
Adafruit74HC595 Library Documentation

Release 1.0

Kattni Rembor

Jan 20, 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_74hc595	13
6.2.1	Implementation Notes	14
7	Indices and tables	15
	Python Module Index	17
	Index	19

CircuitPython driver for 74HC595 shift register.

CHAPTER 1

Dependencies

This driver depends on:

- [Adafruit CircuitPython](#)
- [Bus Device](#)

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-74hc595
```

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

```
sudo pip3 install adafruit-circuitpython-74hc595
```

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-74hc595
```


CHAPTER 3

Usage Example

```
import board
import adafruit_74hc595
import busio
import digitalio
import time

spi = busio.SPI(board.SCK, MOSI=board.MOSI)

latch_pin = digitalio.DigitalInOut(board.D5)
sr = adafruit_74hc595.ShiftRegister74HC595(spi, latch_pin)

pin1 = sr.get_pin(1)

while True:
    pin1.value = True
    time.sleep(1)
    pin1.value = False
    time.sleep(1)
```


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/74hc595_simpletest.py

```
1  # SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
2  # SPDX-License-Identifier: MIT
3
4  import time
5  import board
6  import busio
7  import digitalio
8  import adafruit_74hc595
9
10 spi = busio.SPI(board.SCK, MOSI=board.MOSI)
11
12 latch_pin = digitalio.DigitalInOut(board.D5)
13 sr = adafruit_74hc595.ShiftRegister74HC595(spi, latch_pin)
14
15 pin1 = sr.get_pin(1)
16
17 while True:
18     pin1.value = True
19     time.sleep(1)
20     pin1.value = False
21     time.sleep(1)
```

6.2 adafruit_74hc595

CircuitPython driver for 74HC595 shift register.

- Author(s): Kattni Rembor, Tony DiCola

6.2.1 Implementation Notes

Hardware:

“* 74HC595 Shift Register - 3 pack”

Software and Dependencies:

- Adafruit CircuitPython firmware for the supported boards: <https://github.com/adafruit/circuitpython/releases>
- Adafruit’s Bus Device library: https://github.com/adafruit/Adafruit_CircuitPython_BusDevice

class `adafruit_74hc595.DigitalInOut` (*pin_number, shift_register_74hc595*)

Digital input/output of the 74HC595. The interface is exactly the same as the `digitalio.DigitalInOut` class, however note that by design this device is OUTPUT ONLY! Attempting to read inputs or set direction as input will raise an exception.

direction

Direction can only be set to OUTPUT.

pull

Pull-up/down not supported, return None for no pull-up/down.

switch_to_input (***kwargs*)

`switch_to_input` is not supported.

switch_to_output (*value=False, **kwargs*)

DigitalInOut `switch_to_output`

value

The value of the pin, either True for high or False for low.

class `adafruit_74hc595.ShiftRegister74HC595` (*spi, latch*)

Initialise the 74HC595 on specified SPI bus.

get_pin (*pin*)

Convenience function to create an instance of the `DigitalInOut` class pointing at the specified pin of this 74HC595 device .

gpio

The raw GPIO output register. Each bit represents the output value of the associated pin (0 = low, 1 = high).

CHAPTER 7

Indices and tables

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

a

adafruit_74hc595, [13](#)

A

`adafruit_74hc595` (*module*), [13](#)

D

`DigitalInOut` (*class in* `adafruit_74hc595`), [14](#)

`direction` (*adafruit_74hc595.DigitalInOut attribute*), [14](#)

G

`get_pin()` (*adafruit_74hc595.ShiftRegister74HC595 method*), [14](#)

`gpio` (*adafruit_74hc595.ShiftRegister74HC595 attribute*), [14](#)

P

`pull` (*adafruit_74hc595.DigitalInOut attribute*), [14](#)

S

`ShiftRegister74HC595` (*class in* `adafruit_74hc595`), [14](#)

`switch_to_input()` (*adafruit_74hc595.DigitalInOut method*), [14](#)

`switch_to_output()` (*adafruit_74hc595.DigitalInOut method*), [14](#)

V

`value` (*adafruit_74hc595.DigitalInOut attribute*), [14](#)