# Sharp GP2Y1010AU0F Optical Dust Sensor
# Hook-Up Guide

## Introduction

The Sharp GP2Y1010AU0F optical dust sensor is a small (46.0 × 30.0 × 17.6 mm) lightweight (~15g) air quality sensor containing an infra-red LED and a phototransistor capable of detecting air-borne dust and smoke. The sensor is available from SparkFun and their distributors (Product ID: COM-09689) and has an approximate sensitivity of 0.5V/0.1mg/m$^3$.

Sharp have produced a detailed datasheet for the sensor:
http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y1010au_e.pdf
together with a rather useful application note:
http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y1010au_appl_e.pdf

## Components

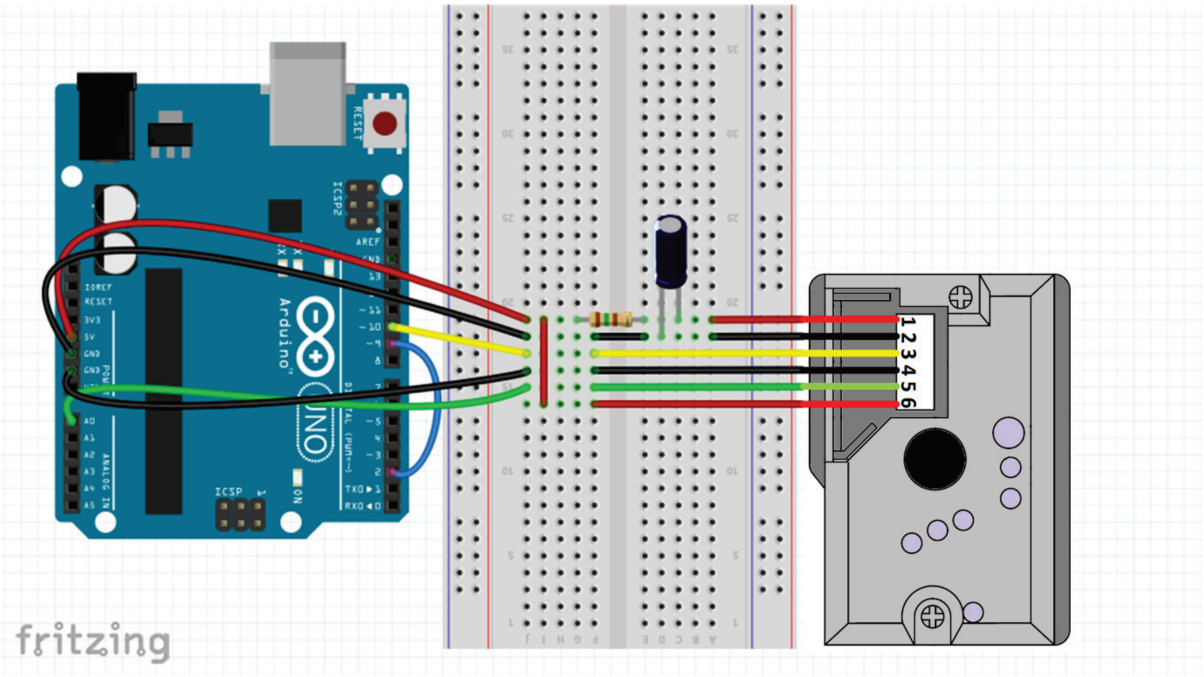
You will need:

- Arduino Uno or equivalent (e.g. the SparkFun RedBoard – product ID: DEV-12757)
- Sharp GP2Y1010AU0F optical dust sensor (SparkFun ID: COM-09689)
- 6-Way JST ZHR-6 connector housing (SparkFun ID: PRT-09690)
- >= 6 JST SZH-002T-P0.5 crimp pins (SparkFun ID: PRT-09728)
- 150R Resistor
- 220µF Electrolytic Capacitor
- Small Breadboard or Veroboard
- Hook-up wires

## Wiring

The tricky bit is crimping the six SZH-002T-P0.5 crimp pins onto suitable wires to interface to the Arduino. JST do of course offer a dedicated crimp tool for the job but it is expensive. The crimp pins are very small but can be crimped using a regular crimp tool so long as care is taken. Wouldn’t it be great if the nice people at SparkFun could offer a ready-crimped cable as a new product?!

Connect the Arduino and dust sensor as follows:



Connect Arduino 5V:

- directly to pin 6 of the dust sensor
- to pin 1 of the dust sensor via a 150R resistor

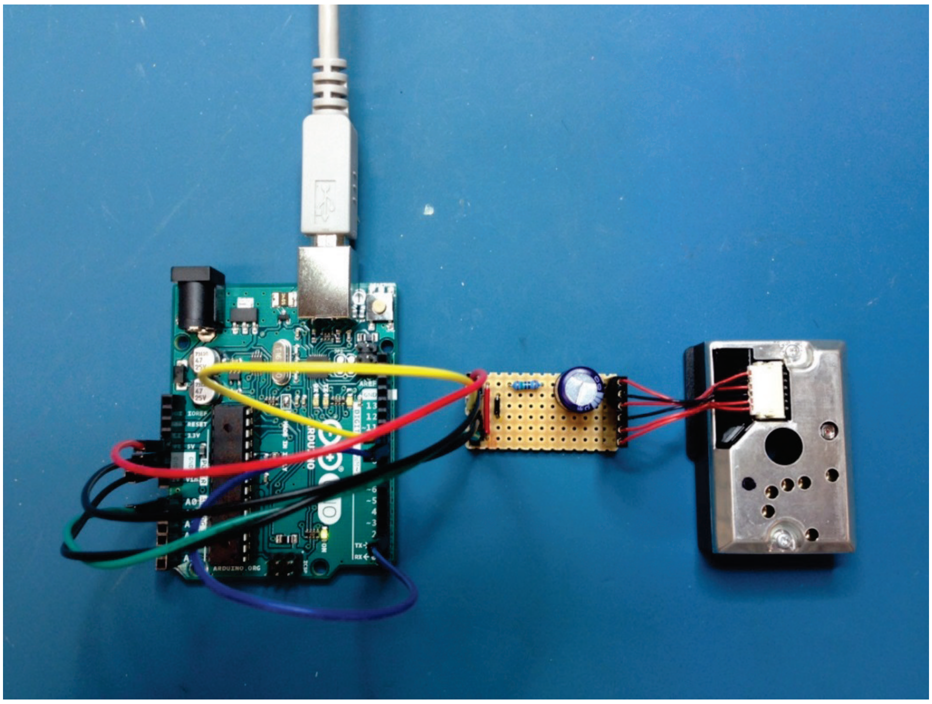
Connect Arduino GND to pins 2 and 4 of the dust sensor

Connect Arduino A0 to pin 5 of the dust sensor

Connect Arduino D10 to pin 3 of the dust sensor

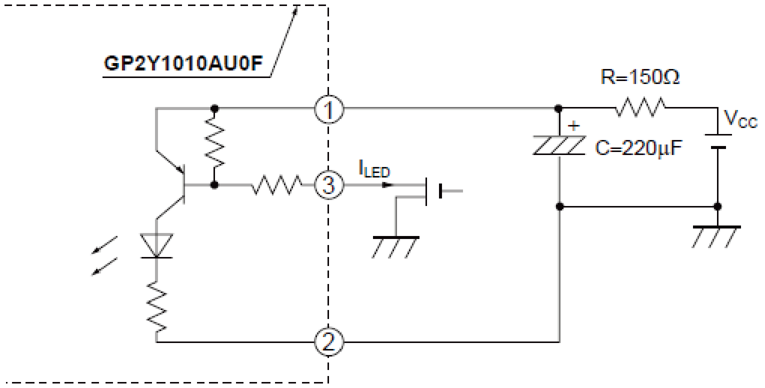
Connect a 220µF capacitor between pin 1 (+ve) and pin 2 (GND) of the dust sensor

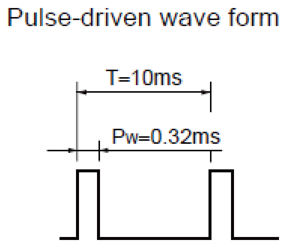
Connect Arduino D9 to D2



## LED Pulses

The GP2Y1010AU0F data sheet calls for the sensor LED (pin 3) to be pulsed on for 0.32msec every 10msec (100Hz). The datasheet schematic includes an additional transistor (FET) between the controller and pin 3, so in reality pin 3 needs to be pulled LOW by the Arduino to turn the LED on.





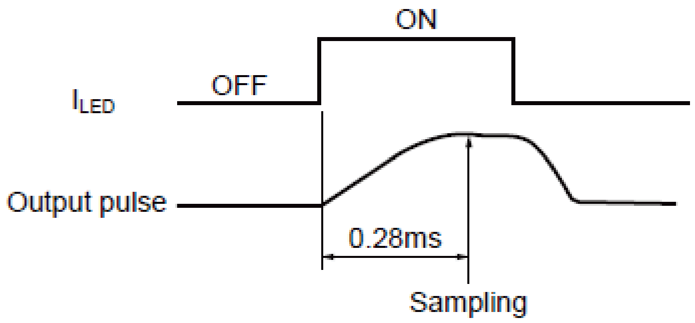
We can use the Arduino Timer 1 OC1B output (D10) to create the correct pulse-width modulated waveform for the LED: low for 0.32msec every 10msec.

Fortunately, Jesse Tane and others have written a rather nice library which lets us access Timer 1 functions with ease. You can download the library from:
http://playground.arduino.cc/Code/Timer1
https://code.google.com/p/arduino-timerone/downloads/list
and can find instructions on how to add this to the Arduino libraries here:
https://www.arduino.cc/en/Guide/Libraries

We need to set Timer 1 to have a pulse length of 10,000 microseconds (10 milliseconds or 100Hz). Pin 10 is put into Pulse Width Modulation mode with a HIGH period of 9.68 milliseconds.

## Phototransistor Samples

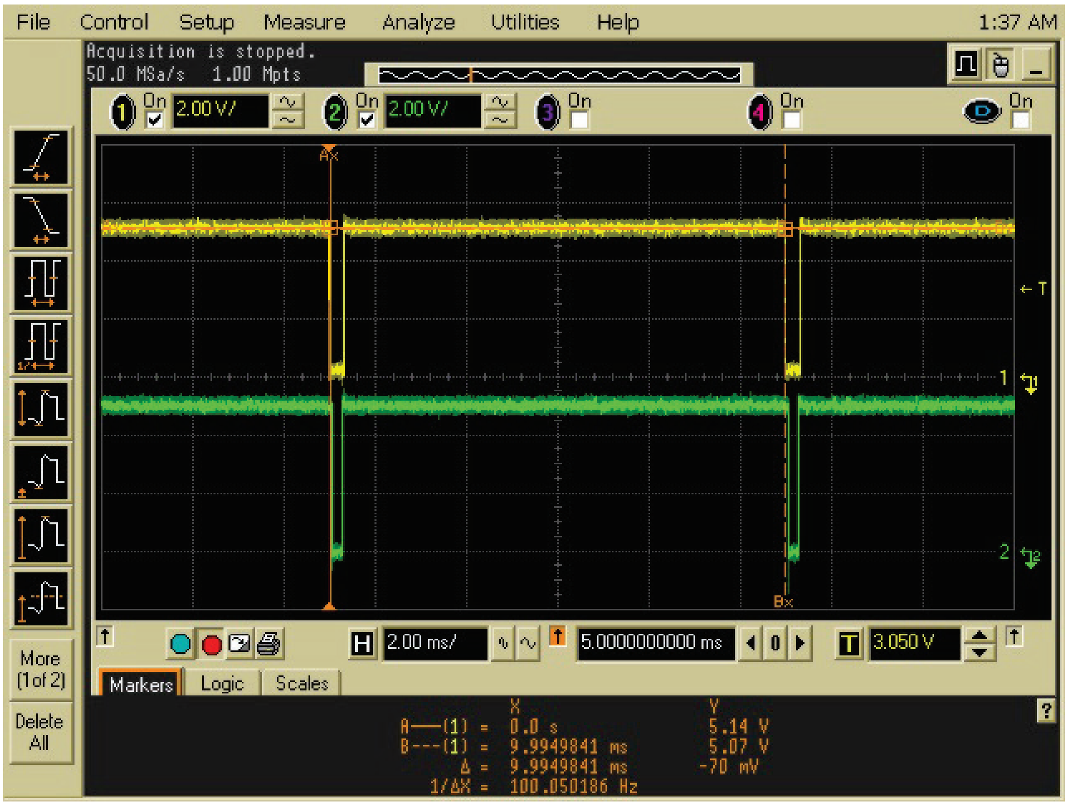
The GP2Y1010AU0F data sheet calls for the phototransistor output (pin 5) to be sampled 0.28msec into the LED pulse.



We can do this quite easily by programming the Timer 1 OC1A output (D9) to produce a PWM output with a LOW period of 0.28msec every 10msec (HIGH for 9.72msec). By connecting D9 to D2 we can generate an interrupt (INT0) at the correct point in the cycle to sample the sensor output voltage via A0. We generate the interrupt on the RISING edge of D2.

A quick check with an oscilloscope confirms that we are generating the correct waveforms on OC1B (D10, Yellow trace) and OC1A (D9, Green trace) :

Correct timing of the OC1B pulses: low every 10msec:



```
Acquisition is stopped.
50.0 MSa/s   1.00 Mpts
```

1: ☑ On 2.00 V/

2: ☑ On 2.00 V/

3: □ On

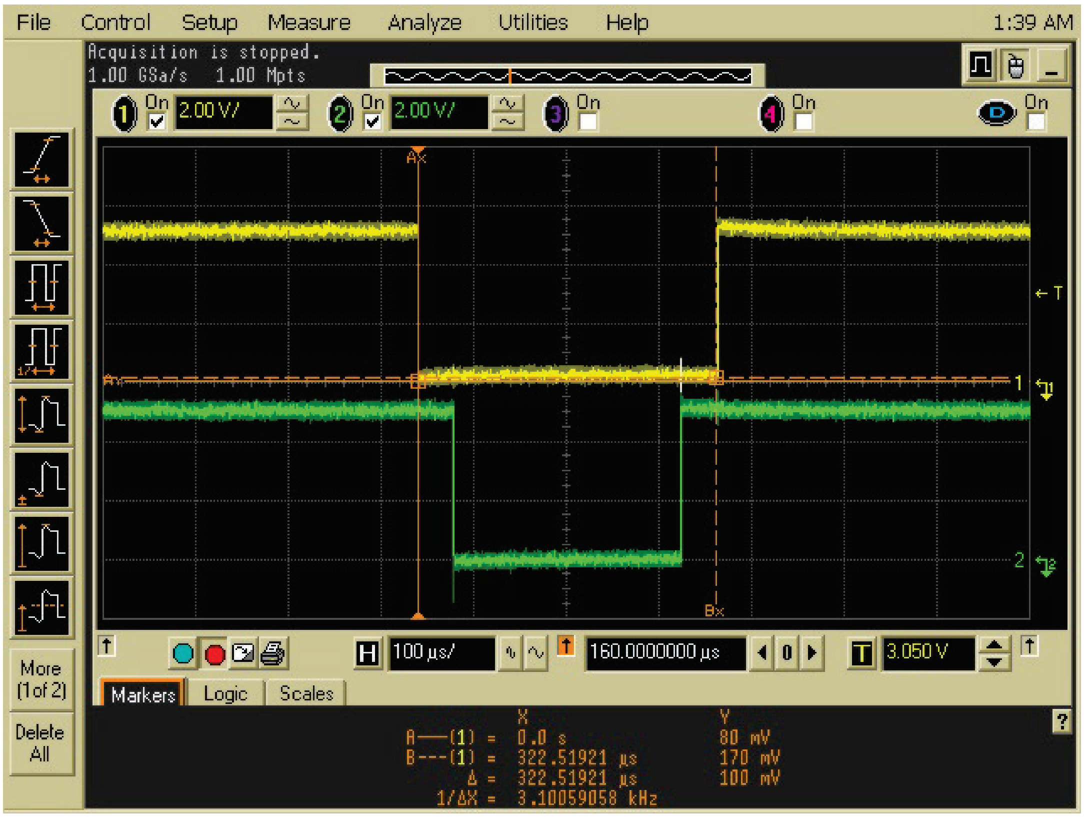
4: □ On

H 2.00 ms/ 5.0000000000 ms

T 3.050 V

```
              X                   Y
A——(1) =  0.0 s                5.14 V
B---(1) =  9.9949841 ms        5.07 V
     Δ =  9.9949841 ms        -70 mV
  1/ΔX =  100.050186 Hz
```

Correct low pulse width of 0.32msec (320µsec) on OC1B (D10):



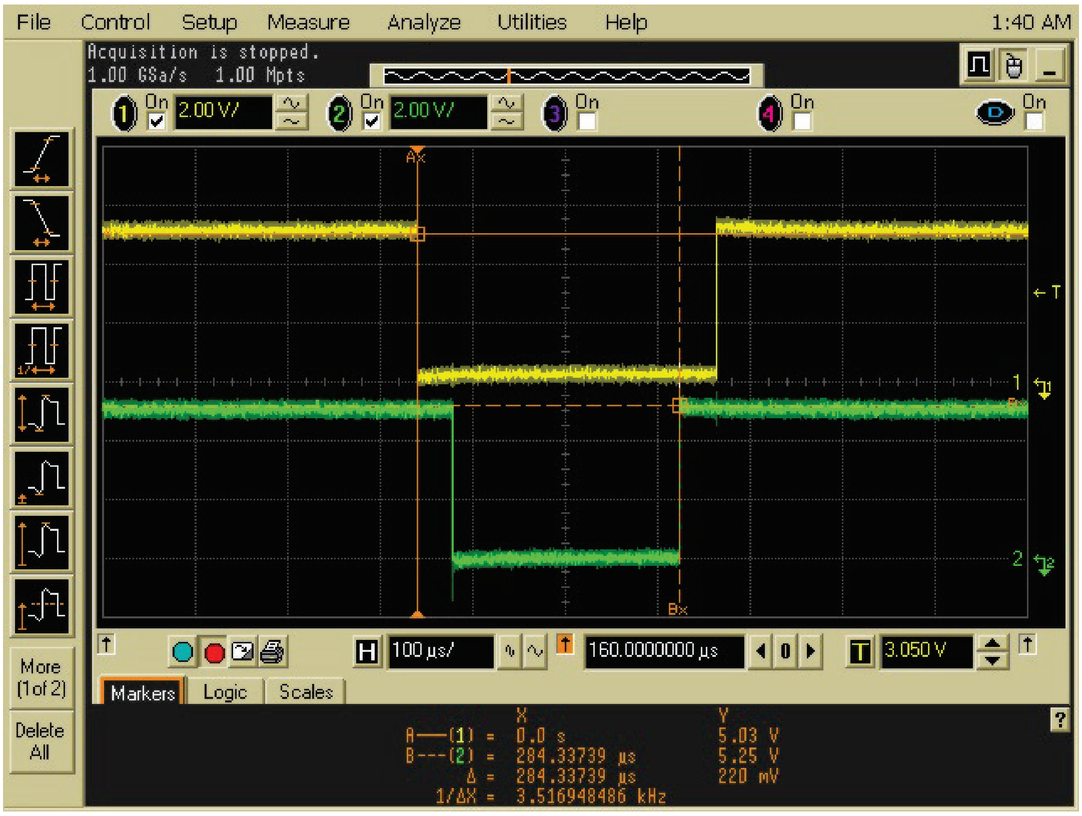
```
Acquisition is stopped.
1.00 GSa/s   1.00 Mpts
```

1 On ☑ 2.00 V/ 2 On ☑ 2.00 V/ 3 On □ 4 On □ On □

H 100 µs/ 160.0000000 µs 0 T 3.050 V

```
                X                     Y
A——(1) =  0.0 s                  80 mV
B---(1) = 322.51921 µs          170 mV
      Δ = 322.51921 µs          100 mV
   1/ΔX = 3.10059058 kHz
```

Correct OC1A (D9) delay of 0.28msec (280µsec) from LED switch on to INT0 interrupt (D2):



```
Acquisition is stopped.
1.00 GSa/s   1.00 Mpts
```

1 On ☑ 2.00 V/ 2 On ☑ 2.00 V/ 3 On □ 4 On □ On □

H 100 µs/ 160.0000000 µs 0 T 3.050 V

```
                X                     Y
A——(1) =  0.0 s                 5.03 V
B---(2) = 284.33739 µs          5.25 V
      Δ = 284.33739 µs          220 mV
   1/ΔX = 3.516948486 kHz
```

## The Code

You can find the full sketch in the Dust_Sensor directory.

Enjoy!

Paul Clark
17th Nov 2015