Application Guide for
Sharp GP2Y1026AU0F Dust Sensor
(Arduino Uno/Mega)
Table of Contents

1. OVERVIEW ..............................................................................................................................................3
2. HOW DO SHARP DUST SENSORS WORK? ...............................................................................................3
3. KEY CONCEPTS AND TERMINOLOGY .....................................................................................................4
4. COMPARING GP2Y1010AU0F AND GP2Y1026AU0F ................................................................................4
5. AIRFLOW DESIGN RECOMMENDATION ...............................................................................................5
6. GP2Y1026AU0F CONNECTOR CABLE ....................................................................................................5
7. USING GP2Y1026AU0F WITH ARDUINO ...............................................................................................6
8. ARDUINO SKETCH (EXAMPLE/DEMO SOURCE CODE) .........................................................................7
9. EXAMPLE SERIAL MONITOR OUTPUT ..................................................................................................10
10. LINEAR COEFFICIENT CALIBRATION ..................................................................................................10
11. REFERENCES ............................................................................................................................................11
1. Overview

Designed for use in commercial air purifiers, Sharp Corporation dust sensors have proven popular over the years with electronics hobbyists and researchers interested in building air quality and pollution monitors. In particular, Sharp’s GP2Y1010AU0F dust sensor model is the subject of many electronics projects due to its low cost and ease of availability through components distributors. Since the introduction of GP2Y1010AU0F, Sharp has released newer models such as GP2Y1014AU0F and GP2Y1026AU0F which offer improvements in design and function. Sharp GP2Y1026AU0F is a digital UART sensor featuring a built-in microcomputer (MCU) which handles some of the processing you would normally need to provide yourself such as offset calibration, temperature correction, LED pulsing, and averaging of data values to minimize noise.

2. How do Sharp dust sensors work?

Sharp dust sensors operate on the principle of light scattering. A photo-detector and LED emitter oppose each other at an angle within the rectangular package of the sensor which has a dust through hole on either side. Air containing dust particles flows into the sensor chamber and causes the light from the LED emitter to be scattered towards the photo-detector. The dust sensor outputs a voltage value which varies according to the intensity of light scattered which in turn corresponds to the level of dust in the sensor chamber. The actual dust density (or dust concentration) can then be calculated from the output voltage value using a linear relation.
3. **Key concepts and terminology**

<table>
<thead>
<tr>
<th><strong>Dust Size</strong></th>
<th><strong>PM(_{2.5})</strong></th>
<th><strong>PM(_{10})</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>is the <strong>diameter</strong> of a dust particle which is measured in <strong>microns</strong> (µm) which is the same as a micrometre.</td>
<td>is <strong>particulate matter</strong> less than 2.5 microns in diameter such as combustion particles. Also called <strong>fine particles</strong> which are hazardous to human health.</td>
<td>is particulate matter less than <strong>10 microns</strong> in diameter such as dust, pollen, and mold.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Dust Density</strong> (or <strong>Dust Concentration</strong>)</th>
<th><strong>Output Voltage</strong></th>
<th><strong>Sensitivity</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>is measured in units of mass / volume such as µg/m(^3) or mg/m(^3).</td>
<td>Sharp dust sensors generally output a voltage which increases with dust density.</td>
<td>is the change in output voltage when the dust density changes by 100 µg/m(^3).</td>
</tr>
</tbody>
</table>

4. **Comparing GP2Y1010AU0F and GP2Y1026AU0F**

<table>
<thead>
<tr>
<th>Specification Parameter</th>
<th>GP2Y1010AU0F</th>
<th>GP2Y1026AU0F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Light emitting element</td>
<td>LED</td>
<td>LED</td>
</tr>
<tr>
<td>Min. detectable dust size</td>
<td>0.5µm</td>
<td>0.5µm</td>
</tr>
<tr>
<td>Dust density sensing range</td>
<td>0 to 600 µg/m(^3)</td>
<td>0 to 1000 µg/m(^3)</td>
</tr>
<tr>
<td>Sensing time</td>
<td>&lt; 1 second</td>
<td>1 second</td>
</tr>
<tr>
<td>Sensitivity / Accuracy</td>
<td>0.5V ± 0.15V per 100 µg/m(^3) change, Accuracy ±30%</td>
<td>0.35V ± 0.06V per 100 µg/m(^3) change, Accuracy ±15%</td>
</tr>
<tr>
<td>Output interface</td>
<td>Analog</td>
<td>Digital (UART)</td>
</tr>
<tr>
<td>Sensing of PM2.5 and PM10 separately</td>
<td>No (measures total dust density only)</td>
<td>No (measures total dust density only)</td>
</tr>
<tr>
<td>LED pulse drive circuit</td>
<td>No (requires external circuit)</td>
<td>Yes</td>
</tr>
<tr>
<td>Offset correction</td>
<td>No (must subtract voltage level at zero dust)</td>
<td>Yes</td>
</tr>
<tr>
<td>Temperature correction</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Averaging of data readings</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

You can find the detailed specifications for both dust sensor models [here](http://www.socle-tech.com).
5. **Airflow design recommendation**

The GP2Y1026AU0F dust sensor does not include any built-in fan or internal resistor heating element to supply airflow to the sensor (and note that the specification does not indicate any airflow is required). While this can be advantageous in terms of providing flexibility in sensor orientation and positioning, it is recommended in practice to design in at least some airflow into your application so that the sensor can react quicker to changes in dust concentration and thereby provide more accurate results. For example, if your application is an air purifier or an air conditioner, you might be able to use the system’s natural airflow for this purpose. Otherwise, you can look at adding a mini 20x20mm or 25x25mm external fan. It is also advisable to use a coarse mesh filter in your application design to prevent large dust particles from entering the Sharp sensor and accumulating inside. While a light vacuum can be used to remove large dust pieces which may be attached inside the sensor, your application design may preclude such kind of maintenance access to the sensor.

6. **GP2Y1026AU0F connector cable**

Sharp GP2Y1026AU0F uses the same type of 6-pin connector as the GP2Y1010AU0F model. Earlier iterations of the GP2Y1010AU0F sensor used the JST (Japan Solderless Terminal) connector S6B-ZR-SM4A. But all newer production versions of both sensors now use JCTC connector 11501W90-6P-S-HF which should be compatible with the old JST connector. What this all means is if you have an old cable harness for GP2Y1010AU0F, the cable should work just fine with GP2Y1026AU0F as well. The following table and figure shows the pin assignments for GP2Y1026AU0F.

<table>
<thead>
<tr>
<th>Pin number</th>
<th>Pin name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>Vcc</td>
</tr>
<tr>
<td>3</td>
<td>NC</td>
</tr>
<tr>
<td>4</td>
<td>NC</td>
</tr>
<tr>
<td>5</td>
<td>RxD</td>
</tr>
<tr>
<td>6</td>
<td>TxD</td>
</tr>
</tbody>
</table>
7. Using GP2Y1026AU0F with Arduino

GP2Y1026AU0F includes a built-in **LED pulse drive circuit**. This eliminates the need for an external circuit and allows you to connect the Sharp sensor directly to your Arduino board. The pulse drive circuit also helps to extend the lifetime of the sensor under continuous operation. Although the Sharp sensor has a **RxD** terminal, you don’t need to send it any commands in order to begin receiving serial UART data from the sensor’s **TxD** terminal. Simply connect the sensor to your Arduino board as shown in the figure below. This example uses Arduino **Uno** but it should work the same with Arduino **Mega 2560**.

![Arduino Uno diagram](http://www.socle-tech.com)
Application Guide for Sharp GP2Y1026AU0F Dust Sensor

In the source code to follow, you will see that the regular hardware serial port is used only for printing output to the Arduino **Serial Monitor** at 9600 baud. The actual UART data from the dust sensor’s `TxD` terminal is received through a **software serial port** connected to Arduino **Pin 10** running at 2400 baud.

Instead of using a software serial port, you can use the regular hardware serial port (Arduino RX0 pin instead of Pin 10) but make sure the sensor TxD terminal is disconnected during uploading to Arduino or else the upload will fail. Another option is to use Serial1 on Arduino Mega 2560 for communicating with the dust sensor.

The GP2Y1026AU0F dust sensor outputs 1 byte every 10ms through its TxD terminal. According to the specification, it produces a 7-byte data frame as follows:

<table>
<thead>
<tr>
<th>Start</th>
<th>Vout (H)</th>
<th>Vout (L)</th>
<th>Vref (H)</th>
<th>Vref (L)</th>
<th>Checksum</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xAA</td>
<td>Ex: 0x01</td>
<td>Ex: 0x3b</td>
<td>Ex: 0x00</td>
<td>Ex: 0x7a</td>
<td>Ex: 0xb5</td>
<td>0xff</td>
</tr>
</tbody>
</table>

Thus it takes 70ms to output a data frame which is equivalent to **14 frames per second**. However, the dust sensor MCU applies **averaging** to the data readings in order to reduce noise and so the output voltage values only change about **once per second**. This means that you will see a lot of consecutive frames with an identical set of bytes (and voltage).

8. **Arduino Sketch (example/demo source code)**

**Disclaimer:** The following code is for demonstration/example purposes only. It is not intended for production or commercial usage.

The example code shows how to read in a data frame, verify the checksum, and calculate the output voltage. A linear calibration coefficient is then applied which converts the output voltage into a dust density value.

The code is listed in two sections, Part A and Part B, which you must combine into the same Arduino sketch file (`sharp_gp2y1026au0f_v2.ino`).
Source Code Listing – sharp_gp2y1026au0f_v2.ino – PART A

#include <SoftwareSerial.h>

#define rxPin 10
#define txPin 11

// Use software serial port for communicating with GP2Y1026.
SoftwareSerial mySerial(rxPin, txPin);

// Helper function to read data from software serial port.
int getSerial() {
  while (!mySerial.available()) {} // Wait for data
  return mySerial.read();
}

void setup() {
  // Start the hardware serial port for the serial monitor.
  Serial.begin(9600);

  // Start the software serial port for receiving data from GP2Y1026.
  mySerial.begin(2400);

  // Wait a second for startup.
  delay(1000);
}

void loop() {
  static int frame[7];
  static int dataStart, VoutH, VoutL, VrefH, VrefL, checksum, dataEnd;

  // Look for end of frame.
  if (getSerial() != 0xff) { return; }

  // Read in next frame.
  for(int i = 0; i < 7; i++) {
    frame[i] = getSerial();
  }

Source Code Listing cont’d – sharp_gp2y1026au0f_v2.ino – PART B

```cpp
// Assign values.
dataStart = frame[0];
VoutH = frame[1];
VoutL = frame[2];
VrefH = frame[3];
VrefL = frame[4];
checksum = frame[5];
dataEnd = frame[6];

// Check the start and end of frame.
if ( dataStart != 0xaa || dataEnd != 0xff ) { return; }

// Verify the checksum.
int testSum = VoutH + VoutL + VrefH + VrefL;
if ( testSum != checksum ) { return; }

// Print the data values.
Serial.print("dataStart=");
Serial.print(dataStart);
Serial.print(",");
Serial.print("VoutH=");
Serial.print(VoutH);
Serial.print(",");
Serial.print("VoutL=");
Serial.print(VoutL);
Serial.print(",");
Serial.print("VrefH=");
Serial.print(VrefH);
Serial.print(",");
Serial.print("VrefL=");
Serial.print(VrefL);
Serial.print(",");
Serial.print("Checksum=");
Serial.print(checksum);
Serial.print(",");
Serial.print("dataEnd=");
Serial.print(dataEnd);
Serial.print(",");

// Calculate Vout.
float Vout = (VoutH * 256 + VoutL) / 1024.0 * 5.0;
Serial.print("Vout=");
Serial.print(Vout * 1000.0);
Serial.print("mV, ");

// Calculate "a" coefficient beforehand by using a reference dust monitor.
// The coefficient will be different depending on the reference dust monitor used
// and your specific application conditions.
float a = 100 / 0.35;

// Calculate dust density.
float dustDensity = a * Vout;
Serial.print("dustDensity=");
Serial.print(dustDensity);
Serial.print("ug/m3");
Serial.println(" ");
```

// END PROGRAM
9. Example Serial Monitor Output

If you are switching between different Arduino boards, make sure you go to the Tools menu and set Board and Port appropriately before compiling or uploading the code. The following screen capture shows example output from the GP2Y1026AU0F dust sensor as displayed on the Serial Monitor.

![Serial Monitor Output](image)

10. Linear Coefficient Calibration

The fixed “a” coefficient in the example code is used to convert output voltage from the sensor into a dust density value. In the code, the coefficient value is set to approximate the following graph from the GP2Y1026AU0F specification.
For practical applications, you should determine a suitable value for the coefficient yourself by measuring different dust density levels and comparing the results against higher accuracy reference equipment such as a TSI DUSTRACK II Aerosol Monitor.

11. References

- [https://pdfs.semanticscholar.org/d641/19160b9effd57448b44d39d5ac5468ed0eff.pdf](https://pdfs.semanticscholar.org/d641/19160b9effd57448b44d39d5ac5468ed0eff.pdf)
- [http://eereview.com/article/gp2y1014au0f-pm25-optical-dust-density-sensor](http://eereview.com/article/gp2y1014au0f-pm25-optical-dust-density-sensor)

**Socle Technology Corporation** is a subsidiary of Foxconn and the official supplier of Sharp optoelectronic components and sensors in North America and China.