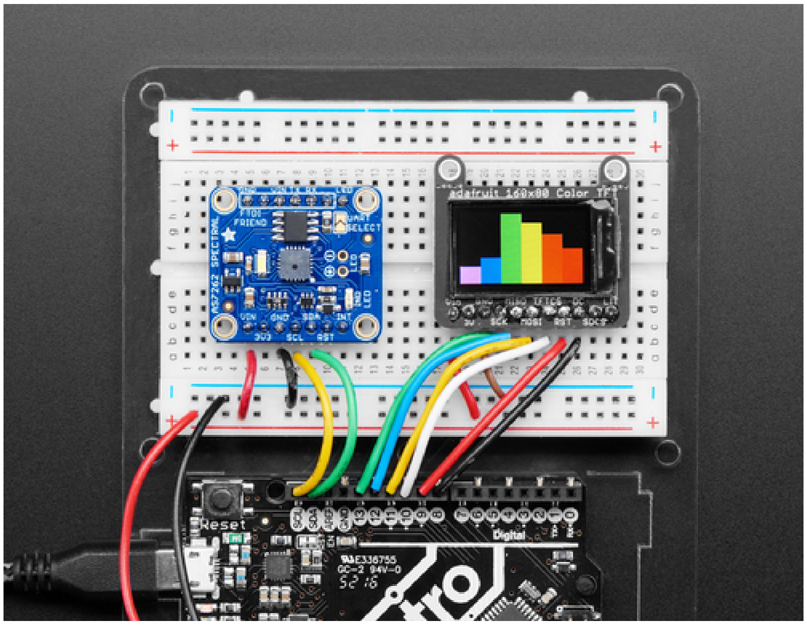
adafruit learning system

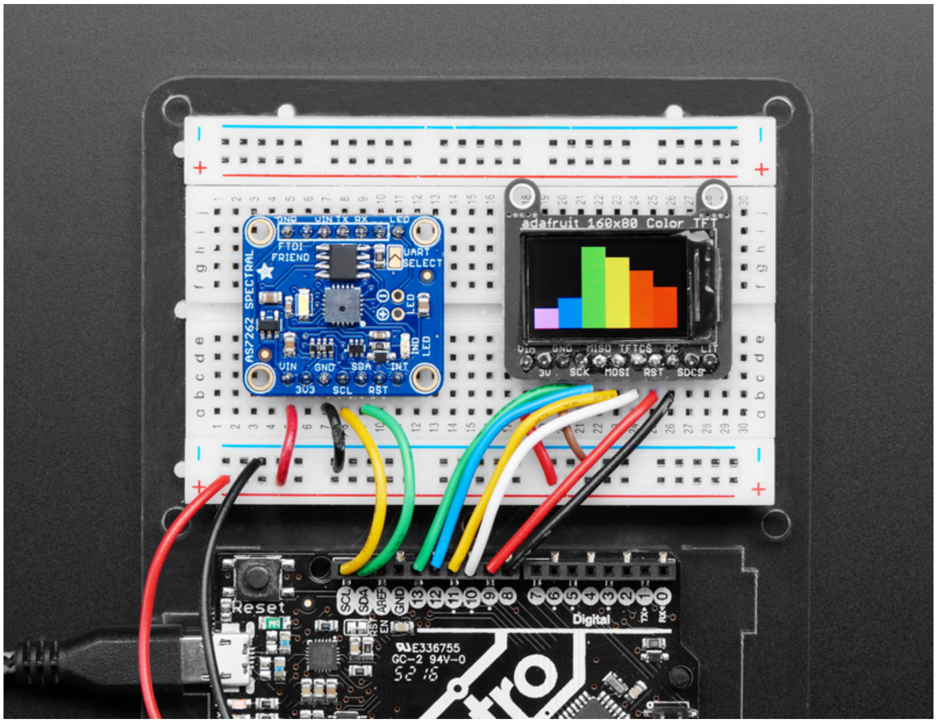
Adafruit AS7262 6-channel Visible Light Sensor

Created by Dean Miller



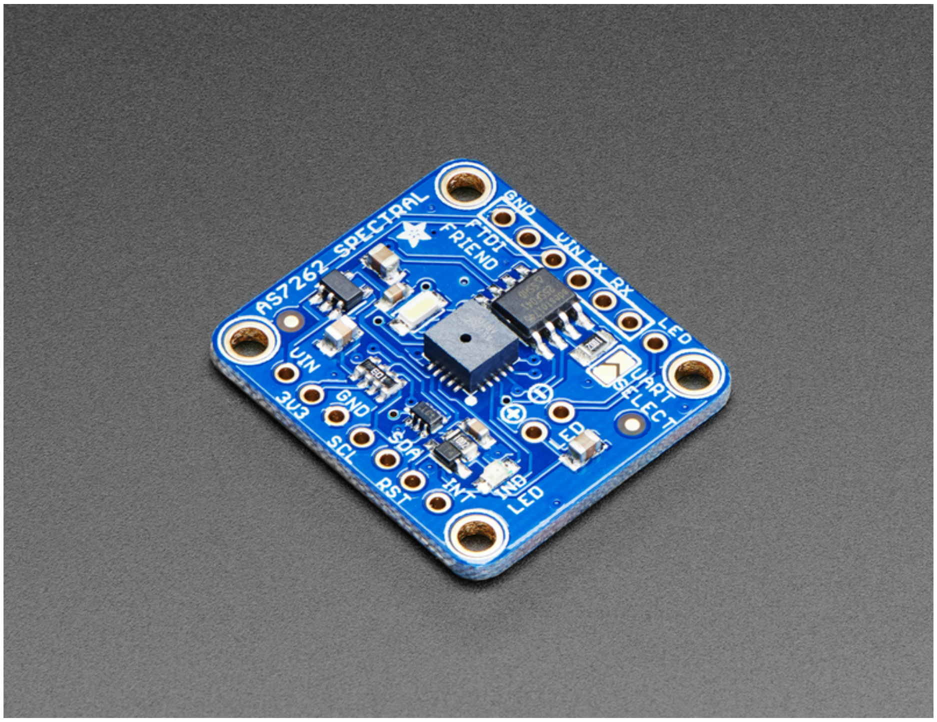
Last updated on 2020-04-24 09:09:05 PM EDT

Overview

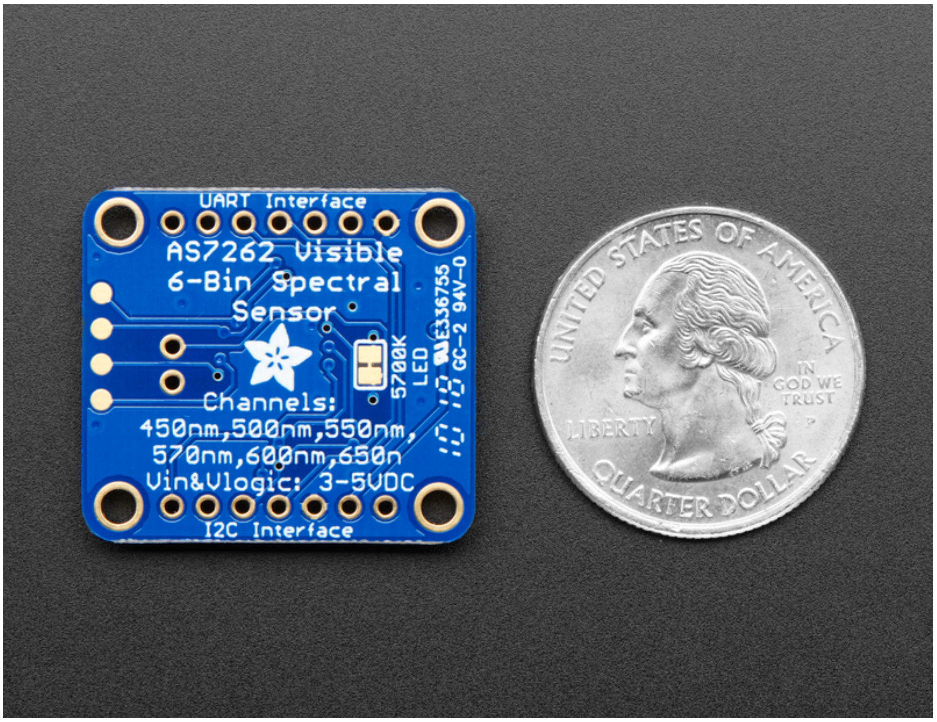


The colors of the rainbow, so pretty in the sky
Are also in this sensor, controlled by you and I!

Detect trees of green, and red roses too with the new **Adafruit AS7262 6-Channel Visible Light / Color Sensor Breakout**. This sensor from AMS has 6 integrated visible light sensing channels for red, orange, yellow, green, blue, and violet. These channels can be read via the I2C bus as either raw 16-bit values or calibrated floating-point values. There is also an on-board temperature sensor that can be used to read the temperature of the chip, and a powerful LED flash to reflect light off objects for better color detection.

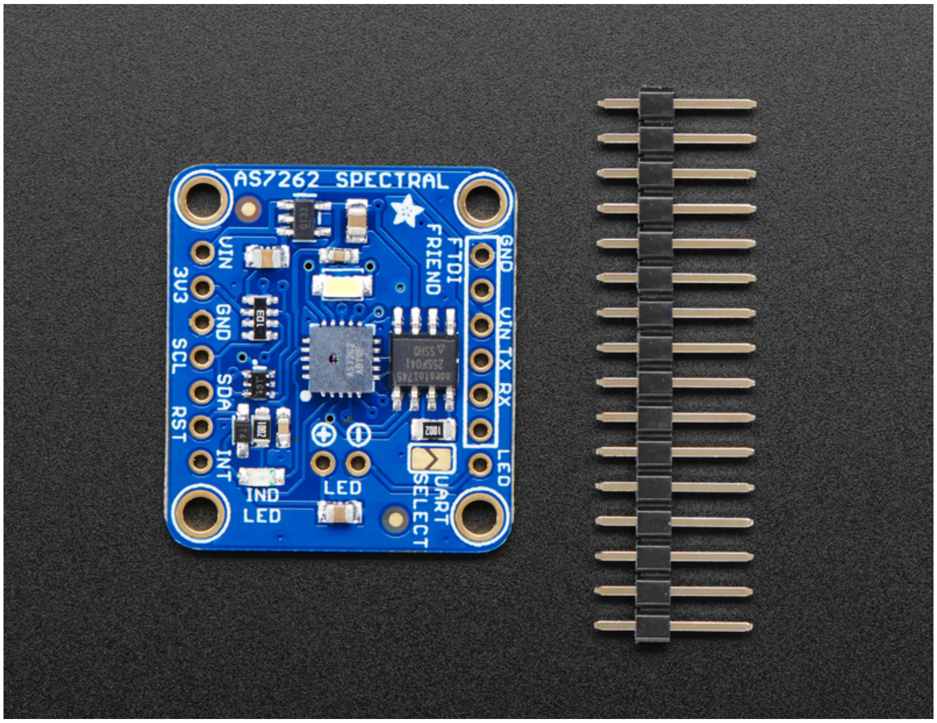


The 6 channels on the AS7262 are realized by silicon interference filters at 450nm, 500nm, 550nm, 570nm, 600nm, and 650nm. This breakout uses the I2C interface on the chip by default, but a UART interface that accepts AT commands is also selectable.

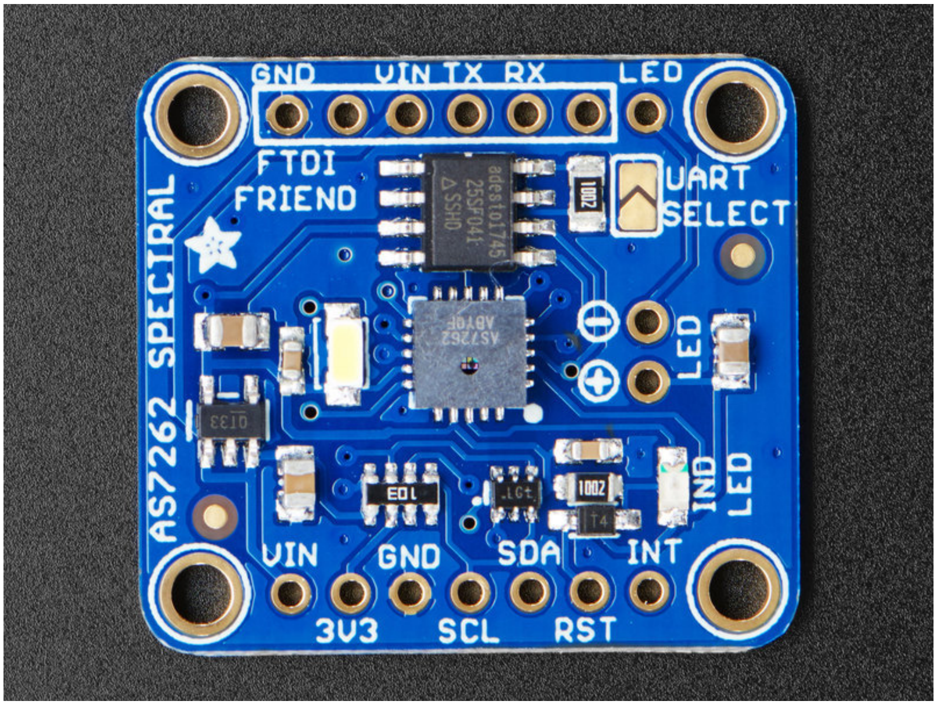


We've placed the sensor on a PCB for you and included an SPI flash chip pre-programmed with the device firmware, a 3.3V regulator, I2C level shifting, and the recommended LED with 5700K color temperature. Both the I2C and UART interfaces are broken out, making this breakout an excellent all-in-one solution for all your color-sensing needs. The pinout for the UART interface is plug-and-play compatible with the Adafruit FTDI Friend.

We've also prepared software libraries to get you up and running in Arduino or CircuitPython with just a few lines of code!



Pinouts



This sensor has 4 mounting holes and 2 header breakout strips.

Power Pins:

- **Vin** - this is the power pin. Since the sensor uses 3.3V, we have included an onboard voltage regulator that will take 3-5VDC and safely convert it down. To power the board, give it the same power as the logic level of your microcontroller - e.g. for a 5V micro like Arduino, use 5V
- **3Vo** - this is the 3.3V output from the voltage regulator, you can grab up to 100mA from this if you like
- **GND** - common ground for power and logic

Logic pins:

- **SCL** - this is the I2C clock pin, connect to your microcontrollers I2C clock line. There is a 10K pullup on this pin and it is level shifted so you can use 3 - 5VDC. If UART mode is selected, this pin acts as RX.
- **SDA** - this is the I2C data pin, connect to your microcontrollers I2C data line. There is a 10K pullup on this pin and it is level shifted so you can use 3 - 5VDC. If UART mode is selected, this pin acts as TX.
- **RST** - this is the reset pin. When it is pulled to ground the sensor resets itself. This pin is level shifted so you can use 3-5VDC logic.

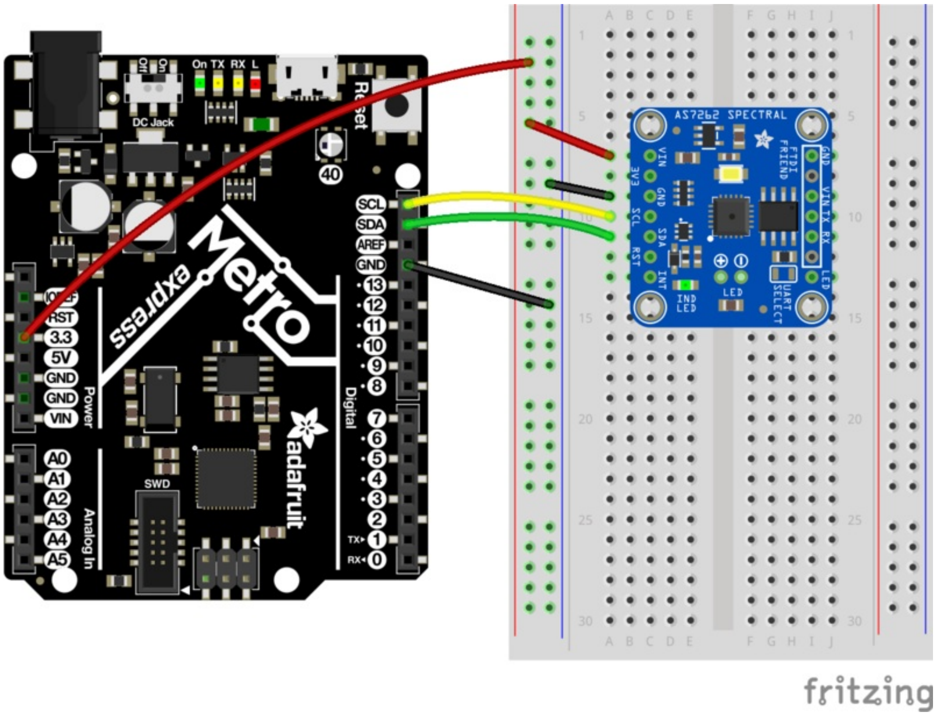
Solder closed the **UART SELECT** jumper to switch the sensor into UART mode.

UART Logic pins:

- **TX** - this is the UART transmit pin, connect to your microcontrollers UART RX line. There is a 10K pullup on this pin and it is level shifted so you can use 3 - 5VDC. If I2C mode is selected, this pin acts as SCL.
- **RX** - this is the UART receive pin, connect to your microcontrollers UART TX line. There is a 10K pullup on this pin and it is level shifted so you can use 3 - 5VDC. If I2C mode is selected, this pin acts as SDA.

Arduino Wiring & Test

You can easily wire this breakout to any microcontroller, we'll be using an Adafruit Metro M0 Express (Arduino compatible) with the Arduino IDE. But, you can use any other kind of microcontroller as well as long as it has I2C clock and I2C data lines.



I2C Wiring

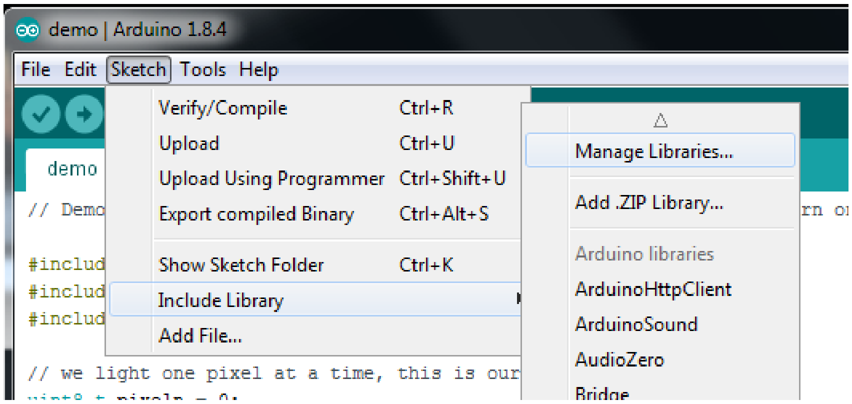
- Connect **Vin** to the power supply, 3-5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A5**, on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**
- Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

This sensor uses I2C address **0x49**.

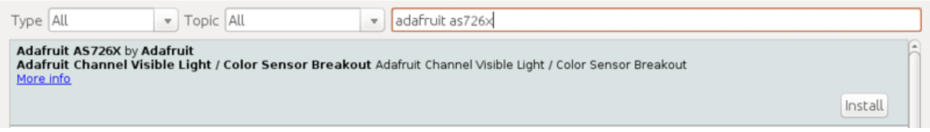
Download Adafruit_AS726x library

To begin reading sensor data, you will need to download Adafruit_AS726x from the Arduino library manager.

Open up the Arduino library manager:



Search for the **Adafruit AS726X** library and install it



Type All Topic All adafruit as726x

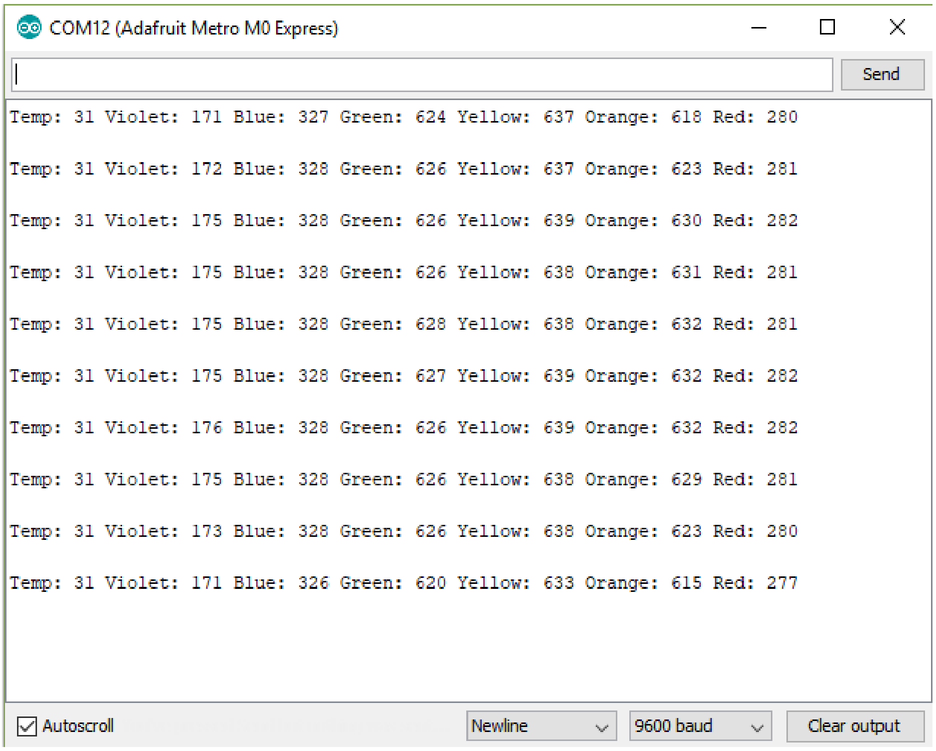
Adafruit AS726X by **Adafruit**
Adafruit Channel Visible Light / Color Sensor Breakout Adafruit Channel Visible Light / Color Sensor Breakout
More info

We also have a great tutorial on Arduino library installation at:
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (https://adafru.it/aYM)

Load Test Example

Open up **File->Examples->Adafruit_AS726x->AS7262_test** and upload to your Arduino wired up to the sensor. This example connects to the sensor and starts taking readings.

Once uploaded to your Arduino, open up the serial console at 9600 baud speed to see the readings. Your sensor will read the temperature and it's 6 visible light channels. Your serial monitor will look something like this:

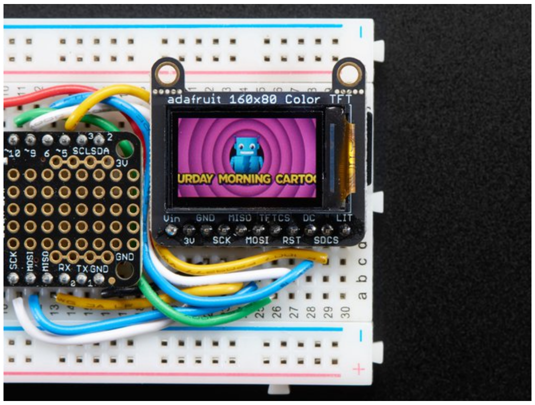


```
Temp: 31 Violet: 171 Blue: 327 Green: 624 Yellow: 637 Orange: 618 Red: 280

Temp: 31 Violet: 172 Blue: 328 Green: 626 Yellow: 637 Orange: 623 Red: 281

Temp: 31 Violet: 175 Blue: 328 Green: 626 Yellow: 639 Orange: 630 Red: 282

Temp: 31 Violet: 175 Blue: 328 Green: 626 Yellow: 638 Orange: 631 Red: 281

Temp: 31 Violet: 175 Blue: 328 Green: 628 Yellow: 638 Orange: 632 Red: 281

Temp: 31 Violet: 175 Blue: 328 Green: 627 Yellow: 639 Orange: 632 Red: 282

Temp: 31 Violet: 176 Blue: 328 Green: 626 Yellow: 639 Orange: 632 Red: 282

Temp: 31 Violet: 175 Blue: 328 Green: 626 Yellow: 638 Orange: 629 Red: 281

Temp: 31 Violet: 173 Blue: 328 Green: 626 Yellow: 638 Orange: 623 Red: 280

Temp: 31 Violet: 171 Blue: 326 Green: 620 Yellow: 633 Orange: 615 Red: 277
```

☑ Autoscroll Newline 9600 baud

TFT Demo

You can create a nice bar graph of the sensors 6 color channels using a 0.90 inch TFT from Adafruit.

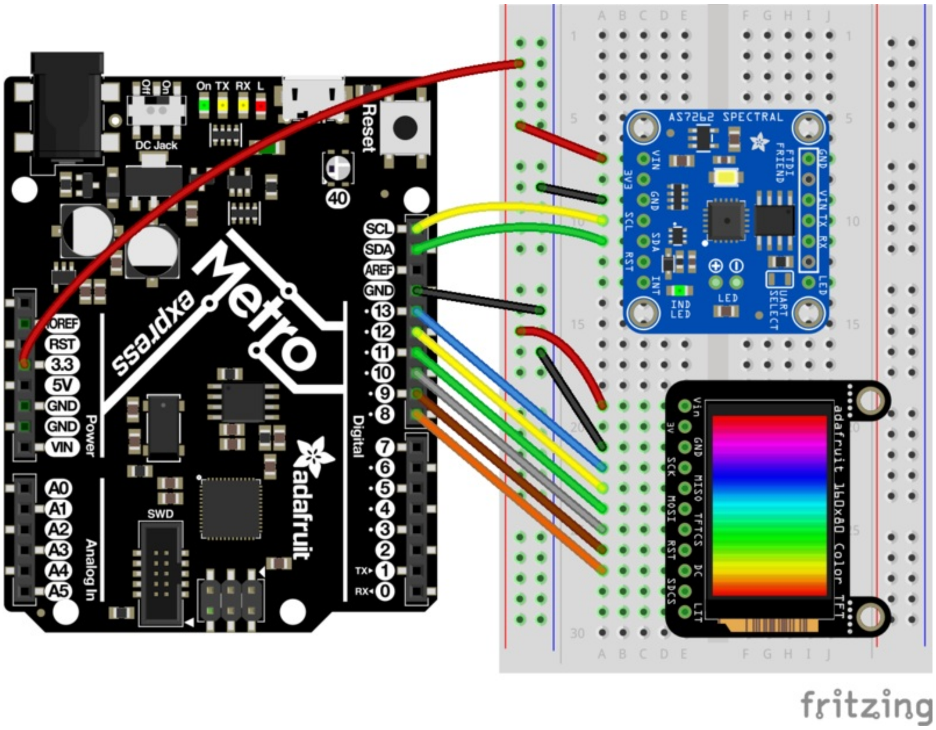


Adafruit 0.96" 160x80 Color TFT Display w/ MicroSD Card Breakout

$14.95
IN STOCK

Add To Cart

Wire the TFT up as shown:



Upload the following code to your board through the Arduino IDE (the code is also available under **File->Examples->Adafruit_AS726x->color_graph**)

```
/***************************************************************************
  This is a library for the Adafruit AS7262 6-Channel Visible Light Sensor

  This sketch reads the sensor and creates a color bar graph on a tiny TFT

  Designed specifically to work with the Adafruit AS7262 breakout and 160x18 tft
  ----> http://www.adafruit.com/products/3779
  ----> http://www.adafruit.com/product/3533
  These sensors use I2C to communicate. The device's I2C address is 0x49
  Adafruit invests time and resources providing this open source code,
  please support Adafruit andopen-source hardware by purchasing products
  from Adafruit!

  Written by Dean Miller for Adafruit Industries.
  BSD license, all text above must be included in any redistribution
 ***************************************************************************/

#include <Wire.h>
#include "Adafruit_AS726x.h"

#include <Adafruit_GFX.h>    // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library
#include <SPI.h>


// For the breakout, you can use any 2 or 3 pins
// These pins will also work for the 1.8" TFT shield
#define TFT_CS     10
#define TFT_RST    9  // you can also connect this to the Arduino reset
                      // in which case, set this #define pin to -1!
#define TFT_DC     8

#define SENSOR_MAX 5000

#define BLACK   0x0000
#define GRAY    0x8410
#define WHITE   0xFFFF
#define RED     0xF800
#define ORANGE  0xFA60
#define YELLOW  0xFFE0
#define LIME    0x07FF
#define GREEN   0x07E0
#define CYAN    0x07FF
#define AQUA    0x04FF
#define BLUE    0x001F
#define MAGENTA 0xF81F
#define PINK    0xF8FF

uint16_t colors[] = {
  MAGENTA,
  BLUE,
  GREEN,
  YELLOW,
  ORANGE,
  RED
};

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,  TFT_DC, TFT_RST);

//create the object
Adafruit_AS726x ams;

//buffer to hold raw values (these aren't used by default in this example)
//uint16_t sensorValues[AS726x_NUM_CHANNELS];

//buffer to hold calibrated values
float calibratedValues[AS726x_NUM_CHANNELS];
```

```
uint16_t barWidth;

void setup() {

  Serial.begin(9600);

  tft.initR(INITR_MINI160x80);   // initialize a ST7735S chip, mini display
  tft.setRotation(3);

  tft.fillScreen(ST7735_BLACK);

  barWidth = tft.width() / AS726x_NUM_CHANNELS;

  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);

  //begin and make sure we can talk to the sensor
  if(!ams.begin()){
    Serial.println("could not connect to sensor! Please check your wiring.");
    while(1);
  }

  ams.setConversionType(MODE_2);

  //uncomment this if you want to use the driver LED (off by default)
  //ams.drvOn();
}

void loop() {

  if(ams.dataReady()){

    //read the values!
    //ams.readRawValues(sensorValues);
    ams.readCalibratedValues(calibratedValues);

    for(int i=0; i<AS726x_NUM_CHANNELS; i++){
      uint16_t height = map(calibratedValues[i], 0, SENSOR_MAX, 0, tft.height());

      tft.fillRect(barWidth * i, 0, barWidth, tft.height() - height, ST7735_BLACK);
      tft.fillRect(barWidth * i, tft.height() - height, barWidth, height, colors[i]);
    }
  }

}
```

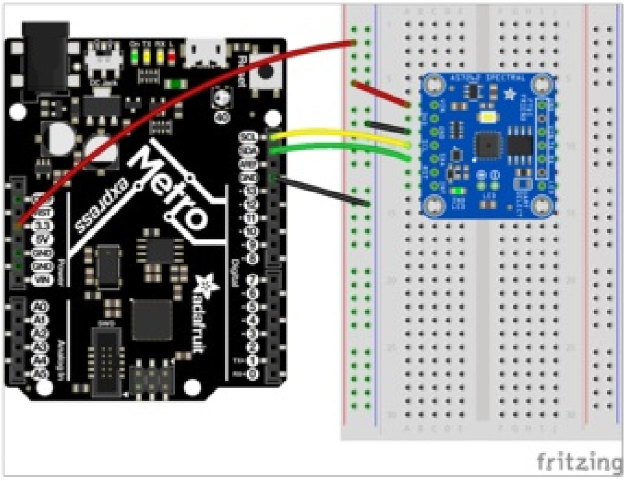
Python & CircuitPython

It's easy to use the AS7262 with Python or CircuitPython and the Adafruit CircuitPython AS726x (https://adafru.it/C2x) module. This module allows you to easily write Python code that reads color data and temperature from the sensor.

You can use this sensor with any CircuitPython microcontroller board or with a computer that has GPIO and Python thanks to Adafruit_Blinka, our CircuitPython-for-Python compatibility library (https://adafru.it/BSN).

CircuitPython Microcontroller Wiring - I2C

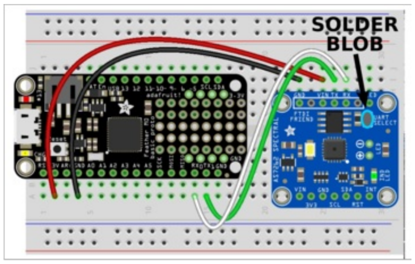
You can easily wire this breakout to a microcontroller running CircuitPython. We will be using a Metro M0 Express.



- Connect **Vin** to the power supply, 3-5V is fine
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Feather or Metro M0 (on a Gemma M0 this would be **Pad #2/ A1**)
- Connect the **SDA** pin to the I2C data **SDA** pin on your Feather or Metro M0 (on a Gemma M0 this would be **Pad #0/A2**)

CircuitPython Microcontroller Wiring - UART

Here's the wiring to use for UART:



- **Feather 3V** to **board VIN**
- **Feather GND** to **board GND**
- **Feather RX** to **board RX***
- **Feather TX** to **board TX***

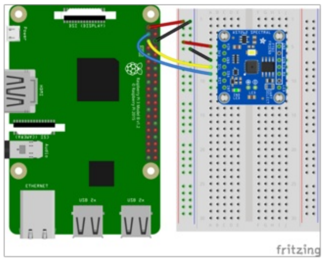
* Yep, TX to TX and RX to RX, that appears to be the way the AS726x has named things.

Don't forget to solder the UART SELECT jumper to enable UART mode.

Python Computer Wiring - I2C

Since there's *dozens* of Linux computers/boards you can use we will show wiring for Raspberry Pi. For other platforms, please visit the guide for CircuitPython on Linux to see whether your platform is supported (https://adafru.it/BSN).

Here's the Raspberry Pi wired with I2C:



- **Pi 3V3** to **sensor VIN**
- **Pi GND** to **sensor GND**
- **Pi SDA** to **sensor SDA**
- **Pi SCL** to **sensor SCL**

This sensor uses I2C address **0x49**.

CircuitPython Installation of AS726x Library

You'll need to install the Adafruit_CircuitPython_AS726x library (https://adafru.it/C1M) library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython (https://adafru.it/Amd) for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (https://adafru.it/zdx). Our introduction guide has a great page on how to install the library bundle (https://adafru.it/ABU) for both express and non-express boards.

Remember for non-express boards you'll need to manually install the necessary libraries from the bundle:

- **adafruit_as726x.mpy**
- **adafruit_bus_device**
- **adafruit_register**

You can also download the **adafruit_as726x.mpy** from it's release page on Github. (https://adafru.it/C1N)

Before continuing make sure your board's lib folder or root filesystem has the **adafruit_as726x.mpy, adafruit_register**, and **adafruit_bus_device** files and folders copied over.

Next connect to the board's serial REPL (https://adafru.it/Awz)so you are at the CircuitPython >>> prompt.

Python Installation of AS726x Library

You'll need to install the Adafruit_Blinka library that provides the CircuitPython support in Python. This may also require enabling I2C on your platform and verifying you are running Python 3. Since each platform is a little different, and Linux changes often, please visit the CircuitPython on Linux guide to get your computer ready (https://adafru.it/BSN)!

Once that's done, from your command line run the following command:

- `sudo pip3 install adafruit-circuitpython-as726x`

If your default Python is version 3 you may need to run 'pip' instead. Just make sure you aren't trying to use CircuitPython on Python 2.x, it isn't supported!

CircuitPython & Python Usage

To demonstrate usage we will initialize the sensor and read it's onboard temperature sensor from the board's Python REPL.

Run the following code to import the necessary modules and initialize the I2C connection with the sensor:

I2C Initialization

For I2C, use the following to create your sensor object:

```
import board
import busio
from adafruit_as726x import AS726x_I2C

i2c = busio.I2C(board.SCL, board.SDA)
sensor = Adafruit_AS726x_I2C(i2c)
```

UART Initialization - CircuitPython

For UART usage on a board (like a Feather) running CircuitPython, use the following to create your sensor object:

```
import board
import busio
from adafruit_as726x import AS726x_UART

uart = busio.UART(board.TX, board.RX)
sensor = Adafruit_AS726x_UART(uart)
```

Usage

Now you can read the temperature property from the sensor:

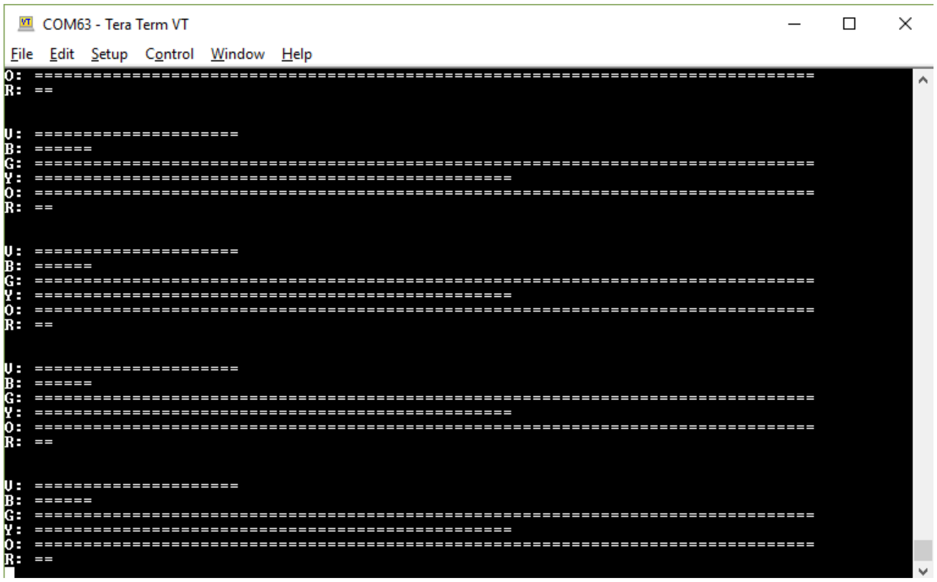
```
print('Temperature: {0}C'.format(sensor.temperature))
```

As long as you get a reasonable temperature (usually around 28 degrees C) you know your sensor is wired up correctly and working!

Below is a complete example that reads all color channels and prints them out as a graph in the REPL. Save this as **code.py** on your board and open the REPL to see the output.

```
Temporarily unable to load content:
```

Running the above code should something like this in your REPL:



```
O: ================================================================================
R: ==


U: =====================
B: ======
G: ================================================================================
Y: =================================================
O: ================================================================================
R: ==


U: =====================
B: ======
G: ================================================================================
Y: =================================================
O: ================================================================================
R: ==


U: =====================
B: ======
G: ================================================================================
Y: =================================================
O: ================================================================================
R: ==


U: =====================
B: ======
G: ================================================================================
Y: =================================================
O: ================================================================================
R: ==
```

Python Docs

Python Docs (https://adafru.it/AUp)

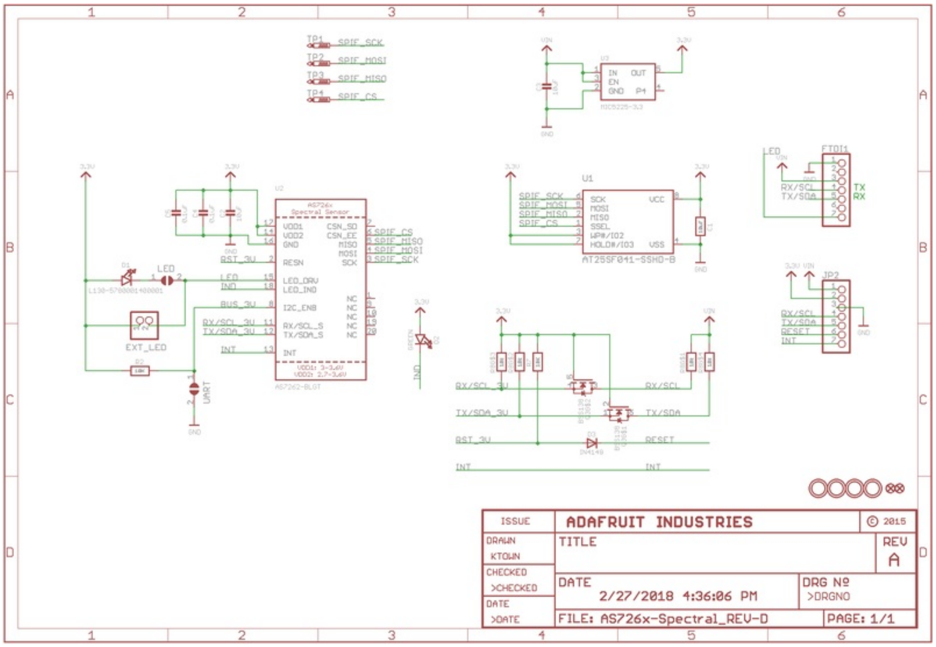
Downloads

Documents

- AS726x Datasheet (https://adafru.it/C1O)
- AS726x Arduino Driver (https://adafru.it/C1P)
- AS726x CircuitPython Driver (https://adafru.it/C1M)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/aP3)
- AS7262 Breakout PCB files (EAGLE format) (https://adafru.it/C1Q)

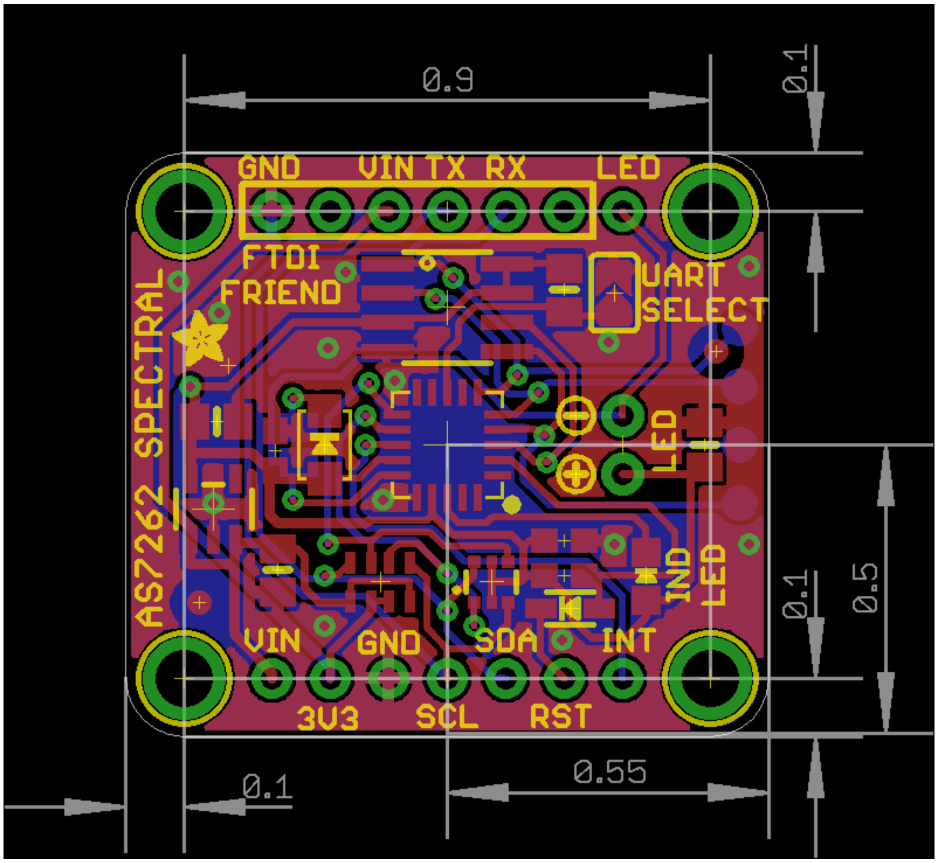
Schematic

click to enlarge



Dimensions

in inches. Click to enlarge


0.9
0.1
GND
VIN
TX
RX
LED
FTDI
FRIEND
UART
SELECT
AS7262 SPECTRAL
LED
IND
LED
0.1
0.5
VIN
GND
SDA
INT
3V3
SCL
RST
0.1
0.55