Adafruit AM2320 Sensor

Created by lady ada
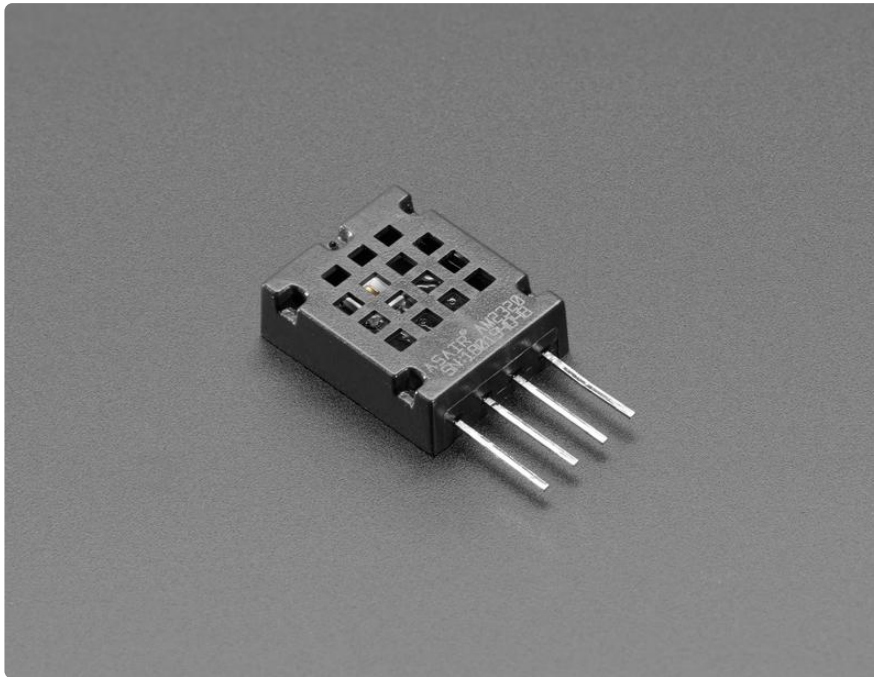

https://learn.adafruit.com/adafruit-am2320-temperature-humidity-i2c-sensor
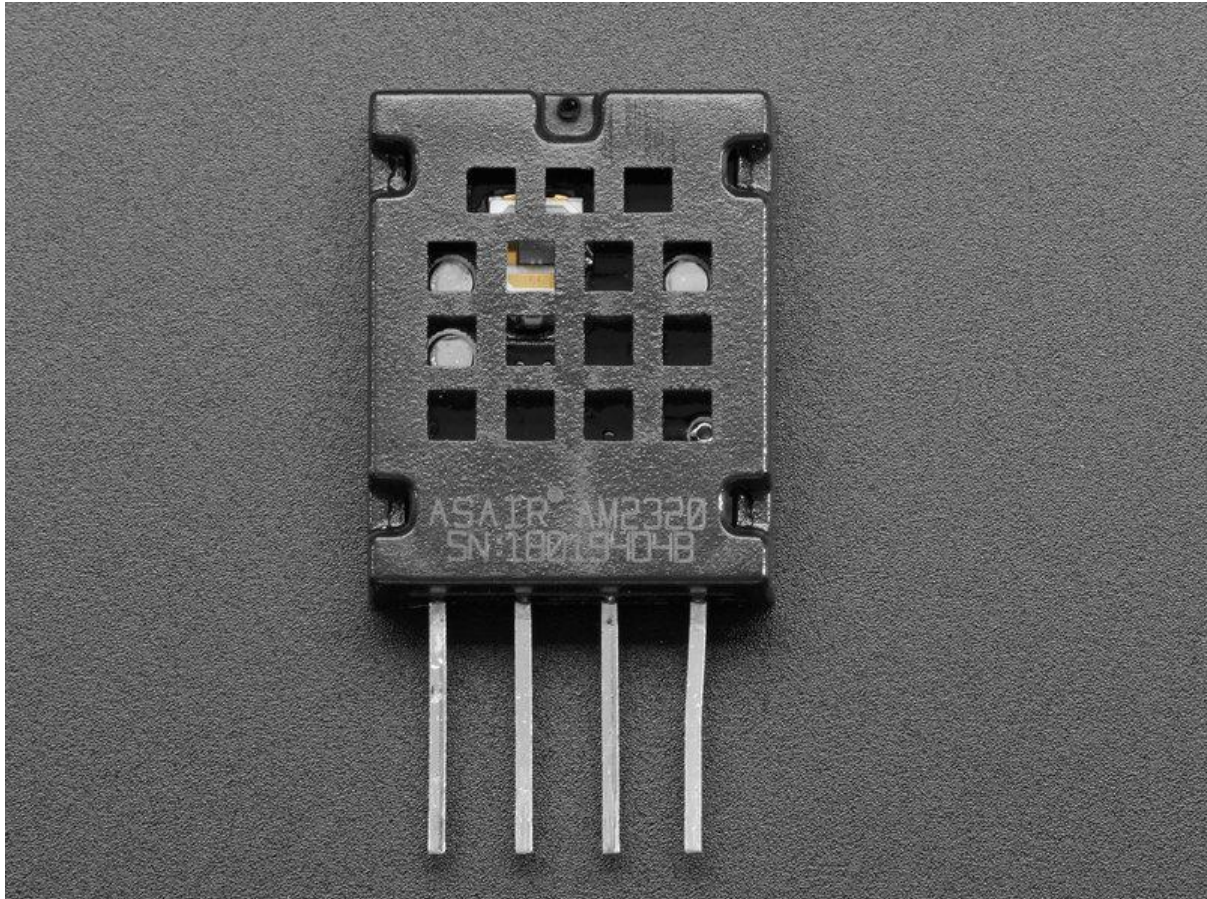
Last updated on 2022-12-01 03:10:50 PM EST

Table of Contents

Overview

This little sensor looks an awful lot like the popular DHT11/DHT22 temperature and humidity sensors, but unlike classic DHT sensors, it has an I2C interface! That's right, you do not need to use a bit-bang timing-specific protocol to talk to the AM2320, it uses plain-old-I2C. Whew, that makes things a little easier, doesn't it?

But! We'll let you know, this sensor is not well documented like our other, fancier I2C temperature & humidity sensors. The datasheet mentions it has 3% humidity accuracy and 0.5C temperature accuracy, but we're not very trusting of the specifications. So, while this sensor does seem to work, it's not recommended for anything where you care about any sort of guaranteed accuracy. Temperature is probably correct to 2-3 degrees Celsius. Humidity is probably within 5-10%.

That said, for maker and IoT projects? You can't beat the simplicity and price! And we've got ready-to-go working Arduino and CircuitPython code to use it.

Each order comes with one AM2320, a low-cost temperature and humidity sensor. You just provide any microcontroller that can run our Arduino or CircuitPython library, and two I2C pull-up resistors (not included).

Pinouts
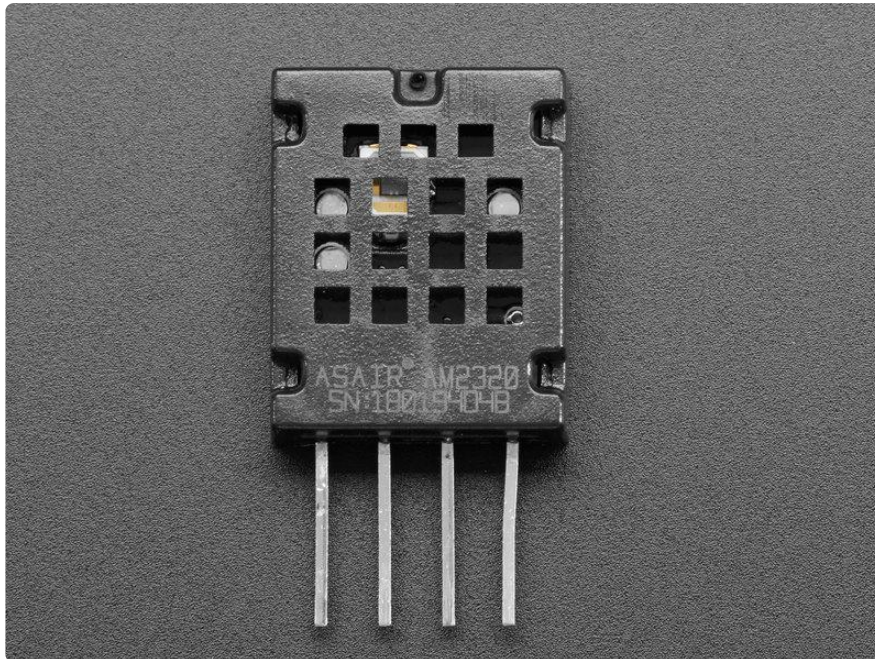
Starting from the left the pins are

1. VDD - this is power in, can be 3-5VDC
2. SDA - I2C data in/out, requires a pullup of 2-10K to VDD
3. GND - this is signal/power ground
4. SCL - I2C clock in, requires a pullup of 2-10K to VDD

Arduino Usage

You can easily wire this breakout to any microcontroller, we'll be using an Arduino. For another kind of microcontroller, just make sure it has I2C, then port the code - its pretty simple stuff!
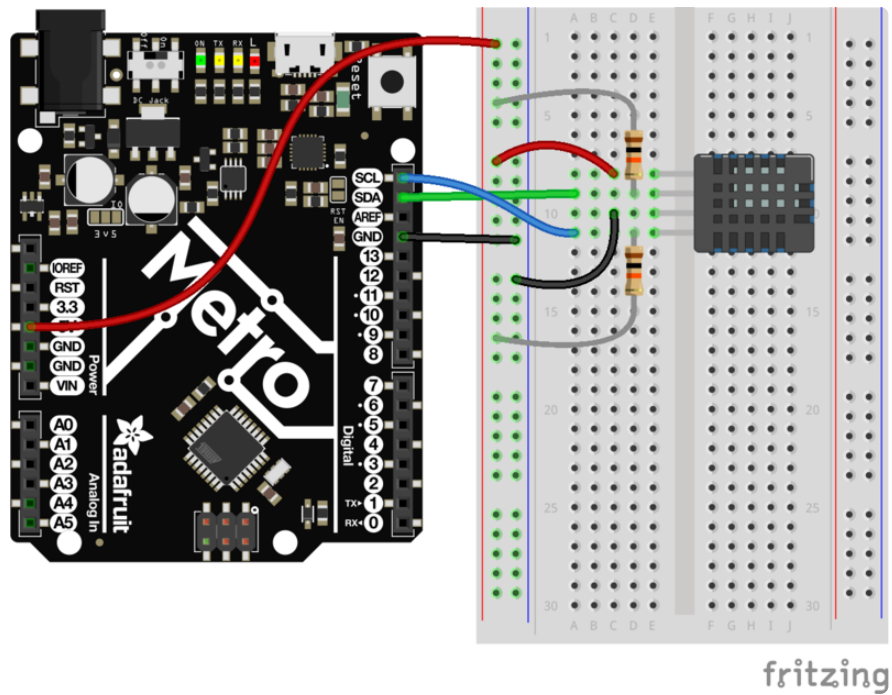
fritzing

[metrowire fritzing]

- 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 microcontroller
- Connect the SDA pin to the I2C data SDA pin on your microcontroller.

You will also need to add two I2C pullup resistors if your board does not already have them. You can use 2.2K - 10K but we will just use 10K. The resistors go from VDD to SCL and SDA each.
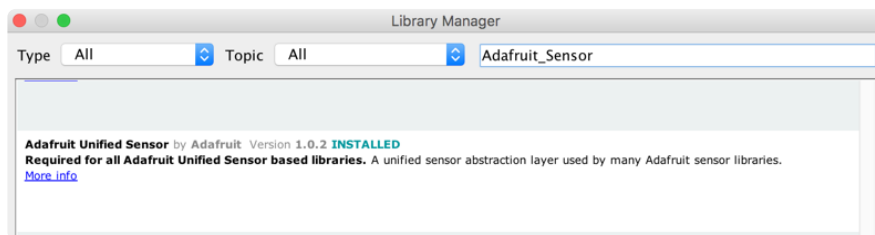
The AM2320 has a default I2C address of 0x5C and cannot be changed.

Install Adafruit Sensor

Open up the Library Manager in the Arduino IDE...
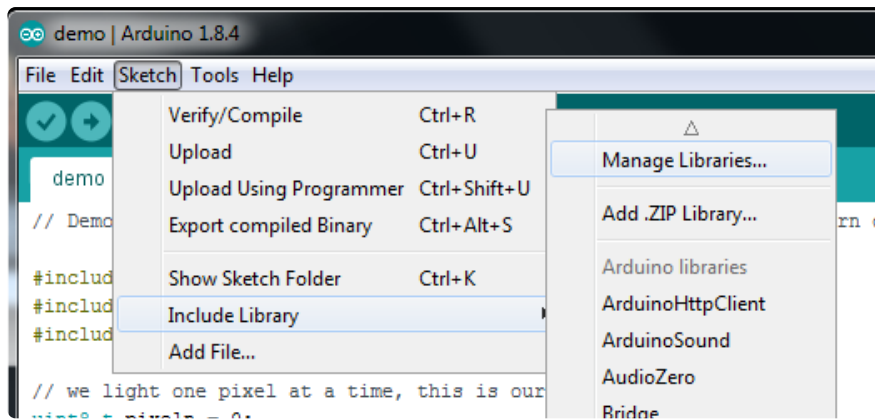
Search for Adafruit_Sensor to locate the Unified Sensor library and Install it

Download Adafruit_AM2320

To begin reading sensor data, you will need to also install the Adafruit_AM2320 library ().
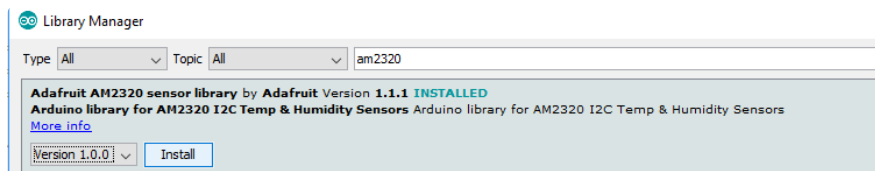
We also have a great tutorial on Arduino library installation at:

http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use ()

Load Demo

Open up File->Examples->Adafruit_AM2320->am2320_basictest and upload to your microcontroller wired up to the sensor

Thats it! Now open up the serial terminal window at 9600 speed to begin the test.

You can try breathing on the sensor to increase the humidity.

Python & CircuitPython

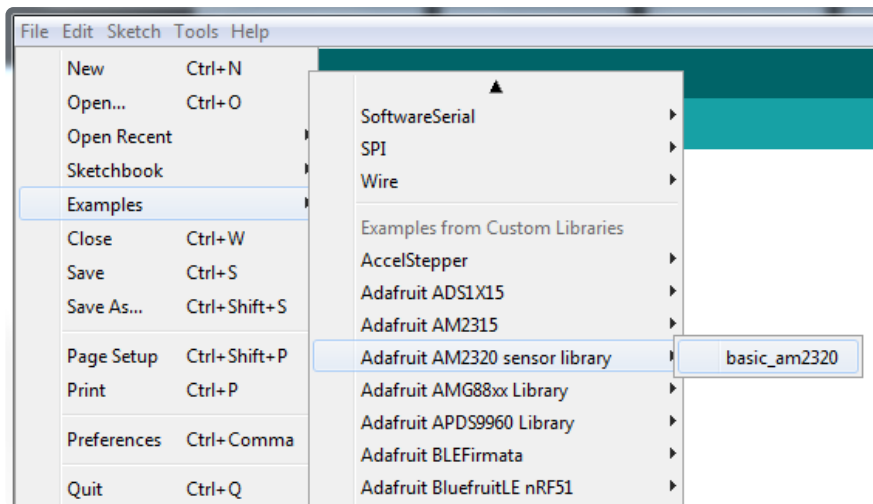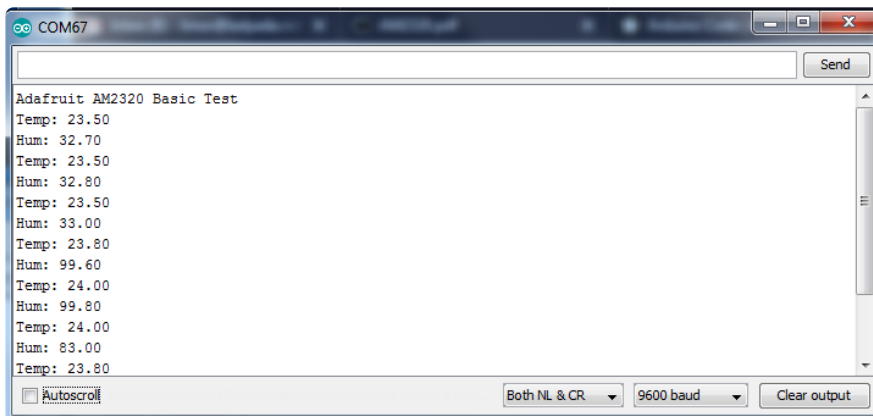It's easy to use the AM2320 sensor with Python or CircuitPython and the Adafruit CircuitPython AM2320 () module. This module allows you to easily write Python code that reads the humidity and temperature from the sensor.

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 ().

One thing to watch out for with the AM2320 is that it goes into 'sleep' mode, you may find if you are scanning the i2c bus that the device does not appear - try scanning again and it will appear the second time!

CircuitPython Microcontroller Wiring

First wire up a AM2320 to your board exactly as shown on the previous pages for Arduino using an I2C connection. Here's an example of wiring a ItsyBitsy M0 to the sensor with I2C:
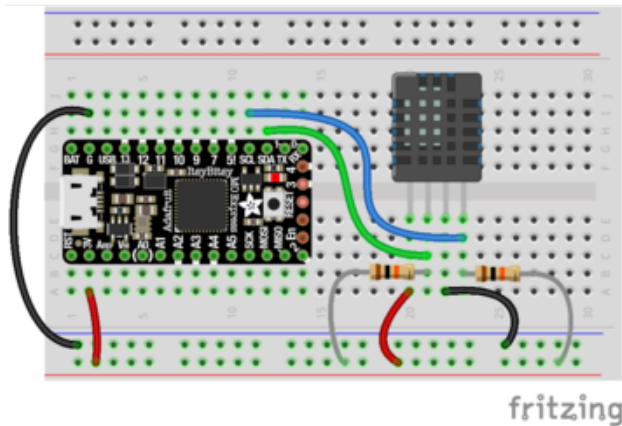
Board 3V to sensor VIN
Board GND to sensor GND
Board SCL to sensor SCL
Board SDA to sensor SDA

itsywire fritzing

Python Computer Wiring

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 ().

Here's the Raspberry Pi wired with I2C:
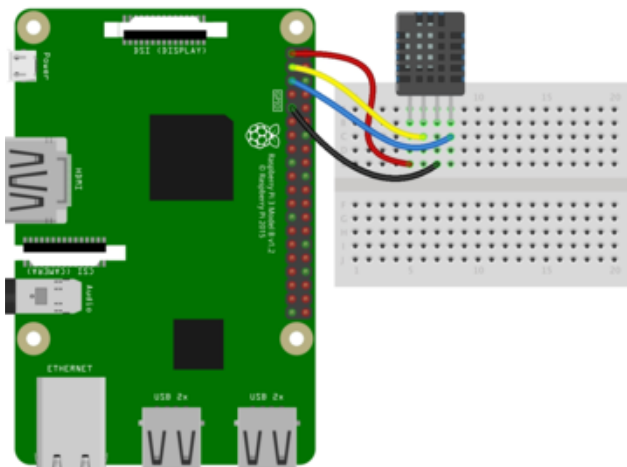
Pi 3V3 to sensor VIN
Pi GND to sensor GND
Pi SCL to sensor SCL
Pi SDA to sensor SDA

The Raspberry Pi has built-in pull up resistors on SDA/SCL so we don't include them here. If you are using some other computer, check if you need to add your own 2-10K pullups to 3.3V!

CircuitPython Installation of AM2320 Library

Next you'll need to install the Adafruit CircuitPython AM2320 () library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () 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 (). Our introduction guide has a great page on how to install the library bundle () for both express and non-express boards.

For non-express boards like the Trinket M0 or Gemma M0, you'll need to manually install the necessary libraries from the bundle:

- adafruit_am2320.mpy
- adafruit_bus_device

You can also download the adafruit_am2320.mpy from its releases page on Github ().

Before continuing make sure your board's lib folder or root filesystem has the adafruit _am2320.mpy, and adafruit_bus_device files and folders copied over.

Next connect to the board's serial REPL () so you are at the CircuitPython >>> prompt.

Python Installation of AM2320 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 ()!

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

- `sudo pip3 install adafruit-circuitpython-am2320`

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 the usage of the sensor we'll initialize it and read the humidity and temperature from the board's Python REPL.

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

```
import board
import adafruit_am2320
i2c = board.I2C()
sensor = adafruit_am2320.AM2320(i2c)
```

Now you're ready to read values from the sensor using any of these properties:

- relative_humidity - The relative humidity measured by the sensor, this is a value from 0-100%.
- temperature - The temperature measured by the sensor, a value in degrees Celsius.

For example, to print the temperature and relative humidity:

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

```
>>> print('Humidity: {0}%'.format(sensor.relative_humidity))
Humidity: 50.9%
>>> print('Temperature: {0}C'.format(sensor.temperature))
Temperature: 21.5C
```

If you want to take another reading, wait 2 seconds before requesting temperature or relative_humidity again. Data in the sensor is only updated every 2 seconds!

That's all there is to using the AM2320 with CircuitPython!

Below is a complete example that measures the sensor readings and prints them every two seconds. Save this as code.py on your board and open the REPL to see the output.

Full Example Code

```python
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import adafruit_am2320

# create the I2C shared bus
i2c = board.I2C()  # uses board.SCL and board.SDA
# i2c = board.STEMMA_I2C()  # For using the built-in STEMMA QT connector on a
microcontroller
am = adafruit_am2320.AM2320(i2c)

while True:
    print("Temperature: ", am.temperature)
    print("Humidity: ", am.relative_humidity)
    time.sleep(2)
```

Python Docs

Python Docs ()