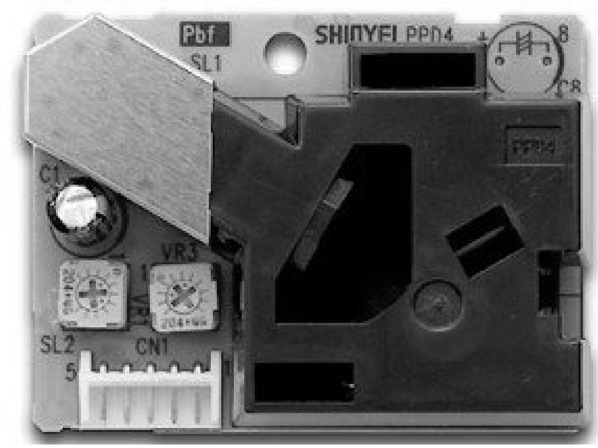
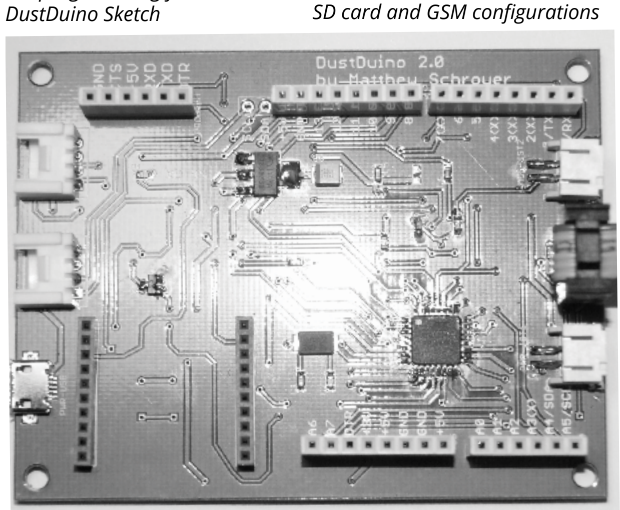
DustDuino Cookbook

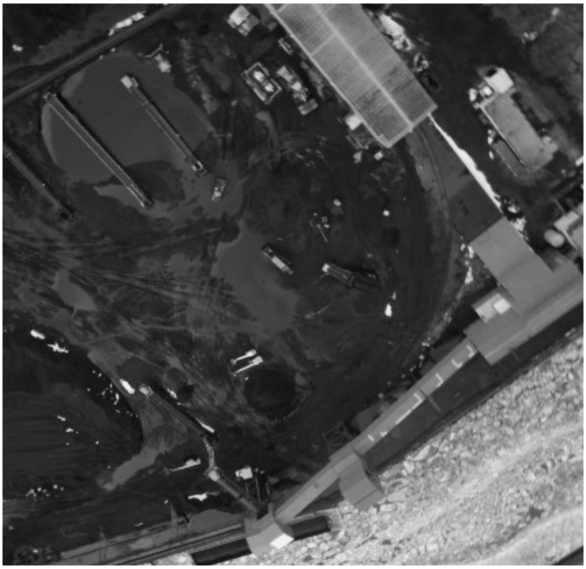
Count and report airborne particles

By Matthew Schroyer and Willie Shubert


Pbf
SHINYEI PPD4
SL1
C1
VR3
SL2
CN1


Programming Headers
For programming your DustDuino Sketch
Arduino Shield Headers
For adding shields needed to create SD card and GSM configurations
Grove Ports
For adding Grove compatible sensors
JST Ports
For Solar Panel and Battery
Mini-USB Port
For Power Only
XBee Headers
For Wi-Fi and Xbee Mesh
Arduino Shield Headers
For adding shields needed to create SD card and GSM configurations
DustDuino 2.0

INTRODUCTION



Development of the DustDuino was inspired by community health concerns about a petcoke site identified using balloon mapping by Public Lab members (left) and a tire fire near a residential neighboorhood in Hoopeston, Illinois (right)

In the summer of 2011, a journalist spoke with a family living in a south-side Chicago neighborhood about the biggest threats they faced on a daily basis. This was a close-knit community, but like many others in that part of the city, it was one that weathered poverty and crime on a daily basis.

The family told stories about shootings, and blood on the streets, but to the surprise of the journalist, this wasn't their biggest concern. Crime, to some extent, can be avoided. As unfortunate as the choice may be, parents can tell their children to avoid spending time on the sidewalk.

But parents can't tell their children not to breathe the air, when it's all around them, every day.

This family was facing extremely rare, multigenerational lupus, and they looked to the ancient coal power plant and smelting operation nearby as contributors to this serious disease. Others in the neighborhood related stories of piles of coal catching fire in the summer, plumes of smoke, and unusual dust gathering on the cars and streets outside.

Like many communities challenged with economic hardships, the community was burdened by an environmental injustice. This was readily apparent to the people living there, but wasn't taken seriously by officials, who were lacking rigorous data. So the community rallied together, and began rounding up that data, sample by sample.

The community sent soil samples to a lab to be examined for lead content. And when the results showed the soil had a lead concentration of 92 times what is considered safe, it confirmed what the citizens already knew, and it spurred environmental authorities into action.

More soil sampling followed. Then, authorities placed an environmental monitor on top of the neighborhood's elementary school. Then, part of neighborhood was declared a "nonattainment" area for lead, meaning enforcement could happen. The smelting operation reduced its lead output, and the power plant was shut down altogether.

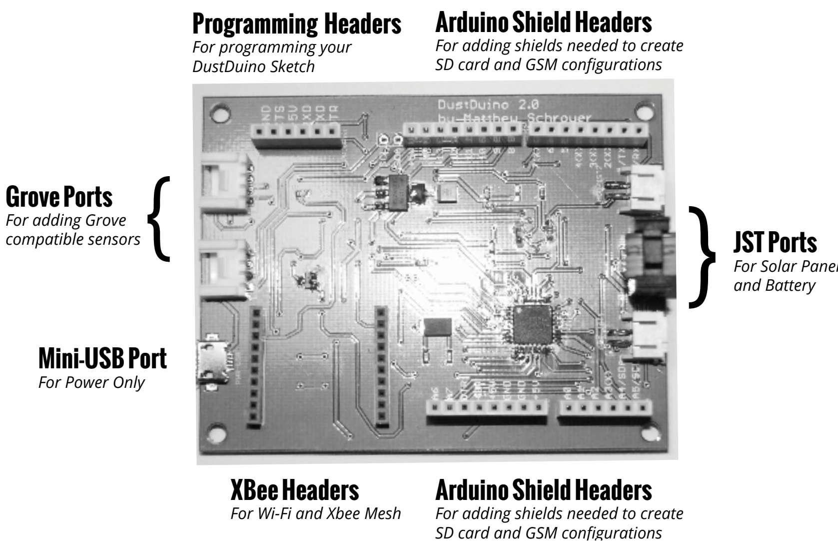
It wasn't immediately apparent to the journalist, Matthew Schroyer, then a graduate student at the University of Illinois, but if more communities and journalists had access to environmental sensing, citizens could have much better understanding about their surroundings. It wasn't until the he stumbled into the makerspaces, fablabs, and other workshops in his town that were providing digital fabrication tools and education to the public, that he realized that environmental sensors could be created in any community with access to tools or civic-minded hackers.

After some months spent tinkering in these workshops with open-source hardware and software, the DustDuino was born. Then, the Tow Center at the Columbia Graduate School of Journalism, and Willie Shubert at the Earth Journalism Network began looking at the DustDuino for research, applications, workshops, and outreach programs.

As of Spring of 2015, more than 30 DustDuinos have been produced and deployed in such diverse places as New York City, New York, and Ulan Bator, Mongolia. But realizing the demand for this type of air quality monitor, the DustDuino collaborators decided to have components mass produced, so that others may build and deploy sensors where they live, even without access to community workshops.

This “cookbook” represents the fruits of years of experimentation with low-cost particulate sensors. The authors hope citizens, journalists, and researchers worldwide can use this text as a guide to make environmental sensors of their own, or even to perfect and customize the DustDuino to get the kind of data they need to be more informed about air quality.

What is a DustDuino?



Put simply, the DustDuino is a device that is meant to measure air quality, or specifically to count and report the number of airborne particles where it is placed.

A DustDuino is comprised of several components. Perhaps the most important of these components is the particulate sensor, which actually is the part that counts the number of coarse and fine particles in the air. The DustDuino as a whole may sometimes be called a “sensor,” but this is the real sensor at the heart of the system.

A sensor is essential for an electronic air quality monitor, but an air quality monitor usually requires other parts in order to make good use of the sensor and its readings. That's where the DustDuino board is useful. The particulate sensor plugs into the board, and the microcontroller chip on the board listens to the sensor and convert the sensor's digital output into numbers that humans can relate to. The microcontroller chip also will create messages, so that the sensor data can be sent over the internet or stored locally on an SD card.

Besides the connector for the dust sensor, there are a number of other connectors on the board. There is a connector for an Xbee-compatible module, which is a small radio that can send data to other Xbee device. Some Xbee-compatible modules can also connect the DustDuino to a WiFi network, enabling internet connectivity.

Power is essential to keep the DustDuino operational, and there are several connectors on the board especially for that purpose. There is a Micro USB-B connector that will allow you to draw the DustDuino's power from most cell phone charging bricks, or also from a USB computer port. Phone charging bricks and USB computer ports all provide 5 volts of power, which is exactly what the DustDuino needs to run properly.

On the opposite side of the board from the Micro USB connector are two **JST connectors**. The PWR-JST1 connector will power the board just as well as the **Micro USB** connector will, as long as 5 volts of power are provided. The JST connector format is very common, often found on small-sized batteries, low-voltage power supplies and solar panels. Because the DustDuino can be powered on a modest solar array, there also is included a **MON-JST2** connector, which can be used to safely monitor the output of solar cells of up to 10 volts.

There may be applications where you want to log the time along with the dust readings. For this purpose, there is a **Seeed Grove connector** for I2C device connection. Seed provides a real time clock (RTK) that can communicate with the DustDuino's microcontroller chip using the I2C protocol. I2C is a common communication protocol used by many other devices and sensors, so you are by no means limited to using a clock on this connector.

The DustDuino board is compatible with Arduino shields, which are separate circuit boards that can add functionality to your monitor. For example, an **SD card shield** could be attached, giving the monitor the ability to store readings on an SD card. If one wishes to add an LCD screen to display particulate readings live, that could be attached as a shield as well. The opportunity for expansion is limited only by one's imagination.

Just as the board is Arduino shield-compatible, the microcontroller chip also is Arduino-compatible. Arduino is an open-source physical computing platform, and is a very flexible, low-power computer that can be used to easily accomplish tasks in the physical word. Best of all, the code to make everything work is freely shared online. People who make Arduino gadgets are encouraged to borrow designs and code from other Arduino makers, so long as credit is given, and all the improvements and changes are made free of charge to the public. We encourage all DustDuino developers to share their improvements, as well.

Why make one?

Being able to see the dust blowing in the wind has broad implications. Today, one in eight people in the world dies from exposure to air pollution, which includes dust. This stunning fact, issued by the World Health Organization in March 2014, adds up to 7 million premature deaths per year. Air pollution is now the single largest environmental risk to human health in the world, and it occurs both indoors and outdoors. In October 2013, the World Health Organization announced they consider particulate matter, a major component of indoor and outdoor air pollution, as a Group 1 carcinogen along with tobacco smoke and asbestos.

What level of accuracy can the DustDuino provide?

There is an enormous need for low-cost air pollution monitoring around the world. In particular, particulate matter has been identified as a globally important public heath problem. The DustDuino measures both PM10 and PM2.5 by utilizing a $15 USD optical sensor, the Shenyei PPD42NS. The construction of the sensor includes an Infrared LED, a focusing lens, and a photodiode to determine the concentration of dust in a partially closed chamber that draws in air from its surroundings.

Design choices for the DustDuino have been based primarily on the research findings of two scientific studies focused on determining the quality and applicable use cases for the Shenyei PPD42NS with reference monitors including the DustTrak, the Dylos laser counter, and a Federal Equivalent Method β-attenuation monitor (BAM-1020 from Met One Instrument).

Testing with reference monitors conducted in Berkeley, California in late 2013 concluded that the results of the low-cost sensor were about equivalent to much more expensive ones when analyzing data at hourly intervals. "Performance at 1 [hour] integration times was comparable to commercially available optical instruments costing considerably more."

Testing in high PM2.5 environments such as the city of Xi'an, China also has shown high correlations between the Shenyei and reference monitors. By setting up a variety of monitors throughout the city, researchers were able to identify the High-technology Zone site as a potential PM2.5 hotspot with sustained high concentrations compared to the city average throughout the day.

Critiques and Limitations

Challenging previously held assumptions, adjusting expectations to match the capabilities of the equipment, and following the scientific method is critical for getting the highest quality data from the DustDuino. There are a number of issues and pitfalls to be aware of when both setting up the monitor and reviewing the data it creates.

Accounting for Weather --Monitoring air-pollution levels is far more involved than the manufacturers and suppliers of cheap sensors suggest. In a January 2015 article in Nature about the current status of DIY air monitoring, Ben Barratt, an air-quality scientist at King's College London stated that the problem is that temperature, humidity and some gases skew the results from sensors making it difficult to compare results between devices. This suggests that calibrating the DustDuino with reference monitors that include local weather information can assist in the modelling process when visualizing the data.

Stationary vs Mobile --The Dustduino is designed to be a stationary monitor since the design of the sensor uses temperature to regulate airflow. If the sensor is moving, the air flow is not consistent and will become an unaccounted for variable. Modifications designed to regulate the airflow are needed to overcome this issue.

High Cost vs Low Cost Sensors --The Air Quality Index of China conducted an experiment comparing a variety of low-cost particluate matter sensors including the Shenyei in Beijing, China. The study noted that “the low-cost optical sensors are not counting individual particles, but instead, counting the amount of time particles are detected by the photo diode sensor.” This can lead to quite noisy data compared to more expensive monitors. This means that the quality of near-real time data is reduced. We suggest following the scientific recommendations and viewing the data as weighted averages at hourly intervals.

What gaps are low-cost sensors like the DustDuino trying to fill?

The core problems the DustDuino is designed to fill is the inability of existing monitoring networks are to properly characterize human exposures to particulate matter due to low resolution spatio-temporal data. Today the majority of official monitoring infrastructure is focused on measuring ambient air at very high resolution and then modelling exposure over a large area with a few high quality stations. However, air quality is not as evenly distributed as the monitoring infrastructure. Effective management is particularly difficult in sprawling megacities. This means that hotspots exist that are invisible to official infrastructure and pollution monitoring devices that are accessible to the citizenry could help fill the knowledge gap.

Progam

A Brief Introduction to the Arduino IDE

The DustDuino is a customization of Arduino designed to help makers create low-power, highly customizable sensor projects. By basing the system on Arduino, the DustDuino can use “out of the box” block of code to program the device to do a number of tasks. Those blocks of code are known as libraries.

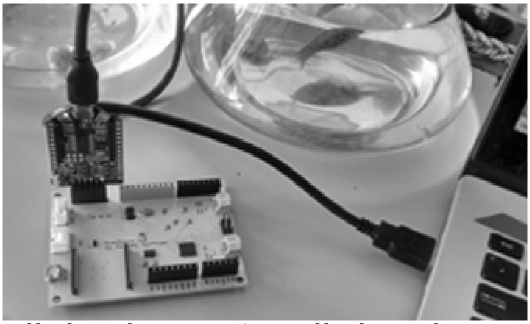
For more information and background on Arduino, visit http://www.arduino.cc/en/Guide/Introduction.

The first step in preparing the DustDuino is to download the Arduino IDE on to your computer. IDE stands for “Integrated Development Environment.” This is the software needed to directly program the board. You can download the latest version of the Arduino IDE at http://www.arduino.cc/en/Main/Software.

DOWNLOAD ARDUINO IDE

Programming the DustDuino board

The DustDuino website (DustDuino.org) has all of the custom code needed to process, transmit, and store particulate matter data. These programs are referredto as a “sketches,” and each configuration recipe listed below corresponds to a sketch available to download and install on your DustDuino. Sketches that are built inside the Arduino IDE may be saved with the “.ino” file extension.



All sketches are installed on the DustDuino using the pragrammer as pictured above.

VISIT DUSTDUINO.ORG FOR CUSTOM CODE

Configuration Recipes

Read dust levels directly on your computer

DOWNLOAD DUSTDUINO SERIAL SKETCH

1. *Attach UartSBee programmer to DustDuino*
2. *Attach UartSBee to computer using USB cable*
3. *Open the Arduino IDE*
4. *Find and select the correct serial port where your DustDuino and UartS-Bee are connected*
5. *Open the DustDuinoSerial sketch in the Arduino IDE*
6. *Compile the DustDuinoSerial sketch and upload to the DustDuino*
7. *Open the serial monitor to watch the output from the DustDuino*

To visualize data copy your dust readings and paste into a spreadsheet program, such as Microsoft Excel or Google Sheets, to save and graph your dust readings.

Log DustDuino readings on an SD card

DOWNLOAD DUSTDUINO SD SKETCH

1. *Attach UartSBee programmer to DustDuino*
2. *Attach UartSBee to computer using USB cable*
3. *Program DustDuino with the SD card sketch using the UartSBee*
4. *Attach SD card shield*
5. *Attach dust sensor*
6. *Insert SD card*
7. *Power with 5 volts*

Transmit DustDuino readings over the internet wirelessly

The DustDuino can be configured to connect with wireless networks, so that it can send data to remote servers, which can then store and visualize the data.

To use the DustDuino to connect and transmit data over a wireless network using an Xbee-compatible wireless module, some setup of the wireless module is required, but the advantage is the monitor can be set up and forgotten. However, the wireless module, not the DustDuino microcontroller chip, does the vast majority of the work when it comes to connecting to networks and sending data.

Set up a Wi-Fi Connected DustDuino

DOWNLOAD DUSTDUINO WIFLY SKETCH

1. *Attach the RN-XV to the UartSBee programmer, and connect the UartSBee programmer to your computer using a USB cable.*
2. *Launch the Arduino IDE, and locate the correct serial port.*
3. *Once the correct serial port is designated, open the serial monitor.*
4. *Select "no line ending" in the serial monitor*
5. *Enter "$$$"*
6. *RN-XV will respond with "CMD"*
7. *Change serial monitor to "carriage return"*
8. *Send "set wlan ssid <id>" (when entering your network's ID here, substitute any spaces with the character "$")*
9. *Send "set wlan phrase <phrase>"*
10. *Send "save"*
11. *Send "reboot"*

The RN-XV should return something like the following:

```
*Reboot*WiFly Ver 2.38, 12-11-2012 on RN-171
MAC Addr=##:##:##:##:##:##
Auto-Assoc ssid chan=5 mode=WPA2 SCAN OK
Joining ssid now..
*READY*
Associated!
DHCP: Start
DHCP in 3195ms, lease=86400s
IF=UP
DHCP=ON
IP=###.###.#.##:####
NM=###.###.###.#
GW=###.###.#.###
Listen on ####
```

If the red LED on your RN-XV disappears after the reboot, congratulations! You've connected to the network.

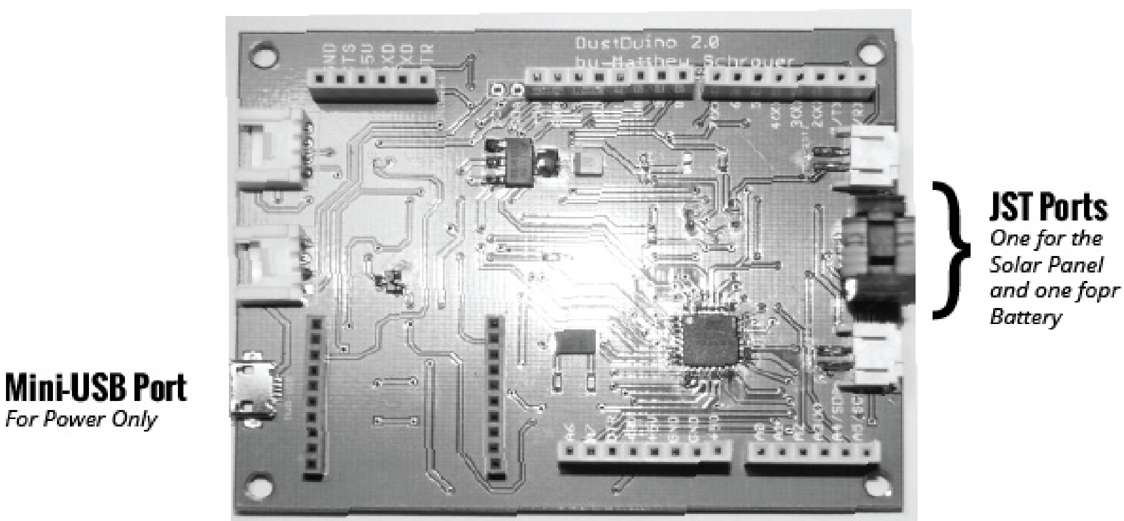
Now, time to load the actual sketch for your sensor node or what have you:

1. ***Exit the Arduino IDE***
2. ***Unplug your UartSBee from the computer***
3. ***Detach the WiFi module from the UartSbee***
4. ***Program the DustDuino board separately with the WiFi DustDuino sketch***
5. ***Attach the the WiFi module and dust sensor to the DustDuino board***
6. ***Plug DustDuino into power source***
7. ***Enjoy!***

Special note: because you are sending your DustDuino data to a server that will store your unique information, you need to make sure that the DustDuino is sending the right identifying information and credentials. Only you will know this identifing information, and the DustDuino sketches provided on-line do not contain your specific credentials. Therefore your DustDuinoWiFly sketch will need to be edited inside the Arduino IDE to include your unique information, such as a Feed ID, API Key, server address, and/or Arduino Key. See the README.md documentation or the comments inside the sketch for information on how to include your credentials in the sketch.

Transmit dust readings over a long distance using Xbee radios

DOWNLOAD DUSTDUINO XBEE SKETCH



You may wish to send particulate readings to a service like Xively, but the place where you want to collect particulate readings, and the nearest eth-ernet port might be some distance away. For this type of application, Xbee radios might be a perfect solution.

Xbee-compatible radios can connect to each other and share messages despite being one or two kilometers apart, in some cases. The range will depend on a number of factors, including the power output of the radio, the type of antenna, and the kinds of obstacles between the two radios. However, many Xbee radios can at least transmit 50 meters indoors.

Earlier, this cookbook described a recipie where an Xbee-compatible WiFi module could connect to a wireless network to send data to the Xively service. In this recipe, the DustDuino is sending its data using an Xbee radio, which connects to the Xbee radio in a base station, which is then connected to the network over an ethernet cable.

This recipe may be handy in instances where you want to have live DustDuino data from a place that is out of WiFi range (such as the outdoors), or in locations that only have a wired internet connection. Please see the solar power suppliment for additional information if interested.

Power Supply

The DustDuino is a versatile environmental monitor, but like everything else it requires power to operate. On the DustDuino board, there are two power options embedded, a mni-USB and JST connectors.

Plug you're DustDuino into a wall outlet.
You may power the DustDuino with a universal cell phone charging adapter that has a female USB connector. Most of these chargers supply 5 volts, but just to be sure, make sure to read any text on your cell charger.

Connect to a cell phone recharging battery
You can use a mobile cell phone power charger to power the DustDuino for a short perior of time away from a wall outlet. These are essentially large lithium batteries that are regulated to output 5 volts, and supply that power over a USB connection.

Keep the DustDuino connected to your Computer's USB plug
Another easy way to power the DustDuino is to keep it plugged in to your computer's USB port, but you have to keep your computer powered at all times.

Manage Data

Save and analyize data on a local computer

Make a copy of your dust readings and paste them into a spreadsheet program, such as Microsoft Excel or Google Sheets, to save and graph your data.

Setting up the connection to OpenDustMap server

DOWNLOAD OPENDUSTMAP DUSTDUINO SKETCH

1. On OpenDustMap.com click "Join this Project."

2. Enter your email address to sign up (if entering multiple sensors, use email+sensorname@gmail.com if available)

3. You will recieve a verification email.

4. Once verified, you will receive a second email with an API key

5. Go to http://opendustmap.com/#manage to configure your sensor profile.

6. Add your sensor's API key to the DustDuino sketch on line 46.

7. Install the sketch on the DustDuino board using the programmer.

Optional: If you are running your own server, replace the server name on line 47 with your server host.

Setting up your own sensor server

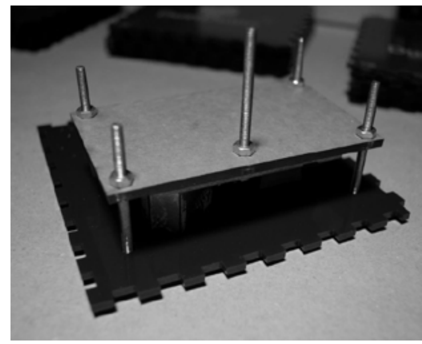
If you'd like to stream your data to the internet but don't want to publish on a public platform you can set up your own sensor server based on the same technology used on OpenDustMap. Follow the instructions at this link: https://github.com/developmentseed/sensor-rest-api

Services such as Xively also provide a free developer account so that you can store the data, and view a graph of your data on a dedicated URL. Data is updated in real time on Xively. Additionally, historical data can be requested from the Xively service, allowing you to export your particulate readings and visualize or host those results using other means. Use the DustDuino WiFly sketch to connect to Xively.

Assemble

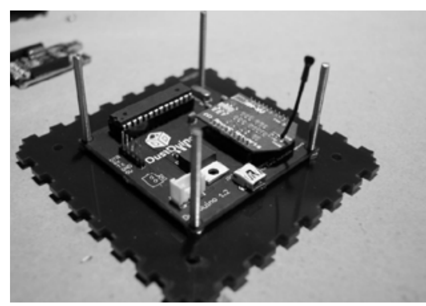
After you have programmed both the DustDuino board and the component that is handling the data storage/transmission, you can put all the pieces togeather within its cube case.

Attach Shinyei PM sensor to the “Front” Plate



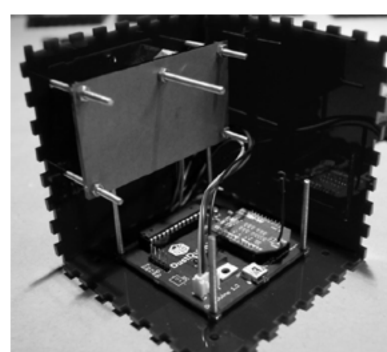
The Shinyei PPD-42 must be placed on a vertical surface, with the connector must be pointing down, and the chrome part of the sensor pointing up. To maintain this orientation, attach to the “Front” plate using a small screw and a nut. Tighten with your fingers.

Secure the DustDuino Board to the “Bottom” Plate



The DustDuino PCB should be secured to the plate labeled “botton.” There are a number of screws and o-rings included in the kit. This creates some space between the board and the bottom plate.screw down the nuts to lock the board in.

Snap the Cube Plates Togeather.



After you have attached the cable to connect the sensor to the DustDuino board you can complete the cube. Each labelled plate will fit togeather by interlocking the

Deploy

While actually deploying a configured DustDunio in the real world is the last step in the process, understanding the conditions of where the monitor will be places should be among your first thoughts. What you are trying to measure will help guide you through the type of connectivity, power, and case you need to help collect the data you need.

The clearest distinction for the deployment is if the dustdunio is being placed outside (exposed to weather) or indoors.

Indoors

The DustDuino kit comes with a laser cut acrylic cube that has a number of holes for mounting the Shenyei sensor and connecting power cables. This means that it is not suitable for outdoor set up without some modifications. Keeping the unit indoors will ensure a long-life for your DustDuino but experimentation is a key part of growing the potential of this open hardware.

The other benefit of an indoor deployment is it is simpler to choose the appropriate connectivity option. Wi-fi is the preferred choice for those who want to watch their DustDuino readings from anywhere with an open WiFi network. If internet isn’t an option, the SD card is a great choice for data logging but the addition of real time clock is required to record the time of each reading.

Outdoors

Since the acrylic case that comes as part of the kit is not weatherized, we suggest purchasing a water resistant outdoor enclosure and modifying it to enable the Shenyei sensor’s air intake ports are open.

NOTE: The Shenyei sensor must be kept in a vertical (upright position) to function properly!

Outdoor deployments can be supported with a variety of data handling and power options. The first consideration for an outdoor deployment should be power. Utilizing a solar panel and/or rechargeable battery greatly extends the range of possible deployments but may not provide as stable a power supply as a plug. Please see the solar power suppliment for details.

For scenarios where a number of DustDuinos are being deployed, transmitting data with Xbee-compatible radios can enable communications of one or two kilometers. The range will depend on a number of factors, including the power output of the radio, the type of antenna, and the kinds of obstacles between the two radios.

If connectivity isn't possible, using a SD card shield will enable you to collect data but the addition of real time clock is required to record the time of each reading.

Stories and Strategies.

Schools: Children are among the most vulnerable populations to poor air quality and childhood impacts often have a lifetime of consequences. In a recent health study by the University of Southern California showed that over a ~20 year period of improving air quality (PM and other pollutants) was correlated with increased lung growth/function in children. Healthier lungs leads to healthier minds and bodies and the environment children live in can effect that last a lifetime

Try collaborating with primary, high school, or university teachers to set up a DustDuino in the classroom or in a covered outdoor space. Building a deploying a monitor can be done as part of a science or technology class, or science club. Working with multiple schools is ideal but requires partnership with teachers in the city so some is outreach needed. If organized, students, parents and teachers will be able to see how their school compares with other schools in terms of its air particle levels, and how those levels compare to the city averages as collected by the government.

Hot Spots – Roadsides, Construction Sites, and more: The way PM2.5 is currently tracked is via high quality monitoring stations managed by governments and major research consortia. These expensive stations are few and far between and specialize in capturing air quality at very high resolution and then modeling exposure over a large area.

The issue with this approach is that air pollution is uneven and hotspots exist that are invisible to the official pollution monitoring infrastructure. Try setting up a DustDuino in an area where you suspect there is higher than average particulate matter and see what the results are. When doing this it is always important to identify reference monitors for calibrating the sensors.

Working with interested citizens to organize citizen scientist sensor meetup is a great way to collect information on important areas. The Public Labratory (publiclac.org) provides tools and community to help publish citizen science research and find like minded groups.

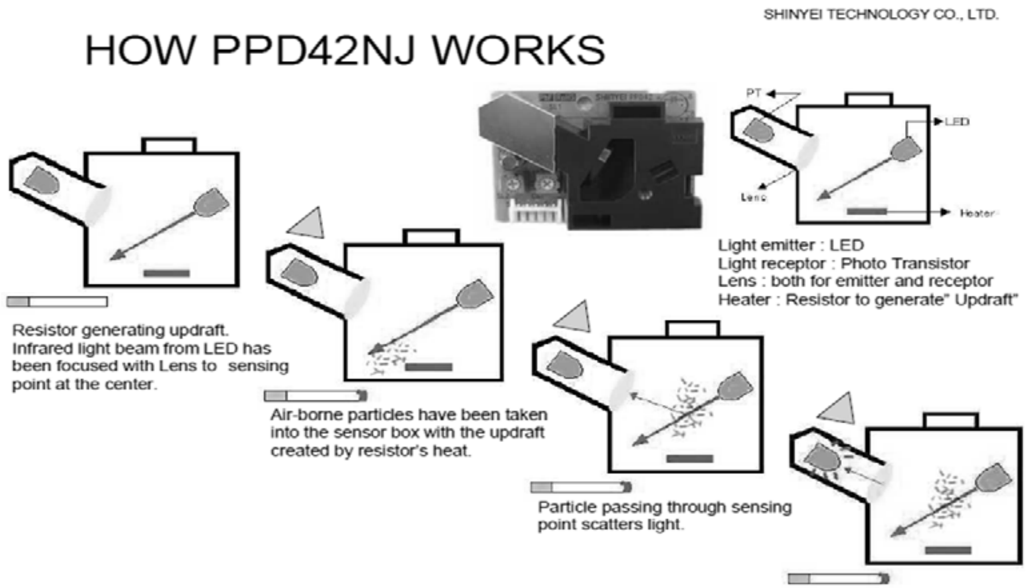
Understanding the readings

As mentioned, the dust sensor consists of an infrared LED, a lens to spread the diode's infrared light, and a photodiode, among other components. The operation of the dust sensor is based on a concept called Mie theory or Mie scattering, named after the German physicist Gustav Mie, who studied how spherical objects can scatter electromagnetic waves.

Mie scattering can be observed with your own eyes each day, during a normal sunrise or sunset. As the sun sets, it appears to get redder. But actually, it's not that the sun is getting more red, but that large particles in the atmosphere such as water droplets, aerosols, and dust are scattering the green, yellow, and blue light from the sun in other directions. Mostly, light in the red part of the visual spectrum remains, which is why the sun seems so red at night and in early morning.

Interesting fact: Mie theory can partly explain the old proverb "red sky at night, sailor's delight; red sky in morning, sailor's warning." A red sky at sunset could indicate water vapor and dust in the west, meaning a calm, high-pressure system is coming in. But a red sky in the east, where the sun rises, could mean the high-pressure system has departed and a stormy, low-pressure system is following behind it.

Just as the sun's light is scattered by particles, so too is the dust sensor's



infrared light. The photodiode will pick up varying levels of infrared light, depending on how much is scattered by airborne dust. So in reality, the photodiode is the real sensor for the DustDuino. This photodiode is a semiconductor device that will convert light into an electrical current, and a small chip on the back of the optical dust sensor will measure the current and create digital output to express how much dust the sensor is picking up.

One neat trick about Mie scattering is the wavelength of light that is scattered by particles will differ depending on the size of the particle. This can be very handy if your goal is to not only to measure the quantity of particles, but also measure the size of those particles (especially if it's the fine PM 2.5 particles you're concerned about). The Shinyei PPD42NS will read its photodiode and communicate its readings using two different "channels," designated as P1 and P2 in the sensor documentation.

Practically speaking, each channel is nothing more than an individual wire whose voltage will rise or fall depending on the type and quantity of particles. The PPD42NS still is a "digital" sensor, however, where 5 volts is "on" or "1" and 0 volts is "off" or "0." Some digital sensors will oscillate between on and off very quickly to generate a string of 1s and 0s (binary) to communicate a message, maybe up to once every 0.0000025 seconds (that's 400kHz). But the Shinyei is a different type of digital sensor. Instead, its message always is 30 seconds long, and the length of time (a.k.a. period) that a channel is sending 5 volts indicates the quantity of dust.

Acknowledgements

The DustDuino has been developed, supported, and extended by a multitute of partners and people who have worked togeather to help change the way people around the world understand and communicate about the air they breathe.

We'd personally like to thank Kat Austen, James Fahn, Gustavo Falieros, Ricardo Guima, Mark Frohardt, Fergus Pitts, Edmund Seto, and Amy Wilson-Chapman for their inspiration, support, and experimentation.

Organizationally, the DustDuino would not have been possible without collaboration with Development Seed, the Earth Journalism Network, Frontline SMS, the GroundTruth Initiative, InfoAmazonia, Makerspace Urbana, the Press Institute of Mongolia, Public Labratory, and SeeedStudio.

Development of the DustDuino has been supported by the Internews Center for Innovation and Learning and Feedback Labs.

Dedicated to the end of unhealthy air. Onwards and upwards!