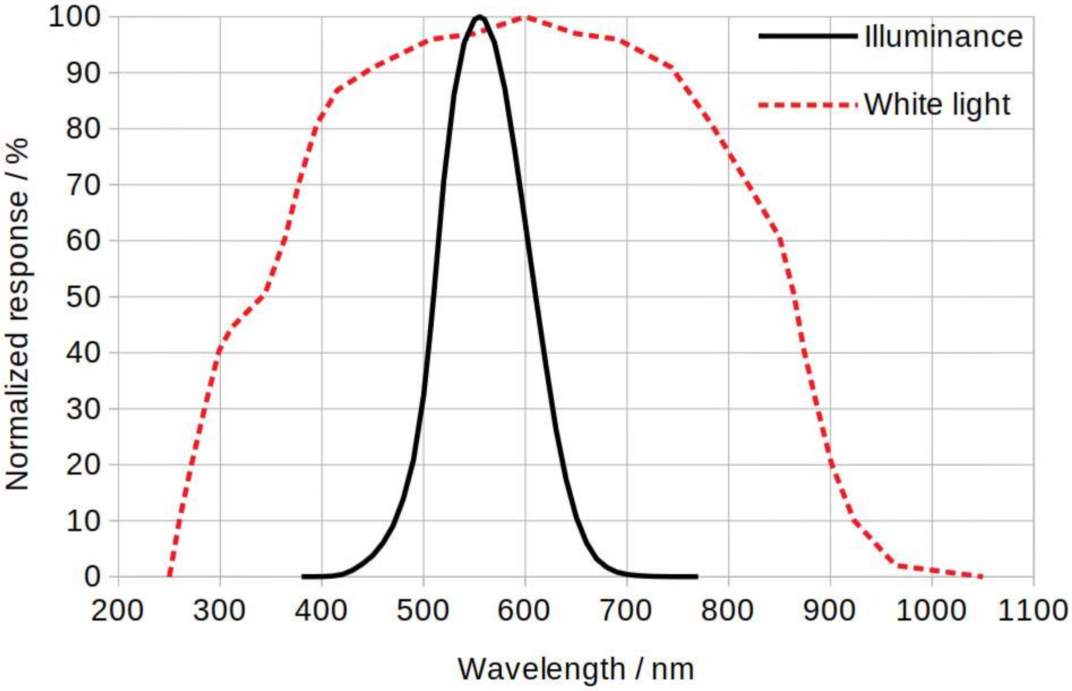


Figure 8 - Spectral response for illuminance and white light measurements.

White light level

A white light measurement (uncalibrated units) is also available from the MS430. This has a broader wavelength response over the range of about 250 to 1000 nm (shown in Figure 8 compared to the illuminance response). Although not provided in standardized units, this variable can be used to detect proportional increases/decreases in light level.

Ideas for further investigation of light

- Check that work areas, stairwells, etc. have light levels suitable for the work or activities being done there. Does this rely on sunlight, or is artificial light sufficient?
- Use the light interrupt system to control room lights in response to low/high ambient light levels.
- At what times of day/night does a bedroom become dark enough for good sleep? Does additional light-proofing help?

Table 7 - Examples of naturally occurring illuminance levels.

Condition	Illuminance / lux
Full daylight	10,000
Overcast day	1,000
Very dark day	100
Twilight	10
Deep twilight	1
Full moon at night	0.1
Quarter moon at night	0.01
Starlight at night	0.001

Table 8 - Recommended illuminance levels[1] for activities and indoor areas (workplace and home). All values are approximate.

Location / activity	Illuminance / lux
Detailed or precise drawing / manual work	750 – 2000
Frequent reading / writing	500 – 1000
General office work, kitchen	300 – 750
Occasional reading / writing	200 – 300
Corridor, lobby, auditorium, dining area	150 – 200
Stairwell	100 – 200
Other home areas	50 – 150
Night-time maximum in bedroom	5 – 10
Night-time maximum, at pillow	0.5

AIR

The MS430 directly measures four variables which depend on properties of the air: temperature, air pressure, humidity and the resistance of a specialized gas sensor.

Temperature

Temperature is an essential measurement for indoor environment monitoring and is a standard feature of heating, air conditioning and home automation systems. The MS430 obtains accurate

[1] Sources: the US government agencies GSA, NOAO, DoE, and the UK CIBSE guide for hospital lighting.

temperature measurements in degrees Celsius. Temperature is especially useful when combined with humidity measurements to determine the comfort level of the environment.

Pressure

Air pressure is caused by the weight of the gases in the atmosphere and decreases with altitude. At a fixed location, the pressure changes due to weather and atmospheric processes. Reduced pressures are often associated with clouds and rain, whereas pressure increases are associated with dry weather and mostly clear skies. The MS430 measures absolute atmospheric pressure, which also allows measurement of relative changes in altitude.

Humidity

Humidity is measured by the MS430 as percentage relative humidity, the most common form of humidity measurement. This quantifies the amount of water vapor in the air as a percentage of the maximum that could exist at the current temperature and pressure. This is a useful definition because it reveals whether water will condense onto objects exposed to air, and also correlates with people’s level of discomfort.

In hot environments, high levels of humidity prevent sweat evaporation from skin, adversely affecting the body’s cooling mechanism. This is uncomfortable and can be dangerous for the elderly or those with heath problems. High humidity at low temperatures is also uncomfortable because heat is conducted away from the skin, making us feel cooler.

At the other extreme, very low humidity levels can be uncomfortable due to the drying effect on the skin and respiratory system.
The World Heath Organization recommends[2] that indoor relative humidity be maintained in the range 30 – 50 %.

Moisture condensation onto surfaces due to high humidity risks the following problems:

- Mold and rot
- Dust mite growth
- Corrosion or degradation of materials
- Windshield fogging inside vehicles

Condensation of the water vapor in the air occurs when the air is cooled lower than the *dew point*, a special temperature which depends on the air humidity and temperature. This can happen when the air is in contact with an object at a temperature lower than the dew point e.g. a cool window, wall or mirror. Table 9 shows dew point values for common household air temperatures and humidities. For example: in a house with air at 25 °C and 50 % humidity, condensation could occur on a surface below 14 °C.

[2] World Health Organization, *Indoor air quality: biological contaminants*, 1988

Table 9 - Dew point temperatures (±1 °C) for common household air temperatures and relative humidities[3].

Dew point temperature / °C		**Air relative humidity / %**					
		30	**40**	**50**	**60**	**70**	**80**
Air temperature / °C	**15**	-3	1	4	7	10	12
	20	2	6	9	12	14	17
	25	6	11	14	17	19	21
	30	11	15	18	21	24	26
	35	15	19	23	26	29	31

Gas (volatile organic compound) sensor resistance

Volatile organic compounds (VOCs) are a class of air impurities that can evaporate at normal indoor temperatures and pressures. This means that VOCs originating in liquid and solid materials can enter the air in everyday conditions. Concentrations of VOCs are generally higher in indoor air than outdoor air due to reduced air volume and flow.

The effects of VOCs on health vary depending on VOC type, concentration and length of exposure. Minor adverse health effects of VOCs can include: eye, nose and throat irritation, headaches, fatigue, nausea and skin problems. Higher concentrations and/or long-term exposure are suspected of causing cancer, as well as damage to the liver, kidney, or central nervous system.

Sources of VOCs include paint, furniture, garbage, plastics, cleaning products, perfumes, clothes, cooking and people (breath and odors). VOCs present in breath include isoprene, ethanol and acetone. Breath can become a dominant source of VOCs, especially in poorly ventilated rooms containing many people.

The MS430 detects VOCs using a highly responsive metal-oxide gas sensor. The gas sensor resistance is modified by VOC adsorption (and subsequent chemical oxidation/reduction) on its surface. Greater VOC concentrations result in a lower sensor resistance, while cleaner air gives a higher resistance.

Ideas for further investigation of air measurements

- Look for correlations of air pressure changes with local weather.
- Check that home heating/cooling systems maintain the correct temperature.
- Use the dew point table, temperature and humidity data to warn when condensation can occur.
- What is the effect on humidity of actions such as: improving home ventilation, drying clothes indoors, using air extraction during cooking and showering?

[3] Source: the US National Oceanic and Atmospheric Administration (NOAA).

AIR QUALITY

The MS430 uses the directly measured air parameters (temperature, humidity and gas sensor resistance) as inputs to an intelligent monitoring algorithm. This produces three derived data values to indicate the quality of the air: Air Quality Index, Estimated CO_2 concentration, and Equivalent breath VOC concentration.

Air Quality Index

The US Environmental Protection Agency (EPA) has identified poor indoor air quality as one of the five major public health risks. The MS430 produces an Air Quality Index (AQI), with values between 0 and 500, to give a qualitative assessment of the air in the surrounding environment.

The AQI value is calculated using the gas sensor output, while also accounting for temperature and humidity changes. Lower values indicate better air quality and higher values indicate worse air quality: Table 10 shows approximate interpretations for the full range.

Table 10 - Interpretation of Air Quality Index values.

Air Quality Index	Interpretation
0 – 50	Good
51 – 100	Acceptable
101 – 150	Substandard
151 – 200	Poor
201 – 300	Bad
301 – 500	Very Bad

The AQI algorithm undergoes a process of automatic self-calibration to ensure that users experience consistent performance. A value of 25 corresponds to typical “good” air, and 250 to typical “polluted” air.

The MS430 provides an accuracy parameter to show the state of the self-calibration process, starting with a value of 0 indicating uncalibrated, and reaching 3 indicating highest accuracy – these values are shown in Table 11 and may take several minutes of operation to achieve, especially for a new sensor. During the code 0 period, values for AQI, equivalent breath VOC, and estimated CO_2 concentration are invalid. If the accuracy is slow to change, provide a variation of air quality e.g. by breathing heavily on the sensor or exposing to a solvent such as marker pen vapor.

Equivalent breath VOC concentration

This measurement makes the assumption that the detected VOCs are due to a typical characteristic mixture in exhaled human breath. This is reasonable in environments where breath is the dominant source of VOCs (e.g. in crowded rooms). The output value is the inferred concentration of this characteristic breath VOC mixture. The typical maximum VOC concentration in human breath is of order 10 ppm (parts per million).

Table 11 - Interpretation of the air quality accuracy code.

Accuracy code	Interpretation
0	Not valid: calibration incomplete
1	Low accuracy: calibration ongoing
2	Medium accuracy: calibration ongoing
3	High accuracy

Estimated carbon dioxide (CO_2) concentration

This measurement also makes the assumption that the detected VOCs are due to the typical characteristic mixture in exhaled human breath. This is reasonable in environments where breath is the dominant source of VOCs (e.g. in crowded rooms). A correlation between exhaled breath VOC concentration and exhaled CO_2 gas is then used to estimate the CO_2 concentration. Note that the MS430 does not detect CO_2 directly.

The global average atmospheric CO_2 concentration is approximately 400 ppm. US federal agency NIOSH considers that indoor CO_2 concentrations exceeding 1,000 ppm indicate inadequate ventilation. Increased CO_2 levels can cause drowsiness, headaches and reduced cognitive performance.

Ideas for further investigation of air quality

- Alert a room's occupants to worsening air quality so that they can take action (open window/door, leave the room, turn on air conditioning) rather than waiting until the air quality is noticeably bad.
- How fast or effective in improving air quality are various actions e.g. opening a window, using an air conditioner, removing garbage, using air extraction when cooking?

PARTICULATES

Airborne particles, sometimes referred to as *particulate matter*, include dust, smoke, pollen, and liquid vapor. Of special interest are those with diameters less than 10 µm, which are easily inhaled and so have the most impact on health. Particle pollution has been linked to problems including respiratory illness (e.g. asthma and bronchitis), cardiovascular disease and lung cancer.

Authorities do not yet provide reference levels for indoor air particulates. Guidelines/standards for **outdoor** air issued by the European Union, US EPA, and World Health Organization are shown in Table 12 for PM10 and Table 13 for PM2.5. Rules differ on the precise averaging and number of times that limits may be exceeded. Note that PM2.5 means the sum of all particles smaller than 2.5 µm and PM10 means the sum of all particles smaller than 10 µm.

Table 12 - Guideline concentration limits for outdoor PM10 pollution

μg/m^3	**24 hour average**	**1 year average**
EU	50	40
US EPA	150	-
WHO	50	20

Table 13 - Guideline concentration limits for outdoor PM2.5 pollution

μg/m^3	**24 hour average**	**1 year average**
EU	-	25
US EPA	35	12
WHO	25	10

The MS430 provides an input connection and data analysis algorithm compatible with Shinyei PPD42 and Nova SDS011 particle sensors. Both sensors use optical scattering to detect particles of around 1 μm diameter. Two related output quantities are produced by the MS430: particle sensor duty cycle and estimated particle concentration.

Particle sensor duty cycle

Both sensors output a digital signal which is active when particles are detected within the sensor. The signal activation time varies depending on the size or quantity of the detected particles. The “duty cycle” is the overall fraction of time during which this detection signal is active. Greater duty cycles indicate larger particulate concentrations.

Estimated particle concentration

The manufacturer’s nominal conversion method is used to obtain the particle concentration value from the duty cycle value. Because this is not a calibrated system, the value obtained should be treated as an estimate only. The PPD42 concentration units are particles per liter (ppL) and the SDS011 units are micrograms per cubic meter (μg/m^3).

Ideas for further investigation of particulates

- What events or activities are associated with spikes in particle concentration?
- Can particle concentration be reduced through steps such as: air extraction during cooking, removal of dust, and removing indoor smoke sources?

Table 14 - Measurement range, resolution and accuracy for all data variables.

Data category	Data variable	Measurement unit (symbol)	Range minimum	Range maximum	Resolution	Measurement accuracy
Sound	A-weighted sound pressure level	decibel (dBA)	26.0	120.0	0.1	±1.0 dBA
	Frequency band sound pressure level	decibel (dB)	21.0	120.0	0.1	±1.0 dB
	Peak sound amplitude	millipascal (mPa)	0.00	28000.00	0.01	*
Light	Illuminance	lux (lx)	0.00	3774.00	0.01	±10%
	White light level	-	0	65535	1	*
Air	Temperature	degree Celsius (°C)	-40.0	85.0	0.1	±0.5 to ±1.0 °C
	Pressure	pascal (Pa)	30000	110000	1	±60 Pa
	Humidity	Percentage relative humidity (% RH)	0.0	100.0	0.1	±3% RH
	Gas sensor resistance	ohm (Ω)	0	4.29×10^9	1	±1.5%
Air Quality	Air quality index	-	0.0	500.0	0.1	*
	Estimated CO_2 concentration	Parts per million (ppm)	0.0	65535.0	0.1	*
	Equivalent breath VOC concentration	Parts per million (ppm)	0.00	1000.00	0.01	*
	Accuracy	-	0	3	1	*
Air particulate matter	Particle sensor duty cycle	Percentage (%)	0.00	100.00	0.01	±0.05%
	Particle concentration (PPD42)	Particles per liter (ppL)	0.00	28000.00	1.00	*
	Particle concentration (SDS011)	micrograms per cubic meter (μg/m^3)	0.00	999.90	0.01	The greater of ±15% and ±10 μg/m^3

*These measurements are either: relative/qualitative values, use a non-standard measurement scale, or are uncalibrated. The values are useful for direct comparisons or further research.

10. DISCLAIMER

ALL PRODUCTS, PRODUCT SPECIFICATIONS AND DATA ARE SUBJECT TO CHANGE WITHOUT NOTICE TO IMPROVE RELIABILITY, FUNCTION OR DESIGN, OR OTHERWISE.

Metriful Ltd., its affiliates, agents, and employees, and all persons acting on its or their behalf (collectively, “Metriful”), disclaim any and all liability for any errors, inaccuracies or incompleteness contained in any datasheet or in any other disclosure.

Metriful makes no warranty, representation or guarantee regarding the suitability of the products for any particular purpose or the continuing production of any product. To the maximum extent permitted by applicable law, Metriful disclaims (i) any and all liability arising out of the application or use of any product or data, (ii) any and all liability, including without limitation special, consequential or incidental damages, and (iii) any and all implied warranties, including warranties of fitness for particular purpose, non-infringement and merchantability.

Statements regarding the suitability of products for certain types of applications are based on Metriful’s knowledge of typical requirements that are often placed on Metriful products in generic applications. Such statements are not binding statements about the suitability of products for a particular application. It is the customer’s responsibility to validate that a particular product is suitable for use in a particular application.

Parameters provided in datasheets and / or specifications may vary in different applications and performance may vary over time. All operating parameters, including typical parameters, must be validated for each customer application by the customer. Product specifications do not expand or otherwise modify Metriful’s terms and conditions of purchase, including but not limited to the warranty expressed therein. Product photographs and diagrams are for illustration purposes only and may differ from the real product appearance.

Metriful products are not designed for use in medical, life-saving, safety, or life-sustaining applications or for any other application in which the failure of the Metriful product could result in personal injury or death. Customers using or selling Metriful products for use in such applications do so at their own risk. No information contained in any Metriful document or disclosure constitutes health or safety advice.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document or by any conduct of Metriful. Product names and markings noted herein may be trademarks of their respective owners.

Document revision: 2.1
Document release date: 16 November 2020
Document number: MET-UG001-03