
AdafruitDS18X20 Library Documentation

Release 1.0

Carter Nelson

Feb 10, 2021

Contents

1	Dependencies	3
2	Installing from PyPI	5
3	Usage Example	7
4	Contributing	9
5	Documentation	11
6	Table of Contents	13
6.1	Simple test	13
6.2	adafruit_ds18x20	14
7	Indices and tables	17
	Python Module Index	19
	Index	21

CircuitPython driver for Dallas 1-Wire temperature sensor.

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Adafruit OneWire](#)

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-ds18x20
```

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

```
sudo pip3 install adafruit-circuitpython-ds18x20
```

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-ds18x20
```


CHAPTER 3

Usage Example

```
import board
from adafruit_owewire.bus import OneWireBus
from adafruit_ds18x20 import DS18X20
ow_bus = OneWireBus(board.D2)
ds18 = DS18X20(ow_bus, ow_bus.scan()[0])
ds18.temperature
```


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](#).

6.1 Simple test

Ensure your device works with these simple tests.

Listing 1: examples/ds18x20_simpletest.py

```
1  # SPDX-FileCopyrightText: 2020 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # Simple demo of printing the temperature from the first found DS18x20 sensor every_
   ↪second.
5  # Author: Tony DiCola
6
7  # A 4.7Kohm pullup between DATA and POWER is REQUIRED!
8
9  import time
10 import board
11 from adafruit_owewire.bus import OneWireBus
12 from adafruit_ds18x20 import DS18X20
13
14
15 # Initialize one-wire bus on board pin D5.
16 ow_bus = OneWireBus(board.D5)
17
18 # Scan for sensors and grab the first one found.
19 ds18 = DS18X20(ow_bus, ow_bus.scan()[0])
20
21 # Main loop to print the temperature every second.
22 while True:
23     print("Temperature: {0:0.3f}C".format(ds18.temperature))
24     time.sleep(1.0)
```

Listing 2: examples/ds18x20_asyncntest.py

```
1  # SPDX-FileCopyrightText: 2020 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  # Simple demo of printing the temperature from the first found DS18x20 sensor every_
   ↪ second.
5  # Using the asynchronous functions start_temperature_read() and
6  # read_temperature() to allow the main loop to keep processing while
7  # the conversion is in progress.
8  # Author: Louis Bertrand, based on original by Tony DiCola
9
10 # A 4.7Kohm pullup between DATA and POWER is REQUIRED!
11
12 import time
13 import board
14 from adafruit_onewire.bus import OneWireBus
15 from adafruit_ds18x20 import DS18X20
16
17
18 # Initialize one-wire bus on board pin D1.
19 ow_bus = OneWireBus(board.D1)
20
21 # Scan for sensors and grab the first one found.
22 ds18 = DS18X20(ow_bus, ow_bus.scan()[0])
23 ds18.resolution = 12
24
25 # Main loop to print the temperature every second.
26 while True:
27     conversion_delay = ds18.start_temperature_read()
28     conversion_ready_at = time.monotonic() + conversion_delay
29     print("waiting", end="")
30     while time.monotonic() < conversion_ready_at:
31         print(".", end="")
32         time.sleep(0.1)
33     print("\nTemperature: {0:0.3f}C\n".format(ds18.read_temperature()))
34     time.sleep(1.0)
```

6.2 adafruit_ds18x20

Driver for Dallas 1-Wire temperature sensor.

- Author(s): Carter Nelson

class adafruit_ds18x20.DS18X20 (bus, address)

Class which provides interface to DS18X20 temperature sensor.

read_temperature()

Read the temperature. No polling of the conversion busy bit (assumes that the conversion has completed).

resolution

The programmable resolution. 9, 10, 11, or 12 bits.

start_temperature_read()

Start asynchronous conversion, returns immediately. Returns maximum conversion delay [seconds] based on resolution.

temperature

The temperature in degrees Celsius.

CHAPTER 7

Indices and tables

- `genindex`
- `modindex`
- `search`

a

adafruit_ds18x20, [14](#)

A

`adafruit_ds18x20` (*module*), [14](#)

D

`DS18X20` (*class in adafruit_ds18x20*), [14](#)

R

`read_temperature()` (*adafruit_ds18x20.DS18X20 method*), [14](#)

`resolution` (*adafruit_ds18x20.DS18X20 attribute*), [14](#)

S

`start_temperature_read()`
(*adafruit_ds18x20.DS18X20 method*), [14](#)

T

`temperature` (*adafruit_ds18x20.DS18X20 attribute*), [14](#)