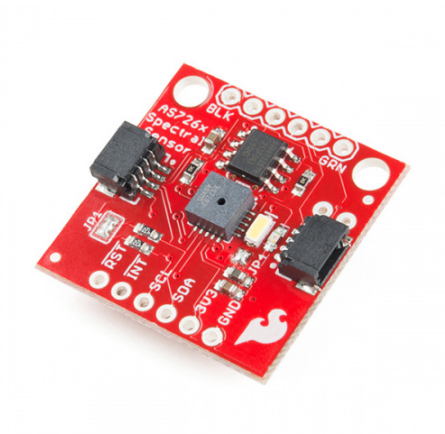
AS726X NIR/VIS Spectral Sensor Hookup Guide

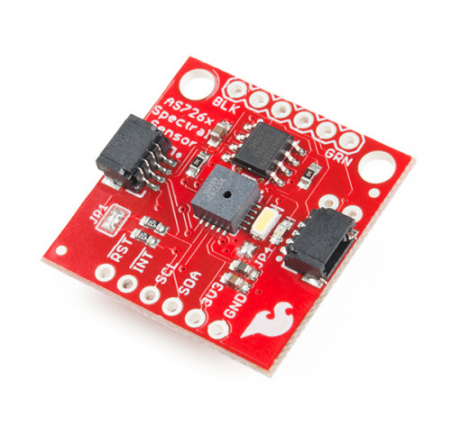
Introduction

The AS726X Spectral Sensors from AMS brings a field of study to consumers that was previously unavailable, spectroscopy! It's now easier than ever to measure and characterize how different materials absorb and reflect different wavelengths of light. Don't know what part of the spectrum you want to look at? You're in luck! Sparkfun carries two different flavors of spectrometer. The AS7262 detects wavelengths in the visible range while the AS7263 detects wavelengths just below the visible range, in the Near Infrared (NIR) range.



SparkFun Spectral Sensor Breakout - AS7263 NIR (Qwiic)

SEN-14351



SparkFun Spectral Sensor Breakout - AS7262 Visible (Qwiic)

SEN-14347

Product Showcase: Qwiic AS726X

Required Materials

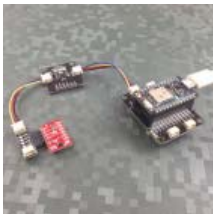
To follow along with this hookup guide, you will need one of the following Qwiic shields to match your preference of microcontroller:

SparkFun Qwiic Shield for Arduino

DEV-14352

Qwiic Shield for Raspberry Pi

SPX-14292



Qwiic Shield for ESP32
SPX-14203

Qwiic Shield for Photon
SPX-14202

You will also need a Qwiic cable to connect the shield to your AS726X, choose a length that suits your needs.

Qwiic Cable - 500mm
PRT-14429

Qwiic Cable - 50mm
PRT-14426

Qwiic Cable - 100mm
PRT-14427

Qwiic Cable - 200mm
PRT-14428

Suggested Reading

If you aren't familiar with our new Qwiic system, we recommend reading here for an overview. We would also recommend taking a look at the following tutorials if you aren't familiar with them.

How to Solder: Through-Hole Soldering
This tutorial covers everything you need to know about through-hole soldering.

Serial Communication
Asynchronous serial communication concepts: packets, signal levels, baud rates, UARTs and more!

Light

I2C

Light is a useful tool for the electrical engineer. Understanding how light relates to electronics is a fundamental skill for many projects.

An introduction to I2C, one of the main embedded communications protocols in use today.

Qwiic Shield for Arduino & Photon Hookup Guide

Get started with our Qwiic ecosystem with the Qwiic shield for Arduino or Photon.

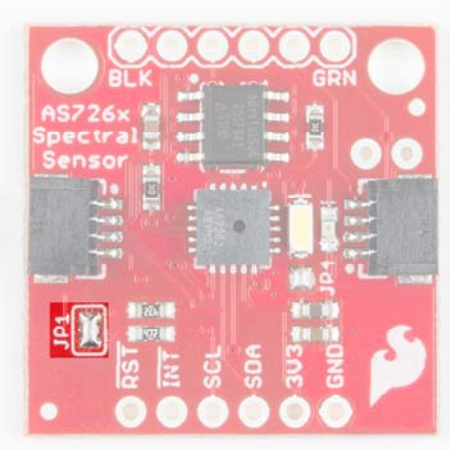
Hardware Overview

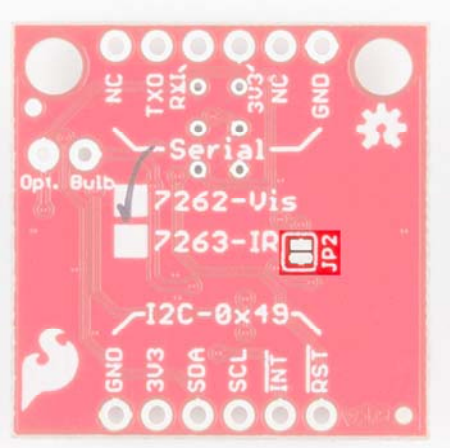
The AS7262 detects 450, 500, 550, 570, 600, and 650nm of light each with 40nm of full-width half-max detection. The AS7263 can detect 610, 680, 730, 760, 810, and 860nm of light each with 20nm of full-width half-max detection.

Communication

The AS726X is unique as it can communicate by both an I²C interface (through the onboard Qwiic connectors, or pins at the bottom of the board) and serial interface using AT commands (pins at the top of the board).

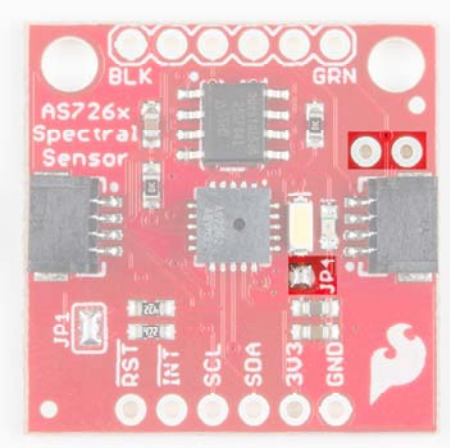
While I²C is the default setting (The default I²C address is 0X49 for both AS7262 and AS7263) Serial communication can be enabled by removing solder from the jumpers labeled JP1, adding solder to the jumper labeled JP2 (on the back of the board), and using Sparkfun's USB-to-Serial breakout to interface directly with the computer.





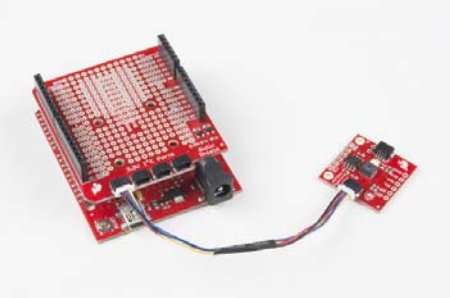
LED

The board also has multiple ways for you to illuminate the object that you are trying to measure for a more accurate spectroscopy reading. There is an onboard, 5700K LED that has been picked out specifically for this task. However, if you aren't satisfied with the onboard LED, you can grab your own through hole incandescent. While you should find a bulb rated for 3.3V, a bulb rated for higher voltage, like 5V, will still work, but will not run as bright as it normally would with 5V. We've found that Mouser is a good place to look for these. If you are going to go that route and use your own bulb, be sure to disable the onboard LED by removing the solder from the JP4 jumper.



Hardware Assembly

The beauty of Sparkfun's new Qwiic environment means that connecting the sensor could not be easier. Just plug one end of the Qwiic cable into the AS726X, the other into one of the Qwiic Shields, and stack the board on a development board. You'll be ready to upload a sketch to start taking spectroscopy measurements. It seems too easy, but thats why we made it this way! Here's an example using the RedBoard Programmed with Arduino.



Library Overview

Before we get started, we'll need to download and install SparkFun's AS726X Arduino library.

DOWNLOAD THE SPARKFUN AS726X LIBRARY https://github.com/sparkfun/Sparkfun_AS726X_Arduino_Library/archive/master.zip

Note: This example assumes you are using the latest version of the Arduino IDE on your desktop. If this is your first time using Arduino, please review our tutorial on installing the Arduino IDE. If you have not previously installed an Arduino library, please check out our installation guide.

Before we get started developing a sketch, let's look at the available functions of the library.

- `void begin(TwoWire &wirePort, byte gain, byte measurementMode);` — Initializes the sensor with user given values for the wire port, gain, and measurement
- `void takeMeasurements();` — Sensor writes spectral measurements to memory locations.
- `void takeMeasurementsWithBulb();` — Illuminates onboard bulb, calls `takeMeasurements();` , then turns off the onboard bulb.
- `void printMeasurements();` — Fetches data from memory and outputs calibrated measurements using `Serial.print();`
- `void printUncalibratedMeasurements();` — Fetches data from memory and outputs uncalibrated measurements using `Serial.print();`
- `byte getTemperature();` — Fetches the temperature in degrees Celsius.
- `float getTemperatureF();` — Fetches the temperature in degrees Fahrenheit.
- `void setMeasurementMode(byte mode);` — Changes the measurement mode to 0, 1, 2, or 3
    - 0: Continuous reading of VBGY (Visible) / STUV (IR)
    - 1: Continuous reading of GYOR (Visible) / RTUX (IR)
    - 2: Continuous reading of all channels
    - 3: One-shot reading of all channels (power-on default)

- `boolean dataAvailable();` — Returns `true` or `false` based on whether or not data is available to be read.
- `void enableIndicator();` — Powers on the surface mounted blue indicator LED.
- `void disableIndicator();` — Powers off the surface mounted blue indicator LED.
- `void setIndicatorCurrent(byte current);` — Sets the current on the indicator LED. The default is `current = 3`, or 8 mA.
    - 0: 1 mA
    - 1: 2 mA
    - 2: 4 mA
    - 3: 8 mA
- `void enableBulb();` — Powers on the surface mounted blue indicator LED.
- `void disableBulb();` — Powers off the surface mounted blue indicator LED.
- `void setBulbCurrent(byte current);` — Sets the current limit on the indicator LED and optional bulb (They are connected in parallel) The default is `current = 0` or 12.5 mA.
    - 0: 12.5 mA
    - 1: 25 mA
    - 2: 50 mA
    - 3: 100 mA
- `void softReset();` — Gives the sensor a 1 second reset.
- `void setGain(byte gain);` — Pass in a 0, 1, 2 or 3 to change the gain.
    - 0: 1x
    - 1: 3.7x
    - 2: 16x
    - 3: 64x (power-on default)
- `void setIntegrationTime(byte integrationValue);` — This sets the time over which samples are taken.
    - Takes a value between 0 and 255.
    - Integration time will be 2.8 ms * `integrationValue`.
- `void enableInterrupt();` — Pulls the interrupt pin low. (Note: not yet implemented)
- `void disableInterrupt();` — Pulls the interrupt pin high.
- If you'd like access to just one channel, getting uncalibrated and calibrated spectral readings for the AS7262 (Visible) sensor can be accomplished with the following commands:
    - `int getViolet();`
    - `int getBlue();`
    - `int getGreen();`
    - `int getYellow();`
    - `int getOrange();`
    - `int getRed();`
    - `float getCalibratedViolet();`
    - `float getCalibratedBlue();`
    - `float getCalibratedGreen();`
    - `float getCalibratedYellow();`
    - `float getCalibratedOrange();`
    - `float getCalibratedRed();`
- A similar set of functions is available for accessing individual channels on the AS7263 (Near Infrared) sensor.
    - `int getR();`
    - `int getS();`
    - `int getT();`
    - `int getU();`
    - `int getV();`
    - `int getW();`

- `float getCalibratedR();`
- `float getCalibratedS();`
- `float getCalibratedT();`
- `float getCalibratedU();`
- `float getCalibratedV();`
- `float getCalibratedW();`

Software

Example 1 – Basic Readings

The below sketch will get you up and running taking calibrated spectral readings on all 6 channels. Once this sketch is uploaded, open the serial monitor with a baud rate of 115200 to display the spectral data from the sensor.

```
#include "AS726X.h"
AS726X sensor;

void setup() {
  sensor.begin();
}

void loop() {
  sensor.takeMeasurements();
  sensor.printMeasurements();//Prints out all measurements (ca
librated)
}
```

If we want, we can change the gain, measurement mode, and Wire that I²C uses by calling the `begin()` function with a few arguments. First, let's look at what values we can assign to which characteristic.

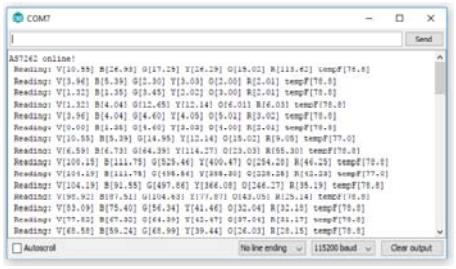
Example 2 – Sensor Settings

The below example code will initialize our sensor with a gain of 16x, measurement mode of 0, and our regular I²C port (you can run the sensor on a different I²C port if you have the right hardware, a Teensy perhaps?).

```
#include "AS726X.h"
AS726X sensor;

byte GAIN = 2;
byte MEASUREMENT_MODE = 0;

void setup() {
    sensor.begin(Wire, GAIN, MEASUREMENT_MODE);
}

void loop() {
    sensor.takeMeasurements();
    sensor.printMeasurements();
}
```

Expected Output

Here is a picture of what to expect when bringing your AS726X online for either of the above sketches. (The program will print "AS7263 online!" instead if you have that sensor)



Resources and Going Further

Now that you've successfully got your AS726X up and running, it's time to incorporate it into your own project!

For more information, check out the resources below:

- AS726X Schematic (PDF) - Schematic for both the AS7262 and AS7263.
- AS726X Eagle Files (ZIP) - Board design files for both the AS7262 and AS7263.
- AS7262 Datasheet (PDF) - Datasheet for AS7262 Visible Sensor.
- AS7263 Datasheet (PDF) - Datasheet for AS7263 NIR Sensor.
- Product Showcase: Qwiic AS726X
- Qwiic System Landing Page
- SparkFun AS726X GitHub Repository – Source and example files for the Arduino library used in this tutorial.

Need some inspiration for your next project? Check out some of these related tutorials:

Qwiic Shield for Arduino & Photon Hookup Guide

Get started with our Qwiic ecosystem with the Qwiic shield for Arduino or Photon.

CCS811/BME280 (Qwiic) Environmental Combo Breakout Hookup Guide

Sense various environmental conditions such as temperature, humidity, barometric pressure, eCO2 and tVOCs with the CCS811 and BME280 combo breakout board.

SparkFun GPS Breakout - XA1110 (Qwiic) Hookup Guide

Figure out where in the world you are with the Qwiic SparkFun GPS Breakout - XA1110.