Application Note

Signal Generators – Configuration and Applications with TAx5x1x/TAx5x1x-Q1

Lakshmi Narasimhan Badrinarayanan, Abin Dany Mathew, Diekoloreoluwa Ogundana

ABSTRACT

The TAx5x1x/TAx5x1x-Q1 family of stereo CODECs and DACs support a range of applications such as jack detection, brown-out limiter, voice activity detection and tone generation. This application note describes one of these features, the tone generator. The document describes the feature's operation, and guides end users on how to configure the different settings that are available for the feature, using examples. This application note applies to the following devices:

- TAC5212, TAC5211
- TAC5211, TAC5111
- TAD5212, TAD5112
- TAC5112-Q1
- TAD5212-Q1, TAD5112-Q1
- TAC5412-Q1, TAC5312-Q1
- TAC5311-Q1, TAC5301-Q1

Table of Contents

Trademarks

1 Introduction

Many end equipments make use of a single-frequency sinusoidal tone for applications like generating an “alert” beep when a certain event has occurred (for instance, a boot-up tone). Similar applications also use a “chirp” signal whose frequency varies over the duration of the signal. Pulses of ultrasonic chirps outside the audible band are used in Ultrasonic Activity Detection (UAD) applications. The TAx5x1x/TAx5x1x-Q1 family of stereo CODECs and DACs have an internal signal chain that can generate sustained single-frequency or chirp signals whose characteristics like amplitude, frequency etc. can be configured into the device.

Each channel on the playback path of the TAx5x1x/TAx5x51x-Q1 devices listed in the abstract follow the signal chain described in Figure 1-1. One of the processing blocks is a side-chain mixer, shown in Figure 1-2 (For more details on the mixer paths, refer to the TAC5x1x and TAC5x1x-Q1 Digital Channel Mixers - Configurations and Applications application note). This allows the signal from the ASI input on the DIN pin to be mixed with:

1. Data from the ADC-to-DAC Loopback Mixer.
2. Two in-built signal generators, Signal Generator-1 (SG1) and Signal Generator-2 (SG2).

SG1, which is a “beep generator” provides a fixed-frequency sinusoidal signal, while the SG2, which is a “chirp generator” provides sinusoidal signal whose frequency is swept over time. The output of these signal generators is then converted to analog and sent to the OUTxP/M pins to the external system.

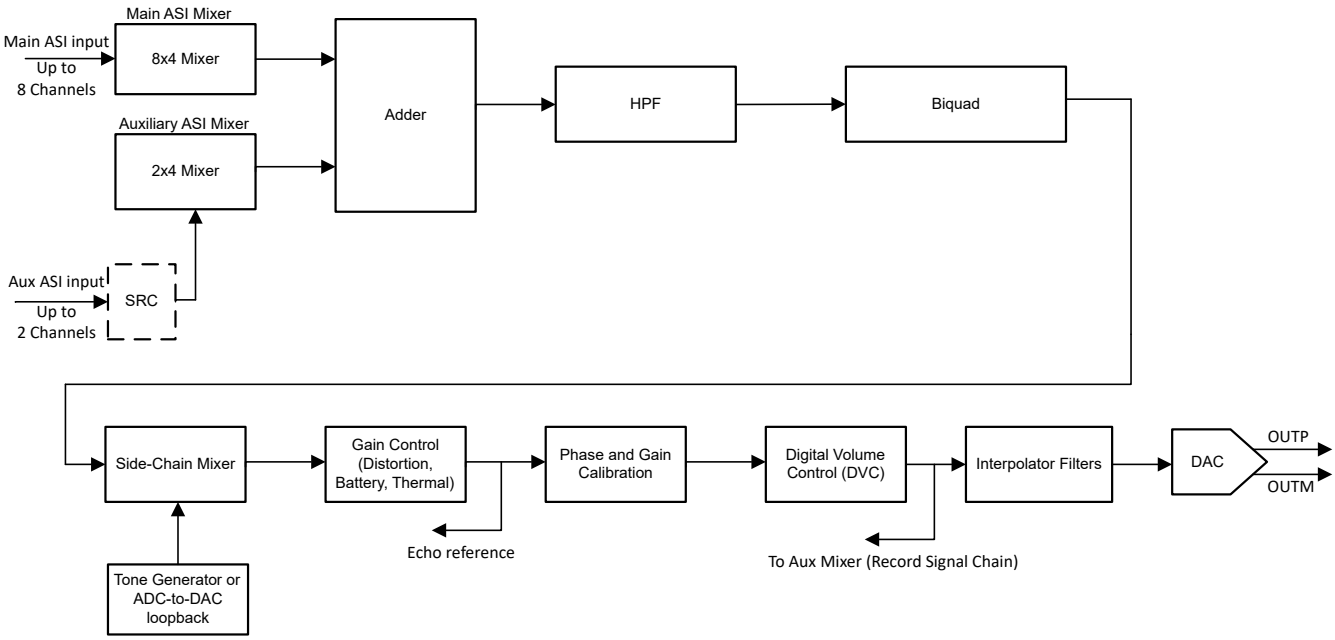

Figure 1-1. Signal Chain of the Playback path

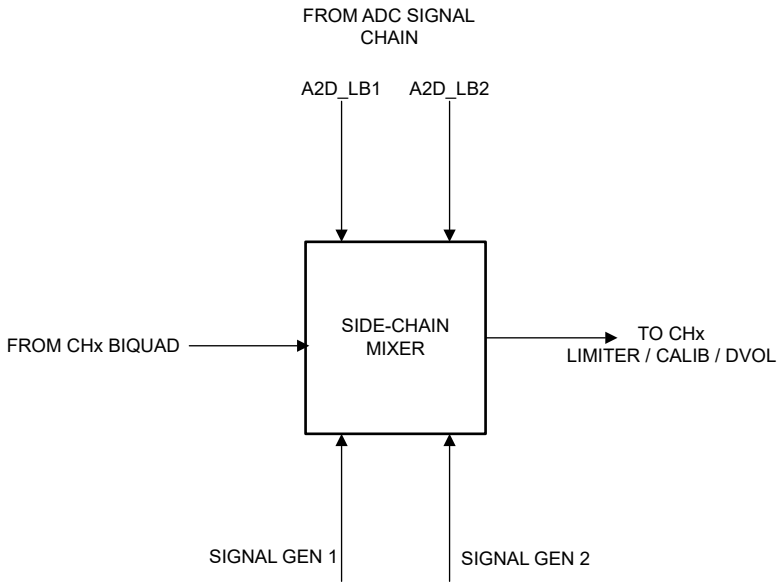

Figure 1-2. Side-Chain Mixer

Table 1-1 shows the register fields that are used to enable the required signal generator along with the side-chain mixer.

Table 1-1. Register Settings to Enable Signal Generator Path

Page	Register Bit	Bit Field Name	Value
0x1	R45[3]	DAC_SIGNAL_GENERATOR_1_ENABLE	0 – Disable Signal Generator 1 (Beep Generator) 1 – Enable Signal Generator 1 (Beep Generator)
0x1	R45[2]	DAC_SIGNAL_GENERATOR_2_ENABLE	0 – Disable Signal Generator 1 (Chirp Generator) 1 – Enable Signal Generator 1 (Chirp Generator)
0x1	R44[6]	EN_SIDE_CHAIN_MIXER	0 – Disable the Side-Chain Mixer 1 – Enable the Side-Chain Mixer

2 Signal Generator 1 (SG1) – “Beep” Generator

Signal Generator 1 (SG1), which is the “Beep” Generator provides a continuous sinusoidal signal of a single frequency, with a programmable output level (up to the full-scale level of 2Vrms differential). This section describes how to configure the key programmable parameters for the SG1 generator:

1. Amplitude, explained in Signal Generator 1 Amplitude and Output Channels.
2. Frequency, explained in Signal Generator 1 Frequency.

Beep Generator Example – Using TAC5212EVM-K and I²C Configuration includes an example configuration using I²C writes in PurePath™ Console 3.

The SG1 can be used for generating bursts of beep signals for short durations. For longer durations, the amplitude of the output signal changes at the rate of ~0.3dB/hr. If the end application needs a sustained tone for a long period of time (over hours or days), it is recommended to use Signal Generator 2 (SG2). Signal Generator 2 Start and Stop Frequency mentions how to configure the SG2 for sustained beep signal.

2.1 Signal Generator 1 Amplitude and Output Channels

SG1 is routed to the DAC outputs through the internal Side-Chain Mixer (refer to the TAC5x1x and TAC5x1x-Q1 Digital Channel Mixers - Configurations and Applications application note). The coefficients for this side-chain mixer (Table 2-1) configure the output amplitude for the signal. These coefficients are programmed as 16-bit numbers, with the coefficient being in 2.14 Q-format as shown in Figure 2-1, with the values ranging from -2 (0x8000) to 1.99994 (0x7FFF).

1. Multiply the number by 2^{14}.

2. Round the number to the nearest integer. If the number is negative, take its twos-complement.
   a. If the desired value is 1.0, the register value needs to be 0x4000.
   b. Similarly, if the desired value is -1, the register value needs to be 0xC000 (twos-complement).
   c. If the desired value is 1.235, the register value needs to be 0x4F0A.
   d. If the desired value needs to be -1.275, the register value needs to be 0xAE66 (twos-complement).

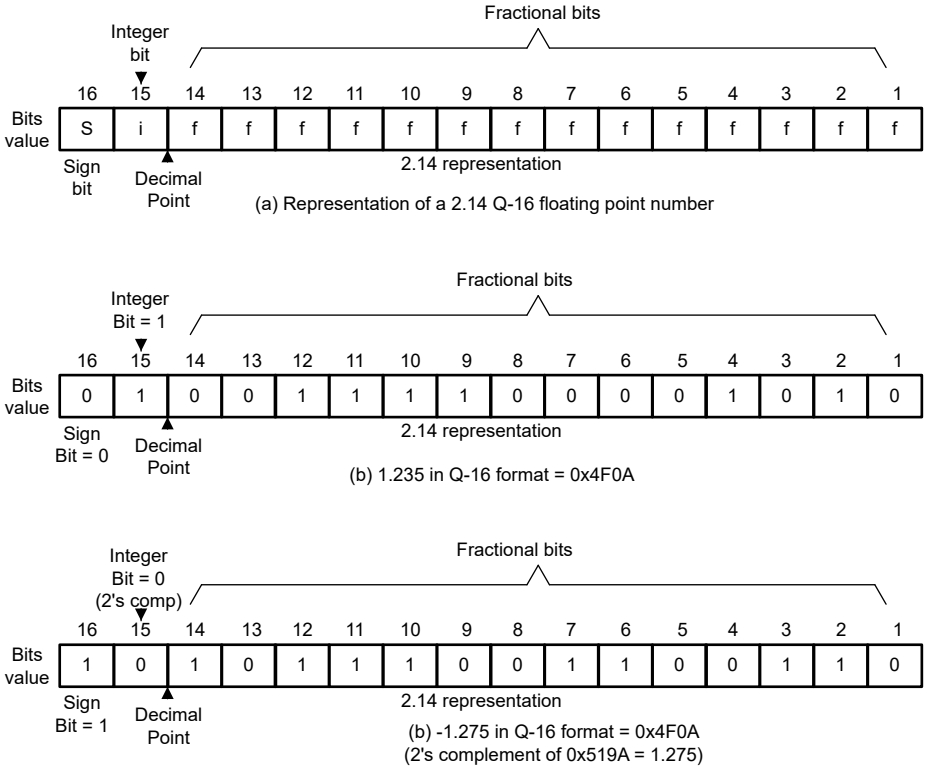

Figure 2-1. Floating Point Numbers in Q-16 Format

The routing of the Side-Chain Mixer’s outputs to the different DAC channels depends on the output configuration (differential/single-ended). This is also described in Table 2-1.

Table 2-1. Programmable Register Map for SG1 Amplitude and Output Channels

Page	Register	Description	Analog Output
0x11	0x68	Side-Chain DAC MIXER, Signal Generator CH1 to RDAC coefficient byte[15:8]	Differential / Mono Single-Ended: OUT2P (OUT2M) Signal Amplitude Stereo Single-Ended: OUT1M Signal Amplitude
0x11	0x69	Side-Chain DAC MIXER, Signal Generator CH1 to RDAC coefficient byte[7:0]	
0x11	0x6A	Side-Chain DAC MIXER, Signal Generator CH1 to LDAC coefficient byte[15:8]	Differential / Mono Single-Ended: OUT1P (OUT1M) Signal Amplitude Stereo Single-Ended: OUT1P Signal Amplitude
0x11	0x6B	Side-Chain DAC MIXER, Signal Generator CH1 to LDAC coefficient byte[7:0]	
0x11	0x6C	Side-Chain DAC MIXER, Signal Generator CH1 to RDAC2 coefficient byte[15:8]	Differential / Mono Single-Ended: To **ADC Loopback Mixer** Stereo Single-Ended: OUT2M Signal Amplitude
0x11	0x6D	Side-Chain DAC MIXER, Signal Generator CH1 to RDAC2 coefficient byte[7:0]	
0x11	0x6E	Side-Chain DAC MIXER, Signal Generator CH1 to LDAC2 coefficient byte[15:8]	Differential / Mono Single-Ended: To **ADC Loopback Mixer** Stereo Single-Ended: OUT2P Signal Amplitude
0x11	0x6F	Side-Chain DAC MIXER, Signal Generator CH1 to LDAC2 coefficient byte[7:0]	

2.2 Signal Generator 1 Frequency

The “beep” signal generated by SG1 is programmed into a series of registers per the following equations:

$$x = \sin\left(2\pi\left(\frac{f_{sig}}{f_s}\right)\right) \quad (1)$$

$$y = \cos\left(2\pi\left(\frac{f_{sig}}{f_s}\right)\right) \quad (2)$$

Where f_{sig} is the frequency of the desired signal, and f_s is the sampling rate of operation.

The maximum value of f_{sig} is $f_s/2$.

The values of x and y range between -1.0 and 1.0. To program this into the device:

1. Multiply the coefficient by 2^{63} (= X or Y).
2. Round X or Y to the nearest integer.
3. Convert this number into a signed 64-bit integer:
   a. If the number is positive, then program the number as is, with the MSB bit = 0.
   b. If the number is negative, then program the twos-complement of the negative number. Here, MSB bit = 1.
4. The 64 bits are split into two 32-bit words and programmed as follows into the locations in Table 2-2.
   a. The MSB 32 bits are programmed as is.
   b. The LSB 32 bits are right-shifted by 1 bit and programmed.
5. For example, to configure a beep frequency of 1kHz with a sampling rate of 48kHz:
   a. The value of X for register configuration will be 0x10B5150F1C1C2300.
   b. The value of Y for register configuration will be 0x7EE7AA4BBC5E1C00.

Table 2-2. Registers to Configure Beep Frequency

Page	Register Address	Register Description	Value
0x12	0x20	Programmable DAC BEEP GEN cos(x) coefficient byte[31:24]	{0, X[31:25]}
0x12	0x21	Programmable DAC BEEP GEN cos (x) coefficient byte[23:16]	X[24:17]
0x12	0x22	Programmable DAC BEEP GEN cos(x) coefficient byte[15:8]	X[16:9]
0x12	0x23	Programmable DAC BEEP GEN cos(x) coefficient byte[7:0]	X[8:1]
0x12	0x24	Programmable DAC BEEP GEN cos(x) coefficient byte[63:56]	X[63:56]
0x12	0x25	Programmable DAC BEEP GEN cos(x) coefficient byte[55:48]	X[55:48]
0x12	0x26	Programmable DAC BEEP GEN cos(x) coefficient byte[47:40]	X[47:40]
0x12	0x27	Programmable DAC BEEP GEN cos(x) coefficient byte[39:32]	X[39:32]
0x12	0x28	Programmable DAC BEEP GEN sin(x) coefficient byte[31:24]	{0, Y[31:25]}
0x12	0x29	Programmable DAC BEEP GEN sin(x) coefficient byte[23:16]	Y[24:17]
0x12	0x2A	Programmable DAC BEEP GEN sin(x) coefficient byte[15:8]	Y[16:9]
0x12	0x2B	Programmable DAC BEEP GEN sin(x) coefficient byte[7:0]	Y[8:1]
0x12	0x2C	Programmable DAC BEEP GEN sin(x) coefficient byte[63:56]	Y[63:56]

Table 2-2. Registers to Configure Beep Frequency (continued)

Page	Register Address	Register Description	Value
0x12	0x2D	Programmable DAC BEEP GEN sin(x) coefficient byte[55:48]	Y[55:48]
0x12	0x2E	Programmable DAC BEEP GEN sin(x) coefficient byte[47:40]	Y[47:40]
0x12	0x2F	Programmable DAC BEEP GEN sin(x) coefficient byte[39:32]	Y[39:32]

2.3 Beep Generator Example – Using TAC5212EVM-K and I²C Configuration

The code example below demonstrates how SG1 can be used to generate a 880Hz sinusoidal signal. The captured signal is shown in Figure 2-2.

```
w a0 00 00 #Page 0
w a0 01 01 #SW Reset
d 01
w a0 00 00 #Page 0
w a0 02 09 #Exit Sleep Mode with DREG and VREF Enabled
w a0 1a 30 #TDM protocol with 32-bit word length
w a0 64 20 #DAC Channel 1 configured for differential output with 0.6*Vref as common mode
w a0 65 20 #DAC OUT1P configured for line out driver and audio bandwidth
w a0 66 20 #DAC OUT1M configured for line out driver and audio bandwidth
w a0 6b 20 #DAC Channel 2 configured for differential output with 0.6*Vref as common mode
w a0 6c 20 #DAC OUT2P configured for line out driver and audio bandwidth
w a0 6d 20 #DAC OUT2M configured for line out driver and audio bandwidth
w a0 26 01 #RX Offset = 1
w a0 28 00 #RX CH1 to DAC CH1 disabled
w a0 29 01 #RX CH2 to DAC CH2 disabled

w a0 00 01 #Page 1
w a0 2d 08 #Enable the beep (SG1)
w a0 2c 40 #Enable side-chain mixer

#Tone generator
#SG1 = 880Hz sine tone
w a0 00 12 #Page 18
w a0 24 7F 26 D6 BA #cos(x)[63:32]
w a0 20 43 91 78 00 #cos(x)[31:0]
w a0 2c 0E B6 42 B2 #sin(x)[63:32]
w a0 28 79 A5 3E 40 #sin(x)[31:0]

w a0 00 11 #Page 17
#DAC output OUT1 = 0.5*SG1
#DAC output OUT2 = -0.8*SG1
w a0 68 cc cd 20 00 #Side-Chain Mixer for SG1

w a0 00 00 #Page 0
w a0 76 0c #enable 2 DAC channels
w a0 78 40 #enable DAC
```

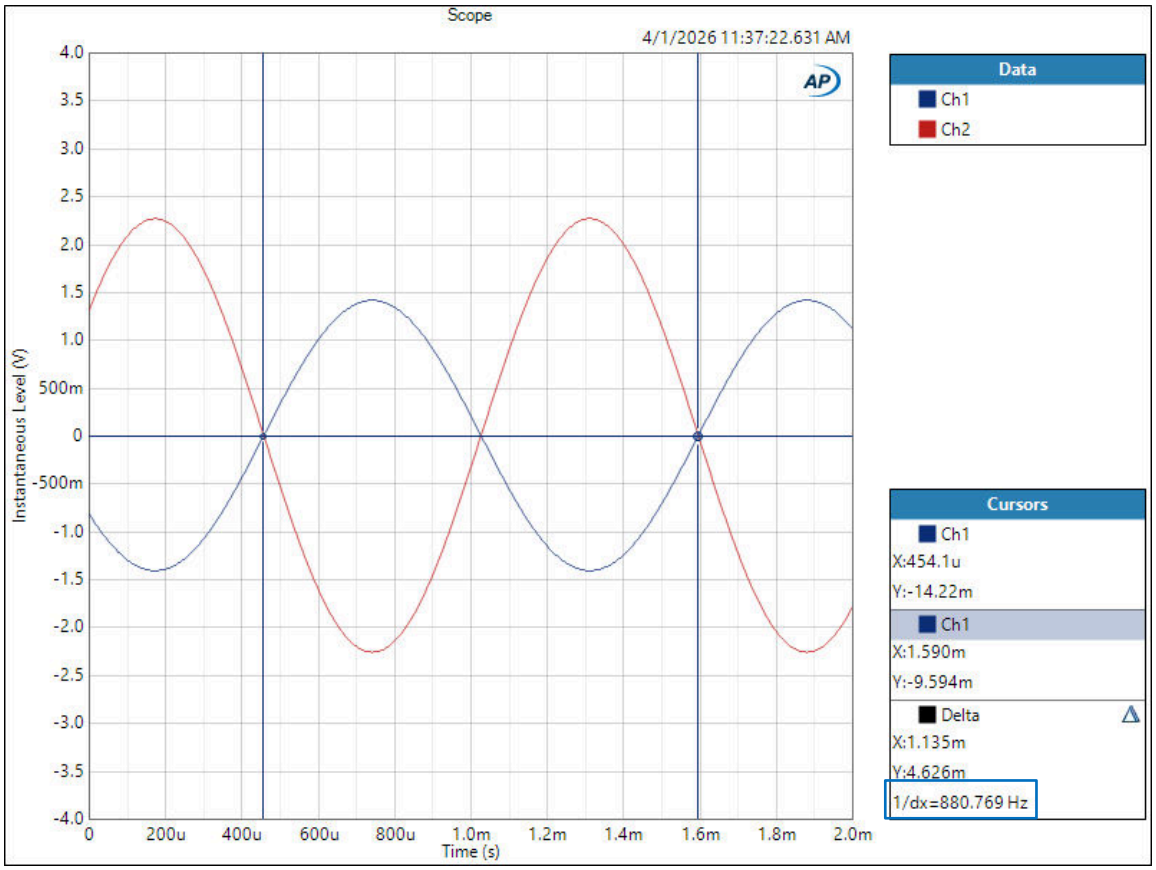

Figure 2-2. Output of the Beep Generator Example

3 Signal Generator 2 (SG2) – "Chirp" Generator

Signal Generator 2 (SG2), which is the "Chirp" Generator provides a sinusoidal output whose frequency is swept over time. This section describes how to configure the key programmable parameters for the SG2 generator:

1. Amplitude, explained in Signal Generator 2 Amplitude and Output Channels.
2. Frequency Sweep, explained in Signal Generator 2 Start and Stop Frequency.
3. ADSR parameters, explained in ADSR Envelope Parameters.
4. Output Modes, described in Signal Generator 2 Output Modes. This section includes example codes for each output mode.

3.1 Signal Generator 2 Amplitude and Output Channels

Like SG1, the signal from SG2 is also sent to the DAC signal chain through the Side-Chain Mixer. The coefficients for the same are programmed in the registers listed in Table 3-1. The computation of these register values is the same as that for SG1, described in Signal Generator 1 (SG1) – "Beep" Generator.

Table 3-1. Programmable Register Map for SG2 Amplitude and Output Channels

Page	Register	Description	Comments
0x11	0x70	SC DAC MIXER, Signal Generator CH2 to RDAC coefficient byte[15:8]	Differential / Mono Single-Ended: OUT2P (OUT2M) Signal Amplitude Stereo Single-Ended: OUT1M Signal Amplitude
0x11	0x71	SC DAC MIXER, Signal Generator CH2 to RDAC coefficient byte[7:0]	
0x11	0x72	SC DAC MIXER, Signal Generator CH2 to LDAC coefficient byte[15:8]	Differential / Mono Single-Ended: OUT1P (OUT1M) Signal Amplitude Stereo Single-Ended: OUT1P Signal Amplitude
0x11	0x73	SC DAC MIXER, Signal Generator CH2 to LDAC coefficient byte[7:0]	

Table 3-1. Programmable Register Map for SG2 Amplitude and Output Channels (continued)

Page	Register	Description	Comments
0x11	0x74	SC DAC MIXER, Signal Generator CH2 to RDAC2 coefficient byte[15:8]	Differential / Mono Single-Ended: To **ADC Loopback Mixer** Stereo Single-Ended: OUT2M Signal Amplitude
0x11	0x75	SC DAC MIXER, Signal Generator CH2 to RDAC2 coefficient byte[7:0]	
0x11	0x76	SC DAC MIXER, Signal Generator CH2 to LDAC2 coefficient byte[15:8]	Differential / Mono Single-Ended: To **ADC Loopback Mixer** Stereo Single-Ended: OUT2P Signal Amplitude
0x11	0x77	SC DAC MIXER, Signal Generator CH2 to LDAC2 coefficient byte[7:0]	

3.2 Signal Generator 2 Start and Stop Frequency

Since the SG2 generates a chirp signal, the user would need to configure a start frequency f_{start} and a frequency increment Δf for each sample. The resulting output is a signal whose frequency is swept from f_{start} to (f_{start} + N_{chirp}Δf), N_{chirp} being the number of samples in a single chirp pulse.

The chirp signal stays at f_{start} until the signal is ramped up and settles to the Sustain Level. Then the signal frequency changes at a rate of Δf per sample. When the signal ramps down, it retains the last updated frequency.

These two frequencies are programmed through coefficients x and y, described in the equations below:

$$x = 2\pi\frac{f_{start}}{f_S} \qquad (3)$$
$$y = 2\pi\frac{\Delta f}{f_S}.$$

with f_s being the sampling rate.

The chirp frequency can range from 0 to $f_s/2$. Hence, the following equations need to be kept in mind when programming the chirp parameters:

$$N_{chirp} = t_{chirp} * f_s \qquad (4)$$

Where $t_{chirp}\ (chirp\ \ duration) = (Sustain\ \ Time - Decay\ \ Time - Attack\ \ Time)$

Hence, $f_{start} + \left(N_{chirp} * \Delta f\right) \leq \frac{f_S}{2}$ **.**

"Sustain Time" is the pulse duration, described in detail in **Sustain/Restart Timers** .

It is not recommended to let the signal frequency at any point exceed beyond $f_s/2$, as this results in aliasing. With sampling rates higher than 48kHz,the SG2 generator can be used as an Ultrasonic Activity Generator (UAG), generating chirp pulses outside of the 20Hz-20kHz audible band.

To generate a sustained single-tone signal, users can configure the SG2 to operate in Manual Mode (refer Section 3.4.1). The f_{start} can be programmed to the desired frequency, with the Δf programmed to 0Hz.

The values of x and y are then written into the registers described in Table 3-2, by:

1. Multiplying each value by 2^{28} (= X or Y).
2. Rounding off X or Y to the nearest integer value.
3. Convert this to 32-bit hexadecimal and program into the registers.

Table 3-2. SG2 Frequency Parameters

Page	Register	Signal Generator Start/Δ Frequency	Value
0x17	0x7C	Start Frequency = 16kHz (0x2182A470)	X[31:24]
0x17	0x7D		X[23:16]
0x17	0x7E		X[15:8]
0x17	0x7F		X[7:0]

Table 3-2. SG2 Frequency Parameters (continued)

Page	Register	Signal Generator Start/Δ Frequency	Value
0x18	0x08	Δ Frequency = 0Hz (0x00000000)	Y[31:24]
0x18	0x09		Y[23:16]
0x18	0x0A		Y[15:8]
0x18	0x0B		Y[7:0]

3.3 ADSR Envelope Parameters

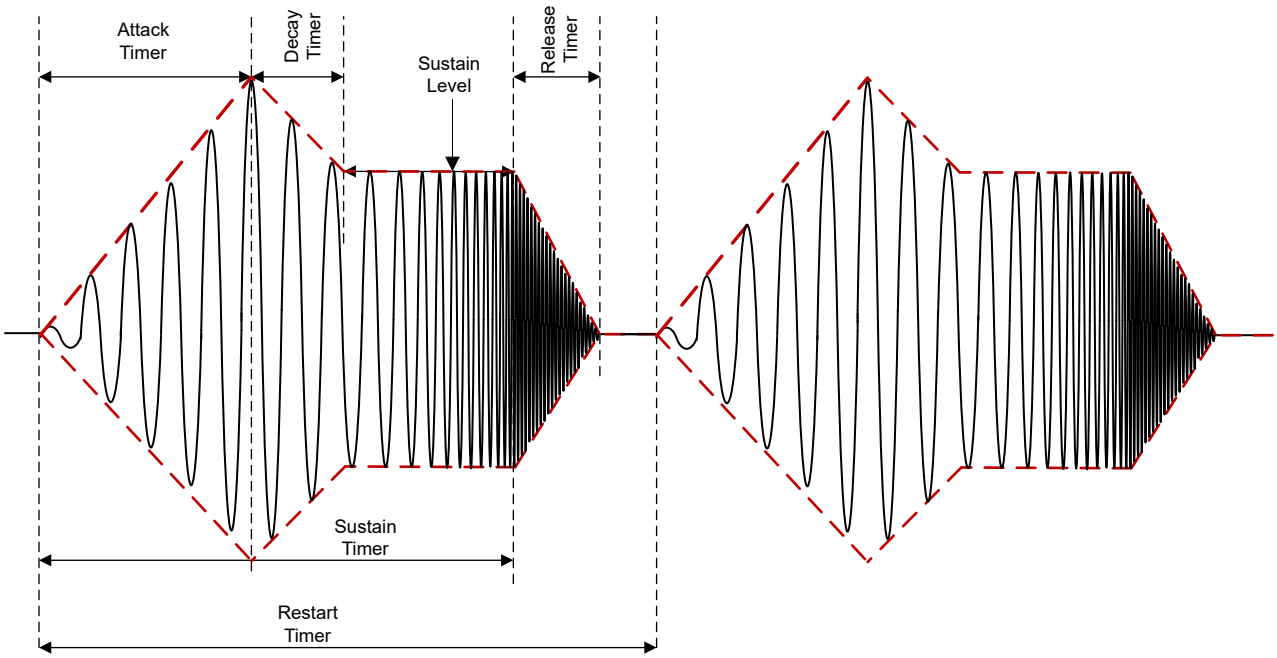

Figure 3-1. Chirp Signal following ADSR Envelope

Figure 3-1 shows an example of a chirp signal, which starts off at a lower frequency, and the frequency increases over time and stops until the pattern repeats itself again. As the figure indicates, the chirp is not just a uniform sine tone with varying frequency, rather it follows a signal envelope, which is characterized by 4 timers (ADSR): **A**ttack, **D**ecay, **S**ustain/Restart, and **R**elease.

3.3.1 Attack/Decay Timers

The attack and decay timers define the ramp-up profile of the chirp signal, until it reaches a steady-state "Sustain Level" (refer **Sustain Level**).

The attack timer is defined by the following equations:

$$t_{attack}\ \left(in\ \ ms\right)\ =\ 1000 * \frac{1}{x} * \frac{1}{f_S\ \ (in\ \ Hz)} \tag{5}$$

For a timer value of t_{attack}, the use can then compute the "attack slope" x, which is a floating positive number < 1.

$$x\ =\ 1000 * \frac{1}{t_{attack}\ \ (in\ \ ms)} * \frac{1}{f_S\ \ (in\ \ Hz)} \tag{6}$$

To program x into the registers listed in Table 3-3:

1. Multiply x with 2^{30} (= X).
2. Round X to the nearest integer value.

Similarly, the decay timer is defined by the following equation:

$$t_{decay}\ \left(in\ \ ms\right)\ =\ 1000*\frac{2*(1\ -\ SustainLevel)}{(-y)}*\frac{1}{f_S\,(in\ \ Hz)} \quad (7)$$

Since it is a ramp-down instead of ramp-up, the decay slope y is a negative floating number.

To program y into the registers listed in Table 3-3:

1. Multiply y with 2^{30} (= Y).
2. Round Y to the nearest integer value.
3. Since Y is a negative number, take its twos-complement value to program the register.
4. If a decay profile is not required in the chirp waveform, program the register to 0x00000000.
5. If the decay timer is set to 0x00000000, the sustain level is 1 irrespective of the value of the register.

Table 3-3. Attack/Decay Timers

Page	Register	Default Attack/Decay time at 48kHz sampling rate	Value
0x1C	0x58	Attack Time = 5 ms (0x0044523F)	X[31:24]
0x1C	0x59		X[23:16]
0x1C	0x5A		X[15:8]
0x1C	0x5B		X[7:0]
0x1C	0x60	Decay Time = 0 ms (0x00000000)	Y[31:24]
0x1C	0x61		Y[23:16]
0x1C	0x62		Y[15:8]
0x1C	0x63		Y[7:0]

3.3.2 Release Timer

The release timer defines the ramp-down profile of the chirp signal from the sustain level. It is defined by the following equation.

$$t_{release}\ \left(in\ \ ms\right)\ =\ 1000*\frac{SustainLevel}{(-z)}*\frac{1}{f_S\ \ (in\ \ Hz)} \quad (8)$$

Since it is a ramp-down instead of ramp-up, the release slope "x" is a negative floating number.

To program z into the registers listed in Table 3-4:

1. Multiply z with 2^{30} (= Z).
2. Round Z to the nearest integer value.
3. Convert Z to twos-complement.

Table 3-4. Release Timer

Page	Register	Default Release time at 48kHz sampling rate	Value
0x1C	0x5C	Release Time = 5 ms (0xFFBBADC1)	Z[31:24]
0x1C	0x5D		Z[23:16]
0x1C	0x5E		Z[15:8]
0x1C	0x5F		Z[7:0]

3.3.3 Powerup Delay Timer

The powerup delay timer is the time by which the user can delay the beginning of the first chirp cycle, from when the playback path is powered up by setting the DAC_PDz bit (B0_P0_R120[6]). For instance, a powerup delay of 500ms would mean that the first chirp cycle starts 500ms after powering up the playback path. **The minimum powerup delay that can be programmed is 127ms.**

The powerup delay follows the equation:

$$a_{pwrup} = t_{pwrup} - 0.007 \quad (9)$$

with a_{pwrup} and t_{pwrup} both in seconds.

To program the powerup delay into the registers listed in Table 3-5:

1. Multiply the computed value of the a_{pwrup} (in seconds) with the sampling rate (in Hz) (= A_{PWRUP}).
2. Round off A_{PWRUP} to the nearest integer.

Table 3-5. Powerup Delay Timer

Page	Register	Default Powerup Delay time at 48kHz sampling rate	Value
0x17	0x74	Powerup Delay = 207 ms (0x00002580)	A_{PWRUP} [31:24]
0x17	0x75		A_{PWRUP} [23:16]
0x17	0x76		A_{PWRUP} [15:8]
0x17	0x77		A_{PWRUP} [7:0]

3.3.4 Sustain/Restart Timers

Sustain timer is the length of a single chirp pulse. Restart timer is the time interval between two subsequent pulses. The effect of both times on the chirp signal is represented in Figure 3-1.

The sustain timer value needs to be less than or equal to the restart timer value.

To program the two timers into the registers described in Table 3-6:

1. Multiply the value of the time (in seconds) with the sampling rate (in Hz) (= $T_{SUSTAIN}$ or $T_{RESTART}$).
2. Round off $T_{SUSTAIN}$ or $T_{RESTART}$ to the nearest integer.

Table 3-6. Sustain/Restart Timers

Page	Register	Default Attack/Decay time at 48kHz sampling rate	Value
0x1C	0x54	Sustain Time = 20 ms (0x000003C0)	$T_{SUSTAIN}$ [31:24]
0x1C	0x55		$T_{SUSTAIN}$ [23:16]
0x1C	0x56		$T_{SUSTAIN}$ [15:8]
0x1C	0x57		$T_{SUSTAIN}$ [7:0]
0x1C	0x50	Restart Time = 200 ms (0x00002580)	$T_{RESTART}$ [31:24]
0x1C	0x51		$T_{RESTART}$ [15:8]
0x1C	0x52		$T_{RESTART}$ [15:8]
0x1C	0x53		$T_{RESTART}$ [7:0]

3.3.5 Sustain Level

Sustain level refers to the steady-state signal level after decay time. The signal remains at this level until the release timer kicks in.

The value of the sustain level **k** can vary from 0 to 1. To program this into the register mentioned in Table 3-7:

1. Multiply k by 2^{30} (= K).
2. Round K to the nearest integer value.

Table 3-7. Sustain Level

Page	Register	Default Sustain Level	Value

Table 3-7. Sustain Level (continued)

0x1C	0x64	Sustain Level = 1 [0 dB] (0x40000000)	K[31:24]
0x1C	0x65		K[23:16]
0x1C	0x66		K[15:8]
0x1C	0x67		K[7:0]

3.4 Signal Generator 2 Output Modes

Based on the desired output when the ADSR envelope is configured, the SG2 can be configured to operate in:

1. Manual Mode, described in Manual Mode.
2. Continuous-Pulse Mode, described in Continuous Pulse Mode.
3. One-Shot Mode, described in One Shot Mode.

3.4.1 Manual Mode

In manual mode, the device generates a single pulse whenever the user enables the ADSR note bit. Hence, the typical sequence followed is:

1. Set ADSR note to 0x0000000 at the beginning of the configuration to initiate an acknowledge.
2. Once DAC is powered up, set ADSR note to 0x00000001 to enable a chirp pulse.
3. The signal is running until the ADSR note is set again to 0x00000000.

Since the pulse start/stop is manual, sustain/restart timer and powerup delay timer configurations are not applicable in manual mode. These registers need to be set to 0xFFFFFFFF.

If the signal is kept running until the frequency exceeds $f_s/2$, aliasing will happen as mentioned in Signal Generator 2 Start and Stop Frequency.

3.4.1.1 Manual Mode Example I²C Configuration using TAC5212EVM-K

The code below is used to manually start and stop the generation of a chirp output. When the ADSR note is set to 0x00000001, the chirp ramps up per attack and decay timers and continues until the ADSR note is set to 0x00000000. When this happens, the signal ramps down per the release timer. Figure 3-2 shows the "attack" profile of the signal, and Figure 3-3 shows the "release" profile of the signal.

```
w a0 00 00 #Page 0
w a0 01 01 #SW Reset
d 10
w a0 00 00 #Page 0
w a0 02 09 #Exit Sleep Mode with DREG and VREF Enabled
w a0 1a 30 #TDM protocol with 32-bit word length
w a0 64 20 #DAC Channel 1 configured for differential output with 0.6*Vref as common mode
w a0 65 20 #DAC OUT1P configured for line out driver and audio bandwidth
w a0 66 20 #DAC OUT1M configured for line out driver and audio bandwidth
w a0 6b 20 #DAC Channel 2 configured for differential output with 0.6*Vref as common mode
w a0 6c 20 #DAC OUT2P configured for line out driver and audio bandwidth
w a0 6d 20 #DAC OUT2M configured for line out driver and audio bandwidth
w a0 26 01 #RX Offset = 1
w a0 28 00 #RX CH1 to DAC CH1 disabled
w a0 29 00 #RX CH1 to DAC CH1 disabled

w a0 00 01 #Page 1
w a0 2d 04 #Enable the chirp (SG2)
w a0 2c 40 #Enable side-chain mixer

w a0 00 17 #Page 23
w a0 7c 01 26 e4 0d #Chirp start frequency of 550 Hz

w a0 00 18 #Page 24
w a0 08 00 00 45 86 #Chirp delta frequency of 0.507 Hz

w a0 00 1c #Page 28
w a0 40 00 00 00 00 #ADSR_Note
w a0 50 ff ff ff ff #Restart Timer = NA
```

```
w a0 54 ff ff ff ff #Sustain Timer = NA
w a0 58 00 01 08 cb #Attack Timer = 330ms
w a0 5c ff ff 8b 7e #Release Timer = 750ms (w.r.t 0dB sustain level)
w a0 64 40 00 00 00 #Sustain Level = 1 (0dB)

w a0 00 17
w a0 74 ff ff ff ff #Powerup Delay = NA

w a0 00 11 #Page 17
w a0 70 40 00 40 00 #Side-Chain Mixer Gain for SG2 = 1

w a0 00 00 #Page 0
w a0 76 0c #enable 2 DAC channels
w a0 78 40 #enable DAC

b # Breakpoint (Writing 0x00000001 into ADSR_Note will start the chirp)
w a0 00 1c #Page 28
w a0 40 00 00 00 01
w a0 00 00

b # Breakpoint (Writing 0x00000000 into ADSR_Note will stop the chirp)
w a0 00 1c #Page 28
w a0 40 00 00 00 00
w a0 00 00
```

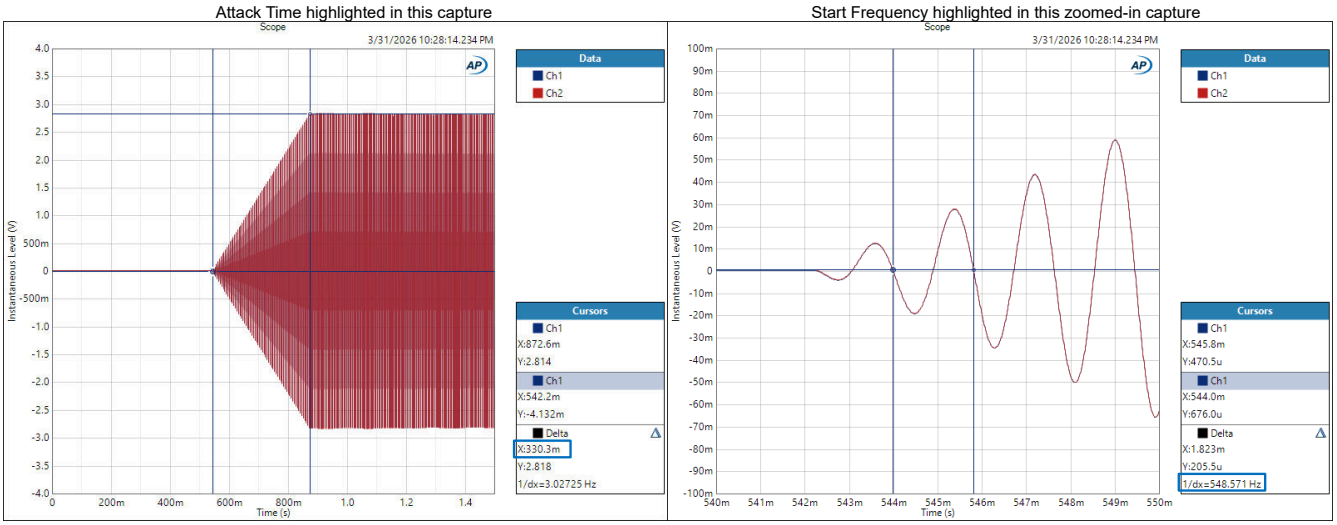

Figure 3-2. Start of manual chirp

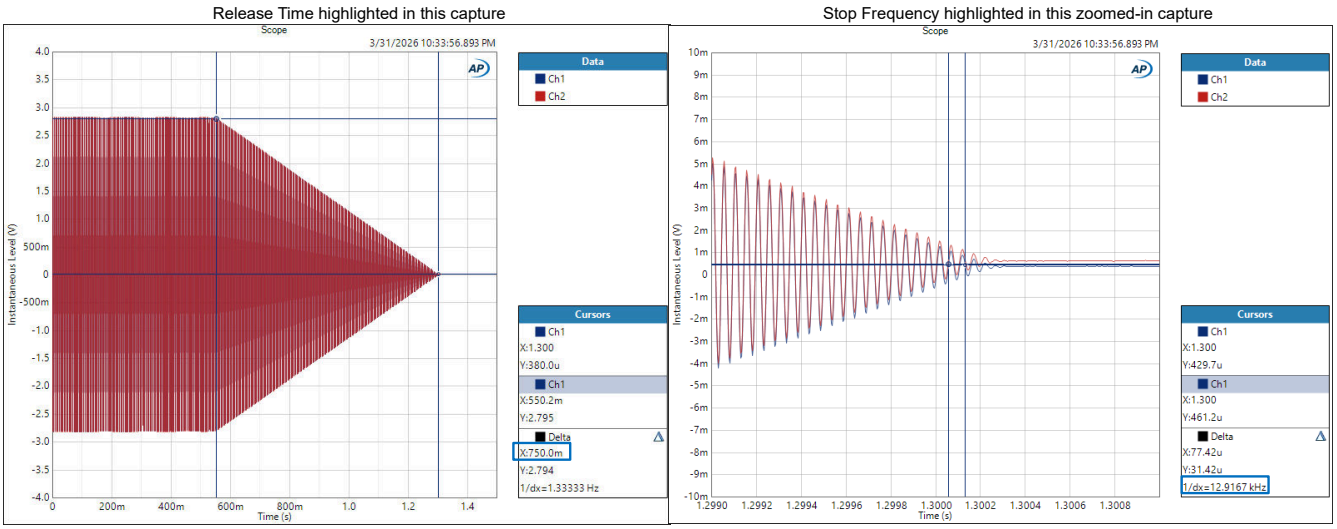

Figure 3-3. Stop of manual chirp

3.4.2 Continuous Pulse Mode

In continuous-pulse mode, the device generates a recurring series of chirp pulses at regular intervals. Hence, the sustain timer and restart timer configurations apply in continuous-pulse mode.

The equations in Signal Generator 2 Start and Stop Frequency are to be accounted for with t_{chirp} = (Sustain Timer) when programming the frequencies to avoid aliasing.

While the ADSR note need not be set to 0x00000001 at the end of the configuration to enable the chirp, it needs to be set to 0xFFFFFFFF during configuration to initiate the ADSR configuration mode.

3.4.2.1 Continuous-Pulse Mode Example I²C Configuration using TAC5212EVM-K

The code below is used to generate a continuous periodic stream of chirp pulses. The ADSR note is set to 0xFFFFFFFF once in the configuration and is kept at this value for continuous generation of chirp pulses.

In this example:

1. t_{attack} = 200ms, t_{decay} = 100ms and $t_{release}$ = 200ms.
2. $t_{sustain}$ = 1sec = 1000ms.
3. f_{start} = 100Hz, Δf = 0.295Hz, f_s = 48kHz.
4. Hence, per the equation in Signal Generator 2 Start and Stop Frequency, the stop frequency f_{stop} = 100 + (0.295 * 48000 * (1000 – 200 – 100)) ~= 10kHz.
5. $t_{restart}$ = 2sec = 2000ms.

Figure 3-2 shows the attack, decay and release timer measurements, while Figure 3-3 shows the sustain and restart timer measurements. Figure 3-6 shows that the signal frequency doesn't change until the decay timer is reached. Figure 3-7 shows that the signal remains at f_{stop} throughout the release phase.

```
w a0 00 00 #Page 0
w a0 01 01 #SW Reset
d 10
w a0 00 00 #Page 0
w a0 02 09 #Exit Sleep Mode with DREG and VREF Enabled
w a0 1a 30 #TDM protocol with 32-bit word length
w a0 64 20 #DAC Channel 1 configured for differential output with 0.6*Vref as common mode
w a0 65 20 #DAC OUT1P configured for line out driver and audio bandwidth
w a0 66 20 #DAC OUT1M configured for line out driver and audio bandwidth
w a0 6b 20 #DAC Channel 2 configured for differential output with 0.6*Vref as common mode
w a0 6c 20 #DAC OUT2P configured for line out driver and audio bandwidth
w a0 6d 20 #DAC OUT2M configured for line out driver and audio bandwidth
w a0 26 01 #RX Offset = 1
w a0 28 00 #RX CH1 to DAC CH1 disabled

w a0 00 01 #Page 1
w a0 2d 04 #Enable the chirp (SG2)
w a0 2c 40 #Enable side-chain mixer

w a0 00 17 #Page 23
w a0 7c 00 35 9d d4 #Chirp start frequency of 100 Hz

w a0 00 18 #Page 24
w a0 08 00 00 28 71 #Chirp delta frequency of 0.295 Hz

w a0 00 1c #Page 28
w a0 40 ff ff ff ff #ADSR_Note
w a0 50 00 01 77 00 #Restart Timer = 2sec
w a0 54 00 00 bb 80 #Sustain Timer = 1sec
w a0 58 00 01 b4 e8 #Attack Timer = 200ms
w a0 5c ff ff 25 07 #Release Timer = 200ms (w.r.t -6dB sustain level)
w a0 60 ff fe 4c 21 #Decay Timer = 100 ms
w a0 64 20 13 73 9e #Sustain Level = 0.501 (~-6dB)

w a0 00 11 #Page 17
w a0 70 40 00 40 00 #Side-Chain Mixer Gain for SG2 = 1

w a0 00 00 #Page 0
w a0 76 0c #enable 2 DAC channels
w a0 78 40 #enable DAC
```

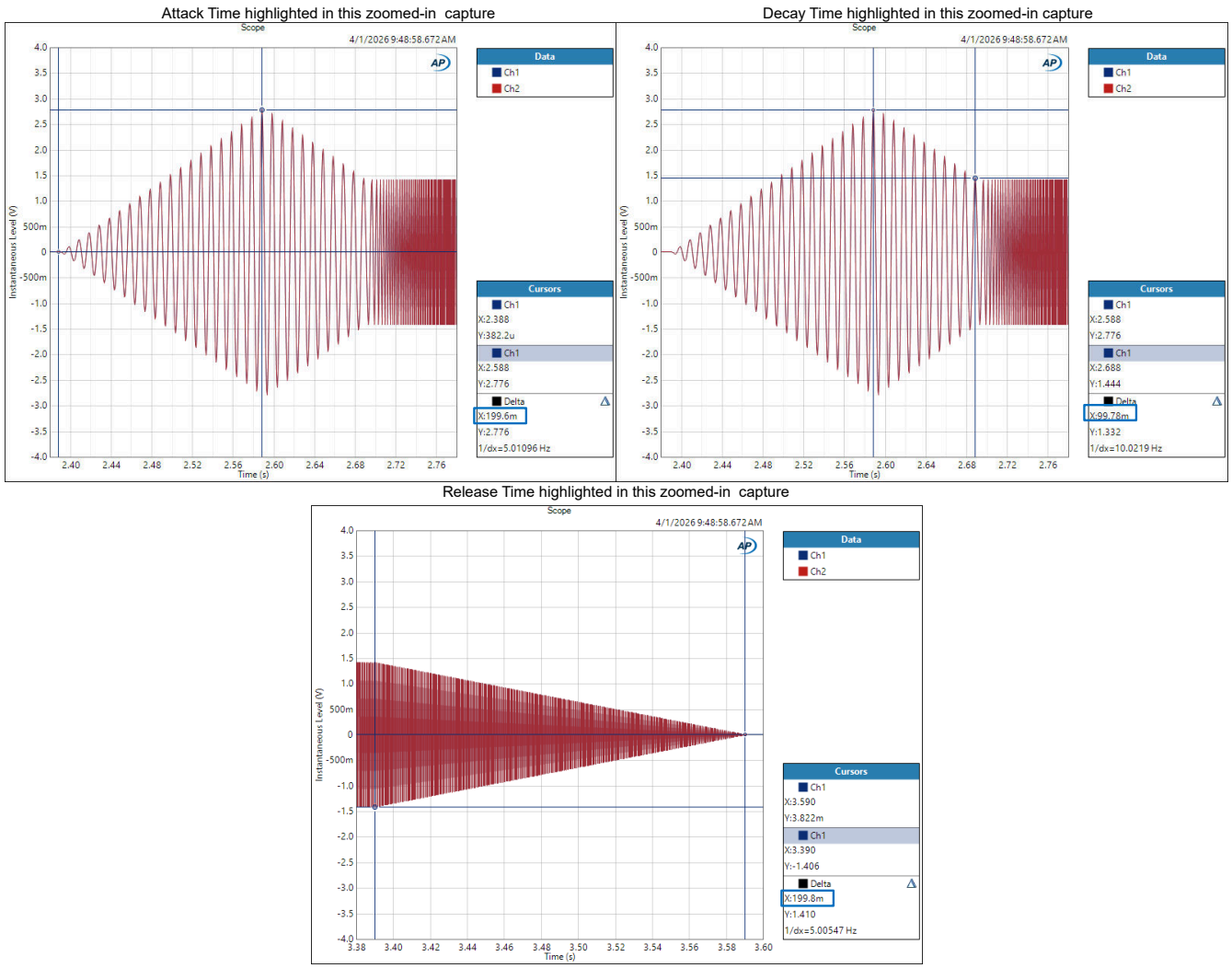

Figure 3-4. Attack, Decay and Release Times

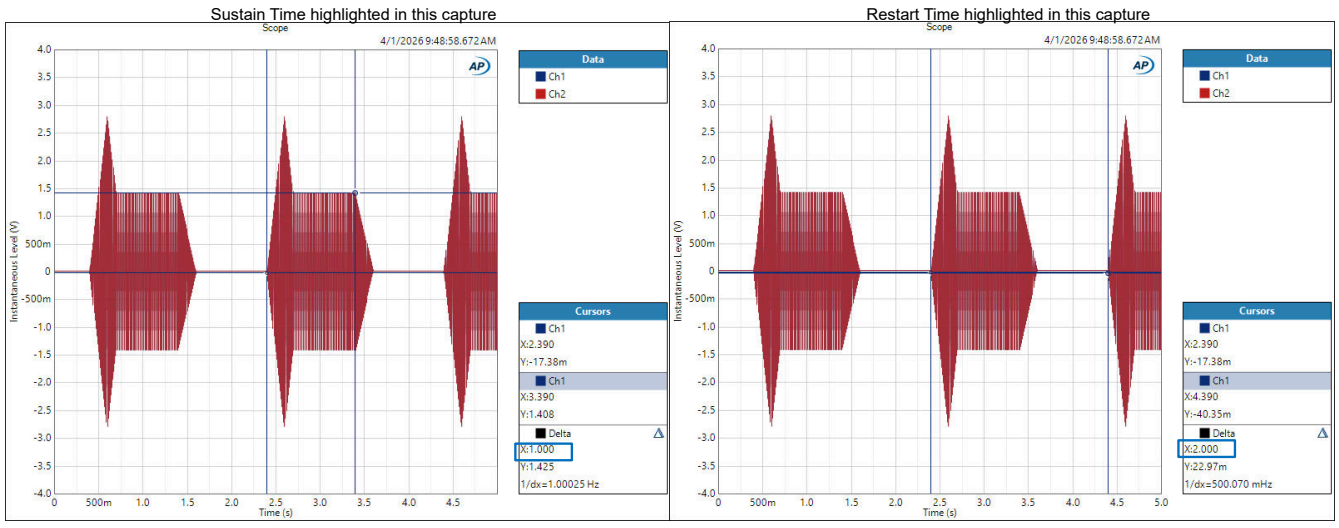

Figure 3-5. Sustain and Restart Times

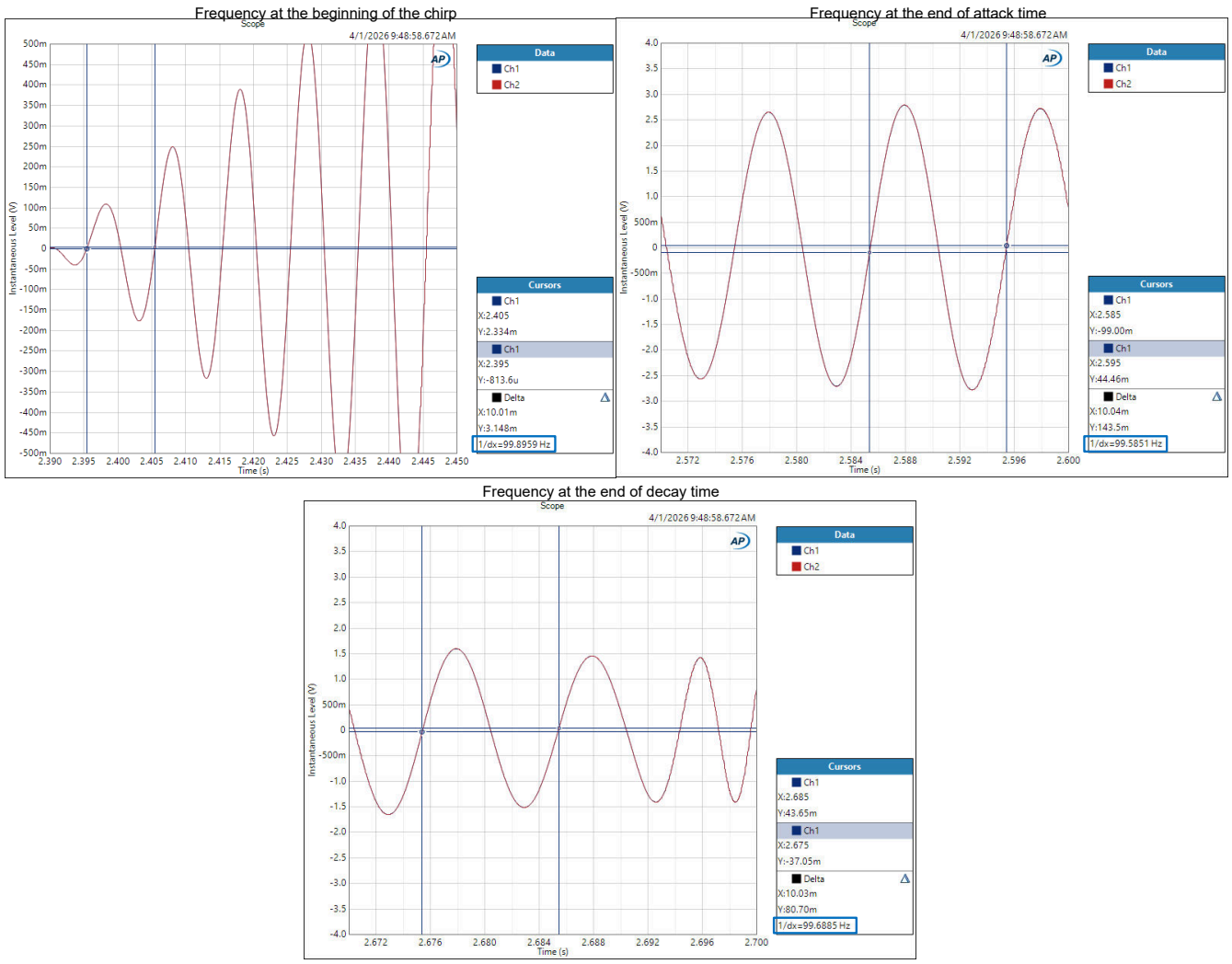

Figure 3-6. Chirp Start Frequency

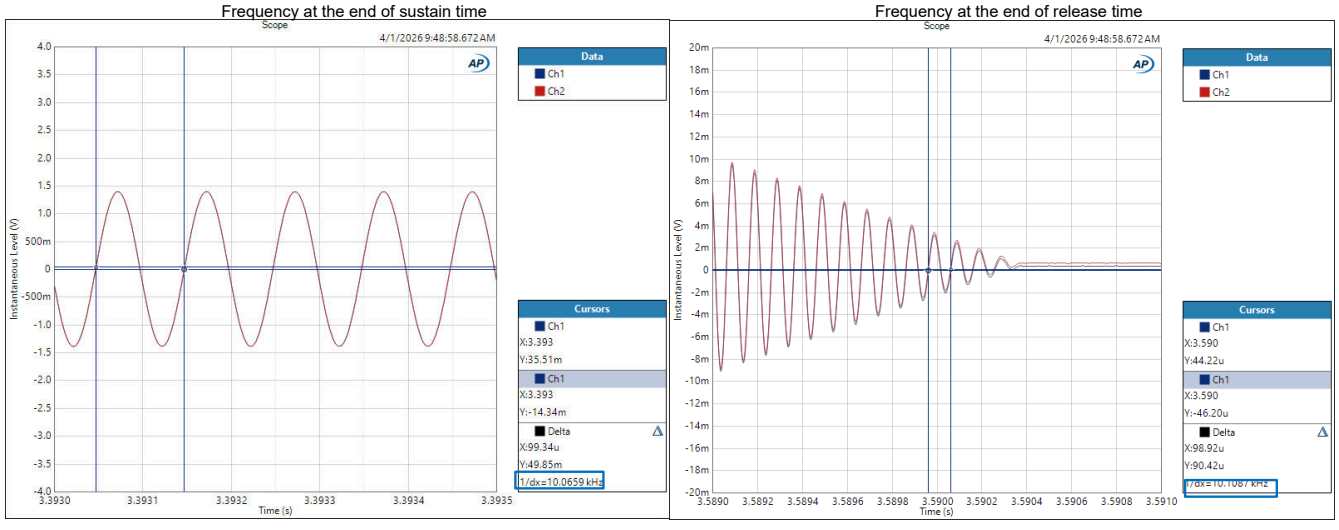

Figure 3-7. Chirp Stop Frequency

3.4.3 One Shot Mode

The one-shot mode is like the manual mode, with regards to ADSR note sequencing. The only difference is that the duration of the chirp pulse t_{chirp} is configured by setting the sustain timer registers.

To operate in one-shot mode, the ADSR note register needs to be set to 0xFFFFFFFF before configuring the ADSR parameters. Once the device is powered up, writing 0x00000001 the ADSR register initiates a single pulse of the chirp signal.

In one-shot mode, the restart and powerup delay timers are not applicable, hence these registers need to be programmed to 0xFFFFFFFF.

3.4.3.1 One-Shot Mode Example I²C Configuration using TAC5212EVM-K

The code below is used to start a single chirp cycle. The ADSR note is set to 0xFFFFFFFF before programming the parameters, and every time it is set to 0x00000001 after the device is powered up, it triggers a single chirp cycle. Figure 3-8shows the profile for a one-shot chirp cycle.

```
w a0 00 00 #Page 0
w a0 01 01 #SW Reset
d 10
w a0 00 00 #Page 0
w a0 02 09 #Exit Sleep Mode with DREG and VREF Enabled
w a0 1a 30 #TDM protocol with 32-bit word length
w a0 64 20 #DAC Channel 1 configured for differential output with 0.6*Vref as common mode
w a0 65 20 #DAC OUT1P configured for line out driver and audio bandwidth
w a0 66 20 #DAC OUT1M configured for line out driver and audio bandwidth
w a0 6b 20 #DAC Channel 2 configured for differential output with 0.6*Vref as common mode
w a0 6c 20 #DAC OUT2P configured for line out driver and audio bandwidth
w a0 6d 20 #DAC OUT2M configured for line out driver and audio bandwidth
w a0 26 01 #RX Offset = 1
w a0 28 00 #RX CH1 to DAC CH1 disabled
w a0 29 00 #RX CH1 to DAC CH1 disabled

w a0 00 01 #Page 1
w a0 2d 04 #Enable the chirp (SG2)
w a0 2c 40 #Enable side-chain mixer

w a0 00 17 #Page 23
w a0 7c 00 35 9d d4 #Chirp start frequency of 100 Hz

w a0 00 18
w a0 08 00 00 1c 4f #Chirp delta frequency of 0.206 Hz

w a0 00 1c #Page 28
w a0 40 ff ff ff ff #ADSR_Note
w a0 50 ff ff ff ff #Restart Timer = NA
w a0 54 00 00 96 00 #Sustain Timer = 800msec
w a0 58 00 03 69 d0 #Attack Timer = 100ms
w a0 5c ff fe 4b 18 #Release Timer = 150ms (w.r.t -2.5dB sustain level)
w a0 60 ff ff c9 62 #Decay Timer = 400 ms
w a0 64 2f ff ae 6b #Sustain Level = 0.75 (~-2.5dB)

w a0 00 17 #Page 23
w a0 74 ff ff ff ff #Powerup Delay = NA

w a0 00 11 Page 17
w a0 70 40 00 40 00 #Side-Chain Mixer Gain for SG2 = 1

w a0 00 00 # goto page 0
w a0 76 0c # enable 2 DAC channels
w a0 78 40 # enable DAC

w a0 00 00 #Page 0
w a0 76 0c #enable 2 DAC channels
w a0 78 40 #enable DAC

b # Breakpoint (Writing 0x00000001 into ADSR_Note will trigger a single chirp pulse)
#(Writing 0x00000001 again will trigger a second pulse)
w a0 00 1c #Page 28
w a0 40 00 00 00 01
w a0 00 00
```

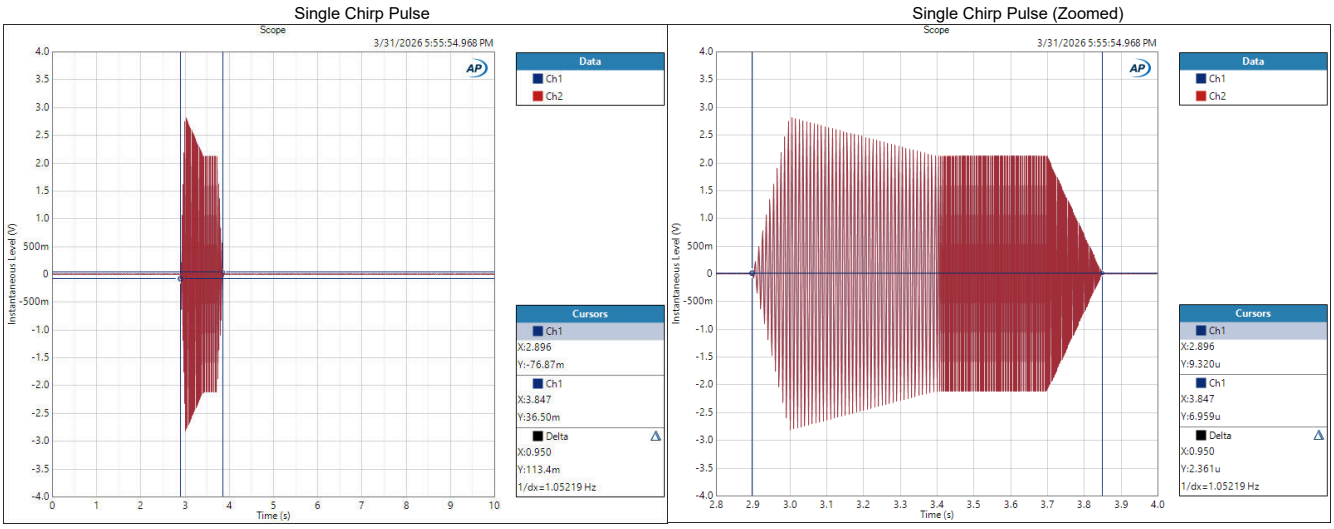

Figure 3-8. Single Chirp Pulse

4 Summary

This app note has described the internal signal generators that are available on the TAx5x1x/TAx5x1x-Q1 family of audio devices, along with guidelines on configuring them. The SG1 generator can be used in applications where a single “beep” is used as an indication for some system event. Likewise, the SG2 generator at higher sampling rates can be used to generate chirp pulses outside the audible band and hence can be used in applications involving ultrasonic generation and detection.

5 References

- TAC5212 High-performance stereo audio codec with 119dB dynamic range ADC and 120dB dynamic range DAC
- TAC5x1x and TAC5x1x-Q1 Digital Channel Mixers - Configurations and Applications

6 Revision History

Changes from Revision * (October 2024) to Revision A (May 2025)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	1

IMPORTANT NOTICE AND DISCLAIMER

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

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

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

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

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

Last updated 10/2025