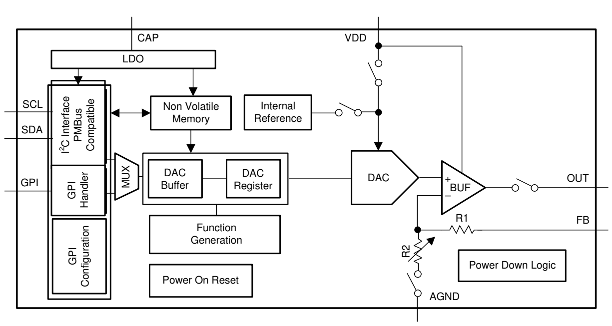
DACx3701 10-Bit and 8-Bit, Voltage-Output Smart DACs With Nonvolatile Memory and PMBus™ Compatible I^2C Interface With GPI Control

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
- General-purpose input (GPI) based function trigger
- PMBus™ compatible I^2C interface
  - Standard, fast, and fast mode plus
  - Four device address options configured using the broadcast address
  - 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, triangular, and sawtooth
- Pulse-width modulation (PWM) output using triangular waveform and FB pin
- Preprogrammed medical-alarm tone-generation mode: low-, medium-, and high-priority alarms
- Digital slew rate control
- Internal reference
- Very low power: 0.2 mA at 1.8 V
- Flexible start-up: High impedance or 10K-GND
- Tiny package: 8-pin WSON (2 mm × 2 mm)

2 Applications

- Oven
- Ventilators
- Infusion pump
- Anesthesia delivery systems
- Surgical equipment
- Rack server
- Exit and emergency lighting



Functional Block Diagram

3 Description

The 10-bit DAC53701 and 8-bit DAC43701 (DACx3701) are a pin-compatible family of buffered voltage-output smart 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 DACx3701 an excellent choice for applications such as appliance door fade-in fade-out, processorless LED dimming with PWM input, general-purpose bias point generation, voltage margining and scaling, PWM signal generation, and medical alarm tone generation.

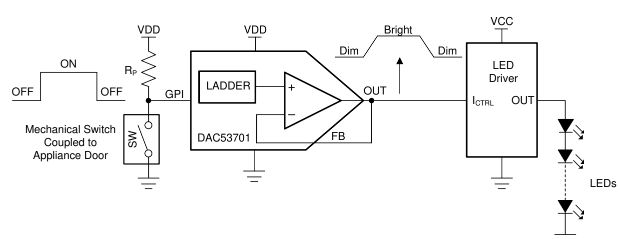
These devices have nonvolatile memory (NVM), an internal reference, a PMBus-compatible I^2C interface, and a general-purpose input. The DACx3701 operates with either an internal reference or with the power supply as a reference, and provides a full-scale output of 1.8 V to 5.5 V.

The DACx3701 are smart DAC devices because of their advanced integrated features. With force-sense output, GPI based function trigger, medical alarm, PWM output, and NVM capabilities, smart DACs enable system performance and control without the use of software.

Device Information

PART NUMBER	PACKAGE(1)	BODY SIZE (NOM)
DAC53701	WSON (8)	2.00 mm × 2.00 mm
DAC43701		

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



Appliance Light Fade-in Fade-out

Table of Contents

4 Revision History

DATE	REVISION	NOTES
December 2020	*	Initial release.

5 Device Comparison Table

DEVICE	RESOLUTION
DAC53701	10-bit
DAC43701	8-bit

6 Pin Configuration and Functions

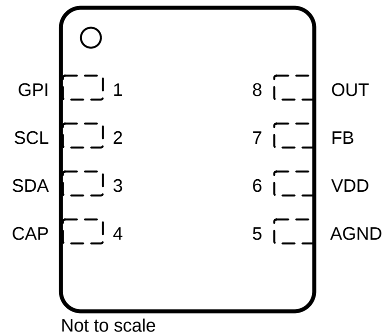


Figure 6-1. DSG Package, 8-Pin WSON, Top View

Table 6-1. Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
AGND	5	Ground	Ground reference point for all circuitry on the device
CAP	4	Input	External capacitor for the internal LDO. Connect a capacitor (approximately 1.5 µF) between CAP and AGND.
FB	7	Input	Voltage-feedback pin
GPI	1	Input	General-purpose input
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

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 except the power pins and the OUT 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	
			pins 2, 3, 6, 7[(2)]	±500	

(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)]		DACx3701 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	DAC53701	10			Bits
		DAC43701	8			
INL	Relative accuracy(1)		–1		1	LSB
DNL	Differential nonlinearity(1)		–1		1	LSB
	Zero-code error	Code 0d into DAC, external reference, V_{DD} = 5.5 V		6	12	mV
		Code 0d into DAC, internal reference, gain = 4x, V_{DD} = 5.5 V		6	15	
	Zero-code-error temperature coefficient			±10		µV/°C
	Offset error(4)		–0.5	0.25	0.5	%FSR
	Offset-error temperature coefficient(4)			±0.0003		%FSR/°C
	Gain error(4)		–0.5	0.25	0.5	%FSR
	Gain-error temperature coefficient(4)			±0.0008		%FSR/°C
	Full-scale error	1.8 V ≤ V_{DD} < 2.7 V, code 1023d into DAC for 10-bit resolution, code 255d into DAC for 8-bit resolution, no headroom	–1	0.5	1	%FSR
		2.7 V ≤ V_{DD} ≤ 5.5 V, code 1023d into DAC for 10-bit resolution, code 255d into DAC for 8-bit resolution, no headroom	–0.5	0.25	0.5	
	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		0.4		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(1) (2)	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 = 8d for 10-bit resolution and code = 2d for 8-bit resolution		0.25		
		DAC output enabled and DAC code = 1016d for 10-bit resolution and code = 254d for 8-bit resolution		0.26		

7.5 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_{FB} dc output impedance(3)	DAC output enabled, DAC reference tied to VDD (gain = 1x) or internal reference (gain = 1.5x or 2x)	160	200	240	kΩ
		DAC output enabled, internal V_{REF}, gain = 3x or 4x	192	240	288	
	V_{OUT} + V_{FB} dc output leakage(2)	At start up, measured when DAC output is disabled and held at V_{DD} / 2 for V_{DD} = 5.5 V			7	nA
	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		8		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		12		
	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		75		mV
		At startup (DAC output disabled), R_L = 100 kΩ		200		
	Output enable glitch magnitude	DAC output disabled to enabled (DAC registers at zero scale, R_L = 100 kΩ		250		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)(3)	Internal V_{REF}, gain = 4x, 200-mV 50-Hz 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)		15		mV
VOLTAGE REFERENCE						
	Initial accuracy	T_A = 25°C		1.212		V
	Reference output temperature coefficient(2)				65	ppm/°C
EEPROM						
	Endurance(2)	–40°C ≤ T_A ≤ +85°C		20000		Cycles
		T_A > 85°C		1000		
	Data retention(2)	T_A = 25°C		50		Years
		T_A = 125°C		20		
	EEPROM programming write cycle time(2)		10		20	ms
DIGITAL INPUTS						
	Digital feedthrough	DAC output static at midscale, fast mode plus, SCL toggling		20		nV-s
	Pin capacitance	Per pin		10		pF

7.5 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
POWER						
	Load capacitor - CAP pin[(2)]		0.5		15	µF
I_{DD}	Current flowing into VDD	Normal mode, DACs at full scale, digital pins static		0.225	0.55	mA
		DAC power-down, internal reference power down		80		µA

(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: 8d to 1016d for 10-bit resolution, 2d to 254d for 8-bit resolution.
(2) Specified by design and characterization, not production tested.
(3) Specified with 200-mV headroom with respect to reference value when internal reference is used.
(4) Measured with DAC output unloaded. For 10-bit resolution, between end-point codes: 8d to 1016d and for 8-bit resolution, between end-point codes: 2d to 254d.

7.6 Timing Requirements: I²C Standard Mode

all input signals are timed from VIL to 70% of V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V, –40°C ≤ T_A ≤ +125°C, and 1.8 V ≤ $V_{pull-up}$ ≤ V_{DD} 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	4000			ns
t_F	Clock and data fall time			300	ns
t_R	Clock and data rise time			1000	ns

7.7 Timing Requirements: I²C Fast Mode

all input signals are timed from VIL to 70% of V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V, –40°C ≤ T_A ≤ +125°C, and 1.8 V ≤ $V_{pull-up}$ ≤ V_{DD} 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 Plus

all input signals are timed from VIL to 70% of V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V, –40°C ≤ T_A ≤ +125°C, and 1.8 V ≤ $V_{pull-up}$ ≤ V_{DD} 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

7.9 Timing Requirements: GPI

all input signals are timed from VIL to 70% of V_{DD}. V_{DD} = 1.8 V to 5.5 V and T_A = –40°C to +125°C (unless otherwise noted)

		MIN	NOM	MAX	UNIT
$t_{GPIDELAY}$	GPI edge to start of operation delay, 1.7 V ≤ V_{DD} ≤ 5.5 V[(1)]		2		μs

(1) The value specified for $t_{GPIDELAY}$ in the timing table is in addition to 2x SLEW_RATE for margin-high, low and function generation operations. The typical value for the total delay is (2xSLEW_RATE + $t_{GPIDELAY}$).

7.10 Timing Diagram

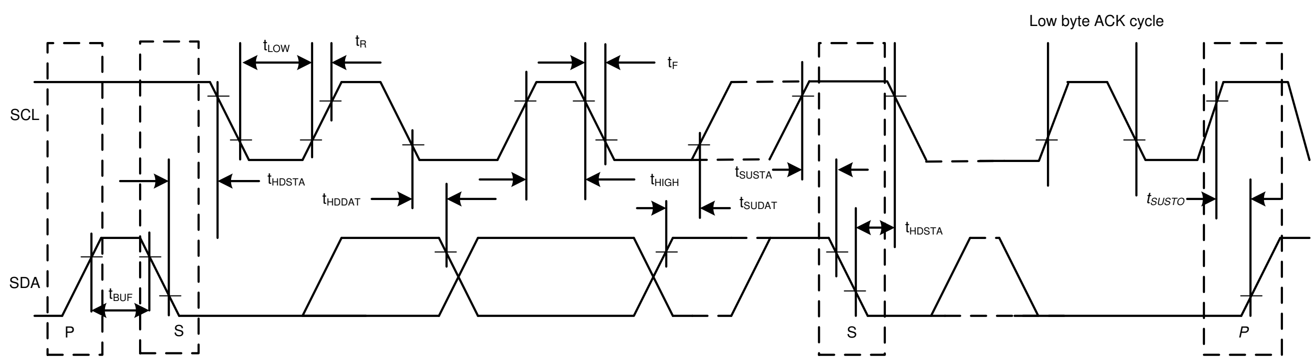


Figure 7-1. I²C Timing Diagram

7.11 Typical Characteristics: V_{DD} = 5.5 V (Reference = V_{DD}) or V_{DD} = 5 V (Internal Reference)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)

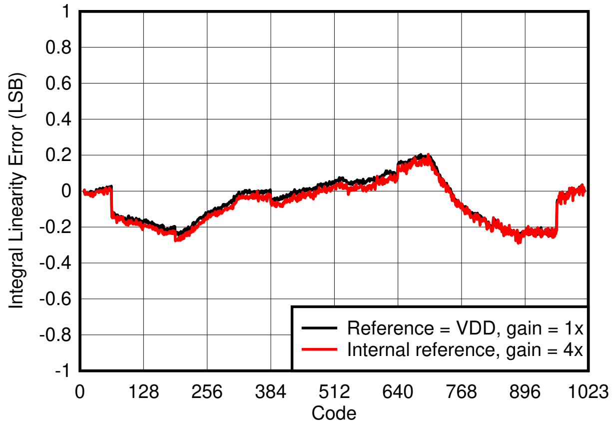


Figure 7-2. Integral Linearity Error vs Digital Input Code

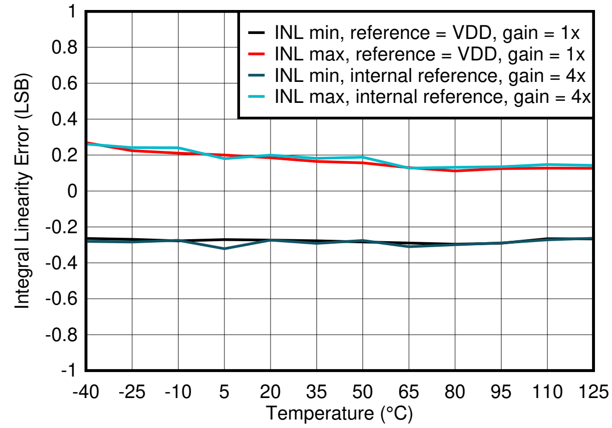


Figure 7-3. Integral Linearity Error vs Temperature

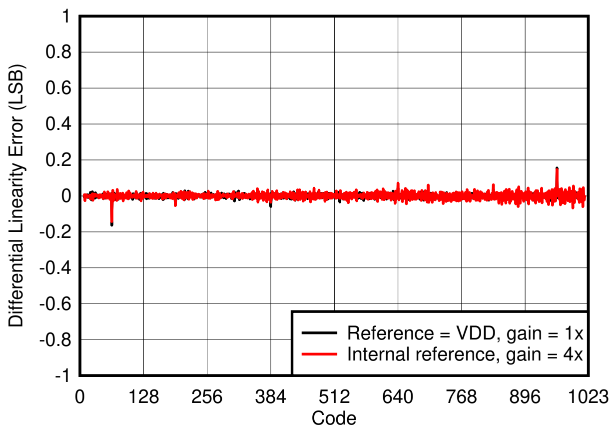


Figure 7-4. Differential Linearity Error vs Digital Input Code

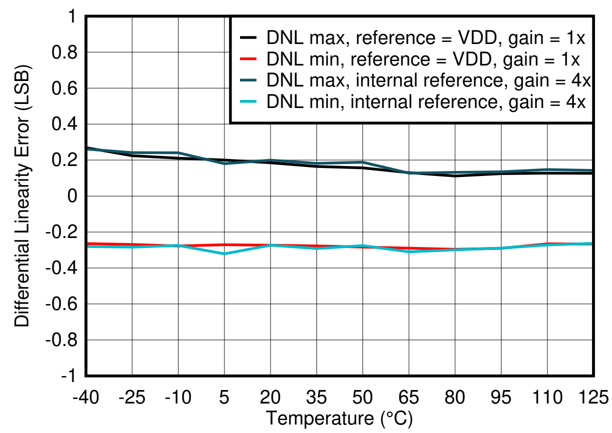


Figure 7-5. Differential Linearity Error vs Temperature

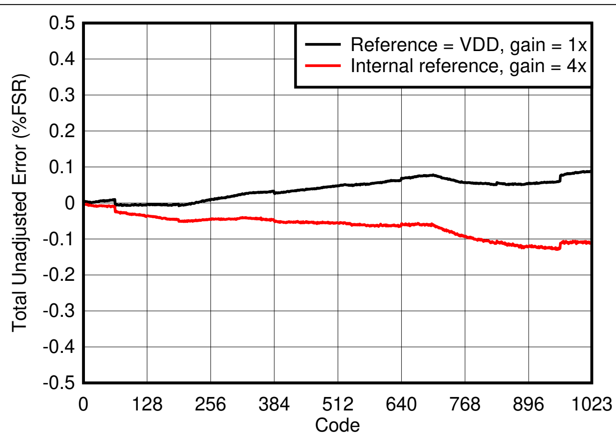


Figure 7-6. Total Unadjusted Error vs Digital Input Code

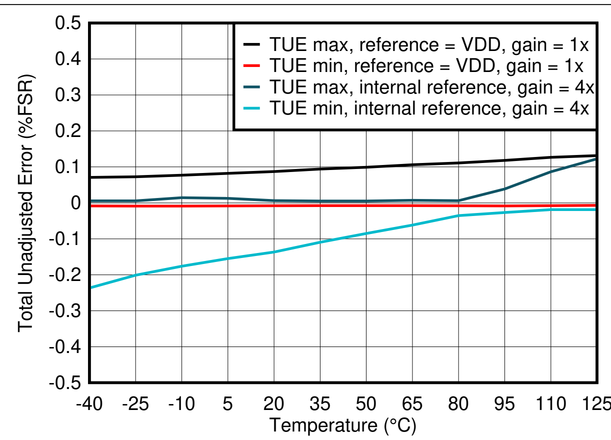
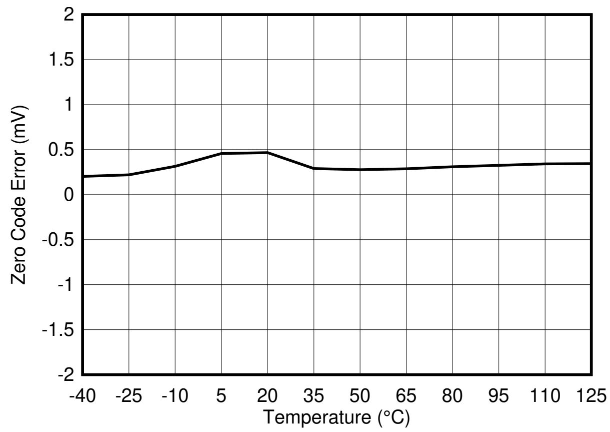


Figure 7-7. Total Unadjusted Error vs Temperature

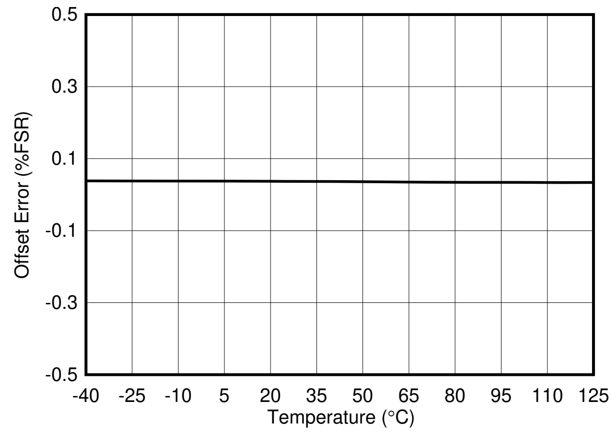
7.11 Typical Characteristics: V_{DD} = 5.5 V (Reference = V_{DD}) or V_{DD} = 5 V (Internal Reference) (continued)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



Reference = V_{DD}

Figure 7-8. Zero Code Error vs Temperature



Reference = V_{DD}

Figure 7-9. Offset Error vs Temperature

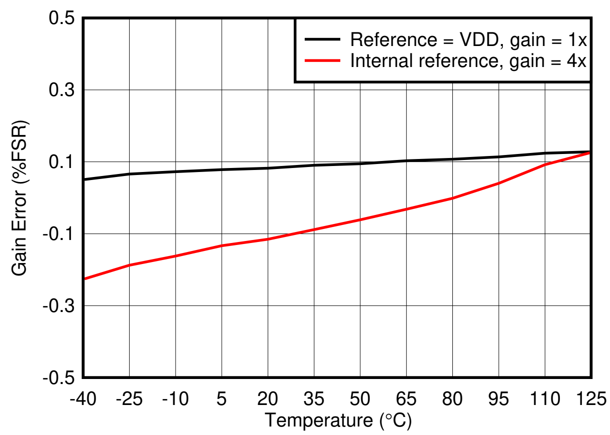


Figure 7-10. Gain Error vs Temperature

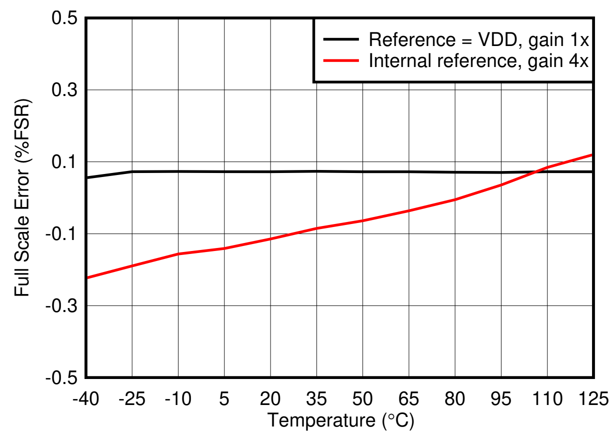


Figure 7-11. Full-Scale Error vs Temperature

7.12 Typical Characteristics: V_{DD} = 1.8 V (Reference = V_{DD}) or V_{DD} = 2 V (Internal Reference)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)

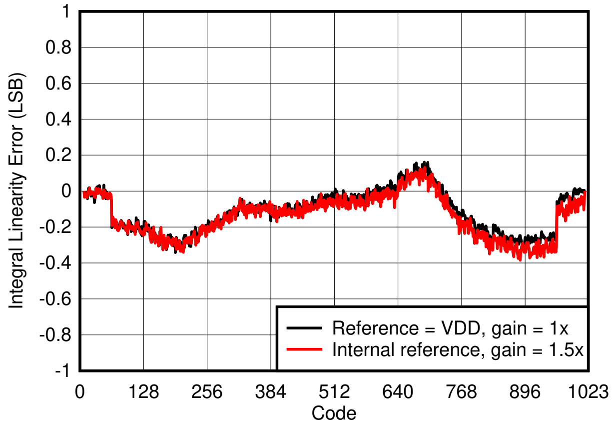


Figure 7-12. Integral Linearity Error vs Digital Input Code

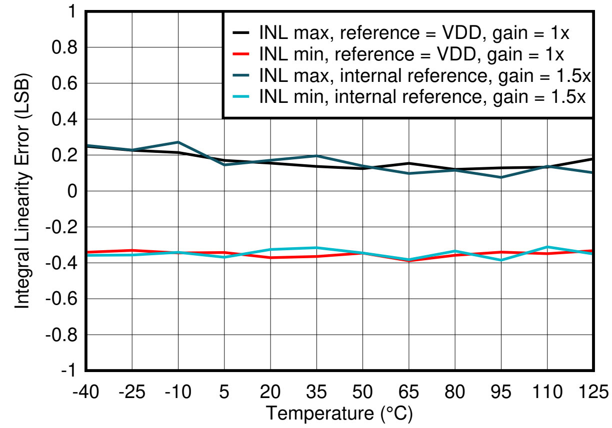


Figure 7-13. Integral Linearity Error vs Temperature

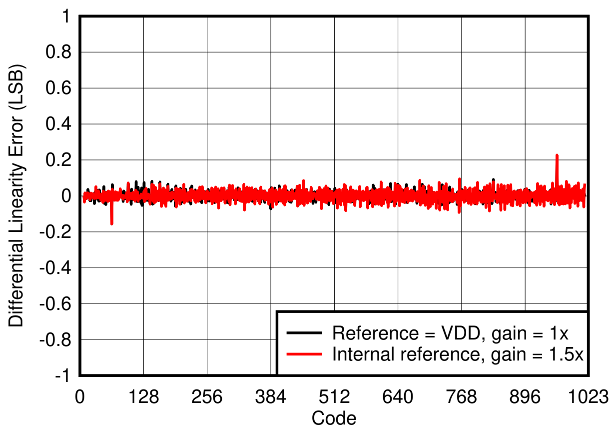


Figure 7-14. Differential Linearity Error vs Digital Input Code

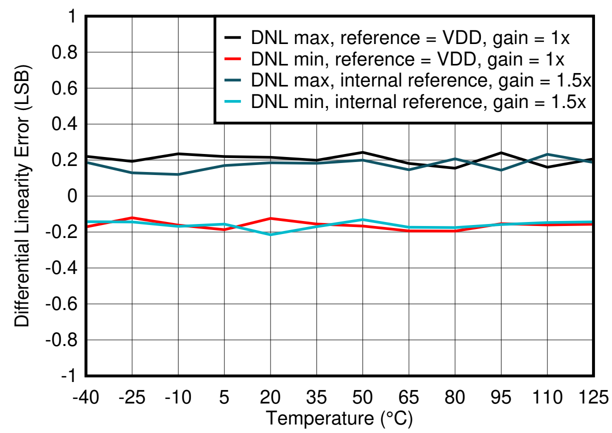


Figure 7-15. Differential Linearity Error vs Temperature

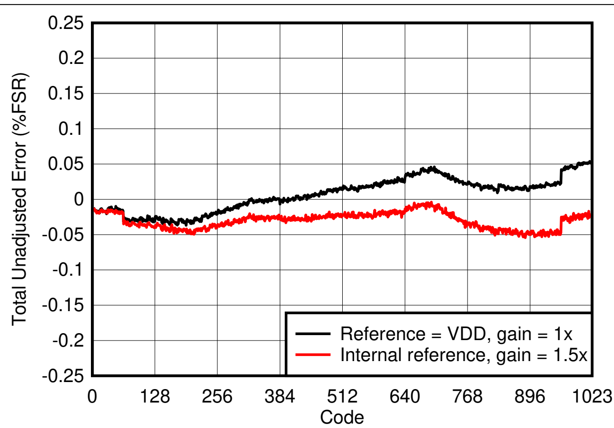


Figure 7-16. Total Unadjusted Error vs Digital Input Code

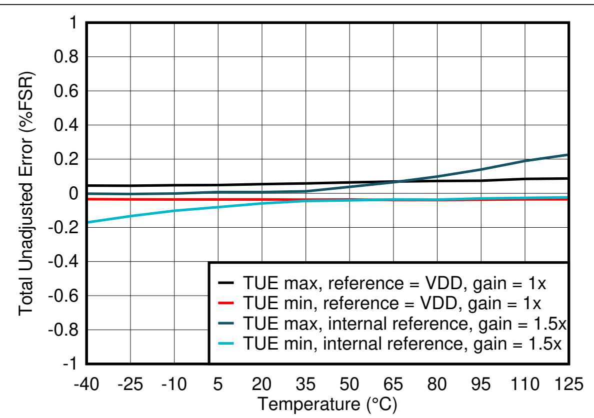
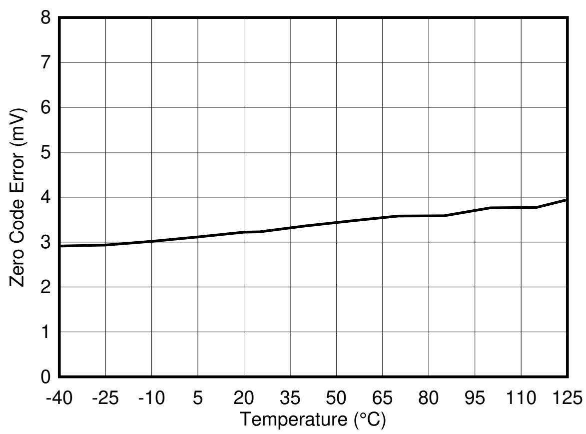


Figure 7-17. Total Unadjusted Error vs Temperature

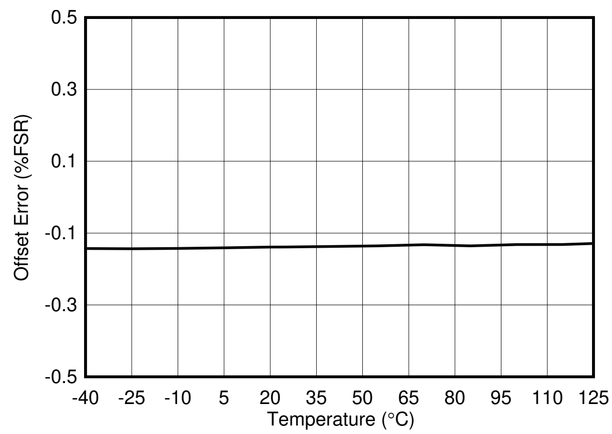
7.12 Typical Characteristics: V_{DD} = 1.8 V (Reference = V_{DD}) or V_{DD} = 2 V (Internal Reference) (continued)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



Reference = V_{DD}

Figure 7-18. Zero Code Error vs Temperature



Reference = V_{DD}

Figure 7-19. Offset Error vs Temperature

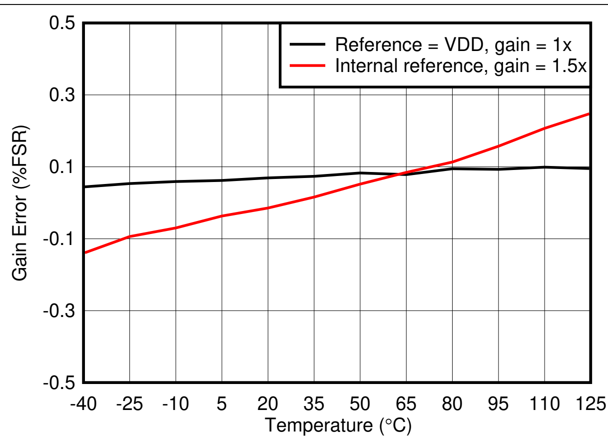


Figure 7-20. Gain Error vs Temperature

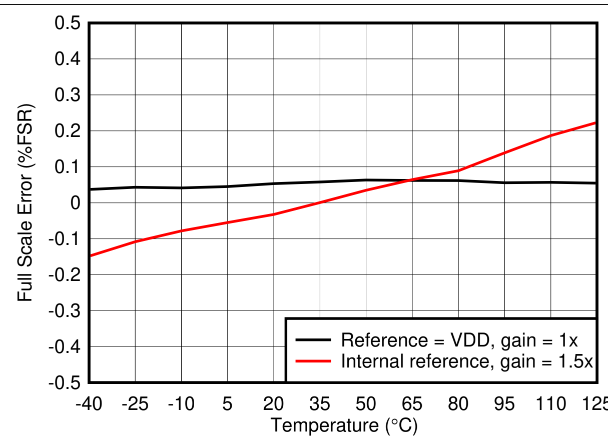
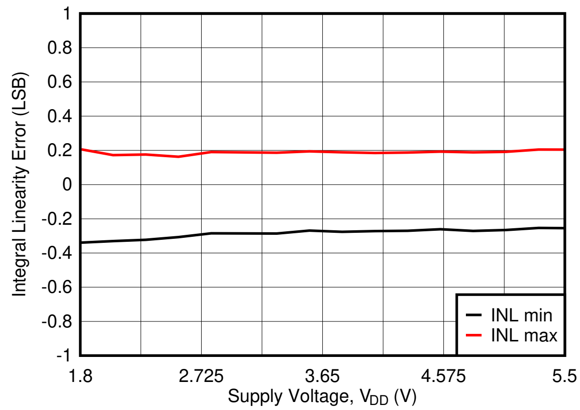


Figure 7-21. Full-Scale Error vs Temperature

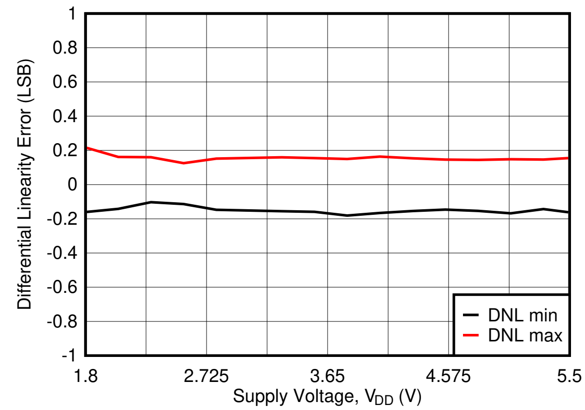
7.13 Typical Characteristics

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



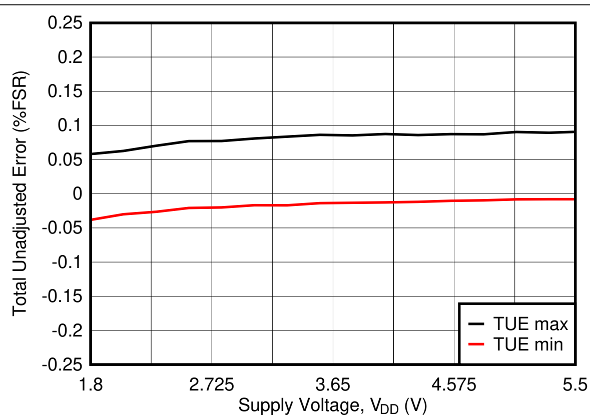
Reference = V_{DD}

Figure 7-22. Integral Linearity Error vs Supply Voltage



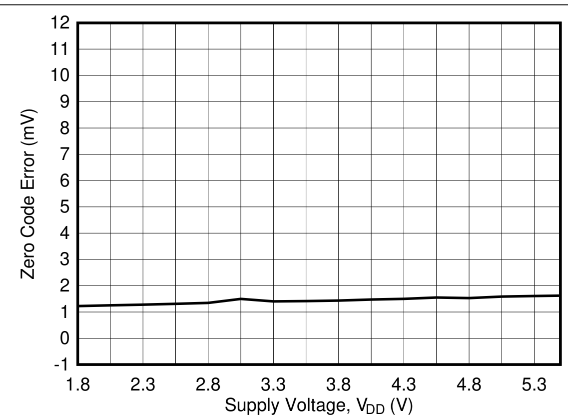
Reference = V_{DD}

Figure 7-23. Differential Linearity Error vs Supply Voltage



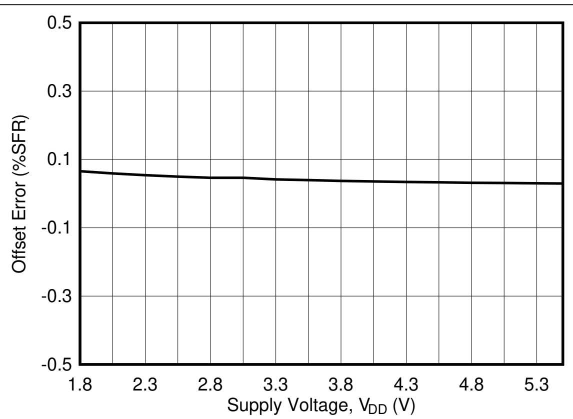
Reference = V_{DD}

Figure 7-24. Total Unadjusted Error vs Supply Voltage



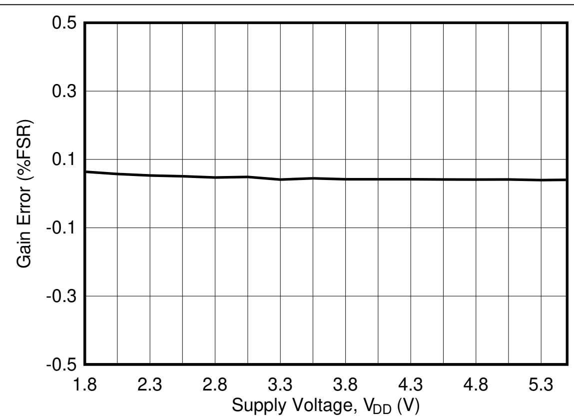
Reference = V_{DD}

Figure 7-25. Zero-Code Error vs Supply Voltage



Reference = V_{DD}

Figure 7-26. Offset Error vs Supply Voltage

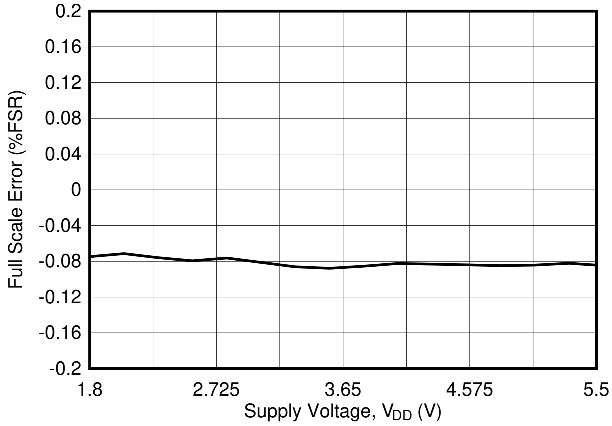


Reference = V_{DD}

Figure 7-27. Gain Error vs Supply Voltage

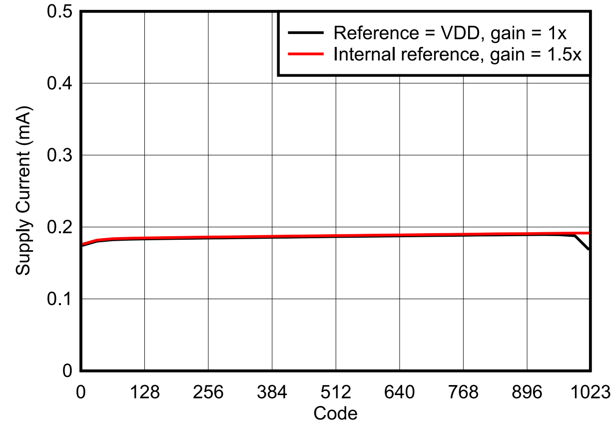
7.13 Typical Characteristics (continued)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



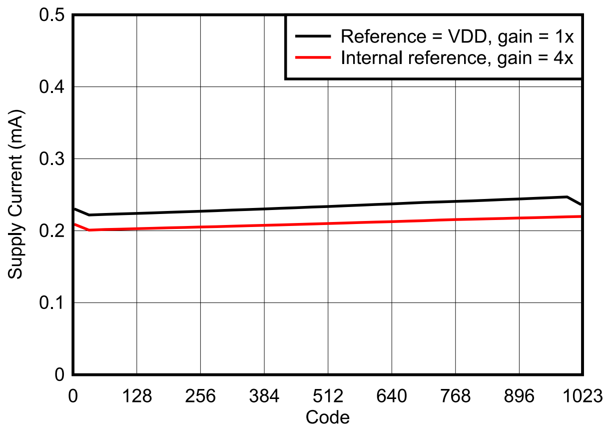
Reference = V_{DD}

Figure 7-28. Full-Scale Error vs Supply Voltage



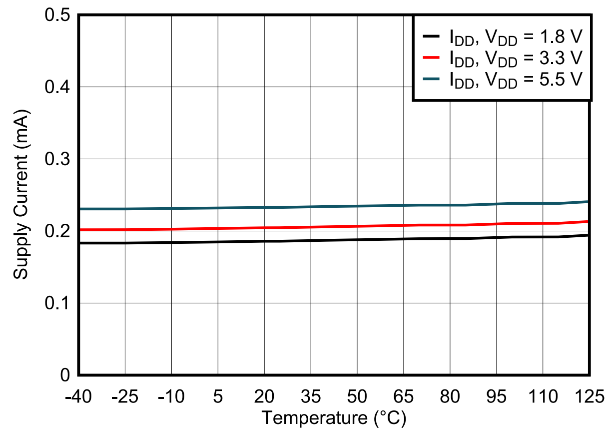
V_{DD} = 1.8 V

Figure 7-29. Supply Current vs Digital Input Code



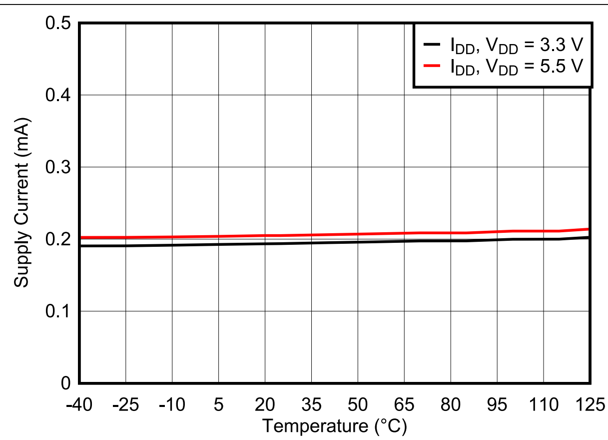
V_{DD} = 5.5 V

Figure 7-30. Supply Current vs Digital Input Code



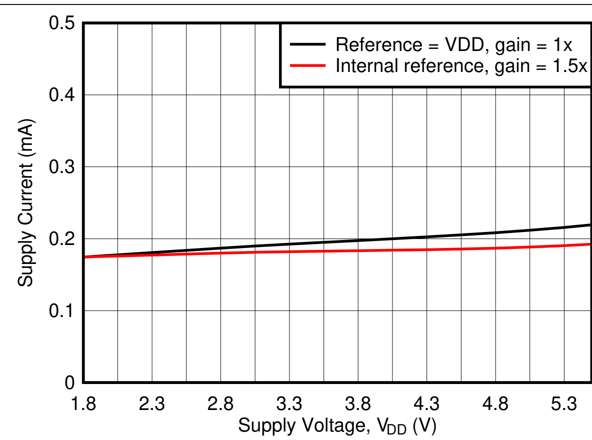
Reference = V_{DD}, DAC at midscale

Figure 7-31. Supply Current vs Temperature



Internal reference (gain = 4x), DAC at midscale

Figure 7-32. Supply Current vs Temperature

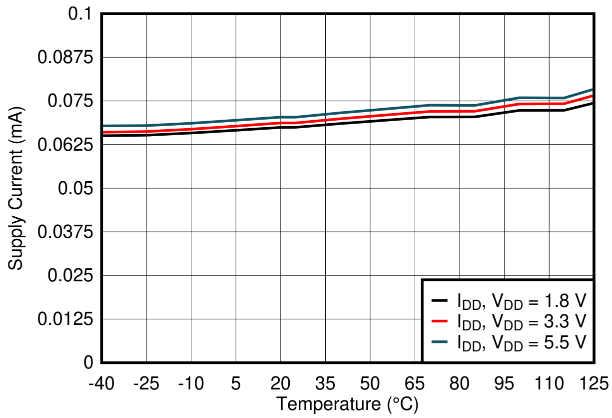


DAC at midscale

Figure 7-33. Supply Current vs Supply Voltage

7.13 Typical Characteristics (continued)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



Reference = V_{DD}, DAC powered down

Figure 7-34. Power-Down Current vs Temperature

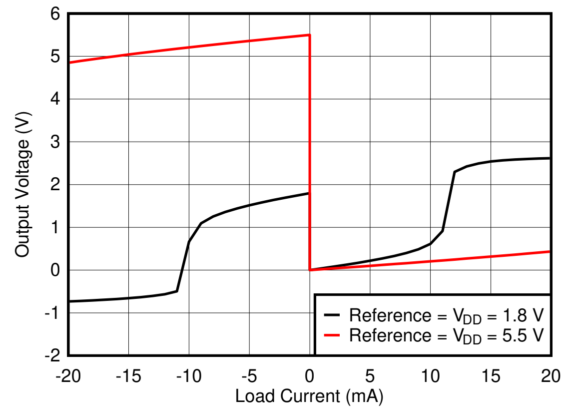
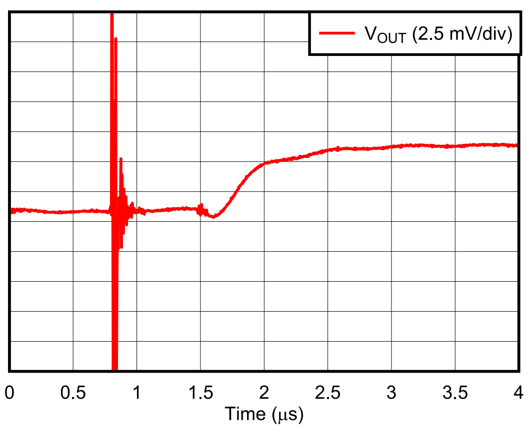
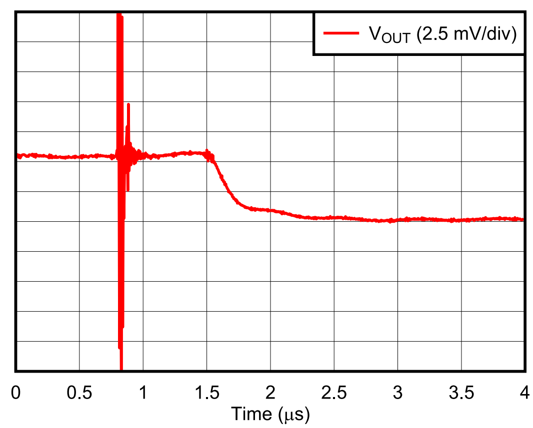


Figure 7-35. Source and Sink Capability



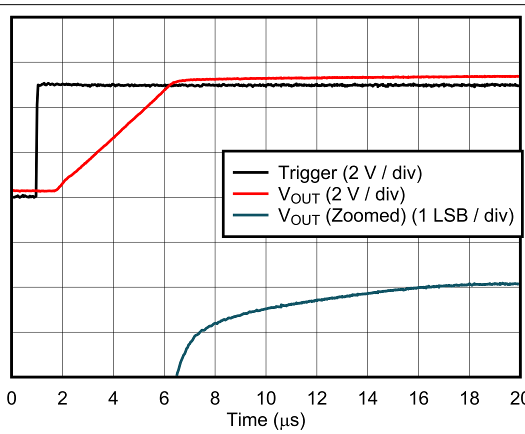
Reference = V_{DD} = 5.5 V, DAC code transition from midscale to midscale + 1 LSB, DAC load = 5kΩ || 200pF

Figure 7-36. Glitch Impulse, Rising Edge, 1-LSB Step



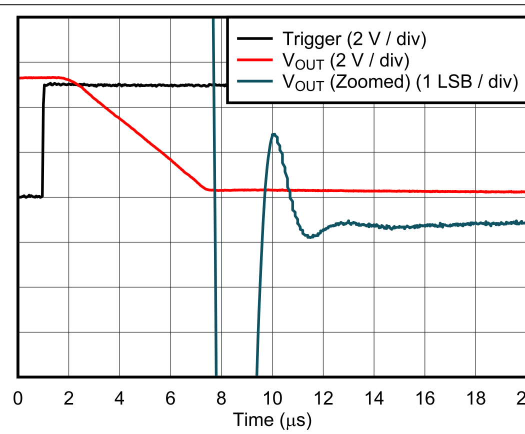
Reference = V_{DD} = 5.5 V, DAC code transition from midscale to midscale – 1 LSB, DAC load = 5kΩ || 200pF

Figure 7-37. Glitch Impulse, Falling Edge, 1-LSB Step



Reference = V_{DD} = 5.5 V, DAC load = 5kΩ || 200pF

Figure 7-38. Full-Scale Settling Time, Rising Edge

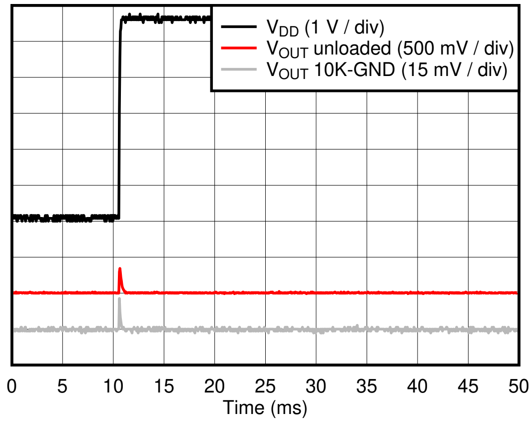


Reference = V_{DD} = 5.5 V, DAC load = 5kΩ || 200pF

Figure 7-39. Full-Scale Settling Time, Falling Edge

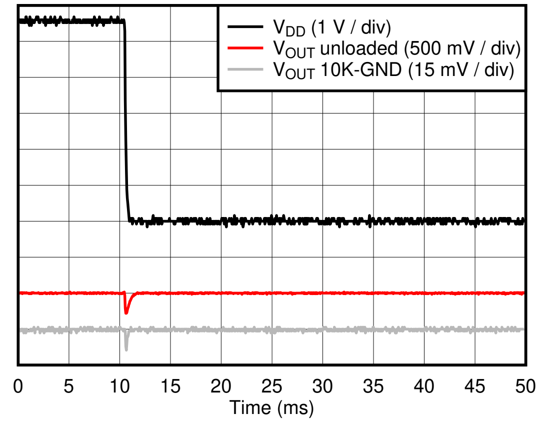
7.13 Typical Characteristics (continued)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



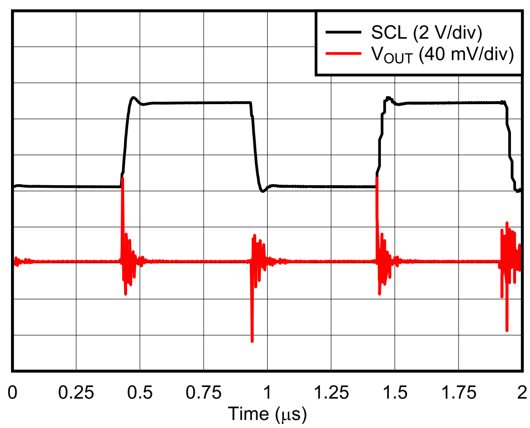
Reference = V_{DD} = 5.5 V

Figure 7-40. Power-on Glitch



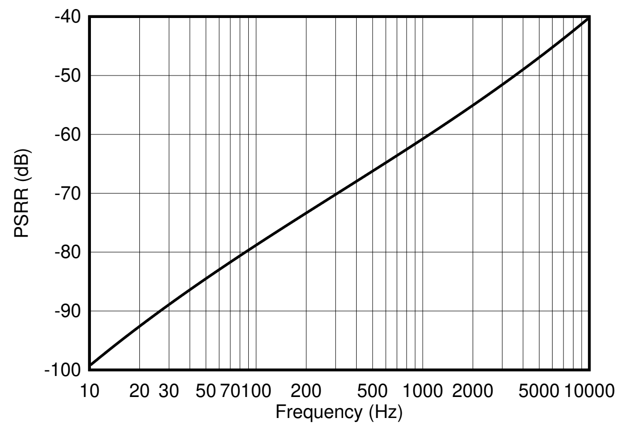
Reference = V_{DD} = 5.5 V

Figure 7-41. Power-off Glitch



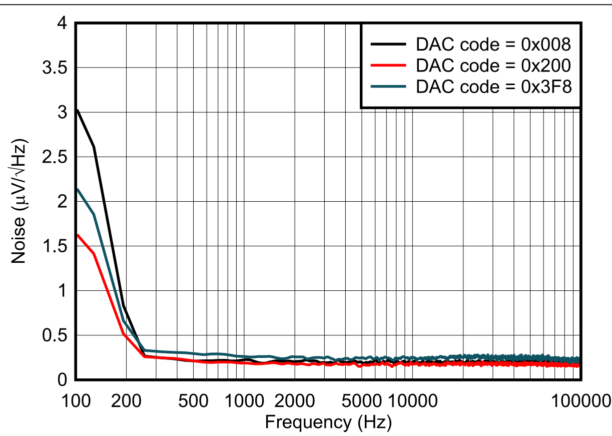
Reference = V_{DD} = 5.5 V, Fast+ mode, DAC at midscale, DAC load = 5kΩ || 200pF

Figure 7-42. Clock Feedthrough



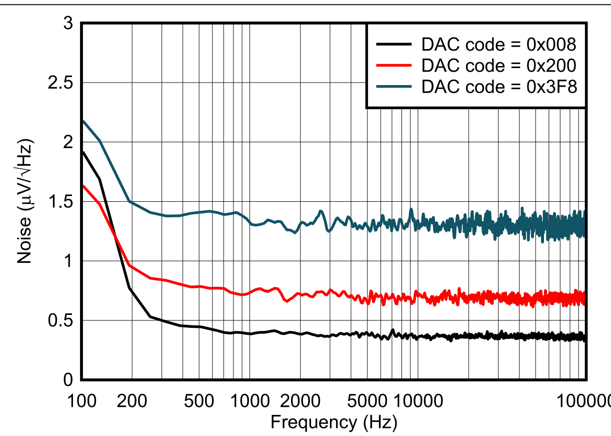
Internal reference (gain = 4x), V_{DD} = 5.25 V + 0.25 V_{PP}, DAC at midscale, DAC load = 5kΩ || 200pF

Figure 7-43. DAC Output AC PSRR vs Frequency



Reference = V_{DD} = 5.5 V

Figure 7-44. DAC Output Noise Spectral Density

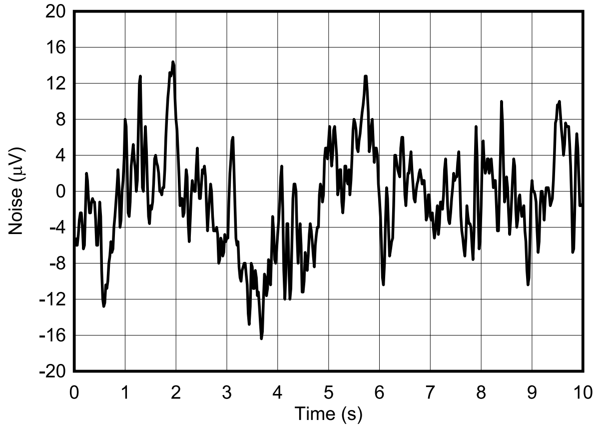


Internal reference (gain = 4x), V_{DD} = 5.5 V

Figure 7-45. DAC Output Noise Spectral Density

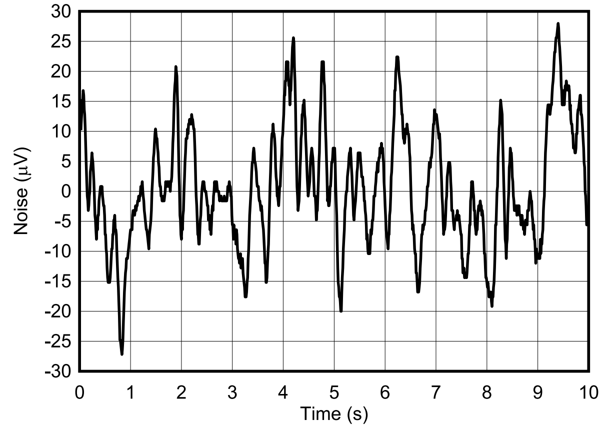
7.13 Typical Characteristics (continued)

at T_A = 25°C, 10-bit DAC, and DAC outputs unloaded (unless otherwise noted)



Reference = V_{DD} = 5.5 V, DAC at midscale

Figure 7-46. DAC Output Noise: 0.1 Hz to 10 Hz



Internal reference (gain = 4x), V_{DD} = 5.5 V, DAC at midscale

Figure 7-47. DAC Output Noise: 0.1 Hz to 10 Hz

8 Detailed Description

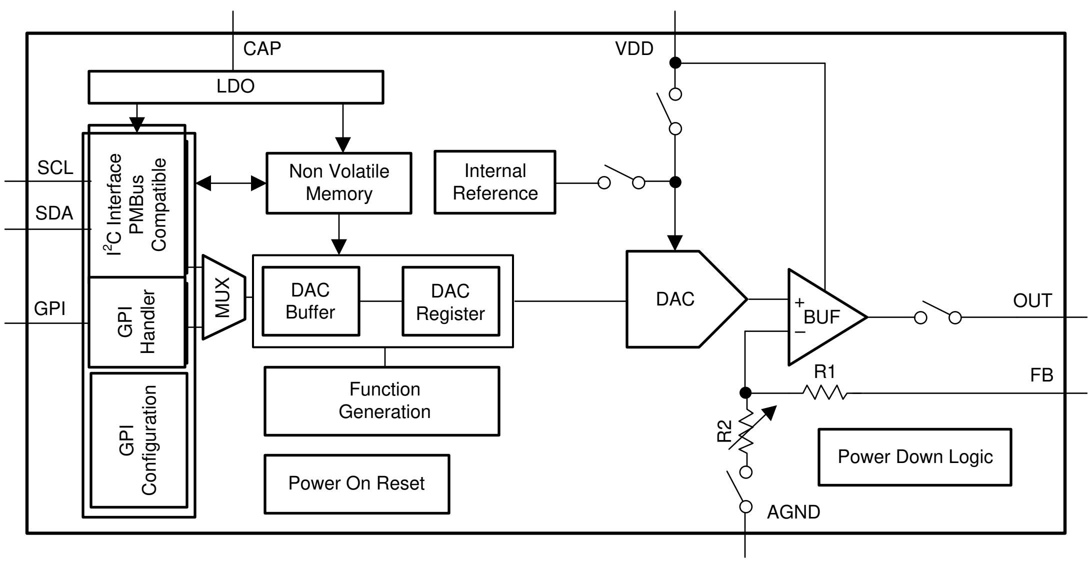
8.1 Overview

The 10-bit DAC53701 and 8-bit DAC43701 (DACx3701) are a pin-compatible family of buffered voltage-output, smart digital-to-analog converters (DACs). These smart DACs contain nonvolatile memory (NVM), an internal reference, a PMBus-compatible I^{2}C interface, force-sense output, and a general-purpose input. The DACx3701 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.

These devices communicate through an I^{2}C interface, and support I^{2}C standard mode (100 kbps), fast mode (400 kbps), and fast mode plus (1 Mbps). These devices also support specific PMBus commands such as *turn on/off*, *margin high or low*, and more. The GPI input can be configured as a power-down trigger, margin-high-low, function trigger, and medical alarm trigger. The DACx3701 also include digital slew rate control, and support basic signal generation such as *square*, *ramp*, and *sawtooth* waveforms. These devices can generate pulse-width modulation (PWM) output with the combination of the triangular or sawtooth waveform and the FB pin. These features enable the DACx3701 to go beyond the limitations of a conventional DAC that depends on a processor to function. Because of processor-less operation and the *smart* feature set, the DACx3701 are called smart DACs.

The DACx3701 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



8.3 Feature Description

8.3.1 Digital-to-Analog Converter (DAC) Architecture

The DACx3701 family of devices consists of string architecture with an output buffer amplifier. *Section 8.2* 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 0.2 mA of current when using a 1.8-V power supply. The DAC output pin starts up in high-impedance mode, making these devices 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. The DACx3701 devices include a *smart* feature set to enable processor-less operation and high-integration. The NVM enables a predictable startup. The GPI triggers the DAC output without the I^2C interface in the absence of a processor or when the processor or software fails. The integrated functions and the FB pin enable PWM output for control applications. The FB pin enables this device to be used as a programmable comparator. The digital slew rate control and the Hi-Z power-down modes enable a hassle-free voltage margining and function.

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 DACx3701 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. The gain at the output stage is always 1x.

$$V_{OUT} = \frac{DAC\_DATA}{2^N} \times V_{DD} \tag{1}$$

where:

- N is the resolution in bits, either 8 (DAC43701) or 10 (DAC53701).
- 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.

8.3.1.1.2 Internal Reference

The DACx3701 also contain 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 1.21-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.

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

where:

- N is the resolution in bits, either 8 (DAC43701) or 10 (DAC53701).
- 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 = 1.21 V.
- GAIN = 1.5x, 2x, 3x, 4x, based on DAC_SPAN (address D1h) bits.

8.3.2 General-Purpose Input (GPI)

The GPI pin of DACx3701 enables processorless operation. The GPI pin can be configured to trigger various functions, as shown in Table 8-1. The GPI_EN bit in the TRIGGER (Section 8.6.4) register enables or disables the GPI input. The GPI_CONFIG field in the CONFIG2 (Section 8.6.3) register maps the GPI pin to various functions. The GPI operations are edge-triggered once the device boots up. Once the power supply ramps up, the device registers the GPI level and executes the associated command. This feature allows the user to configure the initial output state at power-on. By default, the GPI pin is not mapped to any operation. Pull the GPI pin to high or low when not used. When the GPI pin is mapped to a specific function, the corresponding software bit functionality is disabled to avoid a race condition. When the GPI is mapped to margin-high or low trigger function, the output changes dynamically, unlike the behavior with I^{2}C-based programming. This behavior is shown in Section 9.2.1.3. All other constraints of the functions are applied to the GPI-based trigger.

Table 8-1. GPI Configuration

REGISTER NAME	GPI_EN	GPI_CONFIG	PIN FUNCTION	PIN EDGE	COMMAND
D2h, CONFIG2 and D3h, TRIGGER	0	X	None	X	No Operation (Default)
	1	000	Power-Up, Down (Hi-Z)	Rising	Power-Up
				Falling	Hi-Z Power-Down
	1	001	Power-Up, Down (10-kΩ)	Rising	Power-Up
				Falling	10-kΩ Power-Down
	1	010	Margin-High, Low	Rising	Margin High Trigger
				Falling	Margin Low Trigger
	1	011	Function Generation	Rising	Start Function Generation
				Falling	Stop Function Generation
	1	100	High-Priority Medical Alarm	Rising	Start High-Priority Medical Alarm
				Falling	Stop High-Priority Medical Alarm
	1	101	Medium-Priority Medical Alarm	Rising	Start Medium-Priority Medical Alarm
				Falling	Stop Medium-Priority Medical Alarm
	1	110	Low-Priority Medical Alarm	Rising	Start Low-Priority Medical Alarm
				Falling	Stop Low-Priority Medical Alarm
	1	111	I^{2}C Slave Address	Rising	Enable I^{2}C Slave Address Update
				Falling	Disable I^{2}C Slave Address Update

8.3.3 DAC Update

The DAC output pin (OUT) is updated at the end of I²C DAC write frame.

8.3.3.1 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.

8.3.4 Nonvolatile Memory (EEPROM or NVM)

The DACx3701 contain 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 8-2, 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 DACx3701 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 DACx3701 also implement 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 8-2. NVM Programmable Registers

REGISTER ADDRESS	REGISTER NAME	BIT ADDRESS	BIT NAME
D1h	GENERAL_CONFIG	13	DEVICE_LOCK
		11:9	CODE_STEP
		8:5	SLEW_RATE
		4:3	DAC_PDN
		2	REF_EN
		1:0	DAC_SPAN
D2h	CONFIG2	15:14	SLAVE_ADDRESS
		13:11	GPI_CONFIG
		5:4	INTERBURST_TIME
		3:2	PULSE_OFF_TIME
		1:0	PULSE_ON_TIME
D3h	TRIGGER	10	GPI_EN
21h	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)

8.3.4.1 NVM Cyclic Redundancy Check

The DACx3701 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 DACx3701:

- NVM_CRC_ALARM_USER
- 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.

8.3.4.2 NVM_CRC_ALARM_USER Bit

A logic 1 on NVM_CRC_ALARM_USER bit indicates that the user-programmable NVM data are 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 *Section 8.3.7*) command, or cycle power to the DAC. A power cycle also reloads the user-programmable NVM bits. In case of NVM data corruption, program the NVM again.

8.3.4.3 NVM_CRC_ALARM_INTERNAL Bit

A logic 1 on NVM_CRC_ALARM_INTERNAL bit indicates that the internal NVM data are 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. In case of a temporary failure, to reset the alarm bits to 0, issue a software reset (see *Section 8.3.7*) command or cycle power to the DAC.

8.3.5 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 *Section 7.5*. 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 8-3 and Table 8-4 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 8-3. 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 8-4. Slew Rate

REGISTER ADDRESS AND NAME	SLEW_RATE[3]	SLEW_RATE[2]	SLEW_RATE[1]	SLEW_RATE[0]	TIME PERIOD (PER STEP)
D1h, GENERAL_CONFIG	0	0	0	0	25.6 µs
	0	0	0	1	32 µs
	0	0	1	0	38.4 µs
	0	0	1	1	44.8 µs
	0	1	0	0	204.8 µs
	0	1	0	1	256 µs
	0	1	1	0	307.2 µs
	0	1	1	1	819.2 µs
	1	0	0	0	1638.4 µs
	1	0	0	1	2457.6 µs
	1	0	1	0	3276.8 µs
	1	0	1	1	4915.2 µs
	1	1	0	0	12 µs
	1	1	0	1	8 µs
	1	1	1	0	4 µs
	1	1	1	1	0 µs, no slew (default)

8.3.6 Power-on-Reset (POR)

The DACx3701 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 DACx3701 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 8-1, 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.

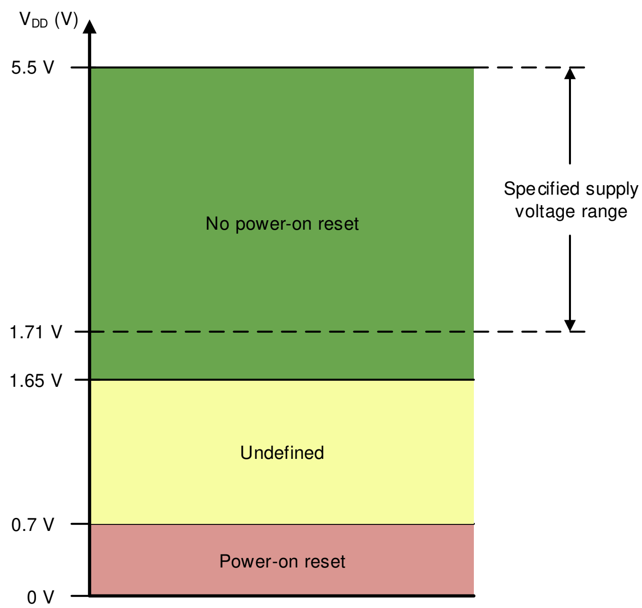


Figure 8-1. Threshold Levels for V_{DD} POR Circuit

8.3.7 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.

8.3.8 Device Lock Feature

The DACx3701 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 bypass the DEVICE_LOCK setting, write 0101 to the DEVICE_UNLOCK_CODE bits (address D3h).

8.3.9 PMBus Compatibility

The PMBus protocol is an I²C-based communication standard for power-supply management. PMBus contains standard command codes tailored to power supply applications. The DACx3701 implement some PMBus commands such as *Turn Off*, *Turn On*, *Margin Low*, *Margin High*, *Communication Failure Alert Bit (CML)*, as well as *PMBUS revision*. Figure 8-2 shows typical PMBus connections. The EN_PMBUS bit (Bit 12, address D1h) must be set to 1 to enable the PMBus protocol.

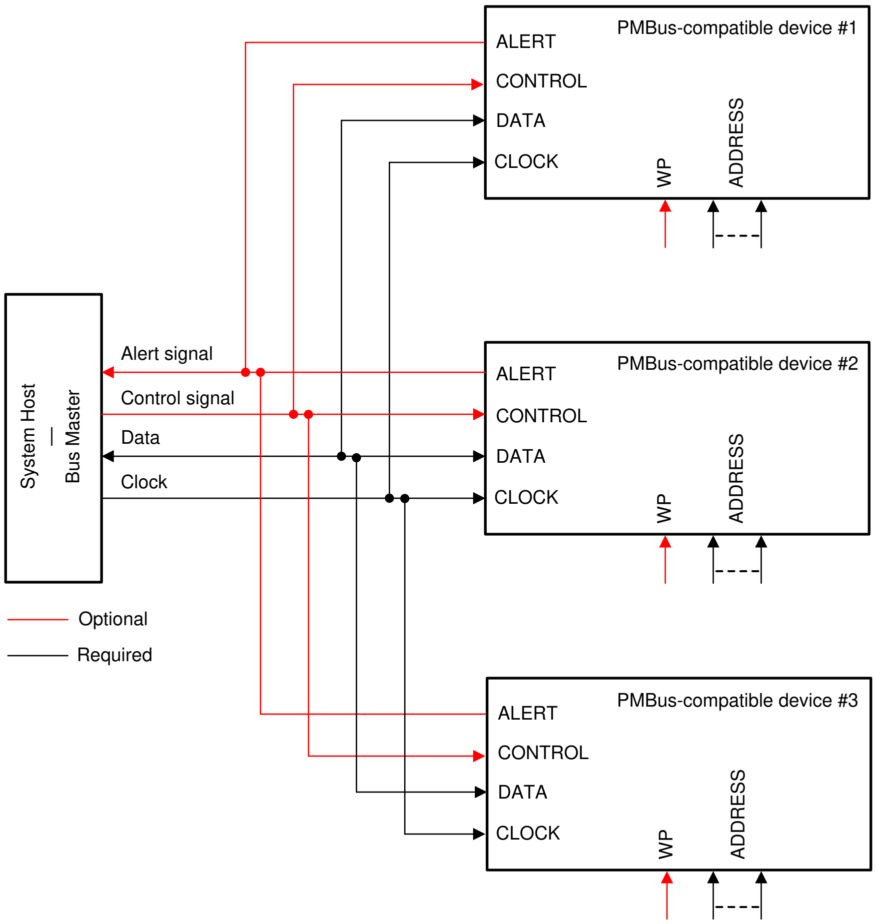


Figure 8-2. PMBus Connections

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. Then the receiver transmits the data following the same most significant byte first format (see Table 8-11).

8.4 Device Functional Modes

8.4.1 Power Down Mode

The DACx3701 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 8-5 shows the DAC power-down bits.

Table 8-5. 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 DACx3701 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 8-6 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.

$$f_{SQUARE-WAVE} = \frac{1}{2 \times SLEW\_RATE} \tag{3}$$

$$f_{TRIANGLE-WAVE} = \frac{1}{2 \times SLEW\_RATE \times \left(\frac{MARGIN\_HIGH - MARGIN\_LOW + 1}{CODE\_STEP}\right)} \tag{4}$$

$$f_{SAWTOOTH-WAVE} = \frac{1}{SLEW\_RATE \times \left(\frac{MARGIN\_HIGH - MARGIN\_LOW + 1}{CODE\_STEP}\right)} \tag{5}$$

where:

- SLEW_RATE is the programmable DAC slew rate specified in Table 8-4.
- MARGIN_HIGH and MARGIN_LOW are the programmable DAC codes.
- CODE_STEP is the programmable DAC step code in Table 8-3.

Table 8-6. 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 and CODE_STEP (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 and CODE_STEP (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 and CODE_STEP (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 DACx3701 I^{2}C interface implements some of the PMBus commands. Table 8-7 shows the supported PMBus commands that are implemented in DACx3701.The DAC uses MARGIN_LOW (address 26h), MARGIN_HIGH (address 25h) bits, SLEW_RATE, and CODE_STEP bits (address D1h) for PMBUS_OPERATION_CMD. The EN_PMBUS bit (Bit 12, address D1h) must be set to 1 to enable the PMBus protocol.

Table 8-7. PMBus Operation Commands

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

The DACx3701 also implement PMBus features such as group command protocol and communication time-out failure. The CML bit (address 78h) indicates a communication fault in the PMBus. This bit is reset by writing 1.

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

8.4.4 Medical Alarm Generation Mode

The DACx3701 are also 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 DACx3701 allow 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. The GPI pin can be used for trigerring an alarm directly without using the I^{2}C interface. This feature helps when the processor fails or the software crashes. This feature is also helpful when there is a power failure and the alarm circuit is driven by a battery or a super capacitor.

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.

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 8-8 lists the INTERBURST_TIME settings.

Table 8-8. 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, CONFIG2	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 8-9 lists the PULSE_OFF_TIME settings.

Table 8-9. 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, CONFIG2	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 8-10 lists the PULSE_ON_TIME settings.

Table 8-10. 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, CONFIG2	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 DACx3701 devices have a 2-wire serial interface (SCL and SDA) as shown in the pin diagram of *Section 6*. The I²C 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 I²C-compatible devices connect to the I²C bus through the open drain I/O pins, SDA and SCL.

The I²C 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 I²C bus is typically a microcontroller or digital signal processor (DSP). The DACx3701 family operates as a slave device on the I²C bus. A slave device acknowledges master commands, and upon master control, receives or transmits data.

Typically, theDACx3701 family operates as a slave receiver. A master device writes to the DACx3701, a slave receiver. However, if a master device requires the DACx3701 internal register data, the DACx3701 operate as a slave transmitter. In this case, the master device reads from the DACx3701. According to I²C terminology, read and write refer to the master device.

The DACx3701 family is a slave and supports the following data transfer modes:

- Standard mode (100 kbps)
- Fast mode (400 kbps)
- Fast mode plus (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 plus 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 DACx3701 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 rising edge of the ACK bit, following the second byte.

Other than specific timing signals, the I²C interface works with serial bytes. At the end of each byte, a ninth clock cycle generates and detects an acknowledge signal. An 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 8-3.

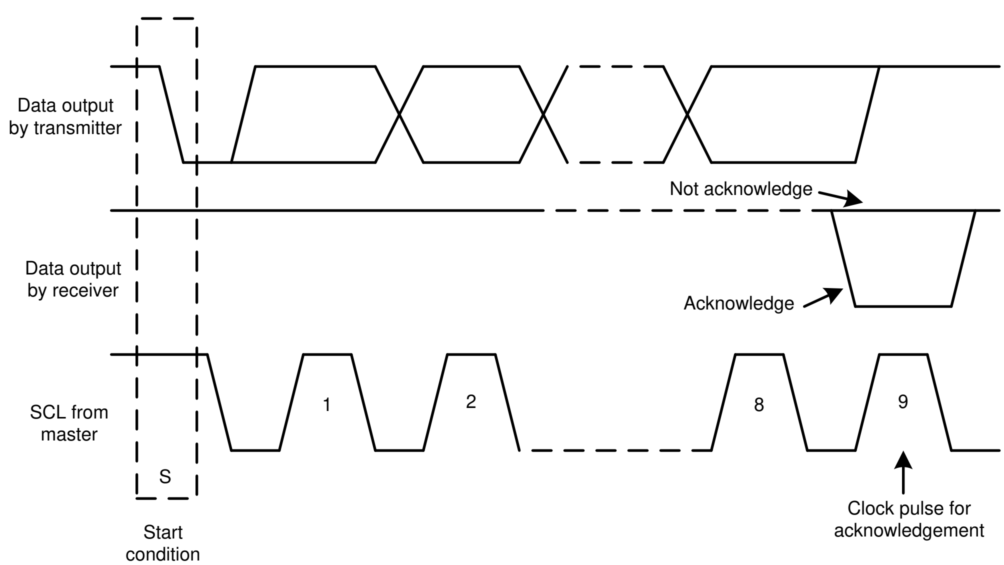


Figure 8-3. Acknowledge and Not Acknowledge on the I²C Bus

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 8-4. All I^{2}C-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/$\overline{W}$) 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 8-5. 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 8-3. 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/$\overline{W}$ bit 0) or receive (R/$\overline{W}$ 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, as shown in Figure 8-4. This action releases the bus and stops the communication link with the addressed slave. All I^{2}C-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.

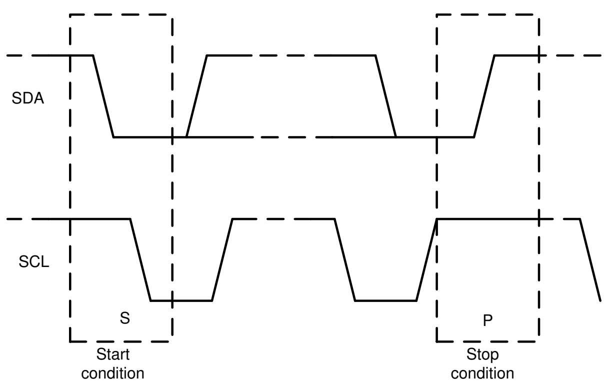


Figure 8-4. Start and Stop Conditions

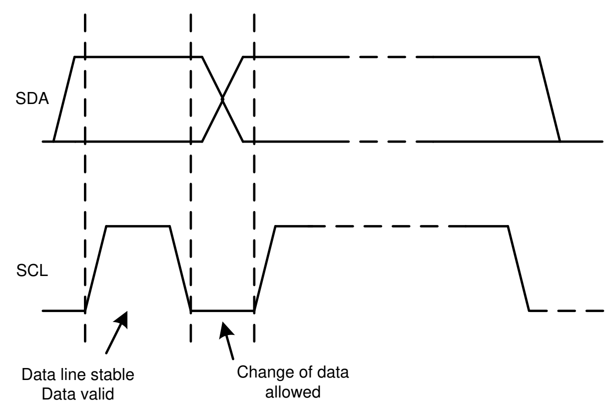


Figure 8-5. Bit Transfer on the I^{2}C Bus

8.5.2 I²C Update Sequence

For a single update, the DACx3701 require a start condition, a valid I²C address byte, a command byte, and two data bytes, as listed in Table 8-11.

Table 8-11. Update Sequence

MSB	....	LSB	ACK	MSB	...	LSB	ACK	MSB	...	LSB	ACK	MSB	...	LSB	ACK
Address (A) byte Section 8.5.2.1				Command byte Section 8.5.2.2				Data byte - MSDB				Data byte - LSDB			
DB [31:24]				DB [23:16]				DB [15:8]				DB [7:0]			

After each byte is received, the DACx3701 family acknowledges the byte by pulling the SDA line low during the high period of a single clock pulse, as shown in Figure 8-6. 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 DACx3701 devices.

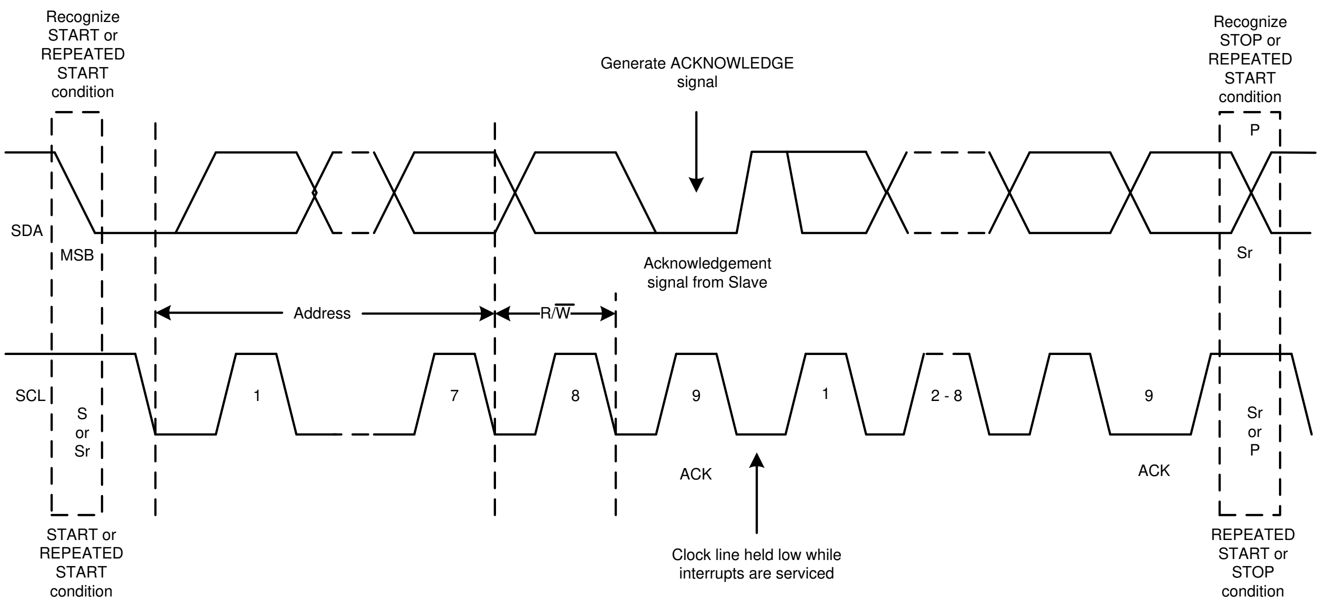


Figure 8-6. I²C Bus Protocol

The command byte sets the operating mode of the selected DACx3701 device. For a data update to occur when the operating mode is selected by this byte, the DACx3701 device must receive two data bytes: the most significant data byte (MSDB) and least significant data byte (LSDB). The DACx3701 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 10 kSPS. Using the fast mode plus (clock = 1 MHz), the maximum DAC update rate is limited to 25 kSPS. When a stop condition is received, the DACx3701 device releases the I²C bus and awaits a new start condition.

8.5.2.1 Address Byte

The address byte, as shown in the following table, is the first byte received following the start condition from the master device. The first five bits (MSBs) of the address are factory preset to 10010. The next two bits of the address are controlled by the SLAVE_ADDRESS field in the CONFIG2 register. Follow the procedure described in the next section to configure the slave address. The possible slave addresses using these bits are also shown in the next section.

Table 8-12. Address Byte

COMMENT	MSB							LSB
—	AD6	AD5	AD4	AD3	AD2	AD1	AD0	R/ $\overline{W}$
General address	1	0	0	1	0	See Table 8-13 (slave address column)		0 or 1
Broadcast address	1	0	0	0	1	1	1	0

The DACx3701 family supports broadcast addressing, which can be used for synchronously updating or powering down multiple DACx3701 devices. The DACx3701 family is designed to work with other members of the family to support multichip synchronous updates. Using the broadcast address, the DACx3701 devices respond regardless of the states of the SLAVE_ADDRESS bits. Broadcast is supported only in write mode.

8.5.2.1.1 Slave Address Configuration

This section provides the step by step procedure to configure the I^{2}C slave addresses for up to four DACs. Use the broadcast address for all the steps.

1. Set GPI pin to 0b for all devices.
2. Set GPI_CONFIG in the CONFIG2 register to 111b.
3. Set GPI_EN in the TRIGGER register to 1b.
4. Set the GPI pin to logic HIGH for the device that needs to be configured.
5. Write data to SLAVE_ADDRESS bit field in the CONFIG2 register. Only the device with GPI pin logic HIGH updates the SLAVE_ADDRESS setting passed in the command. Make sure that the rest of the devices on the same I^{2}C bus have their respective GPI pins set to logic LOW during this process.
6. Toggle the GPI pin of the device bring programmed to logic LOW.
7. Repeat steps (1) through (6) above to program the I2C slave addresses to all the devices on the bus.
8. Set GPI_EN to 0b.
9. Change GPI_CONFIG to 000b.
10. Trigger NVM write operation.

The devices are now ready for use.

Table 8-13. Address Format

SLAVE ADDRESS	SLAVE_ADDRESS FIELD IN CONFIG2 REGISTER
1001000	00 (default)
1001001	01
1001010	10
1001011	11

8.5.2.2 Command Byte

Table 8-14 lists the command byte addresses.

Table 8-14. Command Byte (Register Names)

ADDRESS	REGISTER NAME
D0h	STATUS
D1h	GENERAL_CONFIG
D2h	CONFIG2
D3h	TRIGGER
21h	DAC_DATA
25h	DAC_MARGIN_HIGH
26h	DAC_MARGIN_LOW
01h	PMBUS_OPERATION
78h	PMBUS_STATUS_BYTE
98h	PMBUS_VERSION

8.5.3 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.

The broadcast address cannot be used for reading.

Table 8-15. Read Sequence

<table>
<tr><th>S</th><th>MSB</th><th>…</th><th>R/ $\overline{W}$ (0)</th><th>ACK</th><th>MSB</th><th>…</th><th>LSB</th><th>ACK</th><th>Sr</th><th>MSB</th><th>…</th><th>R/ $\overline{W}$ (1)</th><th>ACK</th><th>MSB</th><th>…</th><th>LSB</th><th>ACK</th><th>MSB</th><th>…</th><th>LSB</th><th>ACK</th></tr>
<tr><td></td><td colspan="3">ADDRESS BYTE
Section 8.5.2.1</td><td></td><td colspan="3">COMMAND BYTE
Section 8.5.2.2</td><td></td><td>Sr</td><td colspan="3">ADDRESS BYTE
Section 8.5.2.1</td><td></td><td colspan="3">MSDB</td><td></td><td colspan="3">LSDB</td><td></td></tr>
<tr><td colspan="4">From Master</td><td>Slave</td><td colspan="3">From Master</td><td>Slave</td><td colspan="4">From Master</td><td>Slave</td><td colspan="3">From Slave</td><td>Master</td><td colspan="3">From Slave</td><td>Master</td></tr>
</table>

8.6 Register Map

Table 8-16. Register Map

ADDRESS	MOST SIGNIFICANT DATA BYTE (MSDB)								LEAST SIGNIFICANT DATA BYTE (LSDB)							
	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
D0h	NVM_CRC_ALARM_USER	NVM_CRC_ALARM_INTERNAL	NVM_BUSY	DAC_UPDATE_BUSY	X(1)						DEVICE_ID				VERSION_ID	
D1h	FUNC_CONFIG		DEVICE_LOCK	EN_PMBUS	CODE_STEP			SLEW_RATE				DAC_PDN		REF_EN	DAC_SPAN	
D2h	SLAVE_ADDRESS		GPI_CONFIG			MED_ALARM_HP	MED_ALARM_MP	MED_ALARM_LP	RESERVED		INTERBURST_TIME		PULSE_OFF_TIME		PULSE_ON_TIME	
D3h	DEVICE_UNLOCK_CODE				X	GPI_EN	DEVICE_CONFIG_RESET	START_FUNC_GEN	PMBUS_MARGIN_HIGH	PMBUS_MARGIN_LOW	NVM_RELOAD	NVM_PROG	SW_RESET			
21h	X				DAC_DATA[9:0] (10-Bit) or DAC_DATA[7:0] (8-Bit)										X	
25h	X				MARGIN_HIGH[9:0] (10-Bit) or MARGIN_HIGH[7:0] (8-Bit)										X	
26h	X				MARGIN_LOW[9:0] (10-Bit) or MARGIN_LOW[7:0] (8-Bit)										X	
01h	PMBUS_OPERATION_CMD								N/A							
78h	X						CML	X	N/A							
98h	PMBUS_VERSION								N/A							

(1) X = Don't care.

Table 8-17. Register Names

ADDRESS	REGISTER NAME	SECTION
D0h	STATUS	Section 8.6.1
D1h	GENERAL_CONFIG	Section 8.6.2
D2h	CONFIG2	Section 8.6.3
D3h	TRIGGER	Section 8.6.4
21h	DAC_DATA	Section 8.6.5
25h	DAC_MARGIN_HIGH	Section 8.6.6
26h	DAC_MARGIN_LOW	Section 8.6.7
01h	PMBUS_OPERATION	Section 8.6.8
78h	PMBUS_STATUS_BYTE	Section 8.6.9
98h	PMBUS_VERSION	Section 8.6.10

Table 8-18. Access Type Codes

Access Type	Code	Description
X	X	Don't care
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

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

Figure 8-7. STATUS Register

15	14	13	12	11 - 6	5 - 2	1 - 0
NVM_CRC_ ALARM_ USER	NVM_CRC_ ALARM_ INTERNAL	NVM_ BUSY	DAC_ UPDATE_ BUSY	X	DEVICE_ID	VERSION_ID
R-0h	R-0h	R-0h	R-0h	X-00h	10-bit: R-3h 8-bit: R-5h	R-0h

Table 8-19. 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 1 : NVM write or load in progress, Write to DAC register map not allowed
12	DAC_UPDATE_BUSY	R	0	0 : DAC outputs updated, Write to DAC registers allowed 1 : DAC outputs update in progress, Write to DAC register map not allowed
11 - 6	X	X	00h	Don't care
5 - 2	DEVICE_ID	R	DAC53701: 3h DAC43701: 5h	Device identifier: DAC53701: 3h DAC43701: 5h
1 - 0	VERSION_ID	R	0h	Silicon version identifier. This field may have a different value based on the silicon revision.

8.6.2 GENERAL_CONFIG Register (address = D1h) [reset = 01F0h]

Figure 8-8. GENERAL_CONFIG Register

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

Table 8-20. 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 and CODE_STEP 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 and CODE_STEP 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 and CODE_STEP 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 bits.
13	DEVICE_LOCK	R/$\overline{W}$	0	0: Device not locked 1: Device locked, the device locks all the registers. This bit can be overwritten (unlock device) by writing 0101 to the DEVICE_UNLOCK_CODE bits (address D3h)

Table 8-20. GENERAL_CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
12	EN_PMBUS	R/$\overline{W}$	0	0: PMBus mode disabled 1: PMBus mode enabled
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: 32 µs (per step) 0010: 38.4 µs (per step) 0011: 44.8 µs (per step) 0100: 204.8 µs (per step) 0101:256 µs (per step) 0110: 307.2 µs (per step) 0111: 819.2 µs (per step) 1000: 1.6384 ms (per step) 1001: 2.4576 ms (per step) 1010: 3.2768 ms (per step) 1011: 4.9152 ms (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 10 kΩ 10: Power down to high impedance (default) 11: Power down to 10 kΩ
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: Internal reference enabled, DAC reference = 1.21 V, DAC output range is a function of DAC_SPAN.
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 CONFIG2 Register (address = D2h) [reset = 0000h]

Figure 8-9. CONFIG2 Register

15 14	13 12 11	10	9	8	7 6	5 4	3 2	1 0
SLAVE_ADDRESS	GPI_CONFIG	MED_ALARM_HP	MED_ALARM_MP	MED_ALARM_LP	RESERVED	MED_ALARM_DEAD_TIME	PULSE_OFF_TIME	PULSE_ON_TIME
R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	RESERVED	R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h

Table 8-21. CONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description		
15 - 14	SLAVE_ADDRESS	R/$\overline{W}$	0h	AD1-AD0 of device address as per Table 8-13		
13-11	GPI_CONFIG	R/$\overline{W}$	0h	Refer to Table 8-1 for the GPI configuration		
10	MED_ALARM_HP	R/$\overline{W}$	0	0: No medical alarm waveform generated 1: High priority medical alarm waveform generated		
9	MED_ALARM_MP	R/$\overline{W}$	0	0: No medical alarm waveform generated 1: Medium priority medical alarm waveform generated		
8	MED_ALARM_LP	R/$\overline{W}$	0	0: No medical alarm waveform generated 1: Low priority medical alarm waveform generated		
7 - 6	RESERVED	Reserved	0	RESERVED		
5 - 4	INTERBURST_TIME	R/$\overline{W}$	00	High priority alarm 00: 2.55 s 01: 2.96 s 10: 3.38 s 11: 3.80 s	Medium priority alarm 00: 2.60 s 01: 3.06 s 10: 3.52 s 11: 4.00 s	Low priority alarm 00: 16 s 01: 16 s 10: 16 s 11: 16 s
3 - 2	PULSE_OFF_TIME	R/$\overline{W}$	00	High priority alarm 00: 15 ms 01: 36 ms 10: 58 ms 11: 80 ms	Medium priority alarm 00: 40 ms 01: 60 ms 10: 80 ms 11: 100 ms	Low priority alarm 00: 40 ms 01: 60 ms 10: 80 ms 11: 100 ms
1 - 0	PULSE_ON_TIME	R/$\overline{W}$	00	High priority alarm 00: 80 ms 01: 103 ms 10: 126 ms 11: 150 ms	Medium priority alarm 00: 130 ms 01: 153 ms 10: 176 ms 11: 200 ms	Low priority alarm 00: 130 ms 01: 153 ms 10: 176 ms 11: 200 ms

8.6.4 TRIGGER Register (address = D3h) [reset = 0008h]

Figure 8-10. TRIGGER Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEVICE_UNLOCK_CODE				X	GPI_EN	DEVICE_CONFIG_RESET	START_FUNC_GEN	PMBUS_MARGIN_HIGH	PMBUS_MARGIN_LOW	NVM_RELOAD	NVM_PROG	SW_RESET			
$\overline{W}$-0h				X-0h	R/$\overline{W}$-0h	$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	R/$\overline{W}$-0h	$\overline{W}$-8h			

Table 8-22. 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	X	X	0h	Don't care
10	GPI_EN	R/$\overline{W}$	0	0: GPI disabled 1: GPI enabled
9	DEVICE_CONFIG_RESET	$\overline{W}$	0	0: Device configuration reset not initiated 1: Device configuration reset initiated. All registers loaded with factory reset values.
8	START_FUNC_GEN	R/$\overline{W}$	0	0: Continuous waveform generation mode disabled 1: Continuous waveform generation mode enabled, device generates continuous waveform based on FUNC_CONFIG (address D1h), MARGIN_LOW (address 26h), MARGIN_HIGH (address 25h), and SLEW_RATE and CODE_STEP (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). This bit automatically resets to 0 after the DAC code reaches MARGIN_HIGH value.
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). This bit automatically resets to 0 after the DAC code reaches MARGIN_LOW value.
5	NVM_RELOAD	R/$\overline{W}$	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.. This bit is self-resetting.
4	NVM_PROG	R/$\overline{W}$	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. This bit is self-resetting.
3 - 0	SW_RESET	$\overline{W}$	1000	1000: 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 8-11. 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-0h				R/W-000h										X-0h	

Table 8-23. 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]	R/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: DACx3701: { DATA[9:0] } DACx3701: { 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 8-12. 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-0h				R/W-000h										X-0h	

Table 8-24. 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	R/W	000h	Margin high code for DAC output. Data are in straight binary format and use the following format: DACx3701: { MARGIN_HIGH[[9:0] } DACx3701: { 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 8-13. 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-0h				R/W-000h										X-0h	

Table 8-25. 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	R/W	000h	Margin low code for DAC output. Data are in straight binary format and follows the format below: DACx3701: { MARGIN_LOW[[9:0] } DACx3701: { MARGIN_LOW[[7:0], X, X } X = Don't care bits
1-0	X	X	0h	Don't care

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

Figure 8-14. PMBUS_OPERATION Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMBUS_OPERATION_CMD								X							
R/W̄-00h								X-00h							

Table 8-26. PMBUS_OPERATION Register Field Descriptions

Bit	Field	Type	Reset	Description
15 - 8	PMBUS_OPERATION_CMD	R/W̄	00h	PMBus operation commands 00h: Turn off 80h: Turn on A4h: Margin high, DAC output margins high to MARGIN_HIGH code (address 25h) 94h: Margin low, DAC output margins low to MARGIN_LOW code (address 26h)
7 - 0	X	X	00h	Not applicable

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

Figure 8-15. PMBUS_STATUS_BYTE Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X						CML	X	N/A							
X-00h						R/W̄-0h	X-0h	X-00h							

Table 8-27. PMBUS_STATUS_BYTE Register Field Descriptions

Bit	Field	Type	Reset	Description
15 - 10	X	X	00h	Don't care
9	CML	R/W̄	0	0: No communication Fault 1: PMBus communication fault for write with incorrect number of clocks, read before write command, invalid command address, and invalid or unsupported data value; reset this bit by writing 1.
8	X	X	0h	Don't care
7 - 0	X	X	00h	Not applicable

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

Figure 8-16. PMBUS_VERSION Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMBUS_VERSION								X							
R-22h								X-00h							

Table 8-28. PMBUS_VERSION Register Field Descriptions

Bit	Field	Type	Reset	Description
15 - 8	PMBUS_VERSION	R	22h	PMBus version
7 - 0	X	X	00h	Not applicable

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, as well as validating and testing their design implementation to confirm system functionality.

9.1 Application Information

The DACx3701 are buffered, force-sense output, single-channel, DACs that include an NVM and internal reference and are available in a tiny 2-mm × 2-mm package . These DACs are designed for general-purpose applications in a wide range of end equipment. Some of the most common applications for these devices are power-supply margining and control, adaptive voltage scaling (AVS), *set-and-forget* LED biasing in mobile projectors, general-purpose function generation, medical alarm generation, and 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 DACx3701: programmable LED biasing, power-supply margining. and medical alarm generation.

9.2.1 Appliance Light Fade-In Fade-Out

Appliances such as toaster ovens, microwave ovens, refrigerators, cloth dryers, and more implement door lights for monitoring the status of the function. These door lights dim and brighten when the door closes and opens, respectively. Appliance manufacturers prefer to provide a smooth-dimming transition for a better user experience. However, a microcontroller is required for such an operation, and implementing a separate microcontroller and associated software is a big overhead. For this reason, only high-end appliances have such features. The DACx3701 provides a simpler way to control the slew of such lights without software. Figure 9-1 shows the simplified circuit diagram of light fade-in fade-out using MOSFET based control and Figure 9-2 shows the circuit with an external LED driver. For high-power LEDs, external LED drivers with headroom control are preferred over MOSFET-based LED control.

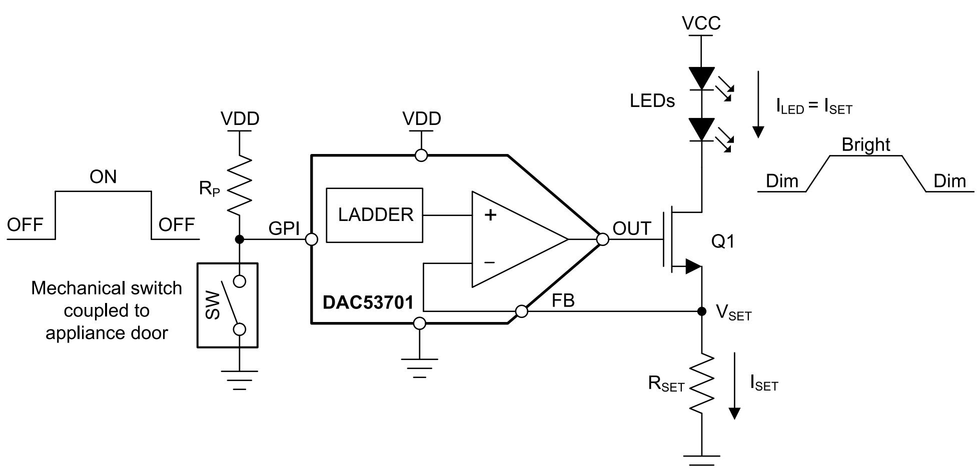


Figure 9-1. Appliance Light Fade-In Fade-Out

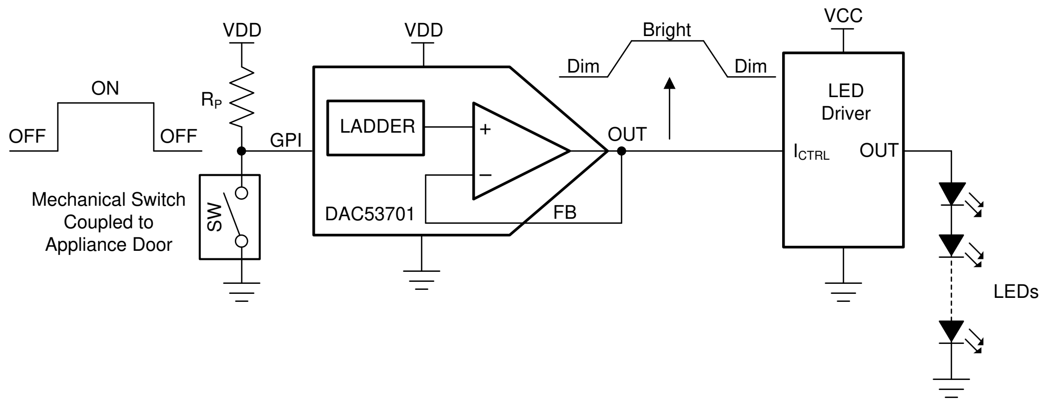


Figure 9-2. Fade-In Fade-Out with Switching LED Driver

9.2.1.1 Design Requirements

- Slew time: approximately, 1.5 s
- Bright LED current: 20 mA
- Dim LED current: 10 mA

9.2.1.2 Detailed Design Procedure

Choose a small V_{SET} so that the power dissipation across R_{SET} is minimum. Choose 1 V for the bright condition, which results in an R_{SET} of (1 V / 20 mA) = 50 Ω. Set the DACx3701 output span to 1.8 V. The output buffer of the DAC is connected in a force-sense configuration to the MOSFET, as shown in Figure 9-1. This configuration compensates the gate-source voltage drop caused by temperature, drain current, and ageing of the MOSFET. Considering a typical gate-source voltage of 1.2 V and a power supply headroom of 200 mV, the VDD for the DAC must be a minimum of (1 V + 1.2 V + 200 mV) = 2.4 V. Use a standard 3.3-V or 5-V power supply for the DAC. A bipolar junction transistor (BJT) provides a much smaller base-emitter voltage drop, but a MOSFET has better matching between the drain and source currents. Choose a BJT over the MOSFET in case there is a less than 2.4-V supply voltage available for the DAC. Configure the MARGIN HIGH value to the code equivalent of 1 V; that is (1 V / 1.8 V) × 1024 = 569d or 0x239. The MARGIN LOW value should be the equivalent of the dim LED current that is 10 mA, which corresponds to a DAC voltage of (10 mA × 50 Ω) = 500 mV. The code for MARGN LOW is (500 mV / 1.8 V) × 1024 = 284d or 0x11C.

For control without the use of software, map the GPI to margin high-low operation as listed in Table 8-1. The rising edge of the GPI maps to the MARGIN HIGH value of the 20-mA LED current, and the falling edge maps to the MARGIN LOW value of the 10-mA LED current. When the DAC output is in the slewing condition, any change in the GPI state changes the direction of the slew after the ongoing SLEW_RATE time, as shown in the Application Curves section.

The slew time is given by (MARGIN_HIGH – MARGIN_LOW) × CODE_STEP × SLEW_RATE. For a 1.5-s slew time, CODE_STEP × SLEW_RATE = 1.5 / (569 – 284) = ~ 5 ms. Choose the CODE_STEP as 1 LSB and SLEW_RATE of 4.9152 ms. This configuration provides a slew time of 1.4 s. Adjust the MARGIN HIGH and MARGIN LOW values for more granular control.

The following pseudocode helps to get started with a light fade-in fade-out application:

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Write MARGIN-HIGH code (12-bit aligned) for bright LED light
//For a 1.8-V output range, the 10-bit hex code for 1 V is 0x0239.
//With 12-bit alignment, it becomes 0x08E4
WRITE DAC_MARGIN_HIGH(0x25), 0x08, 0xE4
//Write MARGIN-LOW code (12-bit aligned) for dim LED light
//For a 1.8-V output range, the 10-bit hex code for 500 mV is 0x11C.
//With 12-bit alignment, it becomes 0x0470
WRITE DAC_MARGIN_LOW(0x26), 0x04, 0x70
//Map GPI to margin high-low function
WRITE CONFIG2(0xD2), 0x10, 0x00
//Enable GPI
WRITE TRIGGER(0xD3), 0x04, 0x08
//Configure internal reference with 1.5x output span, and slew time and power-up the device
//CODE_STEP: 1 LSB, SLEW_RATE: 4.9152 ms
WRITE GENERAL_CONFIG(0xD1), 0x01, 0x64
//Program the EEPROM
WRITE TRIGGER(0xD3), 0x04, 0x18
```

9.2.1.3 Application Curves

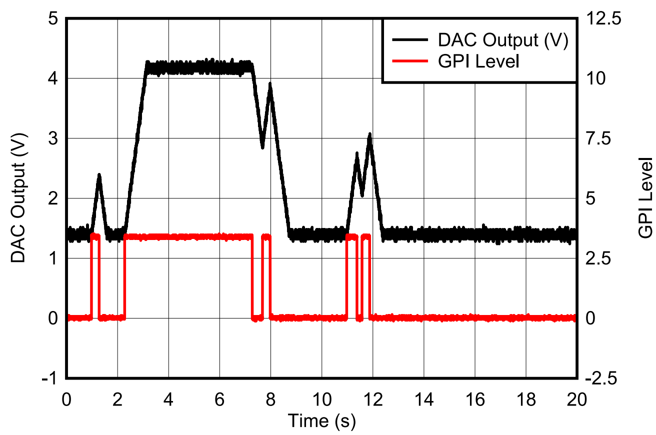


Figure 9-3. Light Fade-In-Fade-Out With GPI

9.2.2 Power-Supply Margining

A power-supply margining or scaling circuit is used to test and trim the output of a power converter. This example circuit is used to test a system by margining the power supplies, for adaptive voltage scaling, 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 the best choice for controlling the power-supply output linearly. Figure 9-4 shows a control circuit for a switch-mode power supply (SMPS) using the DACx3701. Typical applications of power-supply margining are communications equipment, enterprise servers, test and measurement, and general-purpose power-supply modules.

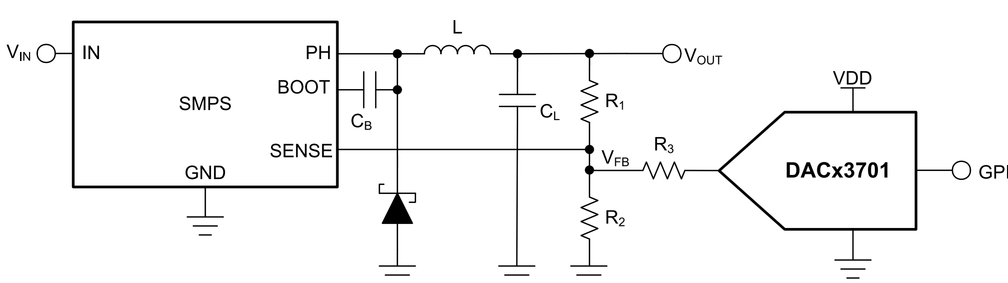


Figure 9-4. Power-Supply Margining

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

9.2.2.2 Detailed Design Procedure

The DACx3701 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, bring 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. Calculate R_1 as (V_{OUT} – V_{FB}) / 100 µA = 27 kΩ.

To achieve ±10% margin-high and margin-low conditions, the DAC must sink or source additional current through R_1. Calculate the current from the DAC (I_{MARGIN}) using Equation 6 as 12 µA.

$$I_{MARGIN} = \left(\frac{V_{OUT} \times (1 + MARGIN) - V_{FB}}{R_1} \right) - I_{NOMINAL} \quad (6)$$

where

- I_{MARGIN} is the margin current sourced or sinked from the DAC.
- MARGIN is the percentage margin value divided by 100.
- $I_{NOMINAL}$ is the nominal current through R_1 and R_2.

To calculate the value of R_3, first decide the DAC output range, and make sure to avoid the codes near zero-scale and full-scale for safe operation in the linear region. A DAC output of 20 mV is a safe consideration as the minimum output, and (1.8 V – 0.6 V – 20 mV = 1.18 V) 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. Calculate the value of R_3 using Equation 7 as 48.3 kΩ. Choose a standard resistor value and adjust the DAC outputs. Choosing R_3 = 47 kΩ makes the DAC margin high code as 1.164 V and the DAC margin low code as 36 mV.

$$R_3 = \frac{|V_{DAC} - V_{FB}|}{I_{MARGIN}} \quad (7)$$

The DACx3701 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 Section 8.6.2 for the slew rate setting details.

Note

The MARGIN HIGH register value in DACx3701 results in the MARGIN LOW value at the power supply output. Similarly, the MARGIN LOW register value in DACx3701 results in the MARGIN HIGH value at the power-supply output.

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

```
//SYNTAX: WRITE <REGISTER NAME (Hex code)>, <MSB DATA>, <LSB DATA>
//Write DAC code (12-bit aligned) for nominal output
//For a 1.8-V output range, the 10-bit hex code for 0.6 V is 0x0155. With 12-bit alignment, it
becomes 0x0554
WRITE DAC_DATA(0x21), 0x05, 0x54
//Write DAC code (12-bit aligned) for margin-low output at the power supply
//For a 1.8-V output range, the 10-bit hex code for 1.164 V is 0x0296. With 12-bit alignment, it
becomes 0x0A58
WRITE DAC_MARGIN_HIGH(0x25), 0x0A, 0x58
//Write DAC code (12-bit aligned) for margin-high output at the power supply
//For a 1.8-V output range, the 10-bit hex code for 36 mV is 0x14. With 12-bit alignment, it becomes
0x50
WRITE DAC_MARGIN_LOW(0x26), 0x00, 0x50
//Power-up the device with enable internal reference with 1.5x output span. This will output the
nominal voltage (0.6 V)
//CODE_STEP: 2 LSB, SLEW_RATE: 25.6 µs
WRITE GENERAL_CONFIG(0xD1), 0x12, 0x14
//Trigger margin-low output at the power supply
WRITE TRIGGER(0xD3), 0x00, 0x80
//Trigger margin-high output at the power supply
WRITE TRIGGER(0xD3), 0x00, 0x40
//Write back DAC code (12-bit aligned) for nominal output
WRITE DAC_DATA(0x21), 0x05, 0x54
```

9.2.2.3 Application Curves

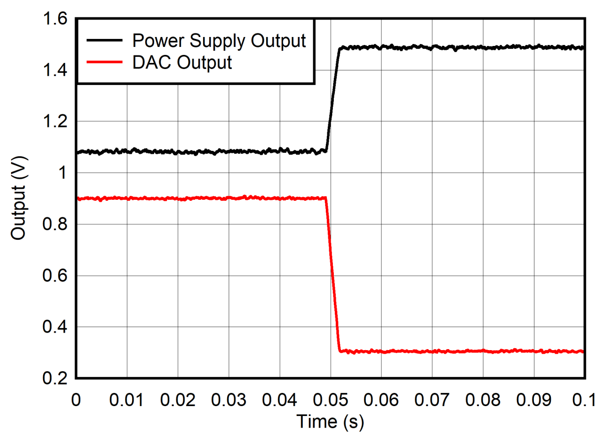


Figure 9-5. Power-Supply Margin High

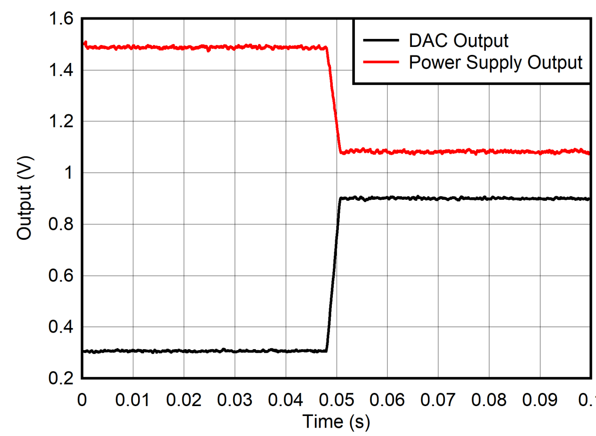


Figure 9-6. Power Supply Margin Low

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 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 DACx3701-based design is aimed at providing a programmable, standalone, and robust implementation.

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.

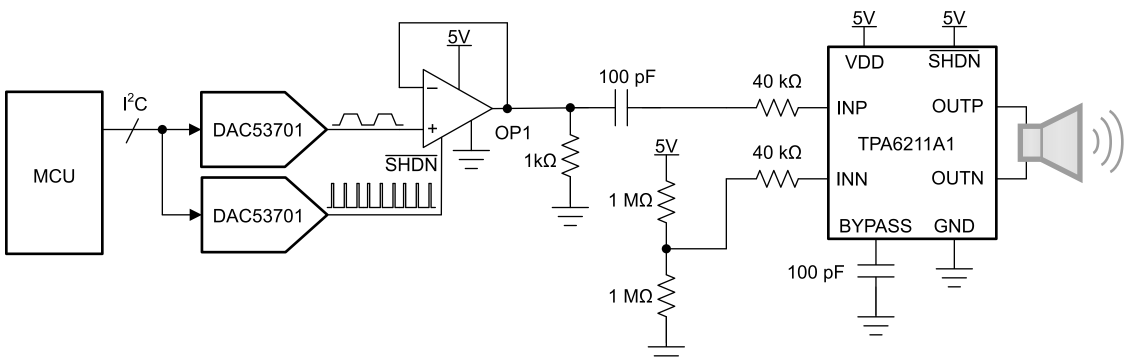


Figure 9-7. Medical Alarm

9.2.3.1 Design Requirements

- Alarm envelope rise and fall time: 26 ms
- Alarm pulse frequency: 610 Hz

9.2.3.2 Detailed Design Procedure

For the auditory alarm implementation, two DAC53701 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 9-7. The signals coming from both DACs are combined together using amplifier OP1 that has a shutdown pin, for example, the TLV9002S or OPA363. The combined signal is then ac-coupled to an audio amplifier, such as the TPA6211A1, to drive the speaker. The TPA6211A1 is an integrated Class-AB amplifier that can drive up to 3 W of output power with very little distortion. As per medical alarm standard IEC60601-1-8, the pulse frequency must be greater than 150 Hz, and must have at least four harmonic components that are within ±15 dB of each other. As a result of the square-wave pulse frequency and the mixing done by OP1, the speaker output generates multiple harmonics of the fundamental pulse frequency. The DACx3701 provide a range of timing options for the pulse frequency and envelope, and various options to program the pulse frequency and envelope timings. See *Section 8.4.4* for the alarm configuration options. Calculate the frequency of a square wave or pulse frequency 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. Generate the triangular waveform using Equation 4. Set the DAC output in the comparator mode by fixing the 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 (see Section 8.6.2).

The following pseudocode helps to get started with a medical alarm application using two DACs:

```
//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 µs x 1.75 = 358.4 µs (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 CONFIG2(0xD2), 0x01, 0x00
//OPTION-2: Configure the second DAC for medium-priority alarm with minimum time settings and trigger
WRITE CONFIG2(0xD2), 0x02, 0x00
//OPTION-3: Configure the second DAC for high-priority alarm with minimum time settings and trigger
WRITE CONFIG2(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
```

9.2.3.3 Application Curves

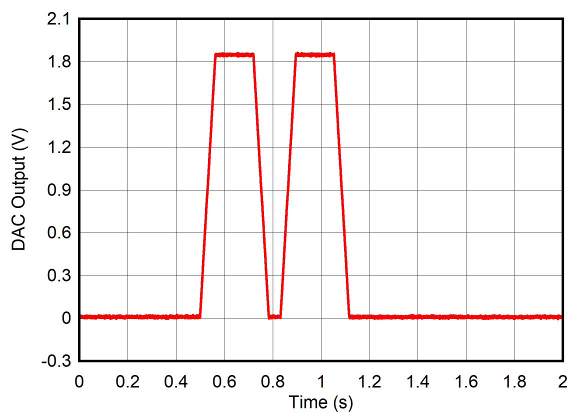


Figure 9-8. Low Priority Alarm

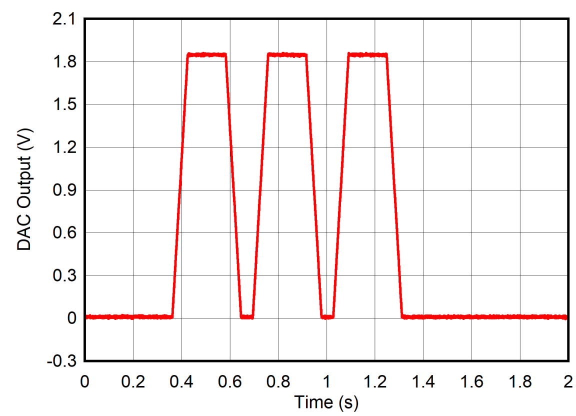


Figure 9-9. Medium Priority Alarm

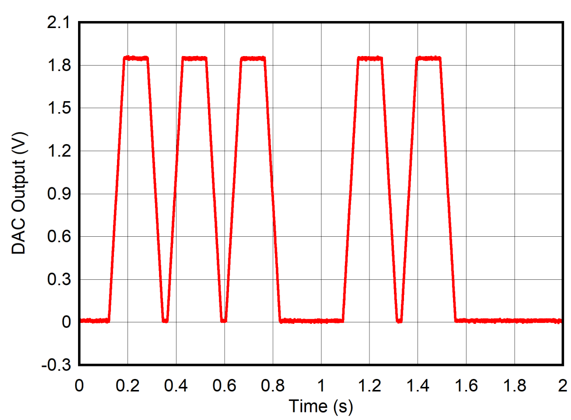


Figure 9-10. High-Priority Alarm

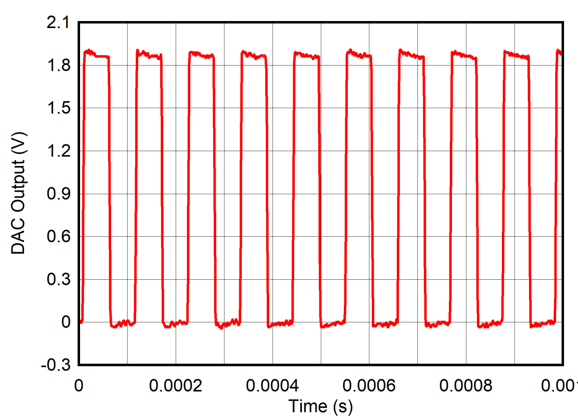


Figure 9-11. Pulse Frequency

10 Power Supply Recommendations

The DACx3701 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 approximately 1.5 μF for the CAP pin.

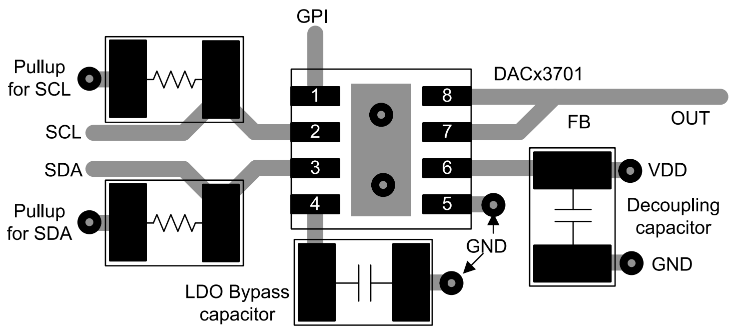
11 Layout

11.1 Layout Guidelines

The DACx3701 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 11-1 shows an example layout drawing with decoupling capacitors and pullup resistors.



(Note: Ground and power planes omitted for clarity)

Figure 11-1. Layout Example

12 Device and Documentation Support

12.1 Documentation Support

12.1.1 Related Documentation

For related documentation see the following:

Texas, Instruments *DAC53701EVM* user's guide

12.2 Receiving Notification of Documentation Updates

To receive notification of documentation updates, navigate to the device product folder on ti.com. Click on *Subscribe to updates* 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.

12.3 Support Resources

TI E2E™ support forums are an engineer's go-to source for fast, verified answers and design help — straight from the experts. Search existing answers or ask your own question to get the quick design help you need.

Linked content is 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.

12.4 Trademarks

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

12.5 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.

12.6 Glossary

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 part number	Status (1)	Material type (2)	Package | Pins	Package qty | Carrier	RoHS (3)	Lead finish/ Ball material (4)	MSL rating/ Peak reflow (5)	Op temp (°C)	Part marking (6)
DAC43701DSGR	Active	Production	WSON (DSG) | 8	3000 | LARGE T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	4371
DAC43701DSGR.A	Active	Production	WSON (DSG) | 8	3000 | LARGE T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	4371
DAC43701DSGRG4	Active	Production	WSON (DSG) | 8	3000 | LARGE T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	4371
DAC43701DSGRG4.A	Active	Production	WSON (DSG) | 8	3000 | LARGE T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	4371
DAC43701DSGT	Active	Production	WSON (DSG) | 8	250 | SMALL T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	4371
DAC43701DSGT.A	Active	Production	WSON (DSG) | 8	250 | SMALL T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	4371
DAC53701DSGR	Active	Production	WSON (DSG) | 8	3000 | LARGE T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	5371
DAC53701DSGR.A	Active	Production	WSON (DSG) | 8	3000 | LARGE T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	5371
DAC53701DSGT	Active	Production	WSON (DSG) | 8	250 | SMALL T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	5371
DAC53701DSGT.A	Active	Production	WSON (DSG) | 8	250 | SMALL T&R	Yes	NIPDAU	Level-1-260C-UNLIM	-40 to 125	5371

(1) **Status:** For more details on status, see our product life cycle.

(2) **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

(3) **RoHS values:** Yes, No, RoHS Exempt. See the TI RoHS Statement for additional information and value definition.

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

(5) **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

(6) **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

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.

OTHER QUALIFIED VERSIONS OF DAC43701, DAC53701 :

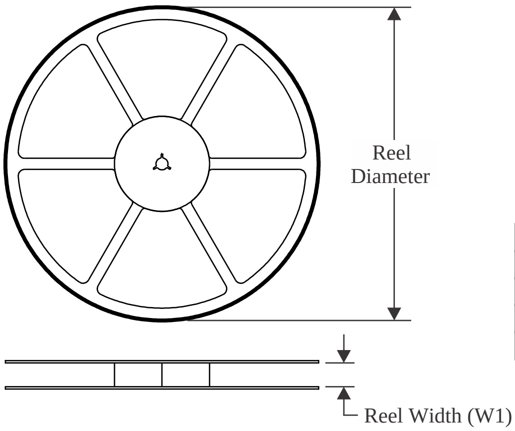
- Automotive : DAC43701-Q1, DAC53701-Q1

NOTE: Qualified Version Definitions:

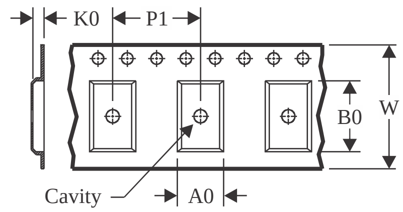
- Automotive - Q100 devices qualified for high-reliability automotive applications targeting zero defects

TAPE AND REEL INFORMATION

REEL DIMENSIONS

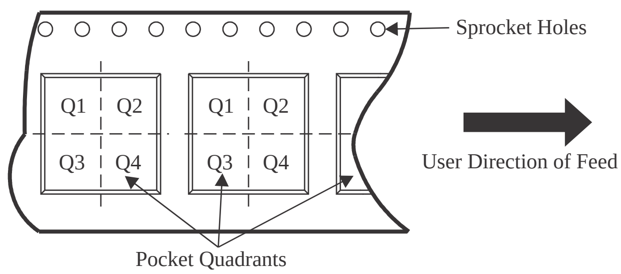


TAPE DIMENSIONS



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

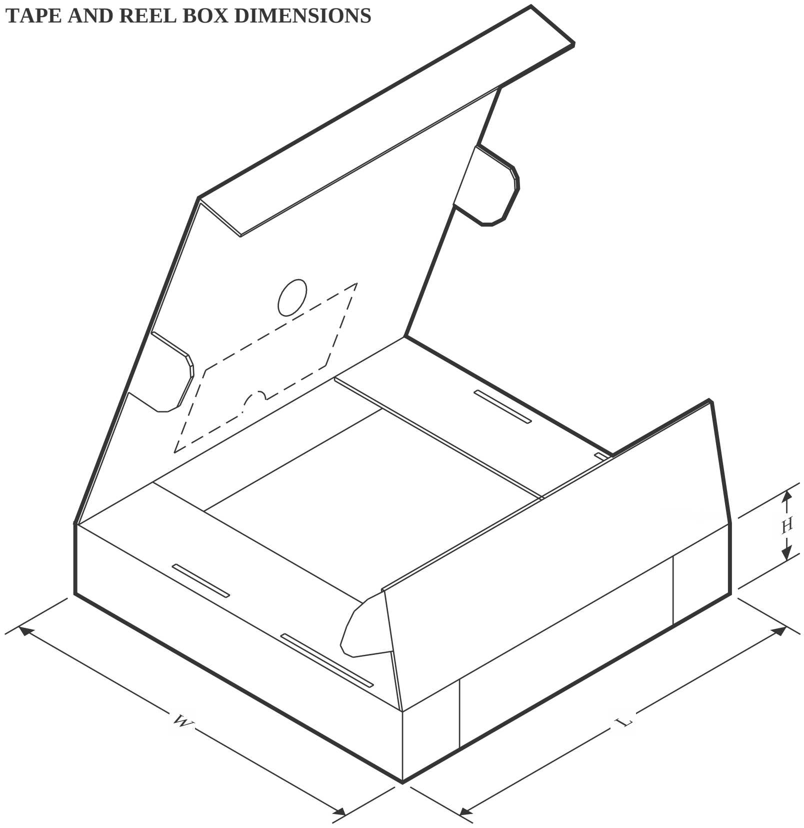
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
DAC43701DSGR	WSON	DSG	8	3000	180.0	8.4	2.3	2.3	1.15	4.0	8.0	Q2
DAC43701DSGRG4	WSON	DSG	8	3000	180.0	8.4	2.3	2.3	1.15	4.0	8.0	Q2
DAC43701DSGT	WSON	DSG	8	250	180.0	8.4	2.3	2.3	1.15	4.0	8.0	Q2
DAC53701DSGR	WSON	DSG	8	3000	180.0	8.4	2.3	2.3	1.15	4.0	8.0	Q2
DAC53701DSGT	WSON	DSG	8	250	180.0	8.4	2.3	2.3	1.15	4.0	8.0	Q2

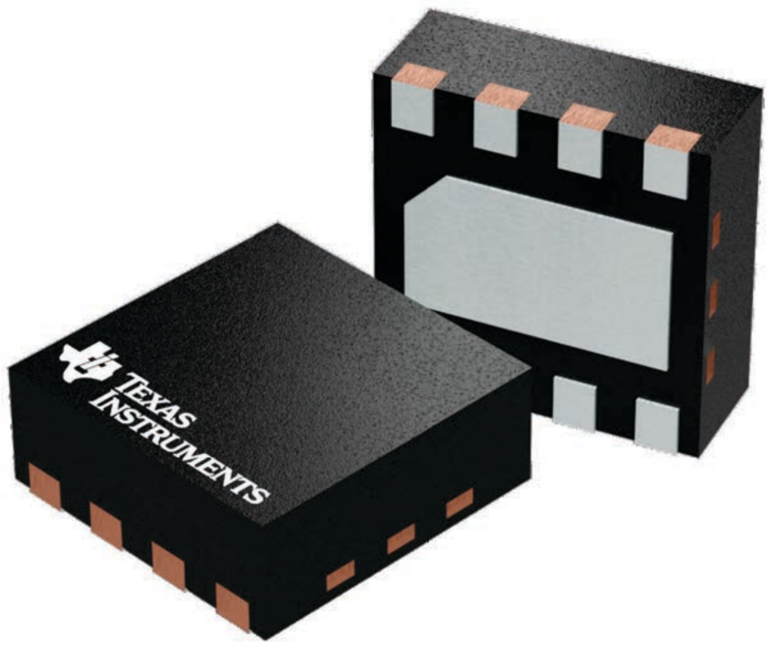
TAPE AND REEL BOX DIMENSIONS

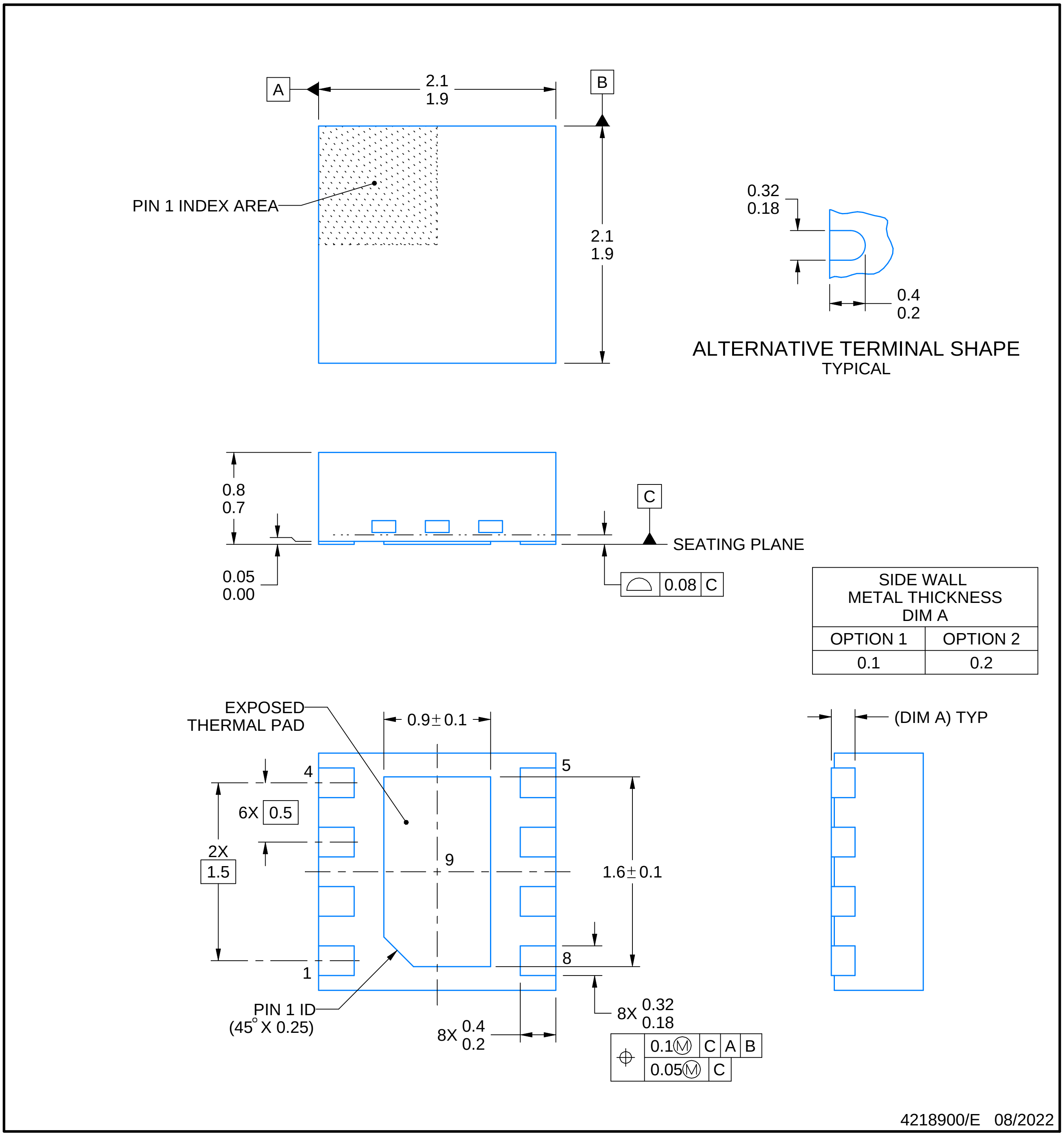


*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
DAC43701DSGR	WSON	DSG	8	3000	210.0	185.0	35.0
DAC43701DSGRG4	WSON	DSG	8	3000	210.0	185.0	35.0
DAC43701DSGT	WSON	DSG	8	250	210.0	185.0	35.0
DAC53701DSGR	WSON	DSG	8	3000	210.0	185.0	35.0
DAC53701DSGT	WSON	DSG	8	250	210.0	185.0	35.0

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

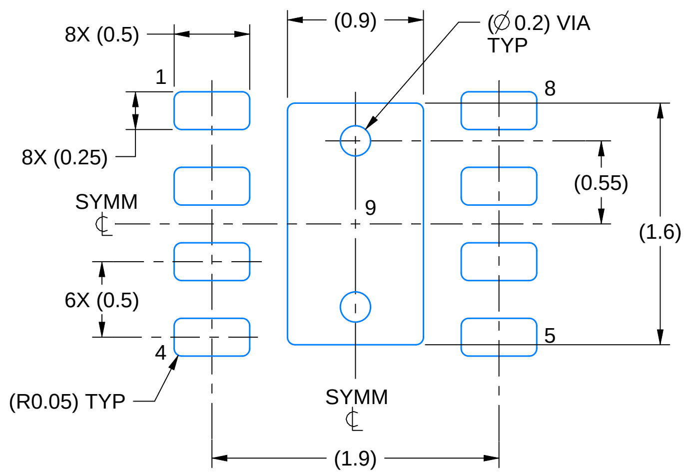




SIDE WALL METAL THICKNESS DIM A	
OPTION 1	OPTION 2
0.1	0.2

NOTES:

1. All linear dimensions are in millimeters. Any dimensions in parenthesis are for reference only. Dimensioning and tolerancing per ASME Y14.5M.
2. This drawing is subject to change without notice.
3. The package thermal pad must be soldered to the printed circuit board for thermal and mechanical performance.



LAND PATTERN EXAMPLE
SCALE:20X

NON SOLDER MASK
DEFINED
(PREFERRED)

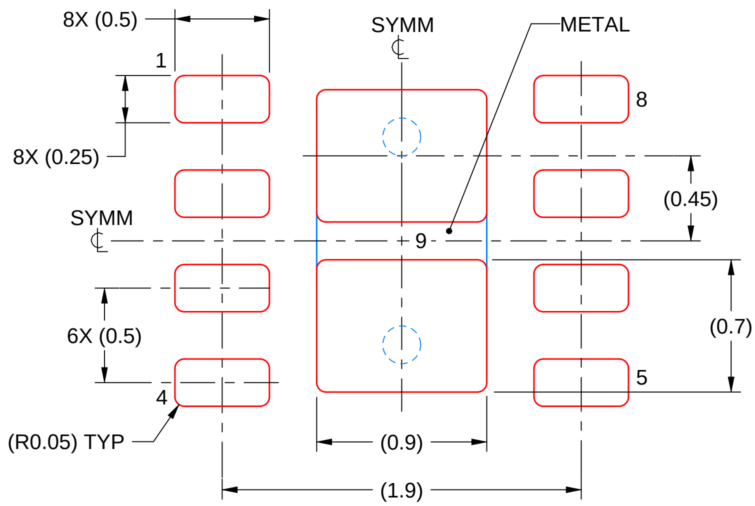
SOLDER MASK
DEFINED

SOLDER MASK DETAILS

4218900/E 08/2022

NOTES: (continued)

4. This package is designed to be soldered to a thermal pad on the board. For more information, see Texas Instruments literature number SLUA271 (www.ti.com/lit/slua271).
5. Vias are optional depending on application, refer to device data sheet. If any vias are implemented, refer to their locations shown on this view. It is recommended that vias under paste be filled, plugged or tented.



SOLDER PASTE EXAMPLE
BASED ON 0.125 mm THICK STENCIL

EXPOSED PAD 9:
87% PRINTED SOLDER COVERAGE BY AREA UNDER PACKAGE
SCALE:25X

4218900/E 08/2022

NOTES: (continued)

6. Laser cutting apertures with trapezoidal walls and rounded corners may offer better paste release. IPC-7525 may have alternate design recommendations.