Adafruit MCP4725 Library Documentation

Release 1.0

Tony DiCola

Aug 27, 2020

Contents

CircuitPython module for the MCP4725 digital to analog converter.

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-mcp4725
```

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

```
sudo pip3 install adafruit-circuitpython-mcp4725
```

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-mcp4725
```

Usage Example

See examples/max4725_simpletest.py for a demo of the usage.

CHAPTER 3

Contributing

Contributions are welcome! Please read our Code of Conduct before contributing to help this project stay welcoming.

Documentation

For information on building library documentation, please check out this guide.

CHAPTER 5

Table of Contents

5.1 Simple test

Ensure your device works with this simple test.

Listing 1: examples/mcp4725_simpletest.py

```python
1  # Simple demo of setting the DAC value up and down through its entire range
2  # of values.
3  # Author: Tony DiCola
4  import board
5  import busio
6
7  import adafruit_mcp4725
8
9
10 # Initialize I2C bus.
11 i2c = busio.I2C(board.SCL, board.SDA)
12
13 # Initialize MCP4725.
14 dac = adafruit_mcp4725.MCP4725(i2c)
15 # Optionally you can specify a different addres if you override the A0 pin.
16 # amp = adafruit_max9744.MAX9744(i2c, address=0x63)
17
18 # There are a three ways to set the DAC output, you can use any of these:
19 dac.value = 65535  # Use the value property with a 16-bit number just like
20 # the AnalogOut class.  Note the MCP4725 is only a 12-bit
21 # DAC so quantization errors will occur.  The range of
22 # values is 0 (minimum/ground) to 65535 (maximum/Vout).
23
24 dac.raw_value = 4095  # Use the raw_value property to directly read and write
25 # the 12-bit DAC value.  The range of values is
26 # 0 (minimum/ground) to 4095 (maximum/Vout).
27
```

```python
28  dac.normalized_value = 1.0  # Use the normalized_value property to set the
29  # output with a floating point value in the range
30  # 0 to 1.0 where 0 is minimum/ground and 1.0 is
31  # maximum/Vout.
32
33  # Main loop will go up and down through the range of DAC values forever.
34  while True:
35      # Go up the 12-bit raw range.
36      print("Going up 0-3.3V...")
37      for i in range(4095):
38          dac.raw_value = i
39      # Go back down the 12-bit raw range.
40      print("Going down 3.3-0V...")
41      for i in range(4095, -1, -1):
42          dac.raw_value = i
```

5.2 `adafruit_mcp4725` - MCP4725 digital to analog converter

CircuitPython module for the MCP4725 digital to analog converter. See examples/mcp4725_simpletest.py for a demo of the usage.

- Author(s): Tony DiCola, Carter Nelson

5.2.1 Implementation Notes

Hardware:

- Adafruit MCP4725 Breakout Board - 12-Bit DAC w/I2C Interface (Product ID: 935)

Software and Dependencies:

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

class `adafruit_mcp4725.`**MCP4725**(*i2c*, ***, *address=98*)

MCP4725 12-bit digital to analog converter. This class has a similar interface as the CircuitPython AnalogOut class and can be used in place of that module.

> **Parameters**
> - **i2c** (`I2C`) – The I2C bus.
> - **address** (`int`) – The address of the device if set differently from the default.

normalized_value

The DAC value as a floating point number in the range 0.0 to 1.0.

raw_value

The DAC value as a 12-bit unsigned value. This is the the true resolution of the DAC and will never peform scaling or run into quantization error.

save_to_eeprom()

Store the current DAC value in EEPROM.

value

The DAC value as a 16-bit unsigned value compatible with the `AnalogOut` class.

Note that the MCP4725 is still just a 12-bit device so quantization will occur. If you'd like to instead deal with the raw 12-bit value use the `raw_value` property, or the `normalized_value` property to deal with a 0. . . 1 float value.

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index

A

M

N

R

S

V