
AdafruitDS18X20 Library Documentation

Release 1.0

Carter Nelson

Oct 21, 2019

Contents

1	Dependencies	3
2	Usage Example	5
3	Contributing	7
4	Building locally	9
4.1	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_ds18x20	12
6	Indices and tables	13
	Python Module Index	15
	Index	17

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

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 3

Contributing

Contributions are welcome! Please read our [Code of Conduct](#) before contributing to help this project stay welcoming.

CHAPTER 4

Building locally

To build this library locally you'll need to install the `circuitpython-build-tools` package.

```
python3 -m venv .env
source .env/bin/activate
pip install circuitpython-build-tools
```

Once installed, make sure you are in the virtual environment:

```
source .env/bin/activate
```

Then run the build:

```
circuitpython-build-bundles --filename_prefix adafruit-circuitpython-ds18x20 --
↪library_location .
```

4.1 Sphinx documentation

Sphinx is used to build the documentation based on rST files and comments in the code. First, install dependencies (feel free to reuse the virtual environment from above):

```
python3 -m venv .env
source .env/bin/activate
pip install Sphinx sphinx-rtd-theme
```

Now, once you have the virtual environment activated:

```
cd docs
sphinx-build -E -W -b html . _build/html
```

This will output the documentation to `docs/_build/html`. Open the `index.html` in your browser to view them. It will also (due to `-W`) error out on any warning like Travis will. This is a good way to locally verify it will pass.

5.1 Simple test

Ensure your device works with these simple tests.

Listing 1: examples/ds18x20_simpletest.py

```
1  # Simple demo of printing the temperature from the first found DS18x20 sensor every_
   ↳second.
2  # Author: Tony DiCola
3  import time
4
5  import board
6
7  from adafruit_owewire.bus import OneWireBus
8  from adafruit_ds18x20 import DS18X20
9
10
11 # Initialize one-wire bus on board pin D5.
12 ow_bus = OneWireBus(board.D5)
13
14 # Scan for sensors and grab the first one found.
15 ds18 = DS18X20(ow_bus, ow_bus.scan()[0])
16
17 # Main loop to print the temperature every second.
18 while True:
19     print('Temperature: {0:0.3f}C'.format(ds18.temperature))
20     time.sleep(1.0)
```

Listing 2: examples/ds18x20_asyncntest.py

```
1  # Simple demo of printing the temperature from the first found DS18x20 sensor every_
   ↳second.
2  # Using the asynchronous functions start_temperature_read() and
```

(continues on next page)

(continued from previous page)

```
3  # read_temperature() to allow the main loop to keep processing while
4  # the conversion is in progress.
5  # Author: Louis Bertrand, based on original by Tony DiCola
6
7  import time
8
9  import board
10
11  from adafruit_owewire.bus import OneWireBus
12  from adafruit_ds18x20 import DS18X20
13
14
15  # Initialize one-wire bus on board pin D1.
16  ow_bus = OneWireBus(board.D1)
17
18  # Scan for sensors and grab the first one found.
19  ds18 = DS18X20(ow_bus, ow_bus.scan()[0])
20  ds18.resolution = 12
21
22  # Main loop to print the temperature every second.
23  while True:
24      conversion_delay = ds18.start_temperature_read()
25      conversion_ready_at = time.monotonic() + conversion_delay
26      print("waiting", end="")
27      while time.monotonic() < conversion_ready_at:
28          print(".", end="")
29          time.sleep(0.1)
30      print('\nTemperature: {0:0.3f}C\n'.format(ds18.read_temperature()))
31      time.sleep(1.0)
```

5.2 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 6

Indices and tables

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

a

adafruit_ds18x20, [12](#)

A

`adafruit_ds18x20` (*module*), [12](#)

D

`DS18X20` (*class in adafruit_ds18x20*), [12](#)

R

`read_temperature()` (*adafruit_ds18x20.DS18X20 method*), [12](#)

`resolution` (*adafruit_ds18x20.DS18X20 attribute*), [12](#)

S

`start_temperature_read()`
(*adafruit_ds18x20.DS18X20 method*), [12](#)

T

`temperature` (*adafruit_ds18x20.DS18X20 attribute*), [12](#)