

---

# **AdafruitDS18X20 Library Documentation**

***Release 1.0***

**Carter Nelson**

**Feb 25, 2021**



---

## Contents

---

<b>1</b>	<b>Dependencies</b>	<b>3</b>
<b>2</b>	<b>Usage Example</b>	<b>5</b>
<b>3</b>	<b>Contributing</b>	<b>7</b>
<b>4</b>	<b>Documentation</b>	<b>9</b>
<b>5</b>	<b>Table of Contents</b>	<b>11</b>
5.1	Simple test . . . . .	11
5.2	adafruit_ds18x20 . . . . .	12
<b>6</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



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

---

### Documentation

---

For information on building library documentation, please check out [this guide](#).



## 5.1 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)
```

## 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](#)