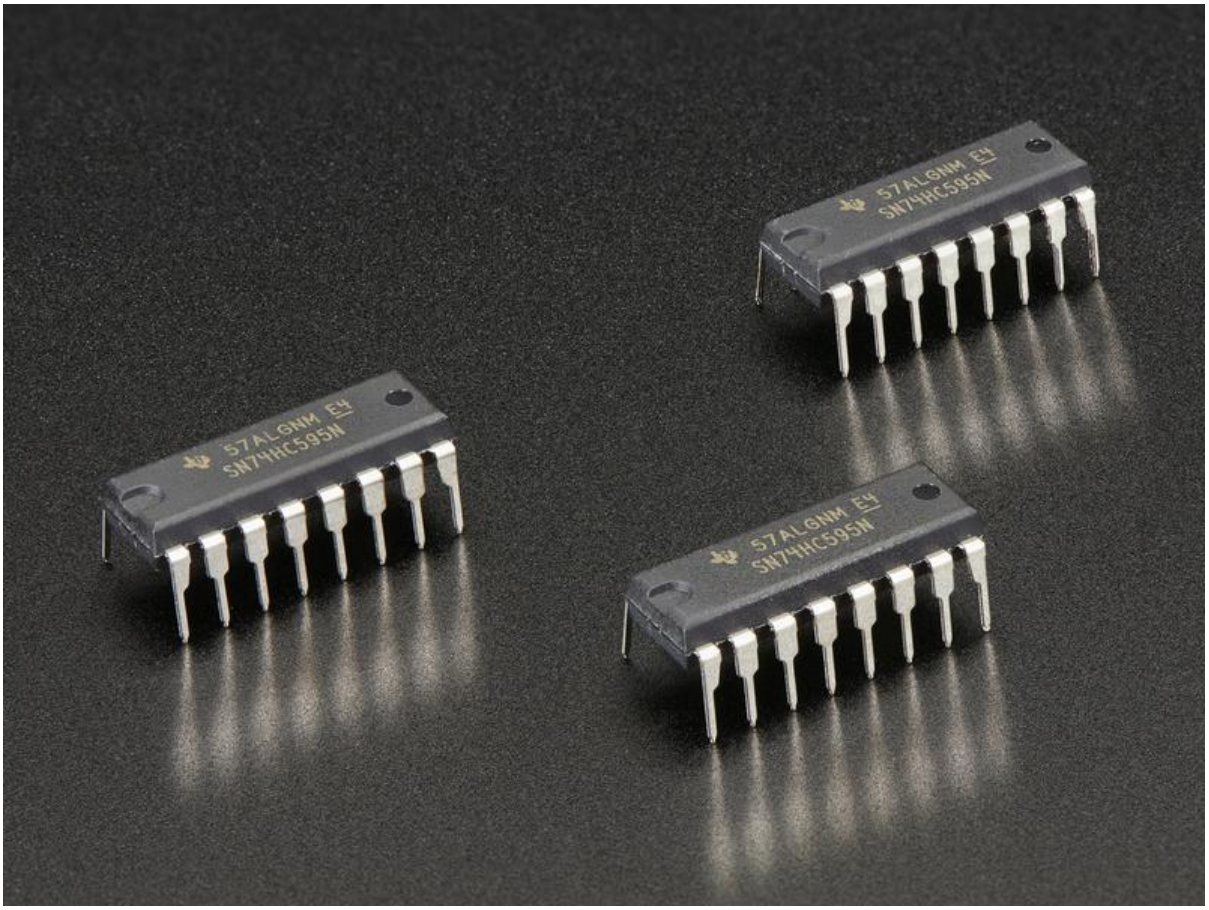
74HC595 Shift Register

Created by Kattni Rembor

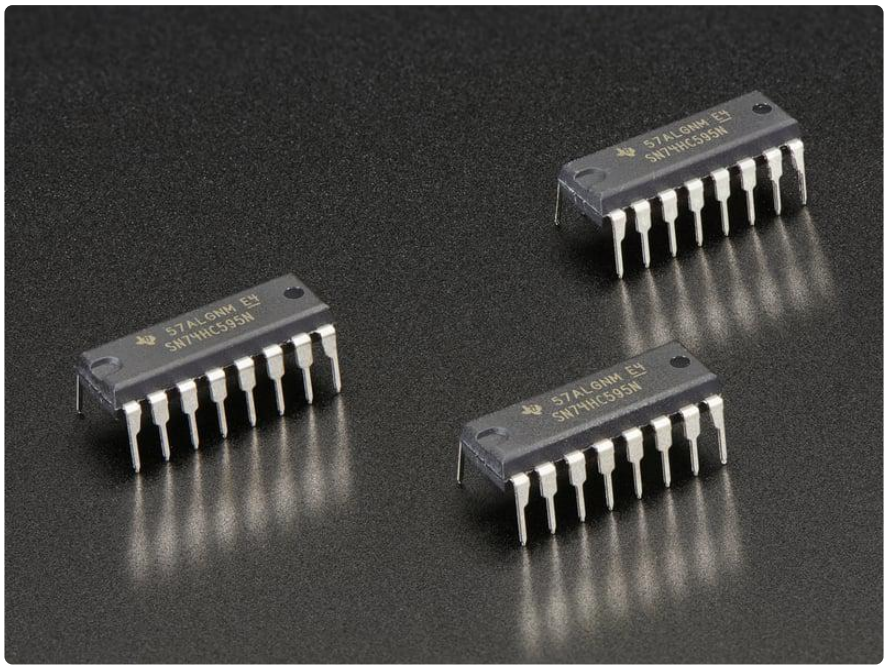


https://learn.adafruit.com/74hc595

Last updated on 2023-08-29 04:28:48 PM EDT

Table of Contents

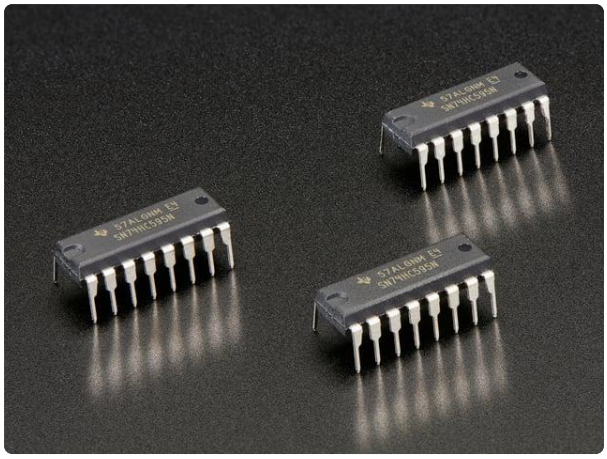
Overview



Add lots more outputs to a microcontroller system with chainable shift registers. These chips take a serial input (kinda like SPI but with a latch pin) of 1 byte (8 bits) and then output those digital bits onto 8 pins. You can chain them together so putting three in a row with the serial output of one plugged into the serial input of another to make 3 x 8 = 24 digital outputs. You can chain pretty much as many as you want. This makes it easy to control a lot of outputs like LEDs from only 3 digital microcontroller pins.

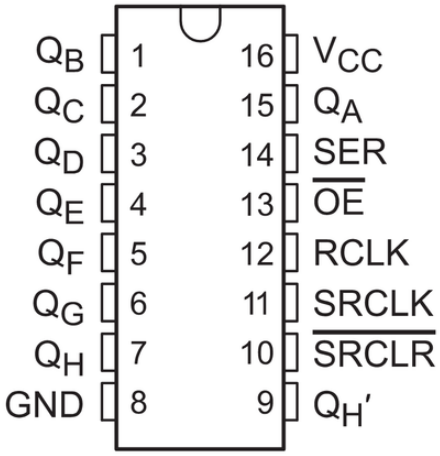
This item () contains three 74HC595 chips! The chips may be NXP or TI brand, they're cross-compatible pinouts. We don't guarantee which brand you'll receive.

These chips are DIP package so you can easily plug them into any breadboard or perfboard with 0.1" spacing. The digital outputs are good for about 20mA, which makes them ideal for LEDs or driving power transistors (say for controlling a lot of solenoids).



74HC595 Shift Register - 3 pack
Add lots more outputs to a microcontroller system with chainable shift registers. These chips take a serial input (SPI) of 1 byte (8 bits) and then output those digital bits onto 8...
https://www.adafruit.com/product/450

Pinouts

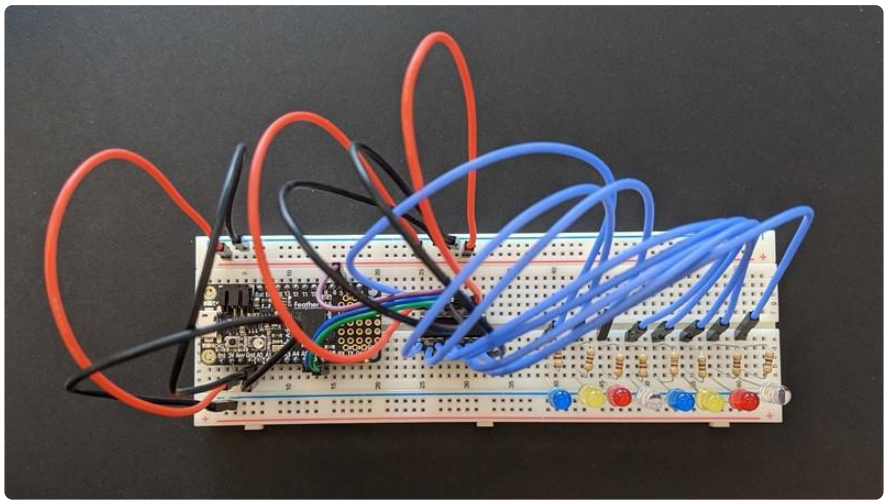


This image is from a TI datasheet. The datasheet uses letters for the output pins, e.g. Q_A. The NXP datasheet uses numbers for the output pins, e.g. Q1.

Pinouts in order of pin number:

- Pins 1-7 - Parallel data output pins B-H/2-7
- Pin 8 - GND (ground)
- Pin 9 - Serial data output pin H`/7`
- Pin 10 - CLR/Reset (active LOW)
- Pin 11 - SRCLK/SHCP (storage register clock pin, SPI clock)
- Pin 12 - RCLK/STCP (shift register clock input, latch pin)
- Pin 13 - OE (output enable, active LOW)
- Pin 14 - SER/DS (serial data input, SPI data)
- Pin 15 - Parallel data output pin A/1
- Pin 16 - VCC (positive voltage supply)

Usage

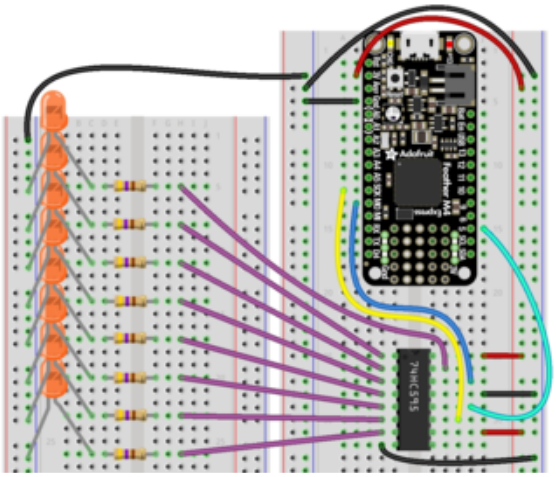


Using the 74HC595 with CircuitPython is simple with the Adafruit CircuitPython 74HC595 () library. You can easily write Python code to control multiple output pins from a single SPI connection. This example will control 8 LEDs from the SPI connection on a Feather M4.

CircuitPython Wiring

First, wire up the 74HC595 to your CircuitPython board. The following shows how to wire it up to a Feather M4.

The following can be done on a single breadboard in practice, however, the wiring diagram uses two breadboards for readability.



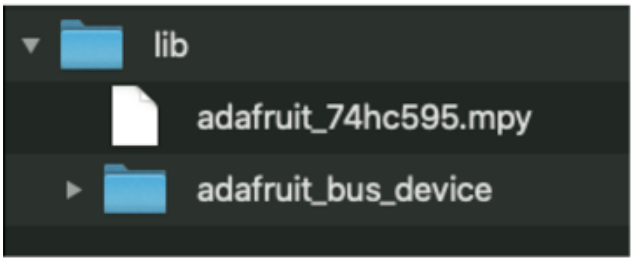
Feather 3V to power rail on breadboard
Feather GND to ground rail on breadboard
Ground rail on breadboard to ground rail on breadboard
74HC595 Pin 16 to power rail on breadboard
74HC595 Pin 10 to power rail on breadboard
74HC595 Pin 13 to ground rail on breadboard
74HC595 Pin 8 to ground rail on breadboard
74HC595 Pin 14 to Feather MO
74HC595 Pin 12 to Feather D5
74HC595 Pin 11 to Feather SCK
8 x LED - (negative) to ground rail on breadboard
8 x LED + (positive) to breadboard (470Ω resistors)
8 x 470Ω resistor FROM LED + (positive) TO 74HC595 output pins
74HC595 Pins 1-7, 15 to 8 x 470Ω resistors

CircuitPython Library Install

You'll need to install the Adafruit CircuitPython 74HC595 () library on your CircuitPython board.

First make sure you are running the latest version of Adafruit CircuitPython () for your board.

Next you'll need to install the necessary libraries to use the hardware--carefully follow the steps to find and install these libraries from Adafruit's CircuitPython library bundle (). Our CircuitPython starter guide has a great page on how to install the library bundle ().



Copy the following libraries from the bundle into your lib folder on the CIRCUITPY drive which appears when you connect your board into your computer via a USB cable:

adafruit_74hc595.mpy
adafruit_bus_device

Before continuing make sure your board's lib folder or root filesystem has the adafruit_74hc595.mpy, and adafruit_bus_device file and folder copied over.

Next connect to the board's serial REPL ()so you are at the CircuitPython >>> prompt.

CircuitPython Usage

This example will display a chase pattern followed by blinking on 8 LEDs wired up to the 74HC595.

First we import the necessary libraries, setup the latch pin, and create the shift register object.

```
import time
import board
import digitalio
import adafruit_74hc595

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

Using the Adafruit CircuitPython 74HC595 library, you access the pins on the 74HC595 using `get_pin(pin_number)`. For example, in this case, to access pin 0, it would be `sr.get_pin(0)`. Since the pin objects need to be in a list to display the chase animation, we'll use a list comprehension () to create the eight pin objects for the eight output pins.

```
pins = [sr.get_pin(n) for n in range(8)]
```

The above code results in the same thing as setting up each pin individually:

```
pins = [sr.get_pin(0), sr.get_pin(1), sr.get_pin(2), sr.get_pin(3),
        sr.get_pin(4), sr.get_pin(5), sr.get_pin(6), sr.get_pin(7)]
```

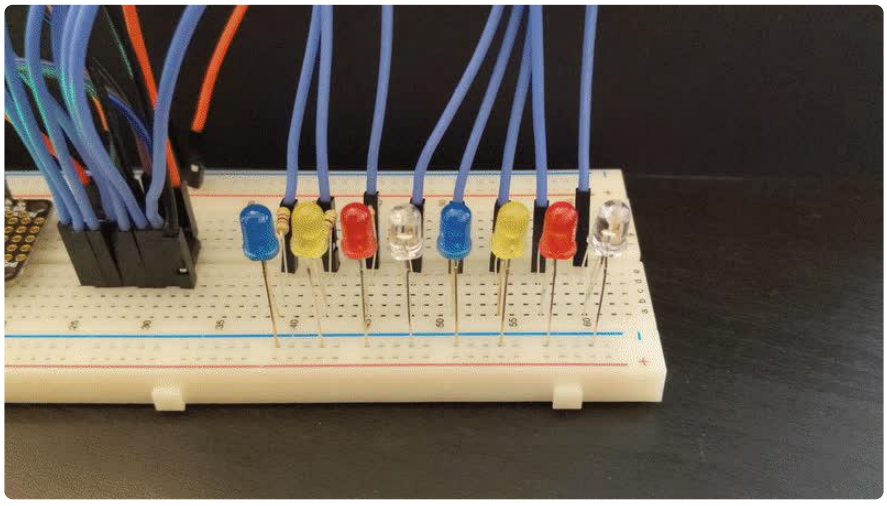
Inside the loop, we begin by running the first code block two times, to run the chase animation twice. It cycles through turning on each LED in sequence.

```
while True:
    for _ in range(2):
        for enabled_pin in range(len(pins)):
            for pin_number, pin in enumerate(pins):
                if pin_number == enabled_pin:
                    pin.value = True
                else:
                    pin.value = False
                time.sleep(0.01)
```

Finally, we blink the LEDs. We turn on the LEDs by setting all the pin values to `True`, wait 0.5 seconds, and then turn off the LEDs by setting all the LED values to `False`. This happens three times.

```
[...]
    for _ in range(3):
        for pin in pins:
            pin.value = True
        time.sleep(0.5)
        for pin in pins:
            pin.value = False
        time.sleep(0.5)
```

That's all there is to controlling eight LEDs using the 74HC595 shift register!



Full Example Code

```
# SPDX-FileCopyrightText: 2021 ladyada for Adafruit Industries
# SPDX-License-Identifier: MIT

import time
import board
import digitalio
import adafruit_74hc595

latch_pin = digitalio.DigitalInOut(board.D5)
```

```
sr = adafruit_74hc595.ShiftRegister74HC595(board.SPI(), latch_pin)

# Create the pin objects in a list
pins = [sr.get_pin(n) for n in range(8)]

while True:
    for _ in range(2):  # Run the chase animation twice
        for enabled_pin in range(len(pins)):
            for pin_number, pin in enumerate(pins):
                if pin_number == enabled_pin:
                    pin.value = True
                else:
                    pin.value = False
                time.sleep(0.01)
    for _ in range(3):  # Run the blink animation three times
        for pin in pins:
            pin.value = True
        time.sleep(0.5)
        for pin in pins:
            pin.value = False
        time.sleep(0.5)
```

Downloads

Files

- 74HC595 datasheet () (Texas Instruments)