AdafruitTLC5947 Library Documentation

Release 1.0

Tony DiCola

Jan 23, 2020

Contents

CircuitPython module for the TLC5947 12-bit 24 channel LED PWM driver.

CHAPTER 1

Dependencies

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.

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-tlc5947
```

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

```
sudo pip3 install adafruit-circuitpython-tlc5947
```

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-tlc5947
```

CHAPTER 3

Usage Example

See examples/tlc5947_simpletest.py for a demo of the usage.
See examples/tlc5947_chain.py for a demo of chained driver usage.

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.

CHAPTER 6

Table of Contents

6.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/tlc5947_simpletest.py

```
# Simple demo of controlling the TLC5947 12-bit 24-channel PWM controller.
# Will update channel values to different PWM duty cycles.
# Author: Tony DiCola

import board
import busio
import digitalio

import adafruit_tlc5947

# Define pins connected to the TLC5947
SCK = board.SCK
MOSI = board.MOSI
LATCH = digitalio.DigitalInOut(board.D5)

# Initialize SPI bus.
spi = busio.SPI(clock=SCK, MOSI=MOSI)

# Initialize TLC5947
tlc5947 = adafruit_tlc5947.TLC5947(spi, LATCH)
# You can optionally disable auto_write which allows you to control when
# channel state is written to the chip.  Normally auto_write is true and
# will automatically write out changes as soon as they happen to a channel, but
# if you need more control or atomic updates of multiple channels then disable
# and manually call write as shown below.
#tlc5947 = adafruit_tlc5947.TLC5947(spi, LATCH, auto_write=False)

```

(continues on next page)

(continued from previous page)

```
# There are two ways to channel channel PWM values.  The first is by getting
# a PWMOut object that acts like the built-in PWMOut and can be used anywhere
# it is used in your code.  Change the duty_cycle property to a 16-bit value
# (note this is NOT the 12-bit value supported by the chip natively) and the
# PWM channel will be updated.

# With an RGB LED hooked up to pins 0, 1, and 2, cycle the red, green, and
# blue pins up and down:

red = tlc5947.create_pwm_out(0)
green = tlc5947.create_pwm_out(1)
blue = tlc5947.create_pwm_out(2)

step = 10
start_pwm = 0
end_pwm = 32767 # 50% (32767, or half of the maximum 65535):

while True:
    for pin in (red, green, blue):
        # Brighten:
        print("Brightening LED")
        for pwm in range(start_pwm, end_pwm, step):
            pin.duty_cycle = pwm

        # Dim:
        print("Dimming LED")
        for pwm in range(end_pwm, start_pwm, 0 - step):
            pin.duty_cycle = pwm

# Note if auto_write was disabled you need to call write on the parent to
# make sure the value is written (this is not common, if disabling auto_write
# you probably want to use the direct 12-bit raw access instead shown below).
#             tlc5947.write()

# The other way to read and write channels is directly with each channel 12-bit
# value and an item accessor syntax.  Index into the TLC5947 with the channel
# number (0-23) and get or set its 12-bit value (0-4095).
# For example set channel 1 to 50% duty cycle.
#tlc5947[1] = 2048
# Or set channel 23 (first channel from the end) to 2/3 duty cycle.
#tlc5947[-1] = 2730
# Again be sure to call write if you disabled auto_write.
#tlc5947.write()
```

6.2 adafruit_tlc5947

CircuitPython module for the TLC5947 12-bit 24 channel LED PWM driver. See examples/simpletest.py for a demo of the usage.

- Author(s): Tony DiCola, Walter Haschka

6.2.1 Implementation Notes

Hardware:

- Adafruit 24-Channel 12-bit PWM LED Driver - SPI Interface - TLC5947 (Product ID: 1429)

Software and Dependencies:

- Adafruit CircuitPython firmware for the ESP8622 and M0-based boards: https://github.com/adafruit/circuitpython/releases

class adafruit_tlc5947.**TLC5947**(*spi*, *latch*, *, *auto_write=True*, *num_drivers=1*)

TLC5947 12-bit 24 channel LED PWM driver. Create an instance of this by passing in at least the following parameters:

Parameters

- **spi** – The SPI bus connected to the chip (only the SCK and MOSI lines are used, there is no MISO/input).
- **latch** – A DigitalInOut instance connected to the chip's latch line.

Optionally you can specify:

Parameters

- **auto_write** – This is a boolean that defaults to True and will automatically write out all the channel values to the chip as soon as a single one is updated. If you set to False to disable then you MUST call write after every channel update or when you deem necessary to update the chip state.
- **num_drivers** – This is an integer that defaults to 1. It stands for the number of chained LED driver boards (DOUT of one board has to be connected to DIN of the next). For each board added, 36 bytes of RAM memory will be taken. The channel numbers on the driver directly connected to the controller are 0 to 23, and for each driver add 24 to the port number printed. The more drivers are chained, the more viable it is to set auto_write=False, and call write explicitly after updating all the channels.

class PWMOut(*tlc5947*, *channel*)

Internal PWMOut class that mimics the behavior of CircuitPython's PWMOut class but is associated with a channel on the TLC5947. You can get and set the instance's duty_cycle property as a 16-bit PWM value (note there will be quantization errors as the TLC5947 is a 12-bit PWM chip, instead use the TLC5947 class item accessor notation for direct 12-bit raw PWM channel access). Note you cannot change the frequency as it is fixed by the TLC5947 to ~2.4-5.6 mhz.

duty_cycle

Get and set the 16-bit PWM duty cycle value for this channel.

frequency

Frequency of the PWM channel, note you cannot change this and cannot read its exact value (it varies from 2.4-5.6 mhz, see the TLC5947 datasheet).

create_pwm_out(*channel*)

Create an instance of a PWMOut-like class that mimics the built-in CircuitPython PWMOut class but is associated with the TLC5947 channel that is specified. This PWMOut class has a duty_cycle property which you can read and write with a 16-bit value to control the channel. Note there will be quantization error as the chip only supports 12-bit PWM, if this is problematic use the item accessor approach to update the raw 12-bit channel values.

write()

Write out the current channel PWM values to the chip. This is only necessary to call if you disabled auto_write in the initializer, otherwise write is automatically called on any channel update.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index