

# Programmable FET LED Biasing Circuit Using Smart DAC



Smart DAC

Katlynn Jones

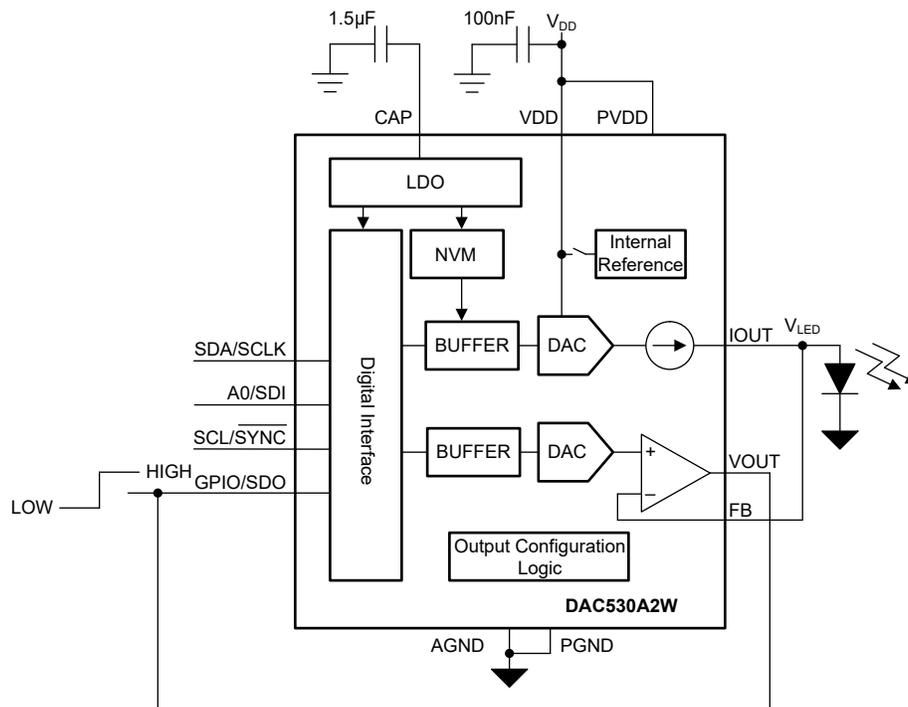
## Design Objective

Key Input Parameter	Key Output Signal	Recommended Device
SPI or I <sup>2</sup> C communication	0mA to 300mA programmable LED current	DAC530A2W, DAC532A3W, AFE532A3W, AFE432A3W

**Objective:** Bias an LED using a programmable current source.

## Design Description

This design uses a buffered voltage and current output smart digital-to-analog converter (DAC) such as the DAC530A2W or DAC532A3W (DAC53xAxW) to bias a light-emitting diode (LED). The current output DAC (IDAC) on the DAC53xAxW can source up to 300mA to bias high-current LEDs. The DAC53xAxW have a general-purpose input-output (GPIO) pin that can be used to switch the LEDs between two current values, or on and off. The voltage channel of the DAC53xAxW can be configured as a comparator to monitor the LED voltage ( $V_{LED}$ ) for software-independent fault management. The output of the comparator can be connected to the DAC53xAxW GPIO pin to turn the IDAC off if  $V_{LED}$  is out of range. All register settings can be saved into the non-volatile memory (NVM) on the smart DAC, meaning that the device can be used without a processor, even after a power cycle. This circuit can be used in applications such as [in-vitro diagnostics](#), [endoscopes](#), and digital microscopes.



## Design Notes

1. This application circuit uses both channels of the DAC530A2W.
2. The [DAC53xAxW 10-Bit Three-Channel and Two-Channel Voltage-Output and Current-Output Smart DAC With I<sup>2</sup>C or SPI](#) data sheet recommends using a 100nF decoupling capacitor for the VDD and PVDD pins, and a 1.5μF or greater bypass capacitor for the CAP pin. The CAP pin is connected to the internal LDO. Place these capacitors close to the device pins.
3. Connect the PV<sub>DD</sub> and V<sub>DD</sub> supplies with a low-impedance PCB trace close to the DAC530A2W supply pins.
4. The DAC530A2W IDAC channel can be configured for a 300mA or 220mA output range in the IOUT-GAIN field of the DAC-2-GAIN-CONFIG register. This application uses the 300mA range.
5. The nominal IDAC output current for this application is 250mA. Set the IDAC code in the DAC-2-DATA register. The IDAC code for a 250mA output is calculated by:

$$\text{DAC\_DATA} = \frac{\text{IOUT}}{\text{GAIN} \times 0.5241\text{mA}} \times 2^{10} = \frac{250\text{mA}}{2/3 \times 0.5241\text{mA}} \times 2^{10} = 733\text{d}$$

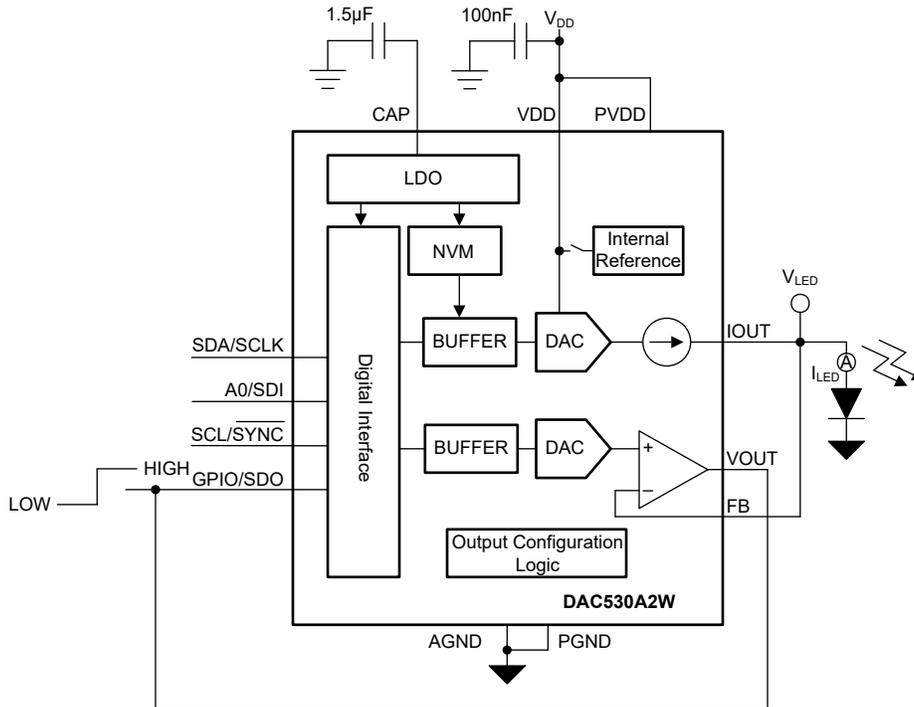
6. The headroom voltage ( $V_{\text{HEADROOM}}$ ) is calculated as the difference between  $P_{\text{VDD}}$  and the voltage of the IDAC pin, or  $V_{\text{LED}}$  in this circuit. The IDAC output cannot source the full-scale current output if  $V_{\text{HEADROOM}}$  is lower than the specified voltage. Minimize  $V_{\text{HEADROOM}}$  to reduce the power dissipation of the device while also meeting the minimum  $V_{\text{HEADROOM}}$  requirement. The IDAC output contributes to power dissipation proportionally to the output current multiplied by  $V_{\text{HEADROOM}}$ .
7. The voltage output (VOUT) channel of the DAC530A2W is used as a programmable comparator to sense an out-of-range condition of  $V_{\text{LED}}$ . The threshold ( $V_{\text{THRESH}}$ ) for the comparator is set in the DAC-1-DATA register. The comparator can use  $V_{\text{DD}}$  as the reference, or the internal 1.21V reference with a configurable gain. In the DAC-1-GAIN-CMP-CONFIG register:
  - a. Select the reference and gain for the comparator
  - b. Enable the channel for comparator mode
  - c. Enable the comparator output
  - d. Disable Hi-Z input mode
8. Using the 3.3V  $V_{\text{DD}}$  as the voltage reference with a 1 × gain, the code for a 1V threshold is calculated by:

$$\text{DAC\_DATA} = \frac{V_{\text{THRESH}}}{V_{\text{REF}} \times \text{GAIN}} \times 2^{10} = \frac{1\text{V}}{3.3\text{V}} \times 2^{10} = 310\text{d}$$

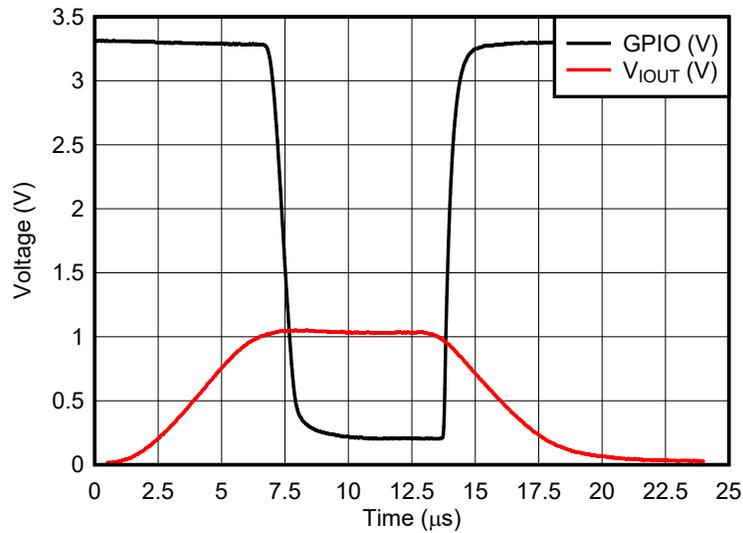
9. The IDAC uses the internal reference. Enable the internal reference, IDAC output, and comparator channel in the COMMON-CONFIG register
10. In this application circuit, the comparator output is connected to the GPIO pin to clear the IDAC output to zero-scale. When  $V_{\text{LED}}$  is less than  $V_{\text{THRESH}}$ , the comparator output is high and the IDAC output remains at the programmed code in the DAC-2-DATA register. When  $V_{\text{LED}}$  is greater than  $V_{\text{THRESH}}$ , the comparator output is set low and the IDAC output is cleared to zero-scale. This is the default configuration of the comparator. To reverse the comparator output polarity, set the CMP-1-INV-EN bit in the DAC-1-GAIN-CMP-CONFIG register to 1.
11. The GPI-EN bit in the GPIO-CONFIG register enables the GPIO pin as an input. The GPI-CH-SEL field selects which channels are controlled by the GPI. The GPI-CONFIG field selects the GPI function. Write 0b0111 to the GPI-CONFIG field to configure the GPIO pin to trigger the clear function.
12. The DAC530A2W can be programmed with the initial register settings described in the [Register Settings](#) section using I<sup>2</sup>C or SPI. Save the initial register settings in the NVM by writing a 1 to the NVM-PROG field of the COMMON-TRIGGER register. After programming the NVM, the device loads all registers with the values stored in the NVM after a reset or a power cycle.

## Design Results

This schematic is used for the following design results of the DAC530A2W.  $V_{LED}$  and  $I_{LED}$  are measured at the test points marked on the schematic.



This plot shows the low-to-high transition of the DAC530A2W comparator output when  $V_{LED}$  is out of the specified range.



## Register Settings

The following table shows an example register map for this application. The values given here are for the design choices made in the [Design Notes](#) section.

### Register Settings for DAC530A2W

Register Address	Register Name	Setting	Description
0x03	DAC-2-GAIN-CONFIG	0x0000	[15:13] 0b000: Don't care [12:10] 0b000: Selects 2/3 gain [9:0] 0x00: Don't care
0x15	DAC-1-GAIN-CMP-CONFIG	0x040D	[15:13] 0b0: Don't care [12:10] 0b001: Set REF-GAIN-1 to $1 \times V_{DD}$ [9:5] 0b0: Don't care [4] 0b0: Set comparator output pin as push-pull [3] 0b1: Enable comparator output [2] 0x1: Set comparator input pin to finite impedance [1] 0x0: Do not invert the comparator output [0] 0x1: Enable comparator mode
0x19	DAC-2-DATA	0x5740	[15:6] 0x2DD: 10-bit IDAC code [5:0] 0x0: Don't care
0x1C	DAC-1-DATA	0x4D80	[15:6] 0x136: 10-bit comparator threshold [5:0] 0x0: Don't care
0x1F	COMMON-CONFIG	0x13F9	[15] 0b0: Write 0b0 to set the window-comparator output to a non-latching output [14] 0b0: Device not locked [13] 0b0: Fault-dump read enable at address 0x00 [12] 0b1: Enables the internal reference [11:10] 0b00: Powers up VOUT1 [9] 0b1: Reserved. Always write 1 [8:7] 0b11: Reserved. Always write 0b11 [6:3] 0xF: Reserved. Always write 0xF [2:1] 0b00: Powers up IDAC2 [0] 0b1: Reserved. Always write 1
0x20	COMMON-TRIGGER	0x0002	[15:12] 0b0000: Write 0b0101 to unlock the device [11:8] 0b0000: Write 0b1010 to trigger a POR reset [7] 0b0: LDAC is not triggered [6] 0b0: DAC clear is not triggered [5] 0b0: Don't care [4] 0b0: Fault-dump is not triggered [3] 0b0: PROTECT function not triggered [2] 0b0: Fault-dump read not triggered [1] 0b1: Write 0b1 to store applicable register settings to the NVM [0] 0b0: NVM reload not triggered. Write 0b1 to reload applicable registers with existing NVM settings
0x24	GPIO-CONFIG	0x002F	[15] 0b0: Glitch filter disabled for GP input [14] 0b0: Don't care [13] 0b0: Disable output mode for GPIO pin [12:9] 0b0000: Selects the STATUS function setting mapped to GPIO as output [8:5] 0b0001: Enables GPI function on channel 2 [4:1] 0b0111: GP input configured to trigger clear [0] 0b1: Enables input mode for GPIO pin

## Pseudocode Example

The following shows a pseudocode sequence to program the initial register values to the NVM of the DAC530A2W. The values given here are for the design choices made in the [Design Notes](#) section.

### Pseudocode Example for DAC530A2W

```
//SYNTAX: WRITE <REGISTER NAME (Hex Code)>, <MSB DATA>, <LSB DATA>
//Set IDAC gain setting to 2/3
WRITE DAC-2-GAIN-CONFIG(0x03), 0x00, 0x00
//Write DAC code for nominal IDAC output
//The 10-bit hex code for 250mA is 0x2DD. With 16-bit left alignment, this becomes 0xB740
WRITE DAC-2-DATA(0x19), 0x57, 0x40
//Set VOUT1 gain setting to 1x VDD (3.3V), enable comparator mode, enable comparator output,
//disable Hi-z input
WRITE DAC-1-GAIN-CMP-CONFIG(0x15), 0x04, 0x0D
//For a 3.3V output range, the 10bit hex code for 1V is 0x136. With 16-bit left alignment, this
//becomes 0x4D80
WRITE DAC-1-DATA(0x1C), 0x4D, 0x80
//Power-up output on IDAC and VDAC channels, enables internal reference
WRITE COMMON-CONFIG(0x1F), 0x13, 0xF9
//Configure GPI for clear trigger for IDAC channel
WRITE GPIO-CONFIG(0x24), 0x00, 0x2F
//Save settings to NVM
WRITE COMMON-TRIGGER(0x20), 0x00, 0x02
```

## Design Featured Devices

Device	Key Features	Link
DAC530A2W	10-bit 300mA current-source DAC with supervisor comparator or VDAC channel, NVM and GPIO	<a href="#">DAC530A2W</a>
DAC532A3W	10-bit 300mA current-source DAC with 10-bit voltage-output DAC, supervisor comparator, NVM and GPI	<a href="#">DAC532A3W</a>
AFE532A3W	10-bit 300mA current-source DAC with 10-bit SAR ADC, 10-bit VDAC, NVM and GPI	<a href="#">AFE532A3W</a>
AFE432A3W	8-bit 300mA current-source DAC with 10-bit SAR ADC, VDAC, NVM, and GPI	<a href="#">AFE432A3W</a>

Find other possible devices using the [parametric search tool](#).

## Design References

See [Analog Engineer's Circuit Cookbooks](#) for TI's comprehensive circuit library.

## Additional Resources

- Texas Instruments, [Smart DAC Python Examples](#)
- Texas Instruments, [AFE532A3WEVM-Evaluation Module](#)
- Texas Instruments, [AFE532A3WEVM User's Guide](#)
- Texas Instruments, [Precision Labs - DACs](#)

For direct support from TI Engineers, use the E2E™ forums:

[e2e.ti.com](http://e2e.ti.com)

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2024, Texas Instruments Incorporated