

---

# AdafruitUS100 Library Documentation

*Release 1.0*

**ladyada**

**Oct 25, 2021**



---

## Contents

---

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>1</b> | <b>Dependencies</b>            | <b>3</b>  |
| 1.1      | Installing from PyPI . . . . . | 3         |
| <b>2</b> | <b>Usage Example</b>           | <b>5</b>  |
| <b>3</b> | <b>Documentation</b>           | <b>7</b>  |
| <b>4</b> | <b>Contributing</b>            | <b>9</b>  |
| <b>5</b> | <b>Documentation</b>           | <b>11</b> |
| <b>6</b> | <b>Table of Contents</b>       | <b>13</b> |
| 6.1      | Simple test . . . . .          | 13        |
| 6.2      | adafruit_us100 . . . . .       | 14        |
| 6.2.1    | Implementation Notes . . . . . | 14        |
| <b>7</b> | <b>Indices and tables</b>      | <b>15</b> |
|          | <b>Python Module Index</b>     | <b>17</b> |
|          | <b>Index</b>                   | <b>19</b> |



CircuitPython library for reading distance and temperature via US-100 ultra-sonic sensor



This driver depends on:

- [Adafruit CircuitPython](#)

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

## 1.1 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-us100
```

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

```
sudo pip3 install adafruit-circuitpython-us100
```

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-us100
```





## CHAPTER 2

---

### Usage Example

---

```
import time
import board
import busio
import adafruit_us100

uart = busio.UART(board.TX, board.RX, baudrate=9600)
# Create a US-100 module instance.
us100 = adafruit_us100.US100(uart)

while True:
    print("-----")
    print("Temperature: ", us100.temperature)
    print("Distance: ", us100.distance)
    time.sleep(0.5)
```



## CHAPTER 3

---

### Documentation

---

API documentation for this library can be found on [Read the Docs](#).



## 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 this simple test.

Listing 1: examples/us100\_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5
6  # For use with a microcontroller:
7  import board
8  import busio
9  import adafruit_us100
10
11  uart = busio.UART(board.TX, board.RX, baudrate=9600)
12
13  # For use with USB-to-serial cable:
14  # import serial
15  # import adafruit_us100
16  # uart = serial.Serial("/dev/ttyUSB0", baudrate=9600, timeout=1)
17
18  # For use with Raspberry Pi/Linux:
19  # import serial
20  # import adafruit_us100
21  # uart = serial.Serial("/dev/ttyS0", baudrate=9600, timeout=1)
22
23  us100 = adafruit_us100.US100(uart)
24
25  while True:
26      print("-----")
27      print("Temperature: ", us100.temperature)
```

(continues on next page)

(continued from previous page)

```
28     time.sleep(0.5)
29     print("Distance: ", us100.distance)
30     time.sleep(0.5)
```

## 6.2 adafruit\_us100

CircuitPython library for reading distance and temperature via US-100 ultra-sonic sensor

- Author(s): ladyada

### 6.2.1 Implementation Notes

#### Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>

**class** `adafruit_us100.US100` (*uart*)  
Control a US-100 ultrasonic range sensor.

#### **distance**

Return the distance measured by the sensor in cm. This is the function that will be called most often in user code. If no signal is received, return `None`. This can happen when the object in front of the sensor is too close, the wiring is incorrect or the sensor is not found. If the signal received is not 2 bytes, return `None`. This means either the sensor was moving too fast to be pointing in the right direction to pick up the ultrasonic signal when it bounced back (less likely), or the object off of which the signal bounced is too far away for the sensor to handle. In my experience, the sensor can not detect objects over 460 cm away.  
:return: Distance in centimeters. :rtype: float

#### **temperature**

Return the on-chip temperature, in Celsius

## CHAPTER 7

---

### Indices and tables

---

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



**a**

adafruit\_us100, 14



## A

adafruit\_us100 (*module*), 14

## D

distance (*adafruit\_us100.US100 attribute*), 14

## T

temperature (*adafruit\_us100.US100 attribute*), 14

## U

US100 (*class in adafruit\_us100*), 14