Adafruit SI7021 Library Documentation

Release 1.0

Radomir Dopieralski

Jan 25, 2021

Contents

1 Dependencies **3**

2 Installing from PyPI **5**

3 Usage Notes **7**

4 Contributing **9**

5 Documentation **11**

6 Table of Contents **13**

6.1 Simple test 13

6.2 `adafruit_si7021` 13

6.2.1 Implementation Notes 14

7 Indices and tables **15**

Python Module Index **17**

Index **19**

CircuitPython module for use with the SI7021 Temperature and Humidity Sensor.

CHAPTER 1

Dependencies

This driver depends on:

- Adafruit CircuitPython
- Bus Device

Please ensure all dependencies are available on the CircuitPython filesystem. This is easily achieved by downloading the Adafruit library and driver bundle.

CHAPTER 2

Installing from PyPI

On supported GNU/Linux systems like the Raspberry Pi, you can install the driver locally from PyPI. To install for current user:

```
pip3 install adafruit-circuitpython-si7021
```

To install system-wide (this may be required in some cases):

```
sudo pip3 install adafruit-circuitpython-si7021
```

To install in a virtual environment in your current project:

```
mkdir project-name && cd project-name
python3 -m venv .env
source .env/bin/activate
pip3 install adafruit-circuitpython-si7021
```

CHAPTER 3

Usage Notes

Of course, you must import the library to use it:

```
import adafruit_si7021
```

This driver takes an instantiated and active I2C object (from the `busio` or the `bitbangio` library) as an argument to its constructor. The way to create an I2C object depends on the board you are using. For boards with labeled SCL and SDA pins, you can:

```
from busio import I2C
from board import SCL, SDA

i2c = I2C(SCL, SDA)
```

Once you have created the I2C interface object, you can use it to instantiate the sensor object:

```
sensor = adafruit_si7021.SI7021(i2c)
```

And then you can start measuring the temperature and humidity:

```
print(sensor.temperature)
print(sensor.relative_humidity)
```

CHAPTER 4

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

CHAPTER 5

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/si7021_simpletest.py

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

"""
Initializes the sensor, gets and prints readings every two seconds.
"""
import time
import board
import adafruit_si7021

# Create library object using our Bus I2C port
sensor = adafruit_si7021.SI7021(board.I2C())

while True:
    print("\nTemperature: %0.1f C" % sensor.temperature)
    print("Humidity: %0.1f %%" % sensor.relative_humidity)
    time.sleep(2)
```

6.2 adafruit_si7021

This is a CircuitPython driver for the SI7021 temperature and humidity sensor.

- Author(s): Radomir Dopieralski, Chris Balmer, Ian Grant

6.2.1 Implementation Notes

Hardware:

- Adafruit Si7021 Temperature & Humidity Sensor Breakout Board (Product ID: 3251)

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: https://github.com/adafruit/circuitpython/releases
- Adafruit's Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class `adafruit_si7021.`**`SI7021`**(*i2c_bus*, *address=64*)

A driver for the SI7021 temperature and humidity sensor.

Parameters

- **i2c_bus** – The `busio.I2C` object to use. This is the only required parameter.
- **address** (*int*) – (optional) The I2C address of the device.

device_identifier

A device identifier (model type) string.

relative_humidity

The measured relative humidity in percent.

serial_number

The device's unique ID (serial number).

start_measurement(*what*)

Starts a measurement.

Starts a measurement of either `HUMIDITY` or `TEMPERATURE` depending on the `what` argument. Returns immediately, and the result of the measurement can be retrieved with the `temperature` and `relative_humidity` properties. This way it will take much less time.

This can be useful if you want to start the measurement, but don't want the call to block until the measurement is ready – for instance, when you are doing other things at the same time.

temperature

The measured temperature in degrees Celcius.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

adafruit_si7021, 13

Index

A

adafruit_si7021 (*module*), 13

D

device_identifier (*adafruit_si7021.SI7021 attribute*), 14

R

relative_humidity (*adafruit_si7021.SI7021 attribute*), 14

S

serial_number (*adafruit_si7021.SI7021 attribute*), 14

SI7021 (*class in adafruit_si7021*), 14

start_measurement() (*adafruit_si7021.SI7021 method*), 14

T

temperature (*adafruit_si7021.SI7021 attribute*), 14