# AdafruitDS2413 Library Documentation

*Release 1.0*

**Carter Nelson**

**Sep 23, 2019**

# Contents

CircuitPython driver for the DS2413 one wire 2 channel GPIO breakout.

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 time
import board
from adafruit_onewire.bus import OneWireBus
import adafruit_ds2413

ow_bus = OneWireBus(board.D2)
ds = adafruit_ds2413.DS2413(ow_bus, ow_bus.scan()[0])

led = ds.IOA
button = ds.IOB
button.direction = adafruit_ds2413.INPUT

while not button.value:
    led.value = True
    time.sleep(0.5)
    led.value = False
    time.sleep(0.5)
```

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-ds2413 --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.

# CHAPTER 5

# Table of Contents

## 5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/ds2413_simpletest.py

```
# This example shows how to access the DS2413 pins and use them for both input
# and output. In this example, it is assumed an LED is attached to IOA and a
# button is attached to IOB. See the datasheet for details about how to
# interface the external hardware (it is different than most Arduino examples).
import time
import board
from adafruit_onewire.bus import OneWireBus
import adafruit_ds2413

# Create OneWire bus
ow_bus = OneWireBus(board.D2)

# Create the DS2413 object from the first one found on the bus
ds = adafruit_ds2413.DS2413(ow_bus, ow_bus.scan()[0])

# LED on IOA
led = ds.IOA

# button on IOB
button = ds.IOB
button.direction = adafruit_ds2413.INPUT

# Loop forever
while True:
    # Check for button press
    if button.value:
        # Print a message.
```

(continues on next page)

(continued from previous page)

```
        print("Button pressed!")
        # Toggle LED
        led.value = not led.value
        # A little debounce
        time.sleep(0.25)
```

## 5.2 adafruit_ds2413

CircuitPython driver for the DS2413 one wire 2 channel GPIO breakout.

- Author(s): Carter Nelson

**class** adafruit_ds2413.**DS2413**(*bus*, *address*)
Class which provides interface to DS2413 GPIO breakout.

**IOA**
The pin object for channel A.

**IOB**
The pin object for channel B.

**pio_state**
The state of both PIO channels.

**class** adafruit_ds2413.**DS2413Pin**(*number*, *host*, *direction=1*)
Class which provides interface to single DS2413 GPIO pin.

**direction**
The direction of the pin, either INPUT or OUTPUT.

**value**
The pin state if configured as INPUT. The output latch state if configured as OUTPUT. True is HIGH/ON, False is LOW/OFF.

CHAPTER 6

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

## a

# Index