DACx3401 10-Bit and 8-Bit, Voltage-Output Digital-to-Analog Converters With Nonvolatile Memory and PMBus™ Compatible I²C Interface in Tiny 2 × 2 WSON

1 Features

- 1 LSB INL and DNL (10-bit and 8-bit)
- Wide operating range
  - Power supply: 1.8 V to 5.5 V
  - Temperature range: –40˚C to +125˚C
- PMBus™ compatible I²C interface
  - Standard, Fast, and Fast+ modes
  - Digital slew rate control
  - 1.62-V V_{IH} with V_{DD} = 5.5 V
- User-programmable nonvolatile memory (NVM/EEPROM)
  - Save and recall all register settings
- Programmable waveform generation: Square, ramp, and sawtooth
- Preprogrammed medical-alarm tone-generation mode: low, medium, and high priority alarms
- Internal reference
- Very low power: 0.1 mA at 1.8 V
- Flexible startup: High impedance or 10K-GND
- Tiny package: 8-pin WSON (2 mm × 2 mm)

2 Applications

- Rack server
- Laser TV
- Enterprise projector
- ATMs (automated teller machines)
- Currency counter
- Multifunction printer

3 Description

The 10-bit DAC53401 and 8-bit DAC43401 (DACx3401) are a pin-compatible family of buffered voltage-output digital-to-analog converters (DACs). These devices consume very low power, and are available in a tiny 8-pin WSON package. The feature set combined with the tiny package and low power make the DACx3401 an excellent choice for applications such as LED and general-purpose bias point generation, power supply control, digitizers, PWM signal generation, and medical alarm tone generation.

These devices have nonvolatile memory (NVM), an internal reference, and a PMBus-compatible I²C interface. The DACx3401 operates with either an internal reference or the power supply as a reference, and provides full-scale output of 1.8 V to 5.5 V. The devices communicate through the I²C interface. These devices support I²C standard mode, fast mode, and fast+ mode.

The DACx3401 are feature rich, and include PMBus voltage margin commands, user-programmable power up to high impedance, standalone waveform generator, medical alarm tone generator, dedicated feedback pin, and more.

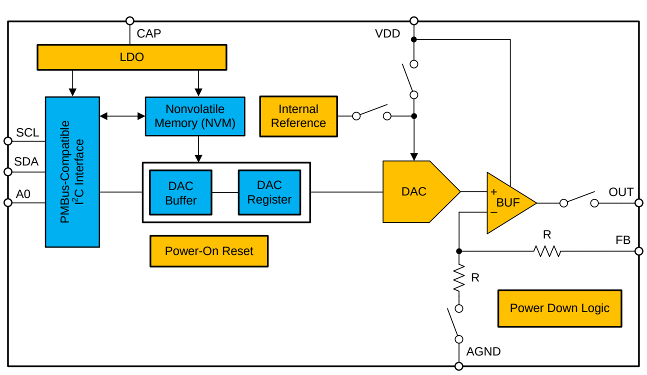
The DACx3401 operate within the temperature range of –40°C to +125°C.

Device Information(1)

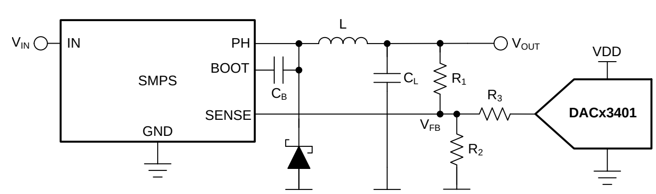
PART NUMBER	PACKAGE	BODY SIZE (NOM)
DACx3401	WSON (8)	2.00 mm × 2.00 mm
DACx3401	WSON (8)	2.00 mm × 2.00 mm

(1) For all available packages, refer to the package option addendum at the end of the data sheet.

Functional Block Diagram



Power-Supply Control With the DACx3401



ADVANCE INFORMATION

Table of Contents

4 Revision History

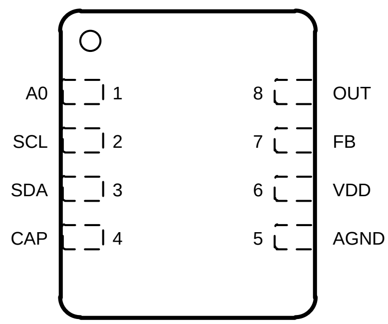
DATE	REVISION	NOTES
July 2019	*	Initial release.

5 Device Comparison Table

DEVICE	RESOLUTION
DAC53401	10-Bit
DAC43401	8-Bit

6 Pin Configuration and Functions

DSG Package
8-Pin WSON
Top View



Not to scale

Pin Functions

PIN		TYPE	DESCRIPTION
NAME	**NO.**		
A0	1	Input	Four-state address input
AGND	5	Ground	Ground reference point for all circuitry on the device
CAP	4	Input	External capacitor for the internal LDO. Connect a capacitor (0.5 µF to 15 µF) between CAP and AGND.
FB	7	Input	Voltage feedback pin
OUT	8	Output	Analog output voltage from DAC
SCL	2	Input	Serial interface clock. This pin must be connected to the supply voltage with an external pullup resistor.
SDA	3	Input/output	Data are clocked into or out of the input register. This pin is a bidirectional, and must be connected to the supply voltage with an external pullup resistor.
VDD	6	Power	Analog supply voltage: 1.8 V to 5.5 V

ADVANCE INFORMATION

7 Specifications

7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)(1)

		MIN	MAX	UNIT
V_{DD}	Supply voltage, V_{DD} to A_{GND}	–0.3	6	V
	Digital input(s) to A_{GND}	–0.3	V_{DD} + 0.3	V
	CAP to A_{GND}	–0.3	1.65	V
	V_{FB} to A_{GND}	–0.3	V_{DD} + 0.3	V
	V_{OUT} to A_{GND}	–0.3	V_{DD} + 0.3	V
	Current into any pin	–10	10	mA
T_J	Junction temperature	–40	150	°C
T_{stg}	Storage temperature	–65	150	°C

(1) Stresses beyond those listed under *Absolute Maximum Rating* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Condition*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

7.2 ESD Ratings

			VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001, all pins(1)	±2000	V
		Charged device model (CDM), per JEDEC specification JESD22-C101, pins 1, 4, 5, 8(2)	±750	V
		Charged device model (CDM), per JEDEC specification JESD22-C101, pins 2, 3, 6, 7(2)	±500	V

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
V_{DD}	Positive supply voltage to ground (A_{GND})	1.71		5.5	V
V_{IH}	Digital input high voltage, 1.7 V < V_{DD} ≤ 5.5 V	1.62			V
V_{IL}	Digital input low voltage			0.4	V
T_A	Ambient temperature	–40		125	°C

7.4 Thermal Information

THERMAL METRIC(1)		DACx3401 DSG (WSON) 8 PINS	UNIT
$R_{\theta JA}$	Junction-to-ambient thermal resistance	49	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	50	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	24.1	°C/W
Ψ_{JT}	Junction-to-top characterization parameter	1.1	°C/W
Ψ_{JB}	Junction-to-board characterization parameter	24.1	°C/W
$R_{\theta JC(bot)}$	Junction-to-case (bottom) thermal resistance	8.7	°C/W

(1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics application report.*

7.5 Electrical Characteristics

all minimum/maximum specifications at T_A = –40°C to +125°C and typical specifications at T_A = 25°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, DAC reference tied to VDD, gain = 1x, DAC output pin (OUT) loaded with resistive load (R_L = 5 kΩ to AGND) and capacitive load (C_L = 200 pF to AGND), and digital inputs at VDD or AGND (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
STATIC PERFORMANCE						
	Resolution	DAC53401	10			Bits
		DAC43401	8			
INL	Relative accuracy (1)		–1		1	LSB
DNL	Differential nonlinearity (1)		–1		1	LSB
	Zero code error	Code 0d into DAC		6	12	mV
		Internal V_{REF}, gain = 4x, V_{DD} = 5.5 V		12	24	
	Zero code error temperature coefficient			±20		µV/°C
	Offset error (1)		–0.5	0.25	0.5	%FSR
	Offset error temperature coefficient (1)			±0.0003		%FSR/°C
	Gain error (1)		–0.5	0.25	0.5	%FSR
	Gain error temperature coefficient (1)			±0.0008		%FSR/°C
	Full scale error	2.7 V ≤ V_{DD} ≤ 5.5 V, code 1023d into DAC, no headroom	–0.5	0.25	0.5	%FSR
		1.8 V ≤ V_{DD} ≤ 2.7 V, code 1023d into DAC, no headroom	–1	0.5	1	
	Full scale error temperature coefficient			±0.0008		%FSR/°C
OUTPUT CHARACTERISTICS						
	Output voltage	Reference tied to V_{DD}	0		5.5	V
C_L	Capacitive load (2)	R_L = Infinite, phase margin = 30°			1	nF
		R_L = 5 kΩ, phase margin = 30°			2	
	Load regulation	DAC at midscale, –10 mA ≤ I_{OUT} ≤ 10 mA, V_{DD} = 5.5 V		1.5		mV/mA
	Short circuit current	V_{DD} = 1.8 V, full-scale output shorted to A_{GND} or zero-scale output shorted to V_{DD}		10		mA
		V_{DD} = 2.7 V, full-scale output shorted to A_{GND} or zero-scale output shorted to V_{DD}		25		
		V_{DD} = 5.5 V, full-scale output shorted to A_{GND} or zero-scale output shorted to V_{DD}		50		
	Output voltage headroom	To V_{DD} (DAC output unloaded, internal reference = 1.21 V), V_{DD} ≥ 1.21 × gain + 0.2 V		0.2		V
		To V_{DD} (DAC output unloaded, reference tied to V_{DD})		0.8		%FSR
		To V_{DD} (I_{LOAD} = 10 mA at V_{DD} = 5.5 V, I_{LOAD} = 3 mA at V_{DD} = 2.7 V, I_{LOAD} = 1 mA at V_{DD} = 1.8 V), DAC code = full scale	10			
	V_{OUT} dc output impedance	DAC output enabled and DAC code = midscale		0.25		Ω
		DAC output enabled and DAC code = 4d		0.25		
		DAC output enabled and DAC code = 1016d		0.26		
	V_{FB} dc output impedance	DAC output enabled	160	200	240	kΩ
		DAC output enabled, internal V_{REF}, gain = 4x	192	240	288	

(1) Measured with DAC output unloaded. For external reference between end-point codes: 8d to 1016d for 10-bit resolution, 2d to 254d for 8-bit resolution. For internal reference V_{DD} ≥ 1.21 x gain + 0.2 V, between end-point codes: 4d to 1016d for 10-bit resolution, 1d to 251d for 8-bit resolution.
(2) Specified by design and characterization, not production tested.

ADVANCE INFORMATION

ADVANCE INFORMATION

Electrical Characteristics (continued)

all minimum/maximum specifications at T_A = –40°C to +125°C and typical specifications at T_A = 25°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, DAC reference tied to VDD, gain = 1x, DAC output pin (OUT) loaded with resistive load (R_L = 5 kΩ to AGND) and capacitive load (C_L = 200 pF to AGND), and digital inputs at VDD or AGND (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
Z_O	V_{OUT} + V_{FB} dc output impedance(2)	At startup, measured when DAC output is disabled and held at V_{DD} / 2 for V_{DD} = 5.5 V	0.5			GΩ
	Power supply rejection ratio (dc)	Internal V_{REF}, gain = 2x, DAC at midscale; V_{DD} = 5 V ±10%		0.25		mV/V
DYNAMIC PERFORMANCE						
t_{sett}	Output voltage settling time	1/4 to 3/4 scale and 3/4 to 1/4 scale settling to 10%FSR, V_{DD} = 5.5 V		10		µs
		1/4 to 3/4 scale and 3/4 to 1/4 scale settling to 10%FSR, V_{DD} = 5.5 V, internal V_{REF}, gain = 4x		20		
	Slew rate	V_{DD} = 5.5 V		1		V/µs
	Power on glitch magnitude	At startup (DAC output disabled), R_L = 5 kΩ, C_L = 200 pF		110		mV
		At startup (DAC output disabled), R_L = 100 kΩ		400		
	Output enable glitch magnitude	DAC output disabled to enabled (DAC registers at zero scale, R_L = 100 kΩ		400		mV
V_n	Output noise voltage (peak to peak)	0.1 Hz to 10 Hz, DAC at midscale, V_{DD} = 5.5 V		34		$µV_{PP}$
		Internal V_{REF}, gain = 4x, 0.1 Hz to 10 Hz, DAC at midscale, V_{DD} = 5.5 V		70		
	Output noise density	Measured at 1 kHz, DAC at midscale, V_{DD} = 5.5 V		0.2		µV/√Hz
		Internal $V_{REF,}$ gain = 4x,, measured at 1 kHz, DAC at midscale, V_{DD} = 5.5 V		0.7		
	Power supply rejection ratio (ac)	Internal V_{REF}, gain = 4x, 200-mV 50 or 60 Hz sine wave superimposed on power supply voltage, DAC at midscale		–71		dB
	Code change glitch impulse	±1 LSB change around mid code (including feedthrough)		10		nV-s
	Code change glitch impulse magnitude	±1 LSB change around mid code (including feedthrough)		25		mV
EEPROM						
	Endurance	–40°C ≤ T_A ≤ 85°C		20000		Cycles
				1000		
	Data retention	T_A = 25°C		50		Years
	EEPROM programming write cycle time(2)		5	10	15	ms
DIGITAL INPUTS						
	Digital feedthrough	DAC output static at midscale, fast+ mode, SCL toggling		20		nV-s
	Pin capacitance	Per pin		10		pF
POWER REQUIREMENTS						
	Load capacitor - CAP pin(2)		0.5		15	µF
I_{DD}	Current flowing into VDD	Normal mode, DACs at full scale, digital pins static		0.5	0.8	mA
		DAC power-down, internal reference power down		80		µA

7.6 Timing Requirements: I^2C^{TM} Standard mode

all input signals are specified with t_R = t_F = 1 ns/V (VIL to VIH) and timed from a voltage level of (VIL + VIH) / 2, 1.8 V ≤ V_{DD} ≤ 5.5 V, –40°C ≤ T_A ≤ +125°C, $V_{pull\text{-}up}$ = V_{DD} for 1.8 V ≤ V_{DD} ≤ 2.7 V, $V_{pull\text{-}up}$ = 2.7 or V_{DD} for 2.7 V ≤ V_{DD} ≤ 5.5 V

		MIN	NOM	MAX	UNIT
f_{SCLK}	SCL frequency			0.1	MHz
t_{BUF}	Bus free time between stop and start conditions	4.7			µs
t_{HDSTA}	Hold time after repeated start	4			µs
t_{SUSTA}	Repeated start setup time	4.7			µs
t_{SUSTO}	Stop condition setup time	4			µs
t_{HDDAT}	Data hold time	0			ns
t_{SUDAT}	Data setup time	250			ns
t_{LOW}	SCL clock low period	4700			ns
t_{HIGH}	SCL clock high period	4700			ns
t_F	Clock and data fall time			300	ns
t_R	Clock and data rise time			1000	ns

7.7 Timing Requirements: I^2C^{TM} Fast mode

all input signals are specified with t_R = t_F = 1 ns/V (VIL to VIH) and timed from a voltage level of (VIL + VIH) / 2, 1.8 V ≤ V_{DD} ≤ 5.5 V, –40°C ≤ T_A ≤ +125°C, $V_{pull\text{-}up}$ = V_{DD} for 1.8 V ≤ V_{DD} ≤ 2.7 V, $V_{pull\text{-}up}$ = 2.7 or V_{DD} for 2.7 V ≤ V_{DD} ≤ 5.5 V

		MIN	NOM	MAX	UNIT
f_{SCLK}	SCL frequency			0.4	MHz
t_{BUF}	Bus free time between stop and start conditions	1.3			µs
t_{HDSTA}	Hold time after repeated start	0.6			µs
t_{SUSTA}	Repeated start setup time	0.6			µs
t_{SUSTO}	Stop condition setup time	0.6			µs
t_{HDDAT}	Data hold time	0			ns
t_{SUDAT}	Data setup time	100			ns
t_{LOW}	SCL clock low period	1300			ns
t_{HIGH}	SCL clock high period	600			ns
t_F	Clock and data fall time			300	ns
t_R	Clock and data rise time			300	ns

7.8 Timing Requirements: I2C™ Fast+ mode

all input signals are specified with t_R = t_F = 1 ns/V (VIL to VIH) and timed from a voltage level of (VIL + VIH) / 2, 1.8 V ≤ V_{DD} ≤ 5.5 V, –40°C ≤ T_A ≤ +125°C, $V_{pull-up}$ = V_{DD} for 1.8 V ≤ V_{DD} ≤ 2.7 V, $V_{pull-up}$ = 2.7 or V_{DD} for 2.7 V ≤ V_{DD} ≤ 5.5 V

		MIN	NOM	MAX	UNIT
f_{SCLK}	SCL frequency			1	MHz
t_{BUF}	Bus free time between stop and start conditions	0.5			µs
t_{HDSTA}	Hold time after repeated start	0.26			µs
t_{SUSTA}	Repeated start setup time	0.26			µs
t_{SUSTO}	Stop condition setup time	0.26			µs
t_{HDDAT}	Data hold time	0			ns
t_{SUDAT}	Data setup time	50			ns
t_{LOW}	SCL clock low period	0.5			µs
t_{HIGH}	SCL clock high period	0.26			µs
t_F	Clock and data fall time			120	ns
t_R	Clock and data rise time			120	ns

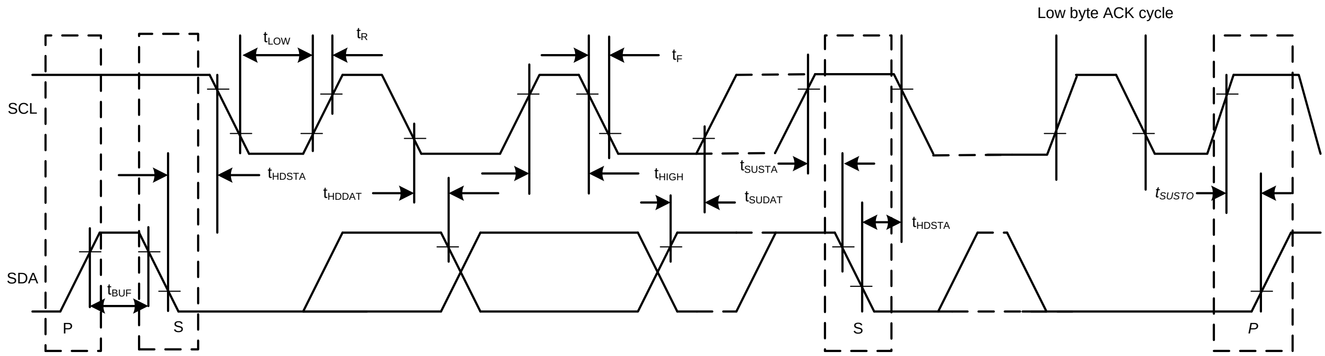


Figure 1. Timing Diagram

ADVANCE INFORMATION

8 Detailed Description

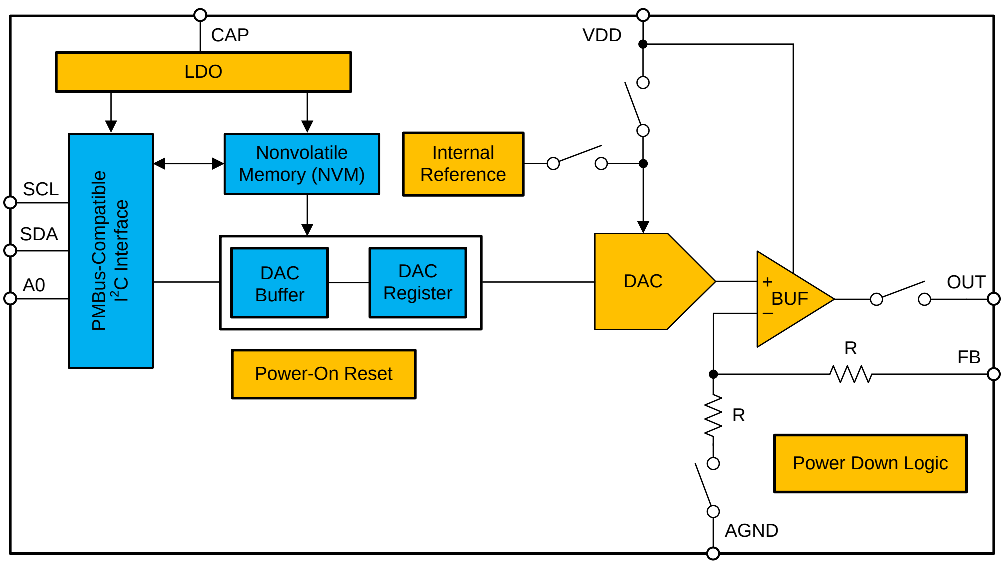
8.1 Overview

The 10-bit DAC53401 and 8-bit DAC43401 (DACx3401) are a pin-compatible family of buffered voltage-output, digital-to-analog converters (DACs). These DACs contain nonvolatile memory (NVM), an internal reference, and a PMBus-compatible I^2C interface. The DACx3401 operate with either an internal reference or with a power supply as the reference, and provide a full-scale output of 1.8 V to 5.5 V.

The devices communicate through an I^2C interface. These devices support I^2C standard mode (100 kbps), fast mode (400 kbps), and fast+ mode (1 Mbps). These devices also support specific PMBus commands such as *turn on/off*, *margin high/low*, and more. The DACx3401 also include digital slew rate control, and support basic signal generation such as *square*, *ramp*, and *sawtooth* waveforms.

The DACx3401 devices have a power-on-reset (POR) circuit that makes sure all the registers start with default or user-programmed settings using NVM. The DAC output powers on in high-impedance mode (default); this setting can be programmed to 10kΩ-GND using NVM.

8.2 Functional Block Diagram



ADVANCE INFORMATION

8.3 Feature Description

8.3.1 Digital-to-Analog Converter (DAC) Architecture

The DACx3401 family of devices consists of string architecture with an output buffer amplifier. The *Functional Block Diagram* section shows the DAC architecture within the block diagram. This DAC architecture operates from a 1.8-V to 5.5-V power supply. These devices consume only 0.1 mA of current when using a 1.8-V power supply. The DAC output pin starts up in high impedance mode making it an excellent choice for power-supply control applications. To change the power-up mode to 10kΩ-GND, program the DAC_PDN bit (address: D1h), and load these bits in the device NVM.

8.3.1.1 Reference Selection and DAC Transfer Function

The device writes the input data to the DAC data registers in straight-binary format. After a power-on or a reset event, the device sets all DAC registers to the values set in the NVM.

8.3.1.1.1 Power Supply as Reference

By default, the DACx3401 operate with the power-supply pin (VDD) as a reference. Equation 1 shows DAC transfer function when the power-supply pin is used as reference.

$$V_{OUT} = \frac{DAC\_DATA}{2^N} \times V_{DD}$$

where:

- N is the resolution in bits, either 8 (DAC43401) or 10 (DAC53401).
- DAC_DATA is the decimal equivalent of the binary code that is loaded to the DAC register.
- DAC_DATA ranges from 0 to $2^N – 1$.
- V_{DD} is used as the DAC reference voltage. (1)

8.3.1.1.2 Internal Reference

The DACx3401 also contains an internal reference that is disabled by default. Enable the internal reference by writing 1 to REF_EN (address D1h). The internal reference generates a fixed 1.22-V voltage (typical). Using DAC_SPAN (address D1h) bits, gain of 1.5X, 2X, 3X, 4X can be achieved for the DAC output voltage (V_{OUT}) Equation 2 shows DAC transfer function when the internal reference is used.

$$V_{OUT} = \frac{DAC\_DATA}{2^N} \times V_{REF} \times GAIN$$

where:

- N is the resolution in bits, either 8 (DAC43401) or 10 (DAC53401).
- DAC_DATA is the decimal equivalent of the binary code that is loaded to the DAC register
- DAC_DATA ranges from 0 to $2^N – 1$.
- V_{REF} is the internal reference voltage = 1.22 V.
- GAIN = 1.5x, 2x, 3x, 4x based on DAC_SPAN (address D1h) bits. (2)

Feature Description (continued)

8.3.2 DAC Update

The DAC output pin (OUT) is updated at the end of I^2C DAC write frame.

8.3.2.1 DAC Update Busy

The DAC_UPDATE_BUSY bit (address D0h) is set to 1 by the device when certain DAC update operations, such as *function generation*, *transition to margin high or low*, or any of the medical alarms are in progress. When the DAC_UPDATE_BUSY bit is set to 1, do not write to any of the DAC registers. After the DAC update operation is completed (DAC_UPDATE_BUSY = 0), any of the DAC registers can be written.

8.3.3 Nonvolatile Memory (EEPROM or NVM)

The DACx3401 contains nonvolatile memory (NVM) bits. These memory bits are user programmable and erasable, and retain the set values in the absence of a power supply. All the register bits, as shown in Table 1, can be stored in the device NVM by setting NVM_PROG = 1 (address D3h). The NVM_BUSY bit (address D0h) is set to 1 by device when a NVM write or reload operation is ongoing. During this time, the device blocks all write operations to the device. The NVM_BUSY bit is set to 0 after the write or reload operation is complete; at this point, all write operations to the device are allowed. The default value for all the registers in the DACx3401 is loaded from NVM as soon as a POR event is issued. Do not perform a read operation from the DAC register while NVM_BUSY = 1.

The DACx3401 also implements NVM_RELOAD bit (address D3h). Set this bit to 1 for the device to start an NVM reload operation. After the operation is complete, the device autoresets this bit to 0. During the NVM_RELOAD operation, the NVM_BUSY bit is set to 1.

Table 1. NVM Programmable Registers

REGISTER ADDRESS	REGISTER NAME	BIT ADDRESS	BIT NAME
D1h	GENERAL_CONFIG	15:14	FUNC_CONFIG
		13	DEVICE_LOCK
		11:9	CODE_STEP
		8:5	SLEW_RATE
		4:3	DAC_PDN
		2	REF_EN
		1:0	DAC_SPAN
D2h	MED_ALARM_CONFIG	10	MED_ALARM_HP
		9	MED_ALARM_MP
		8	MED_ALARM_LP
		5:4	INTERBURST_TIME
		3:2	PULSE_OFF_TIME
		1:0	PULSE_ON_TIME
D3h	TRIGGER	8	START_FUNC_GEN
10h	DAC_DATA	11:2	DAC_DATA
25h	DAC_MARGIN_HIGH	11:4	MARGIN_HIGH (8 most significant bits)
26h	DAC_MARGIN_LOW	11:4	MARGIN_LOW (8 most significant bits)

ADVANCE INFORMATION

8.3.3.1 NVM Cyclic Redundancy Check

The DACx3401 implement a cyclic redundancy check (CRC) feature for the device NVM to make sure that the data stored in the device NVM is uncorrupted. There are two types of CRC alarm bits implemented in DACx3401: NVM_CRC_ALARM_USER and NVM_CRC_ALARM_INTERNAL. The NVM_CRC_ALARM_USER bit indicates the status of user-programmable NVM bits, and the NVM_CRC_ALARM_INTERNAL bit indicates the status of internal NVM bits The CRC feature is implemented by storing a 10-Bit CRC (CRC-10-ATM) along with the NVM data each time NVM program operation (write or reload) is performed and during the device start up. The device reads the NVM data and validates the data with the stored CRC. The CRC alarm bits (NVM_CRC_ALARM_USER and NVM_CRC_ALARM_INTERNAL address D0h) report any errors after the data are read from the device NVM.

8.3.3.2 NVM_CRC_ALARM_USER Bit

A logic 1 on NVM_CRC_ALARM_USER bit indicates that the user-programmable NVM data is corrupt. During this condition, all registers in the DAC are initialized with factory reset values, and any DAC registers can be written to or read from. To reset the alarm bits to 0, issue a software reset (see the *Software Reset* section) command, or cycle power to the DAC. Alternatively, issue a software reset or cycle the power to reload the user-programmable NVM bits.

8.3.3.3 NVM_CRC_ALARM_INTERNAL Bit

A logic 1 on NVM_CRC_ALARM_INTERNAL bit indicates that the internal NVM data is corrupt. During this condition, all registers in the DAC are initialized with factory reset values, and any DAC registers can be written to or read from. To reset the alarm bits to 0, issue a software reset (see the *Software Reset* section) command or cycle power to the DAC.

8.3.4 Programmable Slew Rate

When the DAC data registers are written, the voltage on DAC output (V_{OUT}) immediately transitions to the new code following the slew rate and settling time specified in the *Electrical Characteristics* table. The slew rate control feature allows the user to control the rate at which the output voltage (V_{OUT}) changes. When this feature is enabled (using SLEW_RATE[3:0] bits), the DAC output changes from the current code to the code in MARGIN_HIGH (address 25h) or MARGIN_LOW (address 26h) registers (when margin high or low commands are issued to the DAC) using the step and rate set in CODE_STEP and SLEW_RATE bits. With the default slew rate control setting (CODE_STEP and SLEW_RATE bits, address D1h), the output changes smoothly at a rate limited by the output drive circuitry and the attached load. Using this feature, the output steps digitally at a rate defined by bits CODE_STEP and SLEW_RATE on address D1h. SLEW_RATE defines the rate at which the digital slew updates; CODE_STEP defines the amount by which the output value changes at each update. Table 2 and Table 3 show different settings for CODE_STEP and SLEW_RATE.

When the slew rate control feature is used, the output changes happen at the programmed slew rate. This configuration results in a staircase formation at the output. Do not write to CODE_STEP, SLEW_RATE, or DAC_DATA during the output slew.

Table 2. Code Step

REGISTER ADDRESS AND NAME	CODE_STEP[2]	CODE_STEP[1]	CODE_STEP[0]	COMMENT
D1h, GENERAL_CONFIG	0	0	0	Code step size = 1 LSB (default)
	0	0	1	Code step size = 2 LSB
	0	1	0	Code step size = 3 LSB
	0	1	1	Code step size = 4 LSB
	1	0	0	Code step size = 6 LSB
	1	0	1	Code step size = 8 LSB
	1	1	0	Code step size = 16 LSB
	1	1	1	Code step size = 32 LSB

Table 3. Slew Rate

REGISTER ADDRESS AND NAME	SLEW_RATE[3]	SLEW_RATE[2]	SLEW_RATE[1]	SLEW_RATE[0]	COMMENT
D1h, GENERAL_CONFIG	0	0	0	0	25.6 µs (per step)
	0	0	0	1	25.6 µs × 1.25 (per step)
	0	0	1	0	25.6 µs × 1.50 (per step)
	0	0	1	1	25.6 µs × 1.75 (per step)
	0	1	0	0	204.8 µs (per step)
	0	1	0	1	204.8 µs × 1.25 (per step)
	0	1	1	0	204.8 µs × 1.50 (per step)
	0	1	1	1	204.8 µs × 1.75 (per step)
	1	0	0	0	1.6384 ms (per step)
	1	0	0	1	1.6384 ms × 1.25 (per step)
	1	0	1	0	1.6384 ms × 1.50 (per step)
	1	0	1	1	1.6384 ms × 1.75 (per step)
	1	1	0	0	12 µs (per step)
	1	1	0	1	8 µs (per step)
	1	1	1	0	4 µs (per step)
	1	1	1	1	No slew (default)

ADVANCE INFORMATION

8.3.5 Power-on-Reset (POR)

The DACx3401 family of devices includes a power-on reset (POR) function that controls the output voltage at power up. After the V_{DD} supply has been established, a POR event is issued. The POR causes all registers to initialize to default values, and communication with the device is valid only after a 30-ms, POR delay. The default value for all the registers in the DACx3401 is loaded from NVM as soon as the POR event is issued.

When the device powers up, a POR circuit sets the device to the default mode. The POR circuit requires specific V_{DD} levels, as indicated in Figure 2, in order to make sure that the internal capacitors discharge and reset the device on power up. To make sure that a POR occurs, V_{DD} must be less than 0.7 V for at least 1 ms. When V_{DD} drops to less than 1.65 V, but remains greater than 0.7 V (shown as the undefined region), the device may or may not reset under all specified temperature and power-supply conditions. In this case, initiate a POR. When V_{DD} remains greater than 1.65 V, a POR does not occur.

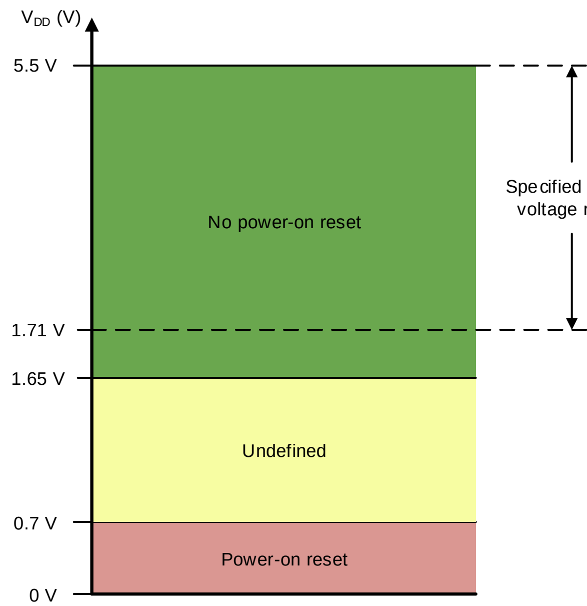


Figure 2. Threshold Levels for V_{DD} POR Circuit

8.3.6 Software Reset

To initiate a device software reset event, write the reserved code 1010 to the SW_RESET (address D3h). A software reset initiates a POR event.

8.3.7 Device Lock Feature

The DACx3401 implement a device lock feature that prevents an accidental or unintended write to the DAC registers. The device locks all the registers when the DEVICE_LOCK bit (address D1h) is set to 1. To reset this bit to 0 (unclock the device), write 0101 to the DEVICE_UNLOCK_CODE bits (address D3h).

ADVANCE INFORMATION

8.3.8 PMBus Compatibility

PMBus is an I²C-based communication standard for power-supply management. PMBus contains standard command codes tailored to power supply applications. The DACx3401 implements some PMBus commands such as *Turn Off*, *Turn On*, *Margin Low*, *Margin High*, *Communication Failure Alert Bit (CML)*, as well as *PMBUS revision*. Figure 3 shows typical PMBus connections

Similar to I²C, PMBus is a variable length packet of 8-bit data bytes, each with a receiver acknowledge, wrapped between a start and stop bit. The first byte is always a 7-bit *slave address* followed by a *write* bit, sometimes called the *even address* that identifies the intended receiver of the packet. The second byte is an 8-bit *command* byte, identifying the PMBus command being transmitted using the respective command code. After the command byte, the transmitter either sends data associated with the command to write to the receiver command register (from most significant byte to least significant byte), or sends a new start bit indicating the desire to read the data associated with the command register from the receiver. After, the receiver transmits the data following the same lowest-byte first format (see Table 10).

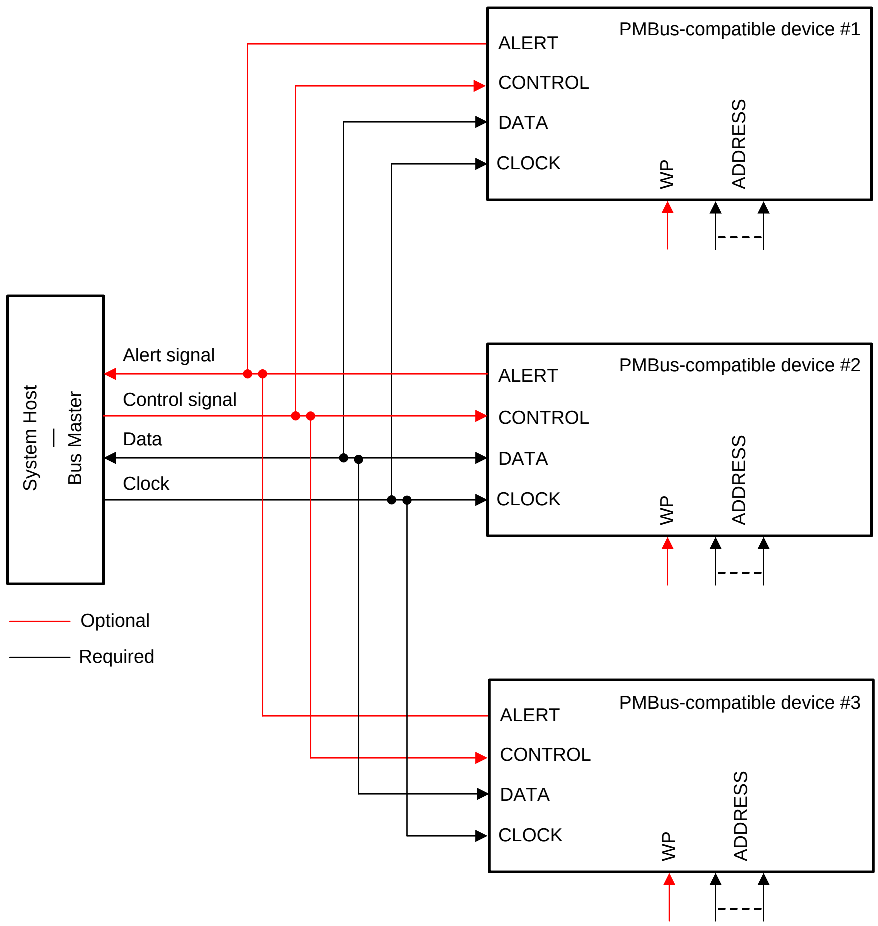


Figure 3. PMBus Connections

ADVANCE INFORMATION

8.4 Device Functional Modes

8.4.1 Power Down Mode

The DACx3401 output amplifier and internal reference can be independently powered down through the DAC_PDN bits (address D1h). At power up, the DAC output and the internal reference are disabled by default. In power-down mode, the DAC output (OUT pin) is in a high-impedance state. To change this state to 10kΩ-A_{GND} (at power up), use the DAC_PDN bits (address D1h).

The DAC power-up state can be programmed to any state (power-down or normal mode) using the NVM. Table 4 shows the DAC power-down bits.

Table 4. DAC Power-Down Bits

REGISTER ADDRESS AND NAME	DAC_PDN[1]	DAC_PDN[0]	DESCRIPTION
D1h, GENERAL_CONFIG	0	0	Power up
	0	1	Power down to 10 kΩ
	1	0	Power down to high impedance (HiZ) (default)
	1	1	Power down to 10 kΩ

8.4.2 Continuous Waveform Generation (CWG) Mode

The DACx3401 implement a continuous waveform generation feature. To set the device to this mode, set the START_FUNC_GEN (address D3h) to 1. In this mode, the DAC output pin (OUT) generates a continuous waveform based on the FUNC_CONFIG bits (address D1h). Table 5 shows the continuous waveforms that can be generated in this mode. The frequency of the waveform depends on the resistive and capacitive load on the OUT pin, high and low codes, and slew rate settings as shown in the following equations.

$$f_{SQUARE-WAVE} = \frac{1}{2 \times SLEW\_RATE}$$

where:
- SLEW_RATE is the programmable DAC slew rate specified in Table 3. (3)

$$f_{TRIANGLE-WAVE} = \frac{1}{2 \times SLEW\_RATE \times \left(\frac{MARGIN\_HIGH - MARGIN\_LOW + 1}{CODE\_STEP} \right)}$$

where:
- SLEW_RATE is the programmable DAC slew rate specified in Table 3.
- MARGIN_HIGH and MARGIN_LOW are the programmable DAC codes.
- CODE_STEP is the programmable DAC step code in Table 2. (4)

$$f_{SAWTOOTH-WAVE} = \frac{1}{SLEW\_RATE \times \left(\frac{MARGIN\_HIGH - MARGIN\_LOW + 1}{CODE\_STEP} \right)}$$

where:
- SLEW_RATE is the programmable DAC slew rate specified in Table 3.
- MARGIN_HIGH and MARGIN_LOW are the programmable DAC codes.
- CODE_STEP is the programmable DAC step code in Table 2. (5)

Table 5. FUNC_CONFIG bits

REGISTER ADDRESS AND NAME	FUNC_CONFIG[1]	FUNC_CONFIG[0]	DESCRIPTION
D1h, GENERAL_CONFIG	0	0	Generates a triangle wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code with slope defined by SLEW_RATE (address D1h) bits
	0	1	Generates Saw-Tooth wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code, with rising slope defined by SLEW_RATE (address D1h) bits and immediate falling edge
	1	0	Generates Saw-Tooth wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code, with falling slope defined by SLEW_RATE (address D1h) bits and immediate rising edge
	1	1	Generates a square wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code with pulse high and low period defined by SLEW_RATE (address D1h) bits

8.4.3 PMBus Compatibility Mode

The DACx3401 I^2C interface implements some of the PMBus commands. Table 6 shows the supported PMBus commands that are implemented in DACx3401.The DAC uses MARGIN_LOW (address 26h), MARGIN_HIGH (address 25h) bits, SLEW_RATE, and CODE_STEP bits (address D1h) for PMBUS_OPERATION_CMD.

Table 6. PMBus Operation Commands

REGISTER ADDRESS AND NAME	PMBUS_OPERATION_CMD[15:8]	DESCRIPTION
01h, PMBUS_OPERATION	00h	Turn off
	80h	Turn on
	A4h	Margin low
	94h	Margin high

The DACx3401 also implements a time-out failure feature for PMBus mode. The CML bit (address 78h) indicates the status of the PMBus. This bit can be reset by writing 1.

To read the PMBus version, read the PMBUS_VERSION bits (address 98h).

8.4.4 Medical Alarm Generation Mode

The DACx3401 can also be used to generate continuous alarm tones for medical devices. Use a suitable analog mixer, audio amplifier, and a speaker to generate low, medium, or high priority alarm tones. See the *Application and Implementation* section for more details. The DACx3401 allows tunability and configurability to support different alarm generation. Using this approach, configurable medical alarm tones can be generated with a simple circuit, and with no need for runtime software.

8.4.4.1 Low-Priority Alarm

The MED_ALARM_LP bit (address D2h) is used to trigger a medical low-priority alarm generation. The DAC generates a continuous-alarm signal until this bit is set back to 0. After the bit is set to 0, the device does not abruptly end the alarm generation; the device stops only after completing the ongoing burst.

8.4.4.2 Medium-Priority Alarm

The MED_ALARM_MP bit (address D2h) is used to trigger a medical medium-priority alarm generation. The DAC generates a continuous-alarm signal until this bit is set back to 0. After the bit is set to 0, the device does not abruptly end the alarm generation; the device stops only after completing the ongoing burst.

ADVANCE INFORMATION

8.4.4.3 High-Priority Alarm

The MED_ALARM_HP bit (address D2h) is used to trigger a medical high-priority alarm generation. The DAC generates a continuous-alarm signal until this bit is set back to 0. After the bit is set to 0, the device does not abruptly end the alarm generation; the device stops only after completing the ongoing burst.

8.4.4.4 Interburst Time

The INTERBURST_TIME bit (address D2h) is used set the time between two adjacent bursts. Table 7 lists the INTERBURST_TIME settings.

Table 7. Interburst Time

REGISTER ADDRESS AND NAME	INTERBURST_TIME[1:0]	HIGH PRIORITY ALARM INTERBURST TIME	MEDIUM PRIORITY ALARM INTERBURST TIME	LOW PRIORITY ALARM INTERBURST TIME
D2h, MED_ALARM_CONFIG	00	2.55 s	2.60 s	16 s
	01	2.96 s	3.06 s	
	10	3.38 s	3.52 s	
	11	3.80 s	4.00 s	

8.4.4.5 Pulse Off Time

The PULSE_OFF_TIME bit (address D2h) is used to control the low period of trapezoid in a medical alarm waveform. Table 8 lists the PULSE_OFF_TIME settings.

Table 8. Pulse Off Time

REGISTER ADDRESS AND NAME	PULSE_OFF_TIME[1:0]	HIGH PRIORITY ALARM PULSE OFF TIME	MEDIUM PRIORITY ALARM PULSE OFF TIME	LOW PRIORITY ALARM PULSE OFF TIME
D2h, MED_ALARM_CONFIG	00	15 ms	40 ms	40 ms
	01	36 ms	60 ms	60 ms
	10	58 ms	80 ms	80 ms
	11	80 ms	100 ms	100 ms

8.4.4.6 Pulse On Time

The PULSE_ON_TIME bit (address D2h) controls the high period of trapezoid in a medical alarm waveform. Table 9 lists the PULSE_ON_TIME settings.

Table 9. Pulse On Time

REGISTER ADDRESS AND NAME	PULSE_ON_TIME[1:0]	HIGH PRIORITY ALARM PULSE ON TIME	MEDIUM PRIORITY ALARM PULSE ON TIME	LOW PRIORITY ALARM PULSE ON TIME
D2h, MED_ALARM_CONFIG	00	80 ms	130 ms	130 ms
	01	103 ms	153 ms	153 ms
	10	126 ms	176 ms	176 ms
	11	150 ms	200 ms	200 ms

8.5 Programming

The DACx3401 devices have a 2-wire serial interface (SCL and SDA), and one address pin (A0), as shown in the *Pin Configuration and Functions* section. The I^2C bus consists of a data line (SDA) and a clock line (SCL) with pullup structures. When the bus is idle, both SDA and SCL lines are pulled high. All the I^2C-compatible devices connect to the I^2C bus through the open drain I/O pins, SDA and SCL.

The I^2C specification states that the device that controls communication is called a *master*, and the devices that are controlled by the master are called *slaves*. The master device generates the SCL signal. The master device also generates special timing conditions (start condition, repeated start condition, and stop condition) on the bus to indicate the start or stop of a data transfer. Device addressing is completed by the master. The master device on an I^2C bus is typically a microcontroller or digital signal processor (DSP). The DACx3401 family operates as a slave device on the I^2C bus. A slave device acknowledges master commands, and upon master control, receives or transmits data.

Typically, the DACx3401 family operates as a slave receiver. A master device writes to the DACx3401, a slave receiver. However, if a master device requires the DACx3401 internal register data, the DACx3401 operates as a slave transmitter. In this case, the master device reads from the DACx3401. According to I^2C terminology, read and write refer to the master device.

The DACx3401 family is a slave and supports the following data transfer modes:
- Standard mode (100 kbps)
- Fast mode (400 kbps)
- Fast+ mode (1.0 Mbps)

The data transfer protocol for standard and fast modes is exactly the same; therefore, both modes are referred to as F/S-mode in this document. The fast+ mode protocol is supported in terms of data transfer speed, but not output current. The low-level output current would be 3 mA; similar to the case of standard and fast modes. The DACx3401 family supports 7-bit addressing. The 10-bit addressing mode is not supported. The device supports the general call reset function. Sending the following sequence initiates a software reset within the device: start or repeated start, 0x00, 0x06, stop. The reset is asserted within the device on the falling edge of the ACK bit, following the second byte.

Other than specific timing signals, the I^2C interface works with serial bytes. At the end of each byte, a ninth clock cycle generates and detects an acknowledge signal. Acknowledge is when the SDA line is pulled low during the high period of the ninth clock cycle. A not-acknowledge is when the SDA line is left high during the high period of the ninth clock cycle as shown in Figure 4.

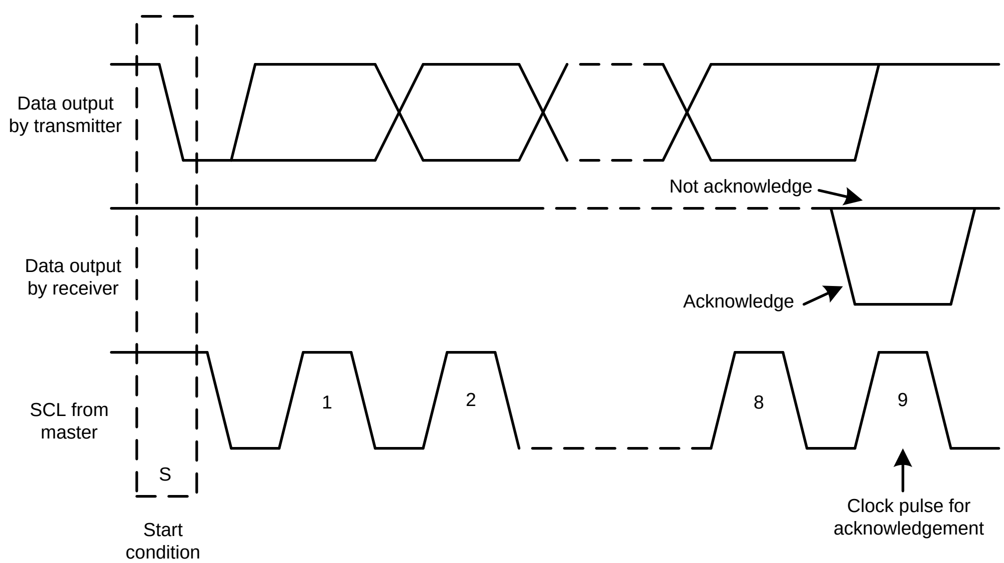


Figure 4. Acknowledge and Not Acknowledge on the I^2C Bus

ADVANCE INFORMATION

Programming (continued)

8.5.1 F/S Mode Protocol

The following steps explain a complete transaction in F/S mode.

1. The master initiates data transfer by generating a start condition. The start condition is when a high-to-low transition occurs on the SDA line while SCL is high, as shown in Figure 5. All I^2C-compatible devices recognize a start condition.
2. The master then generates the SCL pulses, and transmits the 7-bit address and the read/write direction bit (R/W) on the SDA line. During all transmissions, the master makes sure that data are valid. A valid data condition requires the SDA line to be stable during the entire high period of the clock pulse, as shown in Figure 6. All devices recognize the address sent by the master and compare the address to the respective internal fixed address. Only the slave device with a matching address generates an acknowledge by pulling the SDA line low during the entire high period of the 9th SCL cycle, as shown in Figure 4. When the master detects this acknowledge, the communication link with a slave has been established.
3. The master generates further SCL cycles to transmit (R/W bit 0) or receive (R/W bit 1) data to the slave. In either case, the receiver must acknowledge the data sent by the transmitter. The acknowledge signal can be generated by the master or by the slave, depending on which is the receiver. The 9-bit valid data sequences consists of 8-data bits and 1 acknowledge-bit, and can continue as long as necessary.
4. To signal the end of the data transfer, the master generates a stop condition by pulling the SDA line from low-to-high while the SCL line is high (see Figure 5). This action releases the bus and stops the communication link with the addressed slave. All I^2C-compatible devices recognize the stop condition. Upon receipt of a stop condition, the bus is released, and all slave devices then wait for a start condition followed by a matching address.

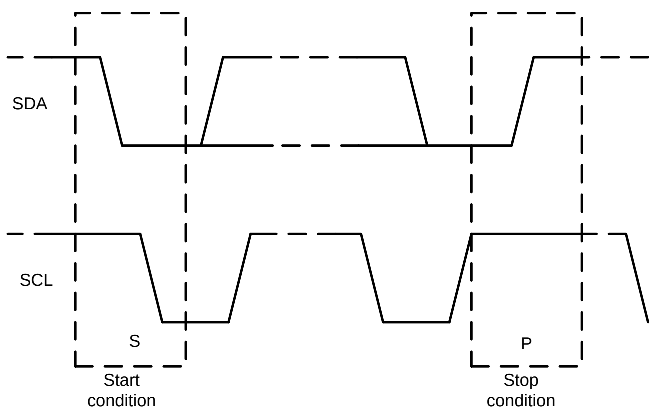


Figure 5. Start and Stop Conditions

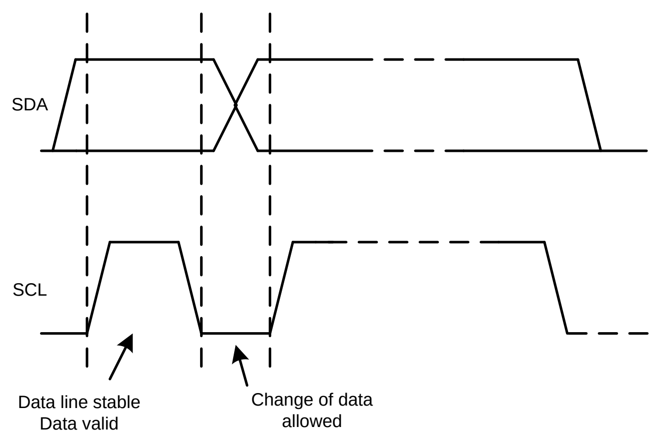


Figure 6. Bit Transfer on the I^2C Bus

ADVANCE INFORMATION

8.5.2 DACx3401 I²C Update Sequence

For a single update, the DACx3401 requires a start condition, a valid I²C address byte, a command byte, and two data bytes, as listed in Table 10.

Table 10. Update Sequence

MSB	....	LSB	ACK	MSB	...	LSB	ACK	MSB	...	LSB	ACK	MSB	...	LSB	ACK
Address (A) byte Address Byte				Command byte Command Byte				Data byte - MSDB Data Byte (MSDB and LSDB)				Data byte - LSDB Data Byte (MSDB and LSDB)			
DB [32:24]				DB [23:16]				DB [15:8]				DB [7:0]			

After each byte is received, the DACx3401 family acknowledges the byte by pulling the SDA line low during the high period of a single clock pulse, as shown in Figure 7. These four bytes and acknowledge cycles make up the 36 clock cycles required for a single update to occur. A valid I²C address byte selects the DACx3401 devices.

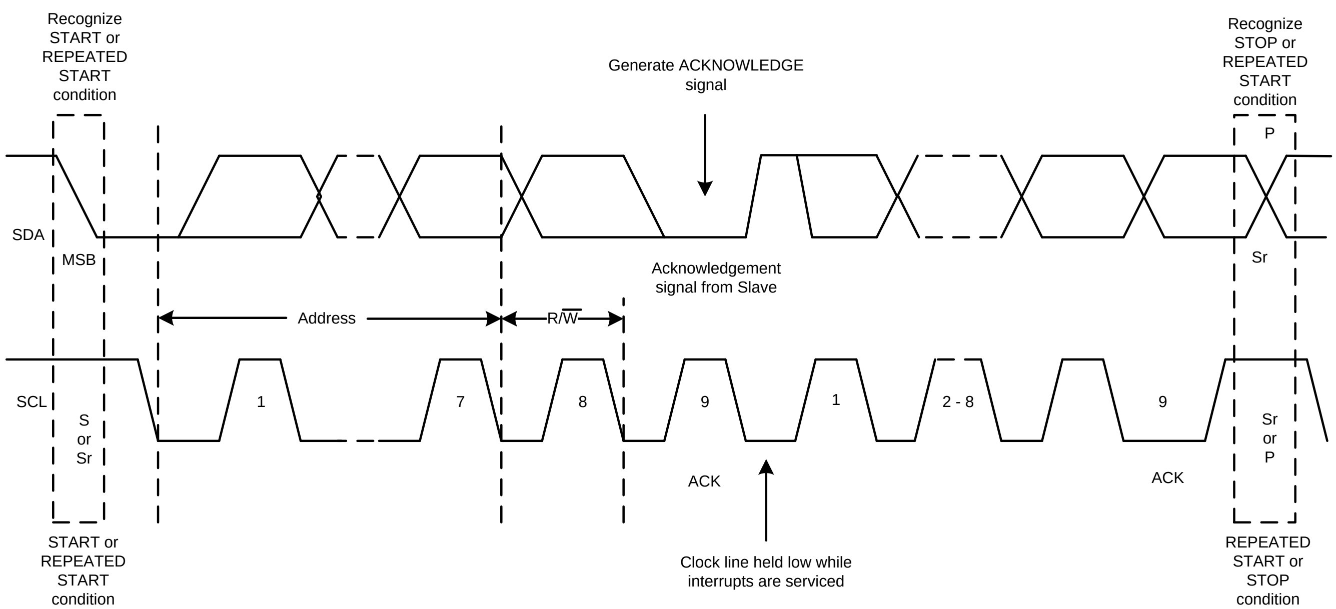


Figure 7. I²C Bus Protocol

The command byte sets the operating mode of the selected DACx3401 device. For a data update to occur when the operating mode is selected by this byte, the DACx3401 device must receive two data bytes: the most significant data byte (MSDB) and least significant data byte (LSDB). The DACx3401 device performs an update on the falling edge of the acknowledge signal that follows the LSDB.

When using fast mode (clock = 400 kHz), the maximum DAC update rate is limited to 22.22 kSPS. Using the fast+ mode (clock = 1 MHz), the maximum DAC update rate is limited to 55.55 kSPS. When a stop condition is received, the DACx3401 device releases the I²C bus and awaits a new start condition.

8.5.3 Address Byte

The address byte, as shown in Table 11, is the first byte received following the start condition from the master device. The first four bits (MSBs) of the address are factory preset to 1001. The next three bits of the address are controlled by the A0 pin. The A0 pin input can be connected to VDD, AGND, SCL, or SDA. The A0 pin is sampled during the first byte of each data frame to determine the address. The device latches the value of the address pin, and consequently responds to that particular address according to Table 12.

The DACx3401 family supports broadcast addressing, which can be used for synchronously updating or powering down multiple DACx3401 devices. The DACx3401 family is designed to work with other members of the family to support multichip synchronous updates. Using the broadcast address, the DACx3401 devices respond regardless of the states of the address pins. Broadcast is supported only in write mode.

ADVANCE INFORMATION

Table 11. Address Byte

COMMENT	MSB							LSB
—	AD6	AD5	AD4	AD3	AD2	AD1	AD0	R/$\overline{W}$
General address	1	0	0	1	See Table 12 (slave address column)			0 or 1
Broadcast address	1	0	0	0	1	1	1	0

Table 12. Address Format

SLAVE ADDRESS	A0 PIN
000	VDD
001	AGND
010	SDA
011	SCL

8.5.4 Command Byte

Table 17 lists the command byte.

Table 13. Command Byte (Register Names)

ADDRESS	REGISTER NAME
D0h	STATUS
D1h	GENERAL_CONFIG
D2h	MED_ALARM_CONFIG
D3h	TRIGGER
21h	DAC_DATA
25h	DAC_MARGIN_HIGH
26h	DAC_MARGIN_LOW
01h	PMBUS_OP
78h	PMBUS_STATUS_BYTE
98h	PMBUS_VERSION

8.5.5 Data Byte (MSDB and LSDB)

Table 14. Data Byte (MSDB and LSDB)(1)

<table>
<tr><th colspan="8">MSDB</th><th colspan="8">LSDB</th></tr>
<tr><th>BIT15</th><th>BIT14</th><th>BIT13</th><th>BIT12</th><th>BIT11</th><th>BIT10</th><th>BIT9</th><th>BIT8</th><th>BIT7</th><th>BIT6</th><th>BIT5</th><th>BIT4</th><th>BIT3</th><th>BIT2</th><th>BIT1</th><th>BIT0</th></tr>
<tr><td>NVM_CRC_ ALARM_ USER</td><td>NVM_CRC_ ALARM_ INTERNAL</td><td>NVM_BUSY</td><td>DAC_ UPDATE_ BUSY</td><td colspan="6">X</td><td colspan="4">DEVICE_ID</td><td colspan="2">VERSION_ID</td></tr>
<tr><td colspan="2">FUNC_CONFIG</td><td>DEVICE_ LOCK</td><td>Reserved</td><td colspan="3">CODE_STEP</td><td colspan="4">SLEW_RATE</td><td colspan="2">DAC_PDN</td><td>REF_EN</td><td colspan="2">DAC_SPAN</td></tr>
<tr><td colspan="5">X</td><td>MED_ ALARM_HP</td><td>MED_ ALARM_MP</td><td>MED_ ALARM_LP</td><td colspan="2">RESERVED</td><td colspan="2">INTERBURST_TIME</td><td colspan="2">PULSE_OFF_TIME</td><td colspan="2">PULSE_ON_TIME</td></tr>
<tr><td colspan="4">DEVICE_UNLOCK_CODE</td><td colspan="2">X</td><td>DEVICE_ CONFIG_ RESET</td><td>START_ FUNC_GEN</td><td>PMBUS_ MARGIN_ HIGH</td><td>PMBUS_ MARGIN_ LOW</td><td>NVM_ RELOAD</td><td>NVM_PROG</td><td colspan="4">SW_RESET</td></tr>
<tr><td colspan="4">X</td><td colspan="10">DAC_DATA[9:0] / DAC_DATA[7:0]</td><td colspan="2">X</td></tr>
<tr><td colspan="4">X</td><td colspan="10">MARGIN_HIGH[9:0] / MARGIN_HIGH[7:0]</td><td colspan="2">X</td></tr>
<tr><td colspan="4">X</td><td colspan="10">MARGIN_LOW[9:0] / MARGIN_LOW[7:0]</td><td colspan="2">X</td></tr>
<tr><td colspan="8">PMBUS_OPERATION_CMD</td><td colspan="8">X</td></tr>
<tr><td colspan="6">X</td><td>CML</td><td colspan="9">X</td></tr>
<tr><td colspan="8">PMBUS_VERSION</td><td colspan="8">X</td></tr>
</table>

(1) X = Don't care.

ADVANCE INFORMATION

8.5.6 I²C Read Sequence

To read any register the following command sequence must be used:

1. Send a start or repeated start command with a slave address and the R/$\overline{W}$ bit set to 0 for writing. The device acknowledges this event.
2. Send a command byte for the register to be read. The device acknowledges this event again.
3. Send a repeated start with the slave address and the R/$\overline{W}$ bit set to 1 for reading. The device acknowledges this event.
4. The device writes the MSDB byte of the addressed register. The master must acknowledge this byte.
5. Finally, the device writes out the LSDB of the register.

An alternative reading method allows for reading back the value of the last register written. The sequence is a start or repeated start with the slave address and the R/$\overline{W}$ bit set to 1, and the two bytes of the last register are read out.

Note that it is not possible to use the broadcast address for reading.

Table 15. Read Sequence

S	MSB	…	R/$\overline{W}$ (0)	ACK	MSB	…	LSB	ACK	Sr	MSB	…	R/$\overline{W}$ (1)	ACK	MSB	…	LSB	ACK	MSB	…	LSB	ACK
	ADDRESS BYTE Address Byte				COMMAND BYTE Command Byte				Sr	ADDRESS BYTE Address Byte				MSDB				LSDB			
From Master				Slave	From Master			Slave	From Master				Slave	From Slave			Master	From Slave			Master

8.6 Register Map

Table 16. Register Map(1)

<table>
<tr><th>Address</th><th>BIT15</th><th>BIT14</th><th>BIT13</th><th>BIT12</th><th>BIT11</th><th>BIT10</th><th>BIT9</th><th>BIT8</th><th>BIT7</th><th>BIT6</th><th>BIT5</th><th>BIT4</th><th>BIT3</th><th>BIT2</th><th>BIT1</th><th>BIT0</th></tr>
<tr><td>D0h</td><td>NVM_CRC_ ALARM_ USER</td><td>NVM_CRC_ ALARM_ INTERNAL</td><td>NVM_BUSY</td><td>DAC_ UPDATE_ BUSY</td><td colspan="6">X</td><td colspan="4">DEVICE_ID</td><td colspan="2">VERSION_ID</td></tr>
<tr><td>D1h</td><td colspan="2">FUNC_CONFIG</td><td>DEVICE_ LOCK</td><td>Reserved</td><td colspan="3">CODE_STEP</td><td colspan="4">SLEW_RATE</td><td colspan="2">DAC_PDN</td><td>REF_EN</td><td colspan="2">DAC_SPAN</td></tr>
<tr><td>D2h</td><td colspan="5">X</td><td>MED_ ALARM_HP</td><td>MED_ ALARM_MP</td><td>MED_ ALARM_LP</td><td colspan="2">RESERVED</td><td colspan="2">INTERBURST_TIME</td><td colspan="2">PULSE_OFF_TIME</td><td colspan="2">PULSE_ON_TIME</td></tr>
<tr><td>D3h</td><td colspan="4">DEVICE_UNLOCK_CODE</td><td colspan="2">X</td><td>DEVICE_ CONFIG_ RESET</td><td>START_ FUNC_ GEN</td><td>PMBUS_ MARGIN_ HIGH</td><td>PMBUS_ MARGIN_ LOW</td><td>NVM_ RELOAD</td><td>NVM_ PROG</td><td colspan="4">SW_RESET</td></tr>
<tr><td>21h</td><td colspan="4">X</td><td colspan="10">DAC_DATA[9:0] (10-Bit) or DAC_DATA[7:0] (8-Bit)</td><td colspan="2">X</td></tr>
<tr><td>25h</td><td colspan="4">X</td><td colspan="10">MARGIN_HIGH[9:0] (10-Bit) or MARGIN_HIGH[7:0] (8-Bit)</td><td colspan="2">X</td></tr>
<tr><td>26h</td><td colspan="4">X</td><td colspan="10">MARGIN_LOW[9:0] (10-Bit) or MARGIN_LOW[7:0] (8-Bit)</td><td colspan="2">X</td></tr>
<tr><td>01h</td><td colspan="8">PMBUS_OPERATION_CMD</td><td colspan="8">X</td></tr>
<tr><td>78h</td><td colspan="6">X</td><td>CML</td><td colspan="9">X</td></tr>
<tr><td>98h</td><td colspan="8">PMBUS_VERSION</td><td colspan="8">X</td></tr>
</table>

(1) X = Don't care.

Table 17. Register Names

ADDRESS	ACRONYM	REGISTER NAME	SECTION
D0h	STATUS	STATUS	STATUS Register (address = D0h) (reset = 000Ch or 0014h)
D1h	GENERAL_CONFIG	GENERAL_CONFIG	GENERAL_CONFIG Register (address = D1h) (reset = 11F0h)
D2h	MED_ALARM_CONFIG	MED_ALARM_CONFIG	MED_ALARM_CONFIG Register (address = D2h) (reset = 0000h)
D3h	TRIGGER	TRIGGER	TRIGGER Register (address = D3h) (reset = 0000h)
21h	DAC_DATA	DAC_DATA	DAC_DATA Register (address = 21h) (reset = 0000h)
25h	DAC_MARGIN_HIGH	DAC_MARGIN_HIGH	DAC_MARGIN_HIGH Register (address = 25h) (reset = 0000h)
26h	DAC_MARGIN_LOW	DAC_MARGIN_LOW	DAC_MARGIN_LOW Register (address = 26h) (reset = 0000h)
01h	PMBUS_OPERATION	PMBUS_OP	PMBUS_OPERATION Register (address = 01h) (reset = 0000h)
78h	PMBUS_STATUS_BYTE	PMBUS_STATUS_BYTE	PMBUS_STATUS_BYTE Register (address = 78h) (reset = 0000h)
98h	PMBUS_VERSION	PMBUS_VERSION	PMBUS_VERSION Register (address = 98h) (reset = 2200h)

8.6.1 STATUS Register (address = D0h) (reset = 000Ch or 0014h)

Figure 8. STATUS Register

15	14	13	12	11 10 9 8 7 6	5 4 3 2	1 0
NVM_CRC_ ALARM_ USER	NVM_CRC_ ALARM_ INTERNAL	NVM_ BUSY	DAC_ UPDATE_ BUSY	X	DEVICE_ID	VERSION_ID
R	R	R	R	X	R	R

Table 18. STATUS Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15	NVM_CRC_ALARM_USER	R	0	0 : No CRC error in user NVM bits 1: CRC error in user NVM bits
14	NVM_CRC_ALARM_INTERNAL	R	0	0 : No CRC error in internal NVM 1: CRC error in internal NVM bits
13	NVM_BUSY	R	0	0 : NVM write or load completed, Write to DAC registers allowed 0 : NVM write or load in progress, Write to DAC registers not allowed
12	DAC_UPDATE_BUSY	R	0	0 : DAC outputs updated, Write to DAC registers allowed 0 : DAC outputs update in progress, Write to DAC registers not allowed
11 - 6	X	X	00h	Don't care
5 - 2	DEVICE_ID	R R	DAC53401: 0Ch DAC43401: 14h	DAC53401: 0Ch DAC43401: 14h
1 - 0	VERSION_ID			

8.6.2 GENERAL_CONFIG Register (address = D1h) (reset = 11F0h)

Figure 9. GENERAL_CONFIG Register

15 14	13	12	11 10 9	8 7 6 5	4 3	2	1 0
FUNC_ CONFIG	DEVICE_ LOCK	EN_ PMBUS	CODE_STEP	SLEW_RATE	DAC_PDN	REF_EN	DAC_SPAN
R/$\overline{W}$	$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$

Table 19. GENERAL_CONFIG Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15 - 14	FUNC_CONFIG	R/$\overline{W}$	00	00 : Generates a triangle wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code with slope defined by SLEW_RATE (address D1h) bits 01: Generates Saw-Tooth wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code, with rising slope defined by SLEW_RATE (address D1h) bits and immediate falling edge 10: Generates Saw-Tooth wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code, with falling slope defined by SLEW_RATE (address D1h) bits and immediate rising edge 11: Generates a square wave between MARGIN_HIGH (address 25h) code to MARGIN_LOW (address 26h) code with pulse high and low period defined by SLEW_RATE (address D1h) bits
13	DEVICE_LOCK	$\overline{W}$	0	0 : Device not locked 1: Device locked, the device locks all the registers. This bit can be reset (unlock device) by writing 0101 to the DEVICE_UNLOCK_CODE bits (address D3h)
12	RESERVED	Reserved	1	Reserved

ADVANCE INFORMATION

ADVANCE INFORMATION

Table 19. GENERAL_CONFIG Register Field Descriptions (continued)

BIT	FIELD	TYPE	RESET	DESCRIPTION
11 - 9	CODE_STEP	R/$\overline{W}$	000	Code step for programmable slew rate control. 000: Code step size = 1 LSB (default) 001: Code step size = 2 LSB 010: Code step size = 3 LSB 011: Code step size = 4 LSB 100: Code step size = 6 LSB 101: Code step size = 8 LSB 110: Code step size = 16 LSB 111: Code step size = 32 LSB
8 - 5	SLEW_RATE	R/$\overline{W}$	1111	Slew rate for programmable slew rate control. 0000: 25.6 µs (per step) 0001: 25.6 µs × 1.25 (per step) 0010: 25.6 µs × 1.50 (per step) 0011: 25.6 µs × 1.75 (per step) 0100: 204.8 µs (per step) 0101: 204.8 µs × 1.25 (per step) 0110: 204.8 µs × 1.50 (per step) 0111: 204.8 µs × 1.75 (per step) 1000: 1.6384 ms (per step) 1001: 1.6384 ms × 1.25 (per step) 1010: 1.6384 ms × 1.50 (per step) 1011: 1.6384 ms × 1.75 (per step) 1100: 12 µs (per step) 1101: 8 µs (per step) 1110: 4 µs (per step) 1111: No slew (default)
4 - 3	DAC_PDN	R/$\overline{W}$	10	00: Power up 01: Power down to 10K 10: Power down to high impedance (default) 11: Power down to 10K
2	REF_EN	R/$\overline{W}$	0	0: Internal reference disabled, V_{DD} is DAC reference voltage, DAC output range from 0 to V_{DD}. 1: 0: Internal reference enabled, DAC reference = 1.21 V
1 - 0	DAC_SPAN	R/$\overline{W}$	00	Only applicable when internal reference is enabled. 00: Reference to V_{OUT} gain 1.5X 01: Reference to V_{OUT} gain 2X 10: Reference to V_{OUT} gain 3X 11: Reference to V_{OUT} gain 4X

8.6.3 MED_ALARM_CONFIG Register (address = D2h) (reset = 0000h)

Figure 10. MED_ALARM_CONFIG Register

15 14 13 12 11	10	9	8	7 6	5 4	3 2	1 0
X	MED_ ALARM_ HP	MED_ ALARM_ MP	MED_ ALARM_ LP	RESERVED	MED_ALARM_ DEAD_TIME	PULSE_ OFF_TIME	PULSE_ ON_TIME
X	$\overline{W}$	$\overline{W}$	$\overline{W}$	RESERVED	$\overline{W}$	$\overline{W}$	$\overline{W}$

Table 20. MED_ALARM_CONFIG Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION		
15 - 11	X	X	00h	Don't care		
10	MED_ALARM_HP	$\overline{W}$	0	0: No medical alarm waveform generated 1: High priority medical alarm waveform generated		
9	MED_ALARM_MP	$\overline{W}$	0	0: No medical alarm waveform generated 1: Medium priority medical alarm waveform generated		
8	MED_ALARM_LP	$\overline{W}$	0	0: No medical alarm waveform generated 1: Low priority medical alarm waveform generated		
7 - 6	RESERVED	Reserved	0	RESERVED		
5 - 4	INTERBURST_TIME	$\overline{W}$	00	High priority alarm 00: 2.55 sec 01: 2.96 sec 10: 3.38 sec 11: 3.80 sec	Medium priority alarm 00: 2.60 sec 01: 3.06 sec 10: 3.52 sec 11: 4.00 sec	Low priority alarm 00: 16 sec 01: 16 sec 10: 16 sec 11: 16 sec
3 - 2	PULSE_OFF_TIME	$\overline{W}$	00	High priority alarm 00: 15 msec 01: 36 msec 10: 58 msec 11: 80 msec	Medium priority alarm 00: 40 msec 01: 60 msec 10: 80 msec 11: 100 msec	Low priority alarm 00: 40 msec 01: 60 msec 10: 80 msec 11: 100 msec
1 - 0	PULSE_ON_TIME	$\overline{W}$	00	High priority alarm 00: 80 msec 01: 103 msec 10: 126 msec 11: 150 msec	Medium priority alarm 00: 130 msec 01: 153 msec 10: 176 msec 11: 200 msec	Low priority alarm 00: 130 msec 01: 153 msec 10: 176 msec 11: 200 msec

8.6.4 TRIGGER Register (address = D3h) (reset = 0000h)

Figure 11. TRIGGER Register

15 14 13 12	11 10	9	8	7	6	5	4	3 2 1 0
DEVICE_UNLOCK_CODE	X	DEVICE_ CONFIG_ RESET	START_ FUNC_ GEN	PMBUS_ MARGIN_ LOW	PMBUS_ MARGIN_ HIGH	NVM_ RELOAD	NVM_ PROG	SW_RESET
$\overline{W}$	X	$\overline{W}$	$\overline{W}$	R/$\overline{W}$	R/$\overline{W}$	$\overline{W}$	$\overline{W}$	$\overline{W}$

Table 21. TRIGGER Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15 - 12	DEVICE_UNLOCK_CODE	$\overline{W}$	0000	Write 0101 to unlock the device to bypass DEVICE_LOCK bit.
11 - 10	X	X	0h	Don't care
9	DEVICE_CONFIG_RESET	$\overline{W}$	0	0: Device configuration reset not initiated 1: Device configuration reset initiated. All registers loaded with default settings. All user NVM bits set to default setting (factory reset).
8	START_FUNC_GEN	$\overline{W}$	0	0: Continuous waveform generation mode disabled 1: Continuous waveform generation mode enabled, device generates continuous waveform based on FUNC_CONFIG (D1h), MARGIN_LOW (address 18h), and SLEW_RATE (address D1h) bits.
7	PMBUS_MARGIN_HIGH	R/$\overline{W}$	0	0: PMBus margin high command not initiated 1: PMBus margin high command initiated, DAC output margins high to MARGIN_HIGH code (address 25h).
6	PMBUS_MARGIN_LOW	R/$\overline{W}$	0	0: PMBus margin low command not initiated 1: PMBus margin low command initiated, DAC output margins low to MARGIN_LOW code (address 26h).
5	NVM_RELOAD	$\overline{W}$	0	0: NVM reload not initiated 1: NVM reload initiated, applicable DAC registers loaded with corresponding NVM. NVM_BUSY bit set to 1 which this operation is in progress.
4	NVM_PROG	$\overline{W}$	0	0: NVM write not initiated 1: NVM write initiated, NVM corresponding to applicable DAC registers loaded with existing register settings. NVM_BUSY bit set to 1 which this operation is in progress.
3 - 0	SW_RESET	$\overline{W}$	0000	00000: Software reset not initiated 1010: Software reset initiated, DAC registers loaded with corresponding NVMs, all other registers loaded with default settings.

8.6.5 DAC_DATA Register (address = 21h) (reset = 0000h)

Figure 12. DAC_DATA Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X				DAC_DATA[9:0] / DAC_DATA[7:0] – MSB Left aligned										X	
X				$\overline{W}$										X	

Table 22. DAC_DATA Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15-12	X	X	0h	Don't care
11-2	DAC_DATA[9:0] / DAC_DATA[7:0]	$\overline{W}$	000h	Writing to the DAC_DATA register forces the respective DAC channel to update the active register data to the DAC_DATA. Data are in straight binary format and use the following format: DAC53401: { DATA[9:0] } DAC43401: { DATA[7:0], X, X } X = Don't care bits
1-0	X	X	0h	Don't care

8.6.6 DAC_MARGIN_HIGH Register (address = 25h) (reset = 0000h)

Figure 13. DAC_MARGIN_HIGH Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X				MARGIN_HIGH[9:0] / MARGIN_HIGH[7:0] – MSB Left aligned										X	
X				$\overline{W}$										X	

Table 23. DAC_MARGIN_HIGH Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15-12	X	X	0h	Don't care
11-2	MARGIN_HIGH[9:0] / MARGIN_HIGH[7:0] – MSB Left aligned	$\overline{W}$	000h	Margin high code for DAC output. Data are in straight binary format and use the following format: DAC53401: { MARGIN_HIGH[[9:0] } DAC43401: { MARGIN_HIGH[[7:0], X, X } X = Don't care bits
1-0	X	X	0h	Don't care

8.6.7 DAC_MARGIN_LOW Register (address = 26h) (reset = 0000h)

Figure 14. DAC_MARGIN_LOW Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X				MARGIN_LOW[9:0] / MARGIN_LOW[7:0] – MSB Left aligned										X	
X				$\overline{W}$										X	

Table 24. DAC_MARGIN_LOW Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15-12	X	X	0h	Don't care
11-2	MARGIN_LOW[9:0] / MARGIN_LOW[7:0] – MSB Left aligned	$\overline{W}$	000h	Margin low code for DAC output. Data is in straight binary format and follows the format below: DAC53401: { MARGIN_LOW[[9:0] } DAC43401: { MARGIN_LOW[[7:0], X, X } X = Don't care bits
1-0	X	X	0h	Don't care

ADVANCE INFORMATION

8.6.8 PMBUS_OPERATION Register (address = 01h) (reset = 0000h)

Figure 15. PMBUS_OPERATION Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMBUS_OPERATION_CMD								X							
R/$\overline{W}$								X							

Table 25. PMBUS_OPERATION Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15 - 8	PMBUS_OPERATION_CMD	R/$\overline{W}$	00h	PMBus operation commands 00h: Turn off 80h: Turn on A4h: Margin low, DAC output margins low to MARGIN_LOW code (address 26h) 94h: Margin high, DAC output margins high to MARGIN_HIGH code (address 25h)
7 - 0	X	X	00h	Don't care

8.6.9 PMBUS_STATUS_BYTE Register (address = 78h) (reset = 0000h)

Figure 16. PMBUS_STATUS_BYTE Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X						CML	X								
X						R/$\overline{W}$	X								

Table 26. PMBUS_STATUS_BYTE Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15 - 10	X	X	00h	Don't care
9	CML	R/$\overline{W}$	0	0: No timeout failure 1: PMBus timeout failure; this bit can be reset by writing 1.
8 - 0	X	X	000h	Don't care

8.6.10 PMBUS_VERSION Register (address = 98h) (reset = 2200h)

Figure 17. PMBUS_VERSION Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMBUS_VERSION								X							
R								X							

Table 27. PMBUS_VERSION Register Field Descriptions

BIT	FIELD	TYPE	RESET	DESCRIPTION
15 - 8	PMBUS_VERSION	R	22h	PMBus version
7 - 0	X	X	00h	Don't care

ADVANCE INFORMATION

9 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

9.1 Application Information

The DACx3401 is a buffered-output, single-channel, low-power DAC available in a tiny 3 × 3 package. The features, low power, and small package makes this DAC suitable for general-purpose applications in a wide range of end equipment. Some of the most common applications for this devices are *set-and-forget* LED biasing in automotive applications and mobile projectors, power-supply control, general-purpose function generation, medical alarm generation, programmable comparator applications (such as smoke detectors, standalone PWM control loops, and offset and gain trimming in precision circuits).

9.2 Typical Applications

This section explains the design details of three primary applications of DACx3401: programmable LED biasing, power-supply control. and medical alarm generation.

9.2.1 Programmable LED Biasing

LED and laser biasing or driving circuits often require accuracy and stability of the luminosity with respect to variation in temperature, electrical conditions, and physical characteristics. This accuracy and stability are most effectively achieved using a precision DAC, such as the DACx3401. The DACx3401 has additional features, such as the V_{FB} pin that compensates for the gate-to-source voltage of the transistor (V_{GS}) drop and the drift of the MOSFET. The NVM allows the microprocessor to *set-and-forget* the LED biasing value, even during a power cycle. Figure 18 shows the circuit diagram for LED biasing.

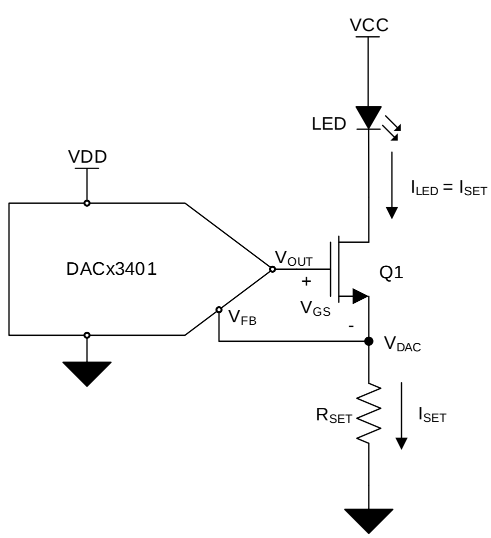


Figure 18. LED Biasing

9.2.1.1 Design Requirements

- DAC output range: 0 V to 2.4 V
- LED current range: 0 mA to 20 mA

ADVANCE INFORMATION

Typical Applications (continued)

9.2.1.2 Detailed Design Procedure

The DAC is used to set the source current of a MOSFET using the integrated unity-gain buffer, as shown in Figure 18. The LED is connected between the power supply and the drain of the MOSFET. This configuration allows the DAC to control or set the amount of current through the LED. The integrated buffer controls the gate-source voltage of the MOSFET inside the feedback loop, thus compensating this drop and corresponding drift due to temperature, current, and ageing of the MOSFET. The current set by the DAC that flows through the LED can be calculated with Equation 6. In order to generate 0 mA to 20 mA from a 0-V to 2.4-V DAC output range, a 120-Ω R_{SET} resistor is required. The internal reference can be selected with a span of 2x. Given a V_{GS} of 1.2 V, the VDD of the DAC must be at least be 3.6 V. Select a VDD of 5 V to allow variation of V_{GS} across temperature. When the VDD headroom is a constraint, use a bipolar junction transistor (BJT) in place of the MOSFET. BJTs have much less V_{BE} drop as compared to a V_{GS} of a MOSFET. A MOSFET provides a much better match between the current through the set register and the LED current, as compared to a BJT.

$$I_{SET} = \frac{V_{DAC}}{R_{SET}} \tag{6}$$

The pseudocode for getting started with an LED biasing application is as follows:

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Power-up the device, enable internal reference with 2x output span
WRITE GENERAL_CONFIG(0xD1), 0x11, 0xE5
//Write DAC code (12-bit aligned)
WRITE DAC_DATA(0x21), 0x07, 0xFC
//Write settings to the NVM
WRITE TRIGGER(0xD3), 0x00, 0x10
```

9.2.2 Power-Supply Control

A power-supply control circuit is used for testing and tuning the output of a power converter. This testing and tuning is done either to test the accuracy of a power supply, adjust the offset and drift of the power-supply output, or to program a desired value at the output. Adjustable power supplies, such as LDOs and DC/DC converters provide a feedback or adjust input that is used to set the desired output. A precision voltage-output DAC is a good choice for controlling the power-supply output linearly. Figure 19 shows a control circuit for a switch-mode power supply (SMPS) using DACx3401. Typical applications of power-supply control are test and measurement, communications equipment, and general-purpose power-supply modules.

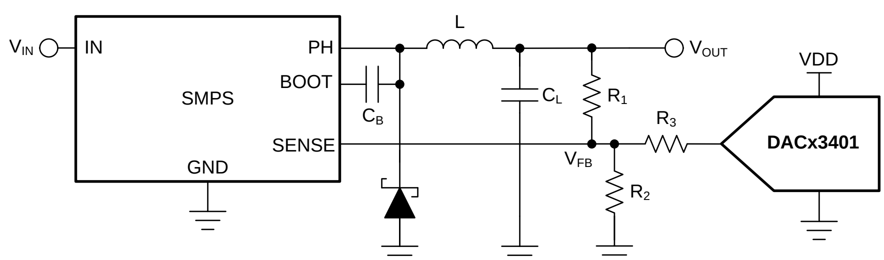


Figure 19. Power-Supply Control

9.2.2.1 Design Requirements

- Power supply nominal output: 3.3 V
- Reference voltage of the converter (V_{FB}): 0.6 V
- Margin: ±10% (that is, 2.97 V to 3.63 V)
- DAC output range: 1.8 V
- Nominal current through R_1 and R_2: 100 µA

ADVANCE INFORMATION

Typical Applications (continued)

9.2.2.2 Detailed Design Procedure

The DACx3401 features a Hi-Z power-down mode that is set by default at power-up, unless the device is programmed otherwise using the NVM. When the DAC output is at Hi-Z, the current through R_3 is zero and the SMPS is set at the nominal output voltage of 3.3 V. To have the same nominal condition when the DAC powers up, make sure to power up the device at the same output as V_{FB} (that is 0.6 V). This configuration makes sure there is no current through R_3 even at power-up. To achieve ±10% margin-high and margin-low conditions, the DAC must draw ±10% of the nominal current (that is, ±10 µA). In order to calculate the value of R_3, first decide the DAC output range, making sure to avoid the codes near zero scale and full-scale for safe operation in the linear region. A DAC output of 20 mV can be safely considered as the minimum output and (1.8 V – 0.6 V – 20 mV = 1.18 V) can be considered as the maximum output. When the DAC output is at 20 mV, the power supply goes to margin high, and when the DAC output is at 1.18 V, the power supply goes to margin low. The value of R_3 can be calculated by Equation 7. For the previous values, R_3 = 58 kΩ.

$$R_3 = \frac{|V_{DAC} - V_{FB}|}{10\ \mu A} \tag{7}$$

The DACx3401 have a slew rate feature that is used to toggle between margin high, margin low, and nominal outputs with a defined slew rate. See the GENERAL_CONFIG register description for the slew rate setting details.

The pseudocode for getting started with a power-supply control application is as follows:

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Power-up the device, enable internal reference with 1x output span
//CODE_STEP: 1-bit, SLEW_RATE: 12µs
WRITE GENERAL_CONFIG(0xD1), 0x11, 0x84
//Write DAC code (12-bit aligned) for nominal output
//For a 1.8V output range, the 10-bit Hex code for 20mV is 0x0B. With 12-bit alignment, it becomes 0x2C
WRITE DAC_DATA(0x21), 0x0F, 0xFC
//Write DAC code (12-bit aligned) for margin-high output
WRITE DAC_MARGIN_HIGH(0x25), 0x0A, 0x7C
//Write DAC code (12-bit aligned) for margin-low output
WRITE DAC_MARGIN_LOW(0x26), 0x00, 0x2C
//Trigger margin-high output
WRITE TRIGGER(0xD3), 0x00, 0x80
//Trigger margin-low output
WRITE TRIGGER(0xD3), 0x00, 0x40
//Write back DAC code (12-bit aligned) for nominal output
WRITE DAC_DATA(0x21), 0x0F, 0xFC
```

Typical Applications (continued)

9.2.3 Medical Alarm Generation

All medical devices implementing an alarm system shall comply to IEC60601-1-8 standard for medical alarms (as per IEC60601-1 Ed 3.1). The regulatory tests are done at a system level; therefore, system level acoustics play a major role in the compliance. A medical alarm is a common functional block in many medical devices. A portable implementation is needed that can also be customized to fit mechanical and audio or acoustic requirements. The DACx3401-based design is aimed at providing a programmable, standalone, and robust implementation at a very low cost.

There are three types of alarms with different timing requirements: low priority, medium priority, and high priority. Usually, for easy identification, different timings are employed for different equipment. Medical device manufacturers prefer using their signature melodies within the limits of the standard.

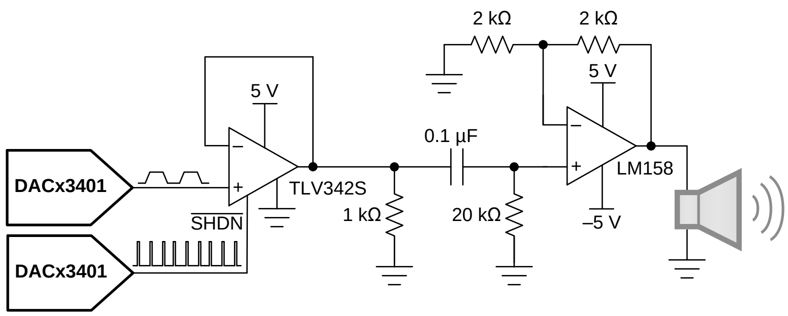


Figure 20. Medical Alarm

9.2.3.1 Design Requirements

- Generate pulse envelope for each of the three alarm priorities, upon trigger
- Generate the programmable pulse frequency
- Mix the envelope and the pulse to create the burst pattern
- Drive a speaker with the selected alarm tone

9.2.3.2 Detailed Design Procedure

Two DACx3401 devices are required: one device to generate the pulse envelope and the burst, and the second device to generate the pulse frequency. As shown in Figure 20, both these signals are mixed together using the TLV342S amplifier with shutdown. The combined signal is then fed to a power amplifier, such as the LM158, to drive the speaker. This design provides a gain of 2 at the speaker amplifier. The actual gain required in a system depends on the acoustic output requirements from the speaker. The RC high-pass filter, designed for a cut-off frequency of approximately 80 Hz at the input of LM158, removes the dc component from the signal so that this signal can be applied to the speaker directly. As per the medical alarm standard, the pulse frequency must be above 150 Hz. As a result of the square-wave pulse frequency and the mixing done by TLV342S, the speaker output has multiple harmonics of the fundamental pulse frequency, thus fulfilling the requirement of the medical alarm standard. The DACx3401 provide various options to program the pulse frequency and envelope timings. See the *Medical Alarm Generation Mode* section for the alarm configuration options. The frequency of a square wave can be calculated using Equation 3. The square wave function has a limited number of frequencies because this function is programmed by the SLEW_RATE bit alone. To get a higher number of frequencies, generate a triangular waveform with comparator mode output. The triangular waveform can be generated using Equation 4. The DAC output can be set to comparator mode by fixing the V_{FB} pin to the midscale of the DAC using a resistive voltage divider from V_{DD}. Select V_{DD} as the reference in this case using the GENERAL_CONFIG register.

ADVANCE INFORMATION

Typical Applications (continued)

The pseudocode for getting started with a medical alarm application using two DACs is as follows:

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Power-up the first DAC, enable VDD reference
//SLEW_RATE: 1.6384 ms (Square wave frequency: 610 Hz)
WRITE GENERAL_CONFIG(0xD1), 0xD1, 0x58
//Set MARGIN_HIGH on the first DAC
WRITE DAC_MARGIN_HIGH(0x25), 0x0F, 0xFC
//Set MARGIN_LOW on the first DAC
WRITE DAC_MARGIN_LOW(0x26), 0x00, 0x00
//Trigger square wave generation on the first DAC
WRITE TRIGGER(0xD3), 0x01, 0x00
//Power-up the second DAC, enable VDD reference
//CODE_STEP: 8 LSB, SLEW_RATE: 204.8 us x 1.75 = 358.4 us (Envelope rise/fall times for full-
scale: ~26 ms)
WRITE GENERAL_CONFIG(0xD1), 0x1A, 0xE8
//OPTION-1: Configure the second DAC for low-priority alarm with minimum time settings and trigger
WRITE MED_ALARM_CONFIG(0xD2), 0x01, 0x00
//OPTION-2: Configure the second DAC for medium-priority alarm with minimum time settings and trigger
WRITE MED_ALARM_CONFIG(0xD2), 0x02, 0x00
//OPTION-3: Configure the second DAC for high-priority alarm with minimum time settings and trigger
WRITE MED_ALARM_CONFIG(0xD2), 0x04, 0x00
//Set MARGIN_HIGH on the second DAC
WRITE DAC_MARGIN_HIGH(0x25), 0x0F, 0xFC
//Set MARGIN_LOW on the second DAC
WRITE DAC_MARGIN_LOW(0x26), 0x00, 0x00
```

ADVANCE INFORMATION

10 Power Supply Recommendations

The DACx3401 family of devices does not require specific supply sequencing. These devices require a single power supply, V_{DD}. Use a 0.1-µF decoupling capacitor for the V_{DD} pin. Use a bypass capacitor with a value greater than 1.5-µF for the CAP pin.

11 Layout

11.1 Layout Guidelines

The DACx3401 pin configuration separates the analog, digital, and power pins for an optimized layout. For signal integrity, separate the digital and analog traces, and place decoupling capacitors close to the device pins.

11.2 Layout Example

Figure 21 shows an example layout drawing with decoupling capacitors and pullup resistors.

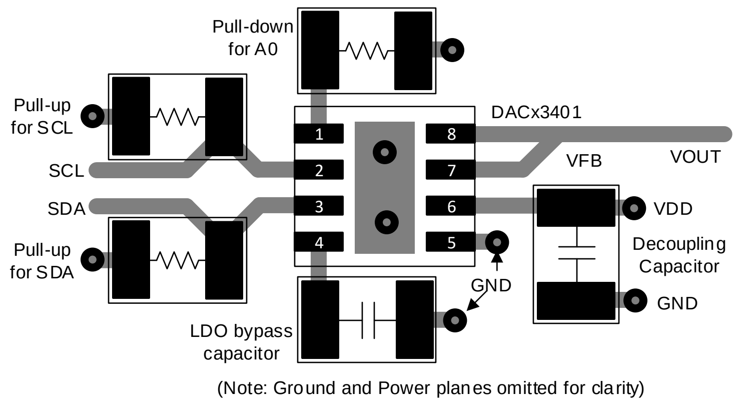


Figure 21. Layout Example

ADVANCE INFORMATION

12 Device and Documentation Support

12.1 Documentation Support

12.1.1 Related Documentation

For related documentation see the following:

Texas, Instruments *DAC53401EVM* user's guide

12.2 Related Links

Table 28 lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to order now.

Table 28. Related Links

PARTS	PRODUCT FOLDER	ORDER NOW	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
DAC53401	Click here	Click here	Click here	Click here	Click here
DAC43401	Click here	Click here	Click here	Click here	Click here

12.3 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. In the upper right corner, click on *Alert me* to register and receive a weekly digest of any product information that has changed. For change details, review the revision history included in any revised document.

12.4 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI E2E™ Online Community ***TI's Engineer-to-Engineer (E2E) Community.*** Created to foster collaboration among engineers. At e2e.ti.com, you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

Design Support ***TI's Design Support*** Quickly find helpful E2E forums along with design support tools and contact information for technical support.

12.5 Trademarks

E2E is a trademark of Texas Instruments.
PMBus is a trademark of SMIF, Inc.
All other trademarks are the property of their respective owners.

12.6 Electrostatic Discharge Caution

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage.

ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications.

12.7 Glossary

SLYZ022 — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

PACKAGING INFORMATION

Orderable Device	Status (1)	Package Type	Package Drawing	Pins	Package Qty	Eco Plan (2)	Lead/Ball Finish (6)	MSL Peak Temp (3)	Op Temp (°C)	Device Marking (4/5)	Samples
DAC43401DSGR	PREVIEW	WSON	DSG	8	3000	TBD	Call TI	Call TI	-40 to 125		
DAC43401DSGT	PREVIEW	WSON	DSG	8	250	TBD	Call TI	Call TI	-40 to 125		
PDAC53401DSGT	ACTIVE	WSON	DSG	8	250	TBD	Call TI	Call TI	-40 to 125		Samples

(1) The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

(2) **RoHS:** TI defines "RoHS" to mean semiconductor products that are compliant with the current EU RoHS requirements for all 10 RoHS substances, including the requirement that RoHS substance do not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, "RoHS" products are suitable for use in specified lead-free processes. TI may reference these types of products as "Pb-Free".
RoHS Exempt: TI defines "RoHS Exempt" to mean products that contain lead but are compliant with EU RoHS pursuant to a specific EU RoHS exemption.
Green: TI defines "Green" to mean the content of Chlorine (Cl) and Bromine (Br) based flame retardants meet JS709B low halogen requirements of <=1000ppm threshold. Antimony trioxide based flame retardants must also meet the <=1000ppm threshold requirement.

(3) MSL, Peak Temp. - The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

(4) There may be additional marking, which relates to the logo, the lot trace code information, or the environmental category on the device.

(5) Multiple Device Markings will be inside parentheses. Only one Device Marking contained in parentheses and separated by a "~" will appear on a device. If a line is indented then it is a continuation of the previous line and the two combined represent the entire Device Marking for that device.

(6) Lead/Ball Finish - Orderable Devices may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead/Ball Finish values may wrap to two lines if the finish value exceeds the maximum column width.

Important Information and Disclaimer: The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

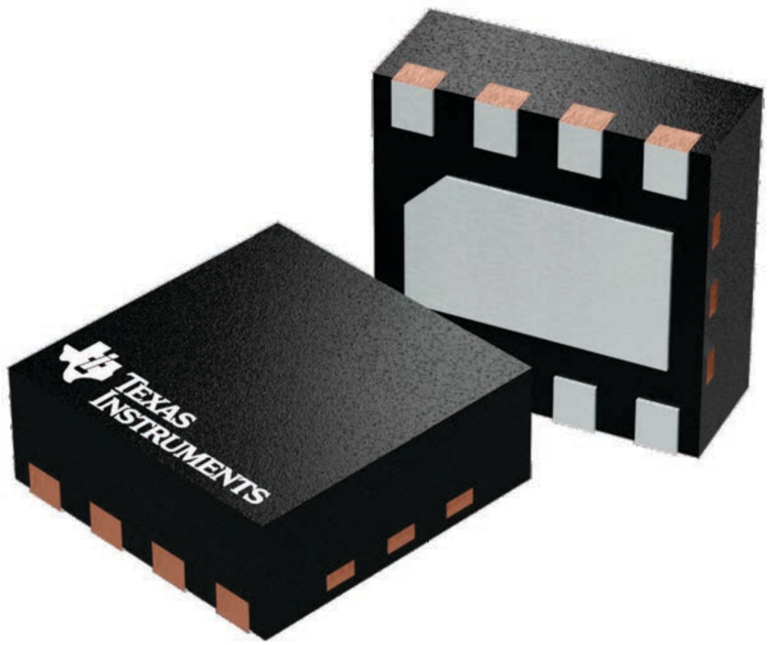
DSG 8

WSON - 0.8 mm max height

2 x 2, 0.5 mm pitch

PLASTIC SMALL OUTLINE - NO LEAD

This image is a representation of the package family, actual package may vary.
Refer to the product data sheet for package details.



IMPORTANT NOTICE AND DISCLAIMER

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265