

# Air Quality Sensor

Steak Electronics

2018 (revised 2019)

## Contents

### 1 Overview

To know when I need to clean my room. I usually vacuum every two or three weeks, but sometimes am lazy and I need a reminder. The dust sensor will be more motivation to get me to clean, when I see the dust levels increase.

<sup>1</sup>

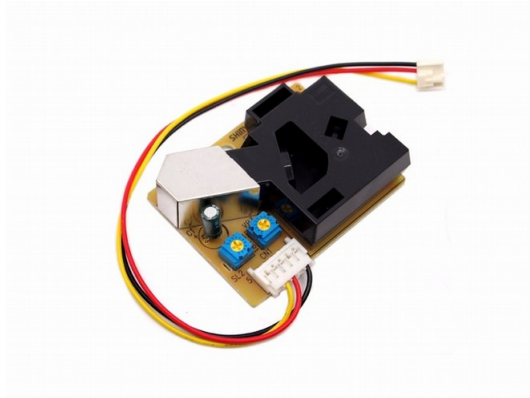


Figure 1: Shinyei Dust sensor PPD42

## 2 Work Log

### 2.1 Parts List

- Shinyei PPD42NS Air Quality Sensor

---

<sup>1</sup>Technically, I also get a rash from excess dust, which is another great motivator.

- MSP-EXP430G2553 V1.5
- 5V adapter (required as MSP is 3.3V only)
- SD Card adapter
- Ethernet Module
- Energia version 0101E0012 (or later, possibly)

## 2.2 Beginning

Here are some things I learned working on this project.

**IDE** I'm using Energia from [energia.nu](http://energia.nu) which is an Arduino IDE clone for TI Launchpads. This will allow for rapid development. And ease me into the TI platform.

**MSP-EXP430G2 V1.5** There are different version of this. My particular board is the and as explained here: [http://energia.nu/pin-maps/guide\\_msp430g2launchpad/](http://energia.nu/pin-maps/guide_msp430g2launchpad/) and [http://energia.nu/pin-maps/guide\\_msp430g2launchpad/](http://energia.nu/pin-maps/guide_msp430g2launchpad/) You need to rotate the UART jumpers to get UART to display correctly. Oddly enough you can choose a different chip in the boards list and serial will print out right... But switch the jumpers and the correct board will work with UART. Otherwise, it does not work out of the box for serial.print. Though the blink example sketch works.

**Dust Sensor** The code for the dust sensor is found online easily. The pinout is tricky as the colours of the wire are nonsense, but the pinout seems to be the same for all sensors, and is: PIN 1 (closest to black box) GND, PIN 3, VDD (5+V), PIN 4 output A. there is also an output B for different readings (I think size) of dust. I'm not that particular (no pun intended) about my dust so I will go with the one most people are using.

**3.3V only on TI** The TI takes input of USB but only outputs 3.3V. Fail. I'll through a 5V PSU on the board. It won't be connected to USB for its use anyways.

**POW function** There is a pow function (power exponent) in arduino. In TI, I changed it to powf, and included math.h.

I'm not sure if my change was correct.

EDIT: looks like there is a LED tied to pin 14. I'm moving to pin 13. I meant to use 13 earlier but accidentally used the wrong pin and kept using it. Oops.

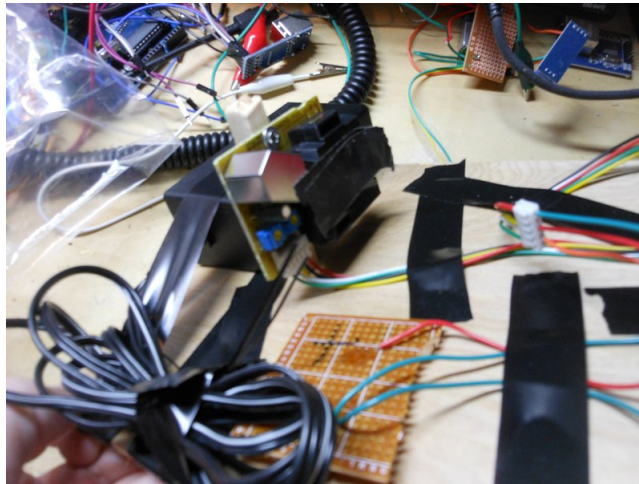
EDIT 2: Actually, I was reading from the wrong pin in software. Maybe I will leave it on the LED pin as you can see when the dust is firing off the sensor. Hm.... Neat.

EDIT 3: Yes, so I had to use powf instead of pow, and it works. The accidental incorrect pin was a bonus as it allows me to see visually how often the sensor is going off. Over time, I should have a vague grasp of the dust levels just looking at the light. Actually, I won't. Nevermind. The dust sensor reading, is too irregular for that.

### 2.3 Work Log 08/2018

I've put everything on a single piece of plywood, and need to do some more work on the code. I've decided to forgo the SD card, as I don't want to deal with reading a 2MB sd card with the buffer provided by the SRAM. I'm not quite sure how to manage that, and whether it's possible to read such large data files. I did some quick research and did not find what I wanted. It is likely possible, but let's do something simpler. Instead, I will use the EEPROM to store the last 5 minutes of data or so, and then have the server read the data every five minutes. This keeps the client simple, and puts the burden of complexity on the server. In addition, I don't want to read SD cards manually, as that is cumbersome, though possibly scripts could be made.

The PPD42 was made sure to be put vertically.



### 2.4 Work Log 12/2018

I have decided to change how I do this slightly. Instead of an SD card, I will connect on the LAN and use thingspeak from a locally hosted instance (and deployed with docker, possibly) or some other aggregating server to pull the

data. I'll also make a shield, to simplify deployment. That might not be necessary, but I can make a shield in a few hours, and pcbs are cheap. These options such as thingspeak sometimes have graphing / plotting included.

I'm going to use this library: <https://github.com/reaper7/EtherEncLib/releases> so git clone that, then git tags -l git checkout tags/v0.4.2 to get the latest release (or a newer one if possible). The pinout for the ENC is viewable at the figure to the right. This is from the 43oh.com forum.

2

---

<sup>2</sup>You will see later, that this library turns out to be not developed enough, and results in overall failure for this project. Next time, use a well vetted library!