
Adafruit74HC595 Library Documentation

Release 1.0

Kattni Rembor

Oct 17, 2019

Contents

1	Dependencies	3
1.1	Installing from PyPI	3
2	Usage Example	5
3	Contributing	7
4	Building locally	9
4.1	Zip release files	9
4.2	Sphinx documentation	9
5	Table of Contents	11
5.1	Simple test	11
5.2	adafruit_74hc595	11
5.2.1	Implementation Notes	12
6	Indices and tables	13
	Python Module Index	15
	Index	17

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](#).

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-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 2

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 3

Contributing

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

4.1 Zip release files

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-74hc595 --
↳library_location .
```

4.2 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 this simple test.

Listing 1: examples/74hc595_simpletest.py

```
1 import time
2 import board
3 import busio
4 import digitalio
5 import adafruit_74hc595
6
7 spi = busio.SPI(board.SCK, MOSI=board.MOSI)
8
9 latch_pin = digitalio.DigitalInOut(board.D5)
10 sr = adafruit_74hc595.ShiftRegister74HC595(spi, latch_pin)
11
12 pin1 = sr.get_pin(1)
13
14 while True:
15     pin1.value = True
16     time.sleep(1)
17     pin1.value = False
18     time.sleep(1)
```

5.2 adafruit_74hc595

CircuitPython driver for 74HC595 shift register.

- Author(s): Kattni Rembor, Tony DiCola

5.2.1 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 6

Indices and tables

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

a

adafruit_74hc595, [11](#)

A

`adafruit_74hc595` (*module*), [11](#)

D

`DigitalInOut` (*class in adafruit_74hc595*), [12](#)

`direction` (*adafruit_74hc595.DigitalInOut attribute*), [12](#)

G

`get_pin()` (*adafruit_74hc595.ShiftRegister74HC595 method*), [12](#)

`gpio` (*adafruit_74hc595.ShiftRegister74HC595 attribute*), [12](#)

P

`pull` (*adafruit_74hc595.DigitalInOut attribute*), [12](#)

S

`ShiftRegister74HC595` (*class in adafruit_74hc595*), [12](#)

`switch_to_input()` (*adafruit_74hc595.DigitalInOut method*), [12](#)

`switch_to_output()` (*adafruit_74hc595.DigitalInOut method*), [12](#)

V

`value` (*adafruit_74hc595.DigitalInOut attribute*), [12](#)