Standalone: Sharp Dust Sensor 82

1 Dec 2012 | Arduino Modules · Arduino-AirQuality
Tags: Arduino · Arduino Fio · Arduino Uno · Dust · Sensor · Sharp

We based our work on the excellent post over here but contrary to the author, we did not try out multiple sensors and we are using an Arduino Fio.

In this tutorial, we'll focus on how to get your Sharp Optical Dust Sensor to work and what to watch out for.

In addition, we will try to provide European shop references and prices in € where possible.

Components

Arduino Fio (BoxTec) – ~21.57 €
FTDI Basic Breakout 3.3V (BoxTec) – ~14.52 €
Sharp Optical Dust Sensor (Robotshop) #GP2Y1010AU0F – 10.75 € – Datasheet
6-pin TE 1.5mm pitch connector cable (DigiKey) #A100196-ND – 1.2 €
220 uF Capacitor
150 Ω Resistor
Breadboard
M/M jumper cables

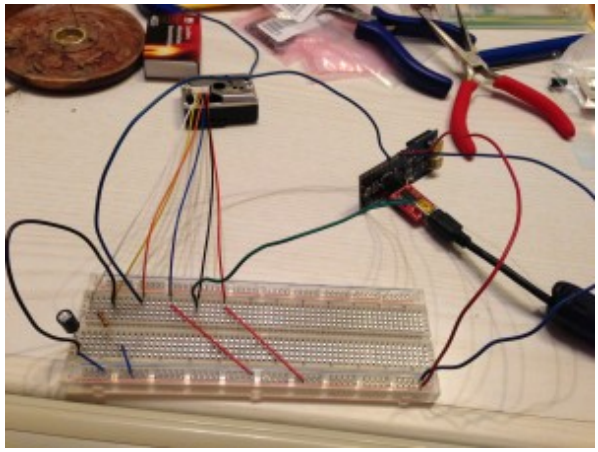
Setup

Prerequisites

Able to upload sketches via the Arduino IDE to your Arduino

In the figure below you can see the overall setup using an Arduino Fio and the Sharp Dust sensor using a classical Breadboard.

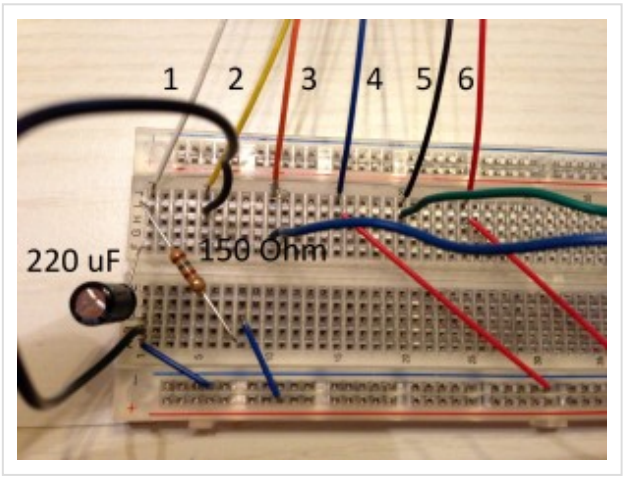
Even though you may use an Xbee shield to load your sketches wirelessly, we use a FTDI breakout board here to connect to our PC.



Overview of the Dust Sensor and the Arduino Fio

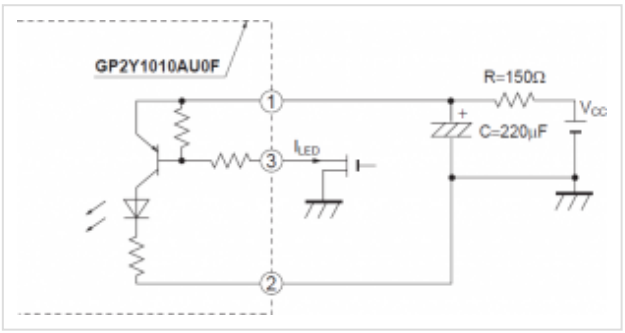
1. Hook up the sensor using the 6 pins in the following way (see Figure below):

Pin numbering on the Sensor

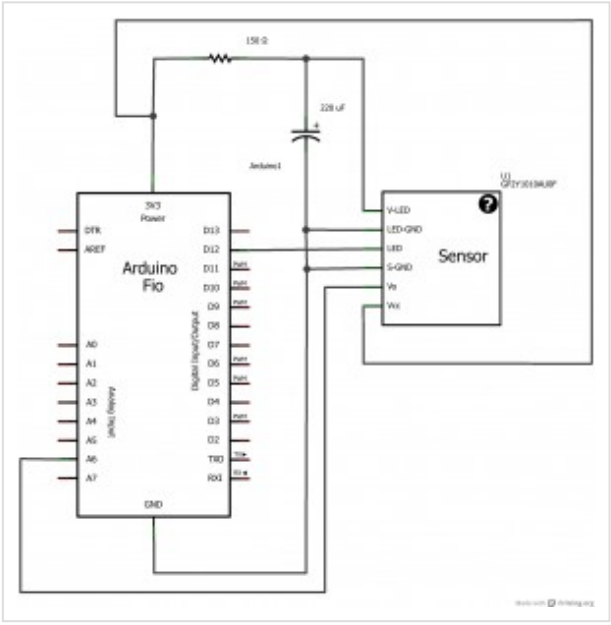


Close-up of breadboard with Dust Sensor and Arduino

You may use the following schema from Sharp and our own drawing to help you in the task:



Official schema from Sharp



Fritzing schema of the Setup using an Arduino Fio

Pins Assignments

Sharp Dust Sensor	Attached To
1 (V-LED)	3.3V Pin (150 Ohm in between)
2 (LED-GND)	GND Pin
3 (LED)	Digital Pin 12
4 (S-GND)	GND Pin
5 (Vo)	Analog Pin A6
6 (Vcc)	3.3V Pin (Direct)

Code

All updated code and schematics can be found on our github dust module project page.

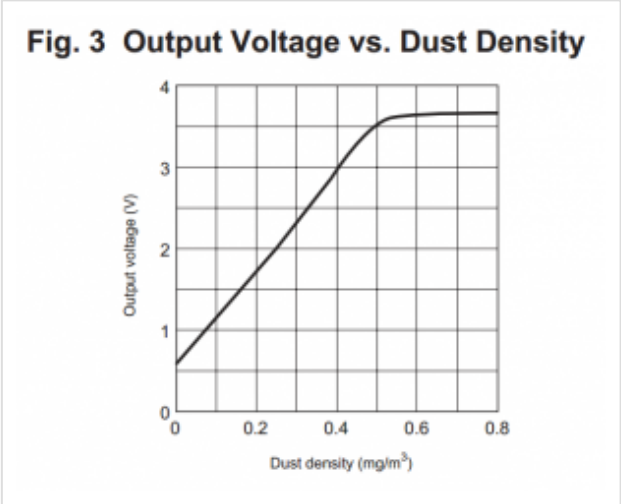
Before launching the code, there are couple of points which are important.

1. How to interpret the output signal

In the figure below taken from the datasheet, you can see that the Dust density grows linearly with respect to the output voltage (Vo).

In line 52 of the code, we implemented the formula of the linear regression that approximatively follows this curve (courtesy of Chris Nafis).

This will allow us to map output voltages to Dust densities in mg/m^3.



Linear range of the Sharp sensor

In addition, on an Arduino, any analog pin will map voltages between integer values from 0-1023 which can be mapped back to a "real" voltage value.

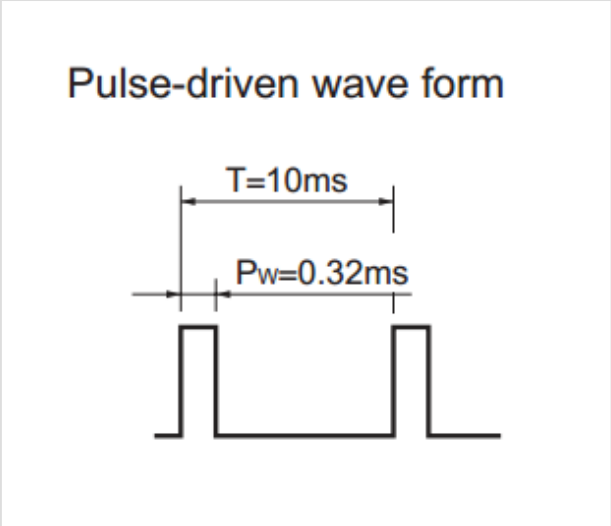
For the Fio, we therefore multiply the analog reading by 3.3/1024.0 and for the Uno, you will want to multiply the reading by 5.0/1024.0.

Important: Be sure to add the trailing zero in these calculations, because if you do not put at least one, you will end up with a nasty bug where all your results will be 0. (Hint: integer division in C)

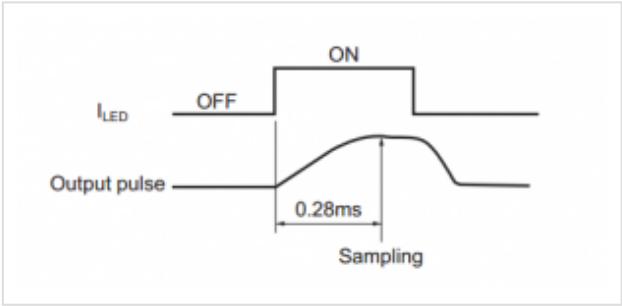
2. Sampling times

According to the datasheet, we need to switch on the internal LED and wait for 280 µs (microsecond) before measuring the output signal and the duration of the whole excitation pulse should be 320 µs.

We therefore pause for another 40 µs before switching off the LED again.



Pulse-driven wave form



Sampling strategy

For convenienve, we provide a code snippet here:

```
/*
 Standalone Sketch to use with a Arduino Fio and a
 Sharp Optical Dust Sensor GP2Y1010AU0F

 Blog: http://arduinodev.woofex.net/2012/12/01/standalone-sharp-d
 Code: https://github.com/Trefex/arduino-airquality/

 For Pin connections, please check the Blog or the github project
 Authors: Cyrille Médard de Chardon (serialC), Christophe Trefois
 Changelog:
   2012-Dec-01:  Cleaned up code

 This work is licensed under the
 Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unport
 To view a copy of this license, visit http://creativecommons.org
 or send a letter to Creative Commons, 444 Castro Street, Suite 9
 Mountain View, California, 94041, USA.
*/

int measurePin = 6;
int ledPower = 12;

int samplingTime = 280;
int deltaTime = 40;
int sleepTime = 9680;

float voMeasured = 0;
float calcVoltage = 0;
float dustDensity = 0;

void setup(){
  Serial.begin(9600);
  pinMode(ledPower,OUTPUT);
}

void loop(){
  digitalWrite(ledPower,LOW); // power on the LED
  delayMicroseconds(samplingTime);

  voMeasured = analogRead(measurePin); // read the dust value

  delayMicroseconds(deltaTime);
  digitalWrite(ledPower,HIGH); // turn the LED off
  delayMicroseconds(sleepTime);

  // 0 - 3.3V mapped to 0 - 1023 integer values
  // recover voltage
  calcVoltage = voMeasured * (3.3 / 1024);

  // linear eqaution taken from http://www.howmuchsnow.com/arduir
```