*Application Note*

# *Tone Generation and Application Modes of TAx5x1x Devices*

*Diekoloreoluwa Ogundana, Abin Dany Mathew*

**ABSTRACT**

The TAx5x1x family of stereo CODECs and DACs support a range of applications such as jack detection, brown-out limiter, voice activity detection, and tone generation. Among these devices are TAC5212, TAC5211, TAC5112, TAC5111, TAD5212, TAD5112, TAC5311, TAC5312, TAC5411, and TAC5412. This application note explains how to use the tone generation feature.

There are two integrated digital tone generators that have a range of capabilities such as frequency and gain sweeping. These generators are high precision and are operable at audio and ultrasonic frequencies. The ADSR (attack-decay-sustain-release) envelope is also a feature within the generators and is available for applications that require sound modulation using the envelope parameters.

TAC5212 stereo software-controlled CODEC is used to demonstrate the capabilities of the tone generators in this application note. Use the EVM user guide to locate the mixer tab containing the tone generation GUI in PPC3.

## Table of Contents

**1 Introduction**....2
**2 Signal Generator 1 (SG1)**....2
2.1 Signal Generator 1 Introduction....2
2.2 Signal Generator 1 Amplitude and Output Channels....2
2.3 Signal Generator 1 Frequency....4
2.4 Generating Register Coefficients Using PPC3....6
**3 ADSR Envelope Parameters**....8
3.1 ADSR Introduction....8
3.2 Restart and Sustain Timers....9
3.3 Attack, Release, and Decay Timers....10
3.4 Sustain Level....11
3.5 ADSR Envelope Example Script....11
3.6 Ultrasonic Activity Detection (UAD) ADSR Mode....13
**4 Signal Generator 2 (SG2)**....15
4.1 Signal Generator 2 Introduction....15
4.2 Signal Generator 2 Amplitude....16
4.3 Signal Generator 2 Frequency....16
4.4 Signal Generator 2 Modes....17
**5 Summary**....23

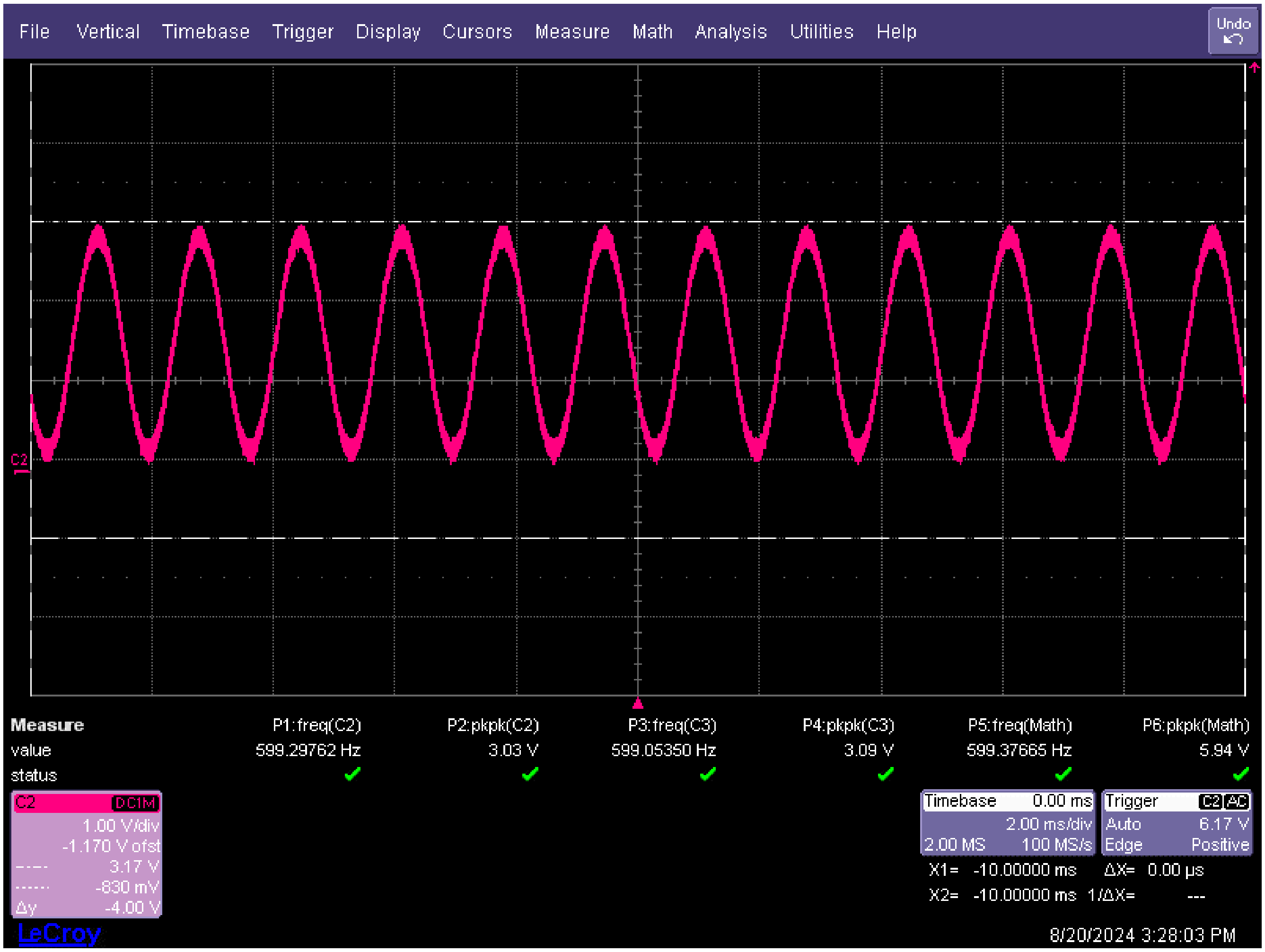
## Trademarks

All trademarks are the property of their respective owners.

# 1 Introduction

Signal generator 1 and signal generator 2 are the digital tone generators in TAC5212. Tone generation is supported in all DAC configurations including fully differential , single-ended, psuedo differential and stereo single-ended modes. All results are monitored on OUT1 of the device unless mentioned otherwise. Throughout this document, signal generator 1 is synonymous with SG1 and beep generator, and signal generator 2 is synonymous with SG2 and chirp generator.

# 2 Signal Generator 1 (SG1)



| Measure | P1:freq(C2) | P2:pkpk(C2) | P3:freq(C3) | P4:pkpk(C3) | P5:freq(Math) | P6:pkpk(Math) |
|---|---|---|---|---|---|---|
| value | 599.29762 Hz | 3.03 V | 599.05350 Hz | 3.09 V | 599.37665 Hz | 5.94 V |
| status | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

C2 DC1M
1.00 V/div
-1.170 V ofst
---- 3.17 V
...... -830 mV
Δy -4.00 V

Timebase 0.00 ms
2.00 ms/div
2.00 MS 100 MS/s

Trigger C2 AC
Auto 6.17 V
Edge Positive

X1= -10.00000 ms ΔX= 0.00 µs
X2= -10.00000 ms 1/ΔX= ---

8/20/2024 3:28:03 PM

**Figure 2-1. Scope Capture from SG1 Example Script: 600Hz Signal at 2Vrms, Full-Scale**

## 2.1 Signal Generator 1 Introduction

Signal Generator 1 (SG1) or Beep Generator generates a continuous pulse with configurable amplitudes up to a full-scale value of 2Vrms. The max frequency obtainable from SG1 is 22KHz. Use SG1 example script to generate signals.

## 2.2 Signal Generator 1 Amplitude and Output Channels

Registers B0_P17_R104 (0x68) to B0_P17_111 (0x6F) in Table 2-1 configure amplitude levels and select output channels for SG1.

Two ways to configure amplitude are executing I2C scripts or using PPC3 mixing volume coefficients.

Table 2-1 demonstrates how these two methods increase or decrease amplitude in steps of 0.56dB gain. Here, I2C scripts require 16-bit registers per channel. Each amplitude and channel selection configuration requires executing a 32-bit I2C command.

This example script template places each analog output channel of OUT1 and OUT2 in a 32-bit I2C commands. This template applies to all SG2 registers as well.

The full-scale value in Vrms and max gain in dB correlates to 0x40000. This makes up 16-bit of the 32-bit I2C command.

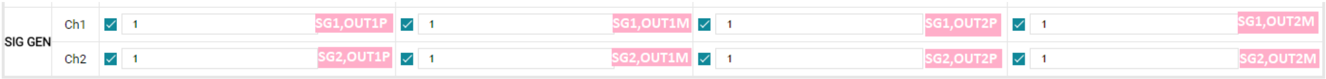
```
#Example template to configure SG1 to OUT1P&M and OUT2P&M. Same format #applies to SG2 registers.
#w a0 68 [OUT1M] [OUT1M] [OUT1P] [OUT1P]
#w a0 6C [OUT2M] [OUT2M] [OUT2P] [OUT2P]
#w xx XX [15:8] [7:0] [15:8] [7:0]

# This is a 32-bit I2C command. Here, OUT1P has full gain while OUT1M has #no gain.
w a0 68 00 00 40 00
```

SG1 example script has an example on amplitude and channel selection using I2C commands. PPC3 provides a range of coefficients between 1 to 0. To predict gain in dB through these coefficients, use...

$$gain = 20\log_{10}(mixing\ \ volume\ \ coefficient),\ \ dB \tag{1}$$

The full-scale value in Vrms and max gain in dB correlates to a coefficient of 1.

Figure 2-2 demonstrates how signal generators use mixing volume coefficients to configure amplitude in PPC3.



| | | | | | |
|---|---|---|---|---|---|
| SIG GEN | Ch1 | ☑ 1 SG1,OUT1P | ☑ 1 SG1,OUT1M | ☑ 1 SG1,OUT2P | ☑ 1 SG1,OUT2M |
| | Ch2 | ☑ 1 SG2,OUT1P | ☑ 1 SG2,OUT1M | ☑ 1 SG2,OUT2P | ☑ 1 SG2,OUT2M |

**Figure 2-2. Amplitude and Output Channel Configuration GUI**

Table 2-4 shows how to navigate to the tone generator GUI.

**Table 2-1. Amplitude Level Configuration**

| Decimal Value | 16-bit I2C command | Mixing Volume Coefficient | Gain dB | Vrms |
|---|---|---|---|---|
| d16384 | 0x4000 | 1 | 0dB | 2Vrms |
| d8192 | 0x2000 | 0.5 | -6dB | 1Vrms |
| d64 | 0x0100 | 0.016 | -35.90dB | - |
| d0 | 0x0000 | - | - | - |

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

| Page | Register | Description | Reset Value |
|---|---|---|---|
| 0x11 | 0x68 | Side Chain DAC Mixer, SG1 to OUT1M coefficient byte [15:8] | 0x00 |
| 0x11 | 0x69 | Side Chain DAC Mixer, SG1 to OUT1M coefficient byte [7:0] | 0x00 |
| 0x11 | 0x6A | Side Chain DAC Mixer, SG1 to OUT1P coefficient byte [15:8] | 0x00 |
| 0x11 | 0x6B | Side Chain DAC Mixer, SG1 to OUT1P coefficient byte [7:0] | 0x00 |
| 0x11 | 0x6C | Side Chain DAC Mixer, SG1 to OUT2M coefficient byte [15:8] | 0x00 |
| 0x11 | 0x6D | Side Chain DAC Mixer, SG1 to OUT2M coefficient byte [7:0] | 0x00 |
| 0x11 | 0x6E | Side Chain DAC Mixer, SG1 to OUT2P coefficient byte [15:8] | 0x00 |

**Table 2-2. Programmable Register Map for SG1 Amplitude and Output Channels (continued)**

| Page | Register | Description | Reset Value |
|---|---|---|---|
| 0x11 | 0x6F | Side Chain DAC Mixer, SG1 to OUT2P coefficient byte [7:0] | 0x00 |

## 2.3 Signal Generator 1 Frequency

Registers B0_P18_R32 (0x20) to B0_P1_R47 (0x2F) configure SG1 frequency. SG1 frequency configuration comes after amplitude and output channel configurations. Refer to Section 2.2.

This set of equations generates register coefficients for SG1 frequency configuration:

$$g(x) \quad = \sin\left(2\pi\left(\frac{\text{desired} \quad beep \quad \text{frequency}}{\text{sampling} \quad frequency}\right)\right) \times \quad 2^{63} \tag{2}$$

$$h \quad = \quad DEC2HEX(g(x)), \quad 64-bit \tag{3}$$

$$(h\_msb) \quad = \quad upper \quad 32-bit \quad of \quad h \tag{4}$$

$$(h\_lsb) \quad = \quad lower \quad 32-bit \quad of \quad h \quad right \quad shifted \quad by \quad 1-bit \tag{5}$$

$$f(x) \quad = \cos\left(2\pi\left(\frac{\text{desired} \quad beep \quad \text{frequency}}{\text{sampling} \quad \text{frequency}}\right)\right) \times \quad 2^{63} \tag{6}$$

$$y \quad = \quad DEC2HEX(f(x)), \quad 64-bit \tag{7}$$

$$(y\_msb) \quad = \quad upper \quad 32-bit \quad of \quad y \tag{8}$$

$$(y\_lsb) \quad = \quad lower \quad 32-bit \quad of \quad y \quad right \quad shifted \quad by \quad 1-bit \tag{9}$$

Pay attention to the *right shift by 1-bit* on the LSB values generated. This is important for executing correct I2C commands. These formulas generate the most significant bit (msb) and least significant bit (lsb) values that make up the I2C command script for SG1 frequency configuration.

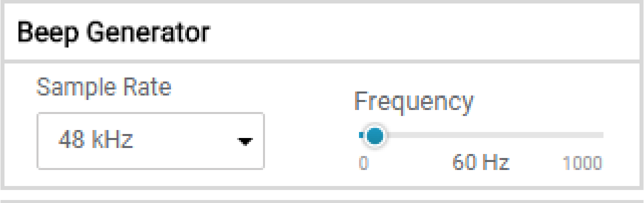
SG1 frequency configuration is a pair of 32-bit I2C commands writing a single frequency to all output channels. No channel is isolated in this frequency configuration. Signals only appear on any output channel based on channel selection configurations. Refer to Section 2.2.

SG1 maintains the same frequency across all channels selected. Generally, one amplitude and frequency configuration is maintained across all channels for both SG1 and SG2.

Below is an example script template for SG1 frequency configuration. This script template shows register coefficients placements. This is based off results from *h_msb, h_lsb, y_msb,* and *y_lsb* equations above and the SG1 example script at the end of this section.

```
#Based off SG1's 600Hz example script.
#64-bit command from SG1 to OUT1
#w a0 20 [31:24] [23:16] [15:8] [7:0] [31:24] [23:16] [15:8] [7:0]
#64-bit command from SG1 to OUT2
#w a0 28 [31:24] [23:16] [15:8] [7:0] [31:24] [23:16] [15:8] [7:0]

#based off SG1 frequency equations.
#w a0 20 [h_msb] [y_msb ]
#w a0 28 [h_lsb, right shift by 1-bit] [y_lsb, right shift by 1-bit]

#600Hz from SG1.
w a0 20 99 7f ec 00 7f 9a fc b8 #[h_msb] [y_msb]
w a0 28 1c ea f1 80 0a 0a f2 99 #[h_lsb] [y_lsb]
```

In PPC3, SG1 frequency configuration is done using the frequency scroll GUI. Figure 2-3 is an image of SG1 frequency scroll GUI.



Beep Generator

Sample Rate: 48 kHz

Frequency: 60 Hz

**Figure 2-3. SG1 Frequency Configuration GUI**

**Table 2-3. Programmable Register Map for SG1 Frequency Configuration**

| Page | Register | Description |
|---|---|---|
| 0x12 | 0x20 | Programmable OUT1 BEEP GEN sin(x) coefficient byte [31:24] |
| 0x12 | 0x21 | Programmable OUT1 BEEP GEN sin(x) coefficient byte [23:16] |
| 0x12 | 0x22 | Programmable OUT1 BEEP GEN sin(x) coefficient byte [15:8] |
| 0x12 | 0x23 | Programmable OUT1 BEEP GEN sin(x) coefficient byte [7:0] |
| 0x12 | 0x24 | Programmable OUT1 BEEP GEN cos(x) coefficient byte [31:24] |
| 0x12 | 0x25 | Programmable OUT1 BEEP GEN cos(x) coefficient byte [23:16] |
| 0x12 | 0x26 | Programmable OUT1 BEEP GEN cos(x) coefficient byte [15:8] |
| 0x12 | 0x27 | Programmable OUT1 BEEP GEN cos(x) coefficient byte [7:0] |
| 0x12 | 0x28 | Programmable OUT2 BEEP GEN sin(x) coefficient byte [31:24] |
| 0x12 | 0x29 | Programmable OUT2 BEEP GEN sin(x) coefficient byte [23:16] |
| 0x12 | 0x2A | Programmable OUT2 BEEP GEN sin(x) coefficient byte [15:8] |
| 0x12 | 0x2B | Programmable OUT2 BEEP GEN sin(x) coefficient byte [7:0] |
| 0x12 | 0x2C | Programmable OUT2 BEEP GEN cos(x) coefficient byte [31:24] |
| 0x12 | 0x2D | Programmable OUT2 BEEP GEN cos(x) coefficient byte [23:16] |
| 0x12 | 0x2E | Programmable OUT2 BEEP GEN cos(x) coefficient byte [15:8] |
| 0x12 | 0x2F | Programmable OUT2 BEEP GEN cos(x) coefficient byte [7:0] |

```
# Key: w a0 XX YY ==> write to I2C address 0xa0, to register 0xXX, data 0xYY
# # ==> comment delimiter
#
#The following list gives an example sequence of items that must be #executed in the time between
powering the device up and reading data #from the device. Note that there are other valid sequences
depending #on which features are used.
#See the corresponding EVM user guide for jumper settings and audio #connections.
#
# Line-Out Fully-Differential 2-channel : OUT1P_M- Ch1, OUT2P_M- Ch2.
# FSYNC = 48 kHz (Output Data Sample Rate), BCLK = 12.288 MHz (BCLK/FSYNC = 256)
###################################################################
#SG1 example script
#SG1; DAC1 OUT1P & OUT1M; Differential 600Hz
w a0 00 00 # locate page x00
w a0 01 01 # device reset

w a0 02 09 # vref and dreg enable

w a0 00 01 # locate page x01
w a0 2d 08 # enable SG1

#Channel selection and amplitude configuration
w a0 00 11 # locate page 0x11
w a0 68 40 00 40 00 # mix SG1 to DAC1 L & R Channels and
                    # set the amplitude to full-scale, 0dB.
#Frequency configuration
w a0 00 00 # locate page 0x00
w a0 7f 00 # locate book 0x00
w a0 00 12 # locate page 0x12

#600Hz
w a0 20 99 7f ec 00 7f 9a fc b8 #sinx and cosx upper bits
w a0 28 1c ea f1 80 0a 0a f2 99 #sinx and cosx lower bits

#Output configuration
w a0 00 00 # locate page 0x00

#differential mode is the default output mode
#w a0 64 24 #option to set to single-ended mode
w a0 76 0c #enable DAC1, CH 1 & 2 180deg out of phase diff. mode
w a0 78 40 #enable DAC
```

## 2.4 Generating Register Coefficients Using PPC3

PPC3 is recommended to generate register values for all forms of tone generation configurations. This section uses SG1 amplitude and frequency configuration as an example to generate I2C scripts.

The mixing volume coefficient and this gain formula configure SG1 and SG2 amplitudes. Fill in coefficient values in the GUI boxes like Figure 2-2 suggests. This applies to all tone generators.

I2C script generation can be accessed offline in PPC3 without connecting a device to the software.

## Table 2-4. PPC3 Guide to Generate Hexadecimal Register Coefficients

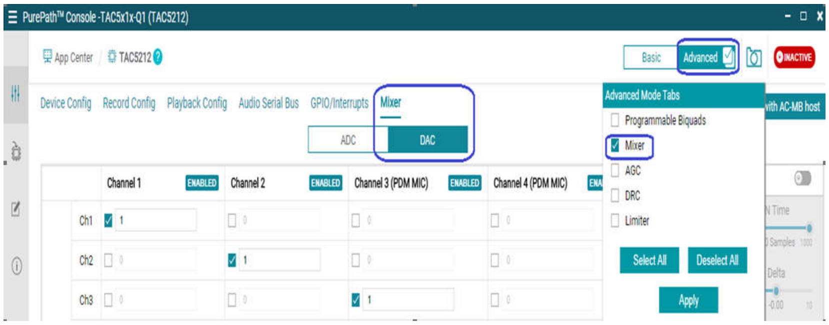
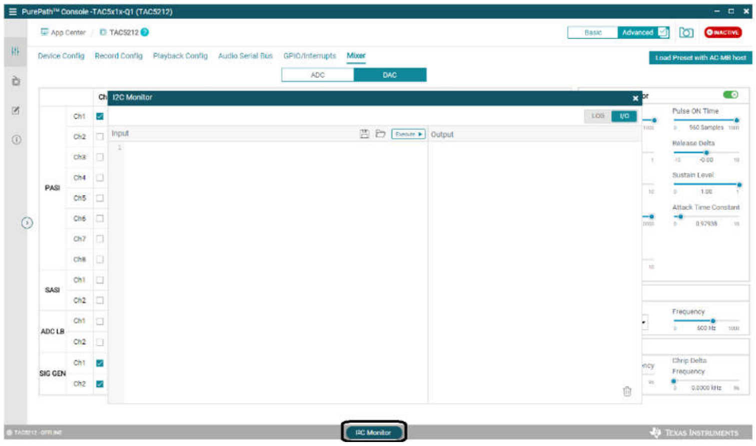
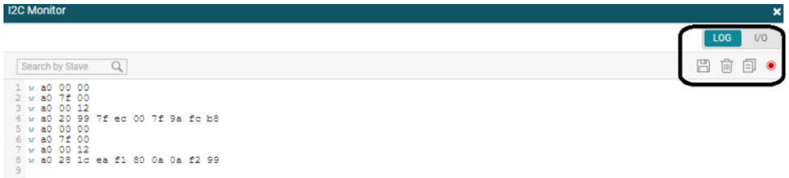
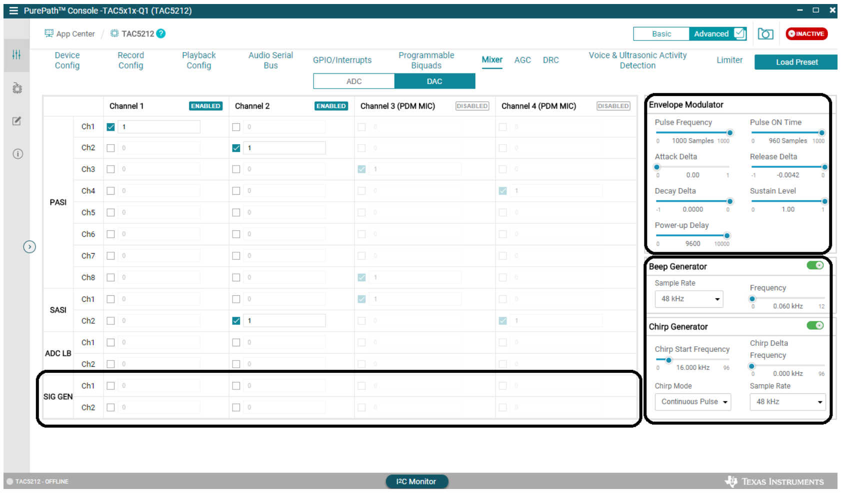
| Instruction | Image |
|---|---|
| Locate DAC Mixer Page in PPC3. |  PurePath™ Console -TAC5x1x-Q1 (TAC5212)<br>App Center<br>TAC5212<br>Basic<br>Advanced<br>INACTIVE<br>Device Config<br>Record Config<br>Playback Config<br>Audio Serial Bus<br>GPIO/Interrupts<br>Mixer<br>ADC<br>DAC<br>Channel 1<br>Channel 2<br>Channel 3 (PDM MIC)<br>Channel 4 (PDM MIC)<br>ENABLED<br>Ch1<br>Ch2<br>Ch3<br>Advanced Mode Tabs<br>Programmable Biquads<br>Mixer<br>AGC<br>DRC<br>Limiter<br>Select All<br>Deselect All<br>Apply<br>**Figure 2-4. PPC3 step 1** |
| Open I2C Monitor Window. |  PurePath™ Console -TAC5x1x-Q1 (TAC5212)<br>App Center<br>TAC5212<br>Basic<br>Advanced<br>INACTIVE<br>Device Config<br>Record Config<br>Playback Config<br>Audio Serial Bus<br>GPIO/Interrupts<br>Mixer<br>ADC<br>DAC<br>Load Preset with AC MB host<br>I2C Monitor<br>LOG<br>I/O<br>Input<br>Execute<br>Output<br>PASI<br>SASI<br>ADC LB<br>SIG GEN<br>Pulse ON Time<br>Release Delta<br>Sustain Level<br>Attack Time Constant<br>Frequency<br>Chirp Delta Frequency<br>I2C Monitor<br>TEXAS INSTRUMENTS<br>**Figure 2-5. PPC3 step 2** |
| Select LOG page. Click on green button. The button turns red. This means PPC3 is ready to record next I2C executions. |  <br>w a0 00 00<br>w a0 7f 00<br>w a0 00 12<br>w a0 20 99 7f ec 00 7f 9a fc b8<br>w a0 00 00<br>w a0 7f 00<br>w a0 00 12<br>w a0 28 1c ea f1 80 0a 0a f2 99<br>**Figure 2-6. PPC3 step 3** |

**Table 2-4. PPC3 Guide to Generate Hexadecimal Register Coefficients (continued)**

| Instruction | Image |
|---|---|
| Return to main PPC3 window. Configure amplitude and frequency using GUI. Observe change in hex values on LOG window. |  PurePath™ Console -TAC5x1x-Q1 (TAC5212)<br>App Center / TAC5212<br>Basic<br>Advanced<br>INACTIVE<br>Device Config<br>Record Config<br>Playback Config<br>Audio Serial Bus<br>GPIO/Interrupts<br>Programmable Biquads<br>Mixer<br>AGC<br>DRC<br>Voice & Ultrasonic Activity Detection<br>Limiter<br>Load Preset<br>ADC<br>DAC<br>TAC5212 - OFFLINE<br>I²C Monitor<br>Texas Instruments<br>Channel 1 ENABLED; Channel 2 ENABLED; Channel 3 (PDM MIC) DISABLED; Channel 4 (PDM MIC) DISABLED<br>PASI Ch1: ☑ 1 · ☐ 0 · ☐ 0 · ☐ 0<br>PASI Ch2: ☐ 0 · ☑ 1 · ☐ 0 · ☐ 0<br>PASI Ch3: ☐ 0 · ☐ 0 · ☑ 1 · ☐ 0<br>PASI Ch4: ☐ 0 · ☐ 0 · ☐ 0 · ☑ 1<br>PASI Ch5: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>PASI Ch6: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>PASI Ch7: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>PASI Ch8: ☐ 0 · ☐ 0 · ☑ 1 · ☐ 0<br>SASI Ch1: ☐ 0 · ☐ 0 · ☑ 1 · ☐ 0<br>SASI Ch2: ☐ 0 · ☑ 1 · ☐ 0 · ☑ 1<br>ADC LB Ch1: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>ADC LB Ch2: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>SIG GEN Ch1: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>SIG GEN Ch2: ☐ 0 · ☐ 0 · ☐ 0 · ☐ 0<br>Envelope Modulator<br>Pulse Frequency: 1000 Samples (0–1000)<br>Pulse ON Time: 960 Samples (0–1000)<br>Attack Delta: 0.00 (0–1)<br>Release Delta: -0.0042 (-1–0)<br>Decay Delta: 0.0000 (-1–0)<br>Sustain Level: 1.00 (0–1)<br>Power-up Delay: 9600 (0–10000)<br>Beep Generator<br>Sample Rate: 48 kHz<br>Frequency: 0.060 kHz (0–12)<br>Chirp Generator<br>Chirp Start Frequency: 16.000 kHz (0–96)<br>Chirp Delta Frequency: 0.000 kHz (0–96)<br>Chirp Mode: Continuous Pulse<br>Sample Rate: 48 kHz<br>**Figure 2-7. PPC3 step 4** |

# 3 ADSR Envelope Parameters

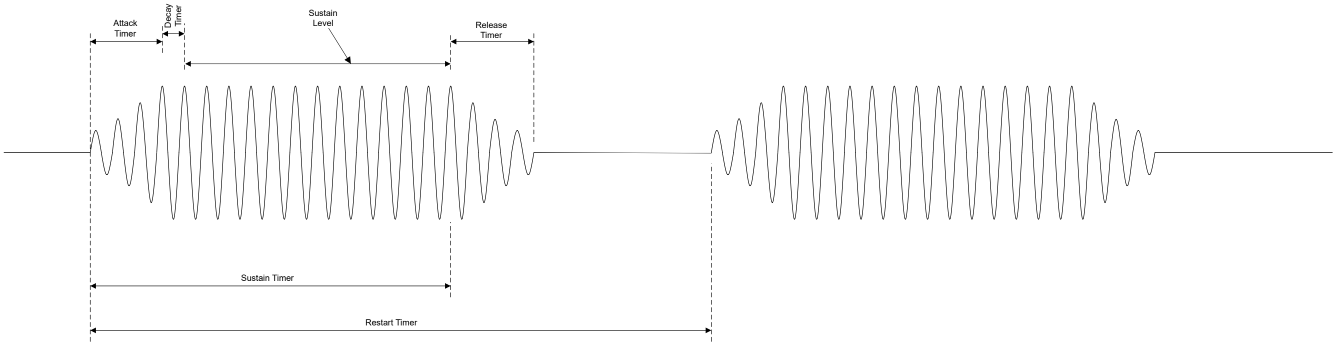


**Figure 3-1. Graphical Representation of High-Precision ADSR Envelope**

## 3.1 ADSR Introduction

ADSR stands for attack, decay, sustain, and release. The ADSR envelope is a commonly used in audio applications that require shaping sound. This feature has high precision and ultrasonic application modes. These parameters are applicable to SG2 pulses only.

Graphical Representation of High-Precision ADSR Envelope highlights the start and end point of each ADSR parameter. In the ADSR Envelope, Sustain level is between the attack and release time intervals. Attack time defines the positive ramp before the sustain level while the release time defines the negative ramp past the sustain level. Decay time controls the settling period after attack. This interval ends just before sustain level.

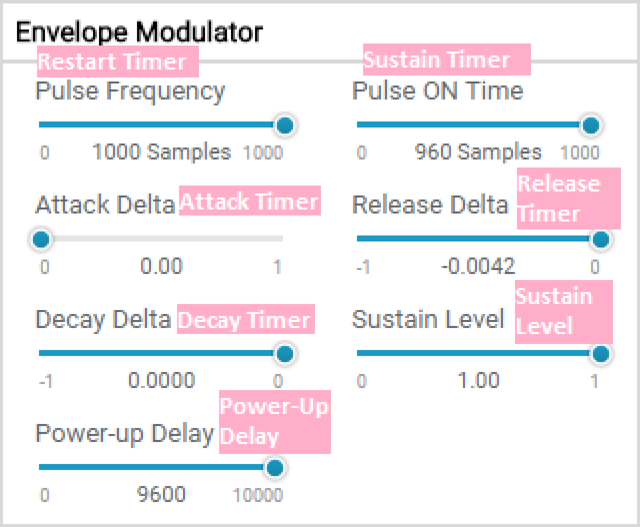
Across all these parameters are the Restart and Sustain timers. Restart Timer is always longer than Sustain timer.

32-bit registers configure each ADSR parameter. Registers B0_P28_R64 (0x40) to B0_P28_R67 (0x43) enable and disable the ADSR envelope. Refer to Programmable Register Map for 32-bit ADSR Parameters. This register map shows which SG2 modes require the ADSR note. Below is an example script template for ADSR note execution. Find ADSR note in use in this SG2 mode example script.

```
#ADSR example script template
#ADSR acknowledgement.
w a0 00 1c # locate ADSR note page
w a0 40 00 00 00 00 #acknowledge ADSR Envelope

#configure all other paramters next
w a0 xx xx
w a0 xx xx
#configure the output channel. Find this in the example script.

#Turn on ADSR envelope
w a0 00 1c # loacate ADSR note page
w a0 40 00 00 00 01 # turn on ADSR Envelope.
```

When building an I2C script, the ADSR enable_disable note is typically configured before other parameters as a 32-bit command, x00 00 00 00. This must be considered as an acknowledgment script to the envelope. After this, other ADSR scripts can be executed. Once all other ADSR parameters have been configured and all output channels enabled, this note is written as x00 00 00 01.

Figure 3-2 is the ADSR envelope PPC3 GUI.



Envelope Modulator

Pulse Frequency: 0 — 1000 Samples — 1000

Pulse ON Time: 0 — 960 Samples — 1000

Attack Delta: 0 — 0.00 — 1

Release Delta: -1 — -0.0042 — 0

Decay Delta: -1 — 0.0000 — 0

Sustain Level: 0 — 1.00 — 1

Power-up Delay: 0 — 9600 — 10000

**Figure 3-2. ADSR PPC3 GUI**

### 3.2 Restart and Sustain Timers

Registers B0_P28_R80 (0x50) to B0_P28_R83 (0x53) configure restart time. Registers B0_P28_R84 (0x54) to B0_P28_R87 (0x57) configure sustain time. Refer to Table 3-2.

The restart time is the time for a configured pulse to repeat, while the sustain time is the length of the pulse configured.

Restart and Sustain timers are based off the number of samples retrieved from an operating sampling frequency, fs. Restart time must always be configured to be longer than sustain time. Table 3-1 generates possible I2C configurations for both parameters based off a 48KHz sampling frequency. The desired time is generated based on scaling samples with respect to the sampling frequency. These two timers are not available in all SG2 modes.

**Table 3-1. Restart and Sustain Timer Correlation Table**

| ksps | Number of Samples Per Timer | I2C Command Script | Desired Time |
|---|---|---|---|
| 48000 | d’480 | 0x000001E0 | 10ms |
| 48000 | d’4,800 | 0x000012C0 | 100ms |
| 48000 | d’48,000 | 0x0000BB80 | 1s |
| 48000 | d’480,000 | 0x00075300 | 10s |

The formula below generates a 32-bit hex register coefficients for both restart and sustain timers.

$$Reg.Coeff = DEC2HEX\left(\left(MOD\left(ROUND\left(k \times \ \left(2^{0}\right), \ \ 0\right), \ \ 2^{32}\right)\right), 8\right) \tag{10}$$

where k is the number of samples per timer.

Restart and sustain timers have been demonstrated in Continuous Pulse Mode example script. These are all 32-bit I2C commands. Refer to this image to locate restart and sustain timer GUI in PPC3. The scrolls generate hex values in I2C monitor.

Find more info on using PPC3 here Generating Register Coefficients Using PPC3.

**Table 3-2. Programmable Restart Timer Registers**

| Page | Register | Description | Reset value |
|---|---|---|---|
| 0x1C | 0x50 | DAC_ADSR_RESTART_BYT1[7:0] | 0x00 |
| 0x1C | 0x51 | DAC_ADSR_ RESTART_BYT2[7:0] | 0x00 |
| 0x1C | 0x52 | DAC_ADSR_ RESTART_BYT3[7:0] | 0x00 |
| 0x1C | 0x53 | DAC_ADSR_ RESTART_BYT4[7:0] | 0x00 |

**Table 3-3. Programmable Sustain Timer Registers**

| Page | Register | Description | Reset Value |
|---|---|---|---|
| 0x1C | 0x54 | DAC_ADSR_SUSTAIN_BYT1[7:0] | 0x00 |
| 0x1C | 0x55 | DAC_ADSR_SUSTAIN_BYT2[7:0] | 0x00 |
| 0x1C | 0x56 | DAC_ADSR_SUSTAIN_BYT3[7:0] | 0x00 |
| 0x1C | 0x57 | DAC_ADSR_SUSTAIN_BYT4[7:0] | 0x00 |

## 3.3 Attack, Release, and Decay Timers

Attack, release and decay define the ramps of ADSR envelope. Refer to Programmable Register Map for 32-bit ADSR Parameters to determine registers that configure these three parameters. When calculating attack time and release time, use the equation below.

$$\frac{2}{2(k)} \times \frac{1}{f_S} \times 1000 \ \ = \ \ desired \ \ time, \ \ ms. \tag{11}$$

Where *k* is an absolute value between 0 to 1 for both timers.

This formula generates register coefficients for attack and release time.

$$Reg.Coeff = DEC2HEX\left(\left(MOD\left(ROUND\left(k \times \ \left(2^{30}\right), \ \ 0\right), \ \ 2^{32}\right)\right), 8\right) \tag{12}$$

The equation can be used as is for attack time. However for release time, *k* must be inserted as a negative number to retrieve accurate hex values. Use the equation below to calculate desired decay time.

$$ABS[(2 - (2 \times Sustain \ \ Level))/(k \times FS\_Vrms)/fs)] \times 1000 \ \ = desired \ \ time, \ \ ms \tag{13}$$

The register coefficient equation, *Reg.Coef f*, for attack and release time apply to decay time as well.

## 3.4 Sustain Level

Sustain level interval must not be mistaken for sustain timer interval. Refer to Figure 3-1 to spot the difference. The sustain level parameter is used to configure amplitude after decay time and before release time. Refer to Programmable Register Map for 32-bit ADSR Parameters to determine registers that configure sustain level.

The equation below generates register coefficient for sustain level:

$$Reg.Coeff = DEC2HEX\left(\left(MOD\left(ROUND\left(k \times \left(2^{30}\right), 0\right), 2^{32}\right)\right), 8\right) \tag{14}$$

where $k$ is the mixing volume coefficient ranging from 0 to 1. Here, 1 correlates to 2Vrms, full-scale. Find an example of the sustain level configuration in ADSR example script.

**Table 3-4. Programmable Register Map for 32-bit ADSR Parameters**

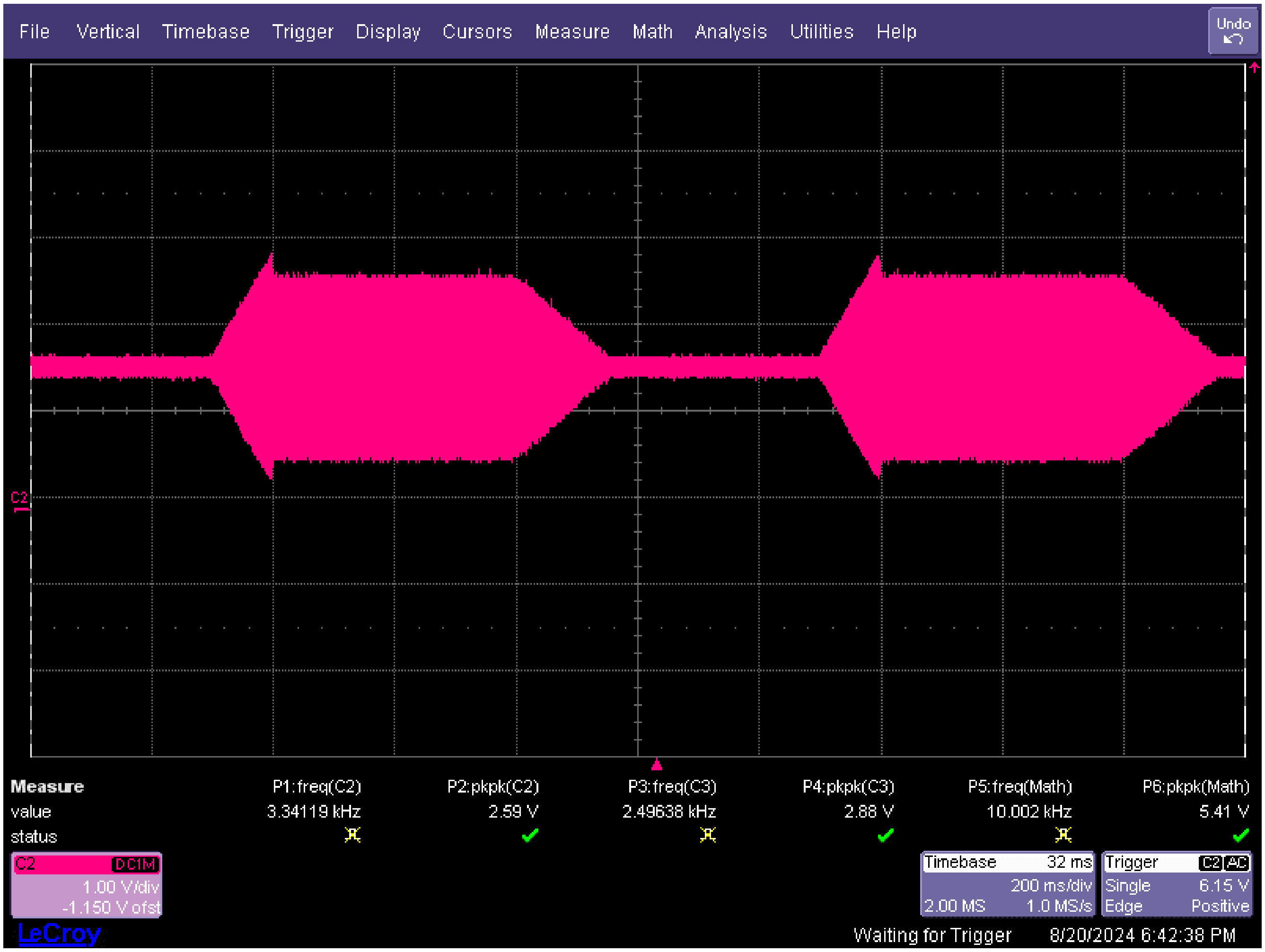
| Parameter Name | Page Address | Register Address | Manual Mode | Continuous Pulse Mode | One Shot | Default Values |
|---|---|---|---|---|---|---|
| ADSR EN_DIS Note | 1C | 40 | cfg | na | na | 0000 0000 (hex) |
| Restart Timer | 1C | 50 | na | cfg | na | 100ms at 48ksps |
| Sustain Timer | 1C | 54 | na | cfg | cfg | 10ms at 48ksps |
| Attack Timer | 1C | 58 | cfg | cfg | cfg | 2.5ms at 48ksps |
| Release Timer | 1C | 5C | cfg | cfg | cfg | 2.5ms at 48ksps |
| Decay Timer | 1C | 60 | cfg | cfg | cfg | 0ms at 48ksps |
| Sustain Level | 1C | 64 | cfg | cfg | cfg | 0dB |
| pwrup_delay | 17 | 74 | na | cfg | na | 100ms at 48ksps |

## 3.5 ADSR Envelope Example Script

The ADSR Envelope example script demonstrates the use of all ADSR parameters.

```
# Key: w a0 XX YY ==> write to I2C address 0xa0, to register 0xXX, data 0xYY
# # ==> comment delimiter
#
#The following list gives an example sequence of items that must be #executed in the time between
powering the device up and reading data #from the device. Note that there are other valid sequences
depending #on which features are used.
#See the corresponding EVM user guide for jumper settings and audio #connections.
#
# Line-Out Fully-Differential 2-channel : OUT1P_M- Ch1, OUT2P_M- Ch2.
# FSYNC = 48 kHz (Output Data Sample Rate), BCLK = 12.288 MHz (BCLK/FSYNC = 256)
#####################################################################
#ADSR Example Script
#ALL ADSR parameters demonstrated on SG2 CPM
#Note: CPM mode does not require ADSR enable script, only acknowledgement #script.
w a0 00 00 #locate page x00
w a0 01 01    # device reset

w a0 00 00    # locate page 0x00
w a0 02 09    # come out of sleep mode with VREF and DREG up

w a0 00 01    # locate page 0x01
w a0 2d 04    # enable chirp only

w a0 00 17    # locate page 0x17

w a0 7c 14 f1 a6 c6 # 10kHz chirp start frequency

w a0 00 18    #locate page 0x18
w a0 08 00 00 00 00    # chirp delta frequency of 0 Hz

w a0 00 1c    # locate page 0x1c
w a0 40 00 00 00 00    # adsr_note
```

```
#restart timer
w a0 50 00 00 bb 80     #48k samples for 1s

#sustain timer
w a0 54 00 00 5d c0     #24000 samples at 500ms

#attack    timer
w a0 58 00 03 69 cd # 100ms

#release timer
w a0 5c ff fe 4b 18 #200ms @ k = 1.04167e-4. Use full-scale susatin level    #to achieve accurate
reading.

#sustian level
w a0 64 33 33 33 33 # sustain lvl at k=0.8

#decay timer
w a0 60 ff f2 58 e3 #5ms

#channel selection
w a0 00 11 #locate page 0x11
w a0 70 40 00 40 00    # Please refer the table to set volumes     #accrodingly
w a0 74 40 00 40 00    # Please refer the table to set volumes  #accordingly

w a0 00 00    # locate page 0x00
w a0 76 0c    # enable 2 DAC channels
w a0 78 40    # enable DAC
```



```
Measure   P1:freq(C2)   P2:pkpk(C2)   P3:freq(C3)   P4:pkpk(C3)   P5:freq(Math)   P6:pkpk(Math)
value     3.34119 kHz   2.59 V        2.49638 kHz   2.88 V        10.002 kHz      5.41 V
status    ✗             ✓             ✗             ✓             ✗               ✓

C2   DC1M                      Timebase        32 ms   Trigger     C2 AC
     1.00 V/div                     200 ms/div         Single      6.15 V
     -1.150 V ofst             2.00 MS    1.0 MS/s     Edge        Positive

                               Waiting for Trigger     8/20/2024 6:42:38 PM
```

**Figure 3-3. ADSR Envelope Scope Capture**

## 3.6 Ultrasonic Activity Detection (UAD) ADSR Mode

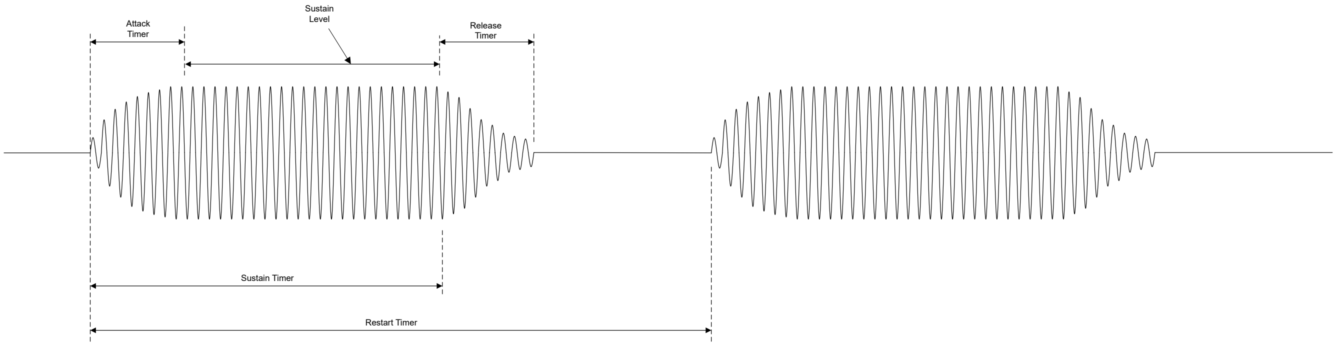


**Figure 3-4. Graphical Representation of UAD ADSR Envelope**

SG2 supports the Ultrasonic Activity Detection (UAD) feature using some ADSR parameters. Refer to this register map to configure UAD mode.

SG2 in this mode can be referred to as Ultrasonic Activity Generator (UAG). UAG is only available to SG2 CPM. All CPM calculations apply in this mode and the default sample rate is 96ksps meaning this mode collects samples at double the regular ADSR mode.

This mode employs all ADSR registers except the attack, release, and decay timer registers. Registers B0_P23_R104 (0x6C) to B0_P23_111 (0x6F) configure the UAG attack time and registers B0_P23_R104 (0x70) to B0_P23_111 (0x74) configure UAG release time. This mode has no decay time.

UAG attack and release time can be determined the same way. The following uses attack time to demonstrate how to retrieve register values:

$$Attack\_Constant = \quad e^{-\left(\frac{1}{f_S \times RT}\right)} \tag{15}$$

Here RT is a the time constant in ms. This takes 5 times *RT* to acquire desired attack time and to reach 99% of 2Vrms, full-scale. For example, *RT* = *1ms* means a desired time of 5ms. This equation also deducts the attack constant needed to retrieve register coefficient values. Use this formula to retrieve register values:

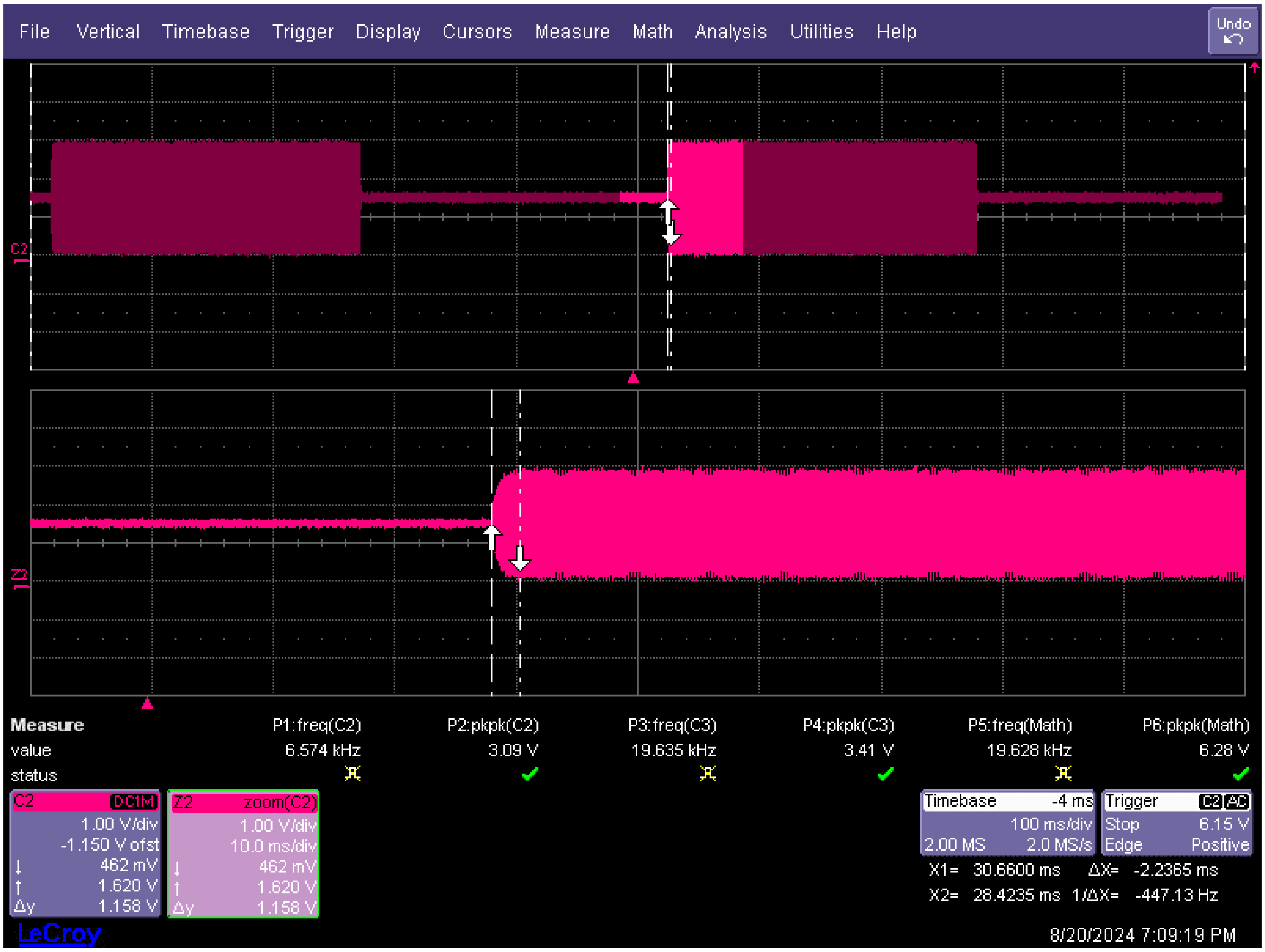
$$Reg.Coeff = DEC2HEX\left(\left(MOD\left(ROUND\left(k \times \quad \left(2^{31}\right), \quad 0\right), \quad 2^{32}\right)\right), 8\right) \tag{16}$$

Where *k* is the attack constant.

The Ultrasonic Activity Detection (UAD) ADSR Mode highlights the UAG ADSR pulse waveform. Reference this UAG/ UAD example script for ultrasonic applications.

```
# Key: w a0 XX YY ==> write to I2C address 0xa0, to register 0xXX, data 0xYY
# # ==> comment delimiter
#
#The following list gives an example sequence of items that must be #executed in the time between
powering the device up and reading data #from the device. Note that there are other valid sequences
depending #on which features are used.
#See the corresponding EVM user guide for jumper settings and audio #connections.
#
# Line-Out Fully-Differential 2-channel : OUT1P_M- Ch1, OUT2P_M- Ch2.
# FSYNC = 48 kHz (Output Data Sample Rate), BCLK = 12.288 MHz (BCLK/FSYNC = 256)
#UAG/UAD Example Script
w a0 00 00 #locate page 0
w a0 01 01 #device reset
w a0 02 09 # come out of sleep mode with VREF and DREG up

w a0 00 17 #locate page 0x17
w a0 7c 14 f1 a6 c6 # 10kHz chirp start frequency

w a0 00 1c #locate page 0x1c
#restart timer
w a0 50 00 00 bb 80     #48k samples for 1s
#sustain timer
```

```
w a0 54 00 00 5d c0     #24000 samples at 500ms
#sustain level
w a0 64 40 00 00 00     #k=0.6

#uag attack
w a0 00 17    #locate page 0x17
#w a0 6c 7D 5C 65 34 #default

#uag release
#w a0 70 7D 5C 65 34 #default

w a0 00 01 #locate page 0x01

w a0 1f 00 #uag on ch1
w a0 20 00 #uag clk cfg

w a0 00 00 #locate page 0x00
w a0 76 08 #en out ch1

w a0 78 02 #en uag
```



| Measure | P1:freq(C2) | P2:pkpk(C2) | P3:freq(C3) | P4:pkpk(C3) | P5:freq(Math) | P6:pkpk(Math) |
|---|---|---|---|---|---|---|
| value | 6.574 kHz | 3.09 V | 19.635 kHz | 3.41 V | 19.628 kHz | 6.28 V |
| status | ✗ | ✓ | ✗ | ✓ | ✗ | ✓ |

C2 DC1M
1.00 V/div
-1.150 V ofst
↓ 462 mV
↑ 1.620 V
Δy 1.158 V

Z2 zoom(C2)
1.00 V/div
10.0 ms/div
↓ 462 mV
↑ 1.620 V
Δy 1.158 V

Timebase -4 ms
100 ms/div
2.00 MS 2.0 MS/s

Trigger C2 AC
Stop 6.15 V
Edge Positive

X1= 30.6600 ms ΔX= -2.2365 ms
X2= 28.4235 ms 1/ΔX= -447.13 Hz

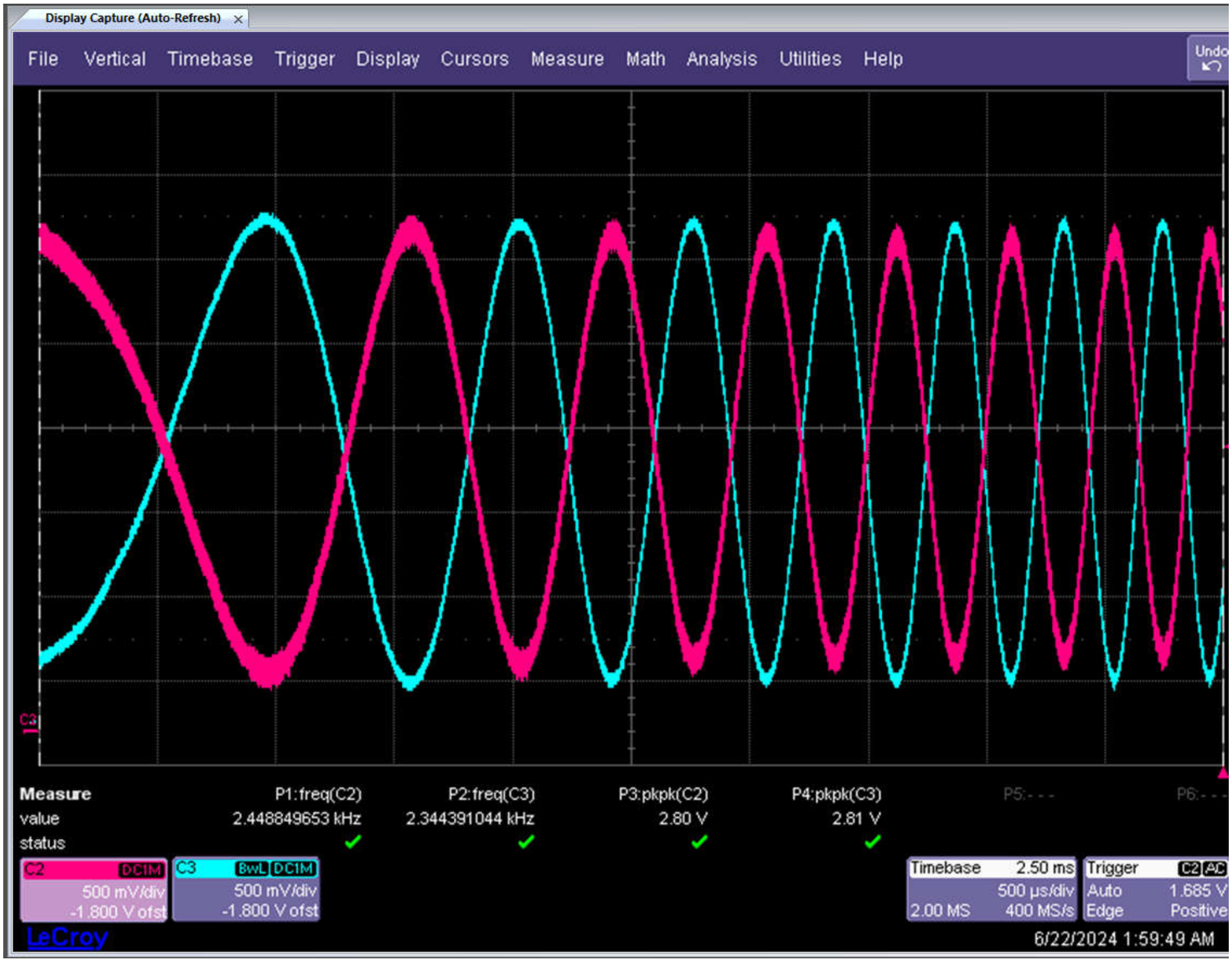
8/20/2024 7:09:19 PM

**Figure 3-5. Ultrasonic Activity Detection Generator Scope Capture**

**Table 3-5. Programmable UAG ADSR Parameters for SG2**

| Parameter | Page Address | Register Address | Default Values |
|---|---|---|---|
| Restart Timer | 0x1C | 0x50 | 100ms at 96Ksps |
| Sustain Timer | 0x1C | 0x54 | 10ms at 96Ksps |
| Sustain Level | 0x1C | 0x64 | 0dB |
| Power delay | 0x17 | 0x74 | 9600 samples |
| Attack Timer | 0x17 | 0x6C | 2.5ms at 96Ksps |
| Release Timer | 0x17 | 0x70 | 2.5ms at 96Ksps |

# 4 Signal Generator 2 (SG2)



```
Measure      P1:freq(C2)      P2:freq(C3)      P3:pkpk(C2)   P4:pkpk(C3)   P5:- - -   P6:- - -
value        2.448849653 kHz  2.344391044 kHz  2.80 V        2.81 V
status       ✓                ✓                ✓             ✓

C2        DC1M   C3   BwL DC1M        Timebase   2.50 ms   Trigger   C2 AC
500 mV/div       500 mV/div                    500 µs/div  Auto      1.685 V
-1.800 V ofst    -1.800 V ofst        2.00 MS  400 MS/s    Edge      Positive
                                                           6/22/2024 1:59:49 AM
```

**Figure 4-1. Scope Capture of SG2 Sample Signal: Differential Chirp Signal at Full-Scale**

## 4.1 Signal Generator 2 Introduction

Signal Generator 2 is synonymous with Chirp Generator and SG2. This generator produces chirps with configurable amplitude and frequency sweeps. SG2 has three different modes namely Manual Mode, Continuous Pulse Mode, and One Shot Mode. The Chirp Start Frequency and Chirp Delta Frequency span 0Hz to 96KHz.

## 4.2 Signal Generator 2 Amplitude

Registers B0_P17_R112 (0x70) to B0_P17_R119 (0x77) configure SG2 amplitude. Refer to Programmable Register Map for SG2 Amplitude and Output Channels.

The methods adopted in Section 2.2 transfer to SG2. Implement these I2C and PPC3 concepts on appropriate SG2 registers.

**Table 4-1. Programmable Register Map for SG2 Amplitude and Output Channels**

| Page | Register | Description | Reset Value |
|---|---|---|---|
| 0x11 | 0x70 | Side Chain DAC Mixer, SG2 to OUT1M coefficient byte [15:8] | 0x00 |
| 0x11 | 0x71 | Side Chain DAC Mixer, SG2 to OUT1M coefficient byte [7:0] | 0x00 |
| 0x11 | 0x72 | Side Chain DAC Mixer, SG2 to OUT1P coefficient byte [15:8] | 0x00 |
| 0x11 | 0x73 | Side Chain DAC Mixer, SG2 to OUT1P coefficient byte [7:0] | 0x00 |
| 0x11 | 0x74 | Side Chain DAC Mixer, SG2 to OUT2M coefficient byte [15:8] | 0x00 |
| 0x11 | 0x75 | Side Chain DAC Mixer, SG2 to OUT2M coefficient byte [7:0] | 0x00 |
| 0x11 | 0x76 | Side Chain DAC Mixer, SG2 to OUT2P coefficient byte [15:8] | 0x00 |
| 0x11 | 0x77 | Side Chain DAC Mixer, SG2 to OUT2P coefficient byte [7:0] | 0x00 |

## 4.3 Signal Generator 2 Frequency

Registers B0_P23_R124 (0x7C) to B0_P23_R127 (0x7F) configure SG2 Chirp Start Frequency. Registers B0_P24_R08 (0x08) to B0_P24_R11 (0x0B) configure SG2 Chirp Delta Frequency. Refer to Table 4-2.

Although amplitude and channel selection methods are the same for SG1 and SG2, both differ in frequency configuration. Chirp Start Frequency and Chirp Delta Frequency make up SG2 frequency parameters. These parameters are 32-bit I2C commands each and are not dependent on each other.

The chirp start frequency is the frequency in the first sample of one second of a sampled signal per an operating sampling rate. This is can be considered the beginning frequency.

The chirp delta frequency is the frequency rise per sample up until the last sample within a defined time period. Per each sample, the next frequency rise is added to the last frequency in the previous sample. This parameter forces frequency sweeps through the start frequency. This can also be configured to not create any sweeps.

Once these parameters are set, this applies to any output channels for SG2. Use these formulas to generate 32-bit I2C scripts to configure SG2 frequency.

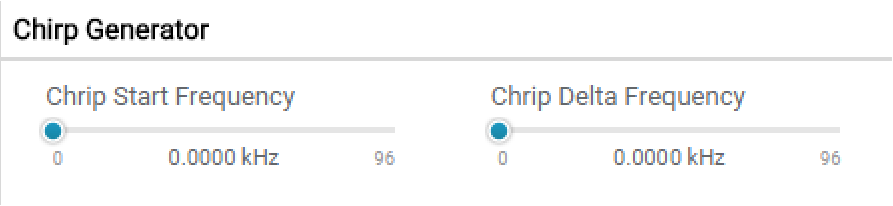
$$Chirp \quad Start \quad Frequency \quad Constant \quad = \frac{2\pi(\mathrm{Chirp \quad Start \quad Frequency})}{\mathrm{F}_{sample\ rate}}, \quad \left(rad\right) \tag{17}$$

$$Chirp \quad Delta \quad Frequency \quad Constant \quad = \frac{2\pi(\mathrm{Chirp \quad Delta \quad Frequency})}{\mathrm{F}_{sample\ rate}}, \quad \left(rad\right) \tag{18}$$

$$Reg.Coeff = DEC2HEX\left(\left(MOD\left(ROUND\left(k \times \quad \left(2^{28}\right), \quad 0\right), \quad 2^{32}\right)\right), 8\right) \tag{19}$$

where *k* is either the Chirp Start Frequency Constant or the Chirp Delta Frequency Constant. This formula applies to generating hex coefficients for both frequency parameters.

In PPC3, SG2 frequency is set using the Chirp Generator frequency scroll. Figure 4-2 is an image of SG2 GUI.



Chrip Start Frequency: 0.0000 kHz

Chrip Delta Frequency: 0.0000 kHz

**Figure 4-2. SG2 Frequency Configuration GUI**

**Table 4-2. Programmable Register Map for SG2 Chirp Start Frequency**

| Page | Register | Description | Reset Value |
|---|---|---|---|
| 0x17 | 0x7C | ASIOUT_BYT1[7:0] | 0x00 |
| 0x17 | 0x7D | ASIOUT_BYT2[7:0] | 0x00 |
| 0x17 | 0x7E | ASIOUT_BYT3[7:0] | 0x00 |
| 0x17 | 0x7F | ASIOUT_BYT4[7:0] | 0x00 |

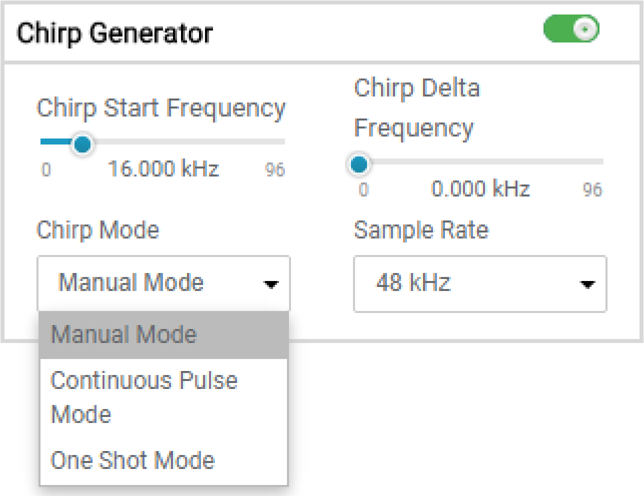
**Table 4-3. Programmable Register Map for SG2 Chirp Delta Frequency**

| Page | Register | Description | Reset Value |
|---|---|---|---|
| 0x18 | 0x08 | ASIOUT__BYT1[7:0] | 0x00 |
| 0x18 | 0x09 | ASIOUT_BYT2[7:0] | 0x00 |
| 0x18 | 0x0A | ASIOUT_BYT3[7:0] | 0x00 |
| 0x18 | 0x0B | ASIOUT_BYT4[7:0] | 0x00 |

## 4.4 Signal Generator 2 Modes

The following sections on SG2 modes describe how the ADSR envelope and SG2 generate different types of chirps. Parameters of the ADSR Envelope are discussed through these sections and in more details in Section 3.

Figure 4-3 shows how to access these modes in PPC3.



Chirp Start Frequency: 16.000 kHz

Chirp Delta Frequency: 0.000 kHz

Chirp Mode: Manual Mode

Sample Rate: 48 kHz

**Figure 4-3. SG2 Mode Selection GUI**

### 4.4.1 Manual Mode

SG2 in manual mode generates a continuous pulse users enable and disable using the ADSR note. This means writing the ADSR note at the beginning and end of the manual mode script. Refer to Section 3.1. Restart and sustain timers are not available here.

In manual mode, pulses generated are sampled into the number of samples based on an operating sampling frequency per second. For example...

$$f_s = \quad 96KHz \tag{20}$$

...means there are 96K samples within a second.

In manual mode, each pulse spans one second and then repeats. Manual mode is subject to aliasing when frequency sweeping. Refer to the formulas below to generate clean signals. These formulas deduct the sample at which aliasing occurs, number of samples required per pulse and time per frequency rise.

$$chirp \;\; start \;\; frequency \; + \; \left(no. \;\; of \;\; samples\right) \times \; chirp \;\; delta \;\; frequency \; < \; \frac{fs}{2}, \;\; Hz \tag{21}$$

$$time \;\; per \;\; delta \;\; frequency \; = \; \frac{delta \;\; frequency}{sampling \;\; frequency}, \;\; sec \tag{22}$$

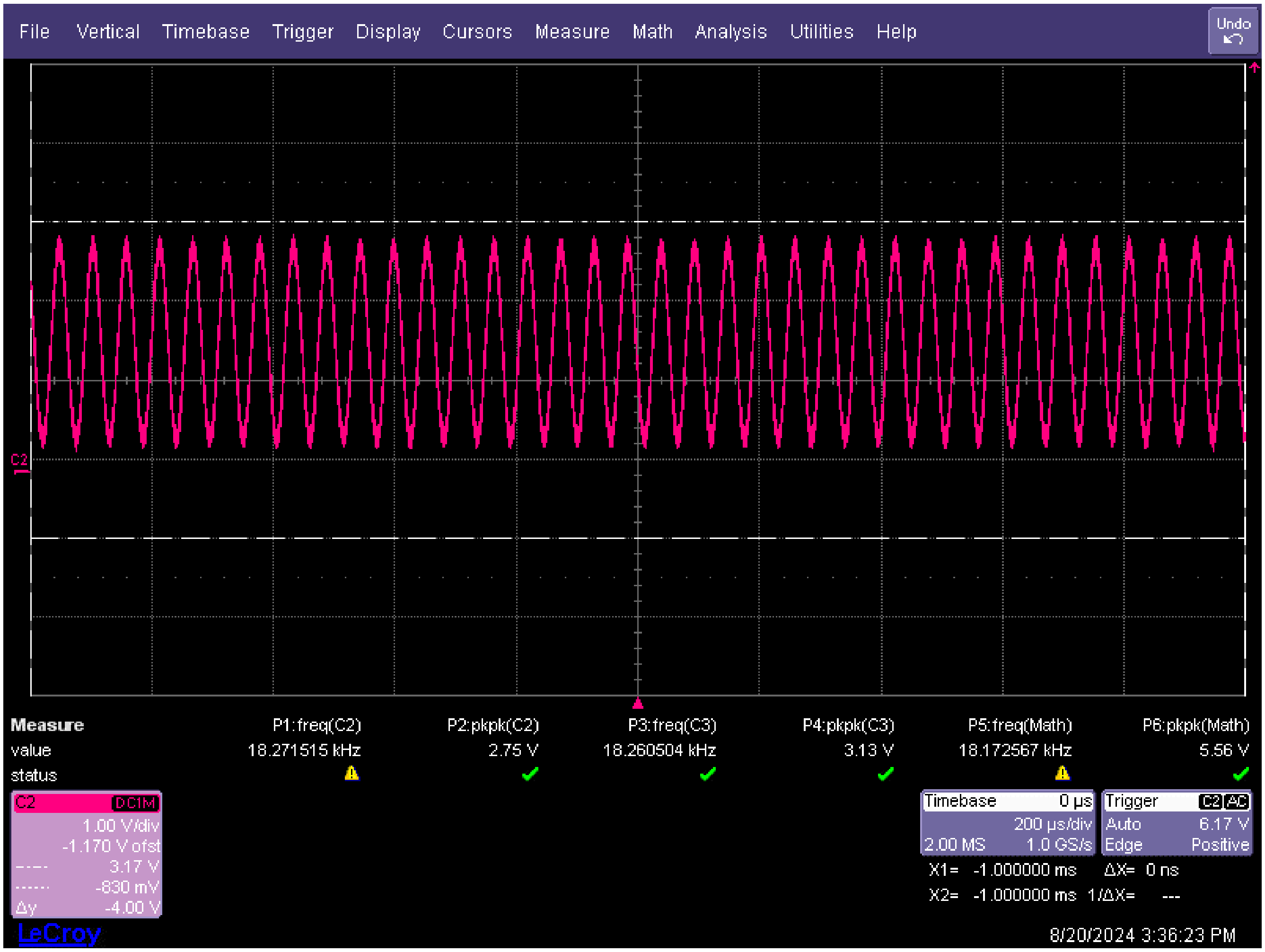
$$frequency \;\; sweep \;\; time \;\; elapsed \; = \; \frac{no. \;\; of \;\; samples \;\; at \;\; desired \;\; end \;\; frequency}{no. \;\; of \;\; samples \;\; per \;\; sec}, \;\; sec \tag{23}$$

This set of formulas are applicable to all SG2 modes.

When programming manual mode, the ADSR acknowledgment script is executed first. Then follows other ADSR parameters and the ADSR enable command at the end of manual mode script.

```
# Key: w a0 XX YY ==> write to I2C address 0xa0, to register 0xXX, data 0xYY
# # ==> comment delimiter
#
#The following list gives an example sequence of items that must be #executed in the time between
powering the device up and reading data #from the device. Note that there are other valid sequences
depending #on which features are used.
#See the corresponding EVM user guide for jumper settings and audio #connections.
#
# Line-Out Fully-Differential 2-channel : OUT1P_M- Ch1, OUT2P_M- Ch2.
# FSYNC = 48 kHz (Output Data Sample Rate), BCLK = 12.288 MHz (BCLK/FSYNC = 256)
####################################################################
#manual mode example script
#acknowledge ADSR engine then enable it with x01 at the end of the script
w a0 00 00    # locate page x00
w a0 01 01    # device reset
w a0 02 09    # come out of sleep mode with VREF and DREG up

w a0 00 01    # locate page 0x01
w a0 2d 04    # enable SG2 only

#Using chirp start frequency formula
w a0 00 17  # locate page 0x17
w a0 7c 02 18 2a 47  # SG2 start frequency at 1 KHz

#Using chirp delta frequency formula
w a0 00 18  # locate page 0x18
w a0 08 00 00 00 44 a1  #delta frequency of 0.5 Hz not causing aliasing

#Acknowledge ADSR envelope
w a0 00 1c  # locate page 0x1c
w a0 40 00 00 00 00  # ADSR note, set 0 when powering
                     # up dac and set it to 1 to enable chirp

#Set sustain and restart timer the same
w a0 50 00 00 00 00  # restart_timer
w a0 54 00 00 00 00  # sustain_timer

w a0 00 17    # locate page 0x17
w a0 74 00 00 00 00  # power up delay
```

```
#SG2 channel selection and amplitude level configuration
w a0 00 11 # locate page 0x11
w a0 70 40 00 40 00  # OUT1
#w a0 74 40 00 40 00 # OUT2

#Output configuration
w a0 00 00    # locate page 0
w a0 76 0c    # enable OUT1
w a0 78 40    # enable all DACs

#Turn on ADSR envelope
w a0 00 1c          #locate page 1c
w a0 40 00 00 00 01 # ADSR note, set 0 when powering
                    # up DAC and set it to 1 to enable chirp
                    # for manual mode
```



```
Measure   P1:freq(C2)     P2:pkpk(C2)   P3:freq(C3)     P4:pkpk(C3)   P5:freq(Math)   P6:pkpk(Math)
value     18.271515 kHz   2.75 V        18.260504 kHz   3.13 V        18.172567 kHz   5.56 V
status    ⚠               ✓             ✓               ✓             ⚠               ✓

C2           DCIM                            Timebase       0 µs    Trigger     C2 AC
          1.00 V/div                                   200 µs/div   Auto       6.17 V
       -1.170 V ofst                         2.00 MS     1.0 GS/s   Edge     Positive
----        3.17 V                            X1=  -1.000000 ms   ΔX=  0 ns
------     -830 mV                            X2=  -1.000000 ms  1/ΔX=  ---
Δy         -4.00 V
                                                              8/20/2024 3:36:23 PM
```

**Figure 4-4. SG2 Manual Mode Scope Capture: 0.5Hz Sweep on 1KHz Signal at 2Vrms, Full-Scale**

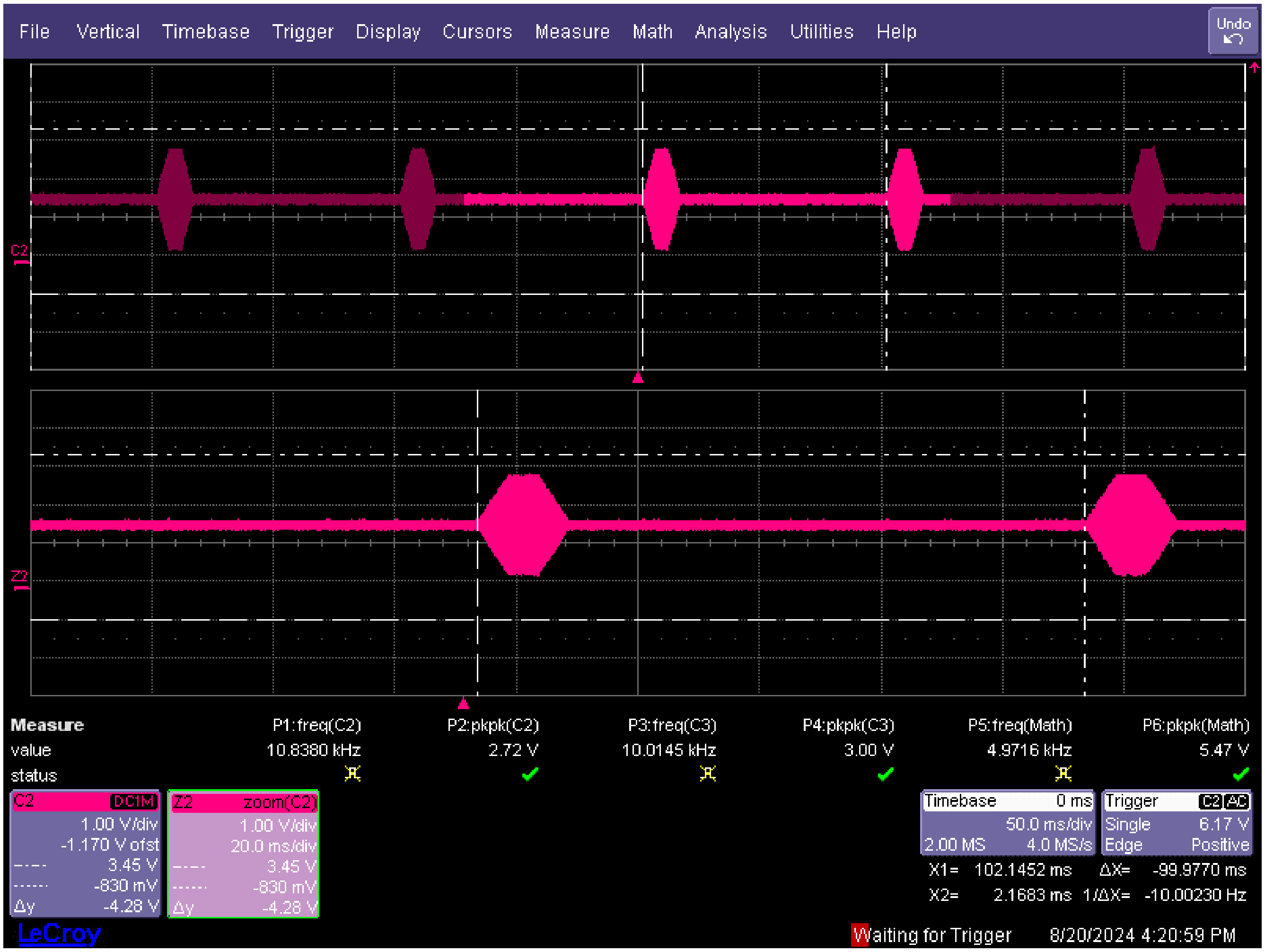
### *4.4.2 Continuous Pulse Mode*

SG2 in Continuous Pulse Mode (CPM) generates repeatable pulses with defined restart and sustain time intervals. These two timers are configured based on the number of samples per second of the operating sampling rate, fs. Refer to Table 3-3.

SG2 in CPM is subject to aliasing if the Nyquist frequency, fs/2 is reached before the end of a sustain time interval. An acceptable end frequency to avoid aliasing can be generated using the timing formulas from Section 4.4.1. These formulas help select an acceptable end-frequency below the Nyquist point.

The ADSR note acknowledgement script must be executed before other ADSR paramters for CPM. However, ADSR enabling script is not needed at the end of CPM script.

```
# Key: w a0 XX YY ==> write to I2C address 0xa0, to register 0xXX, data 0xYY
# # ==> comment delimiter
#
#The following list gives an example sequence of items that must be #executed in the time between
powering the device up and reading data #from the device. Note that there are other valid sequences
depending #on which features are used.
#See the corresponding EVM user guide for jumper settings and audio #connections.
#
# Line-Out Fully-Differential 2-channel : OUT1P_M- Ch1, OUT2P_M- Ch2.
# FSYNC = 48 kHz (Output Data Sample Rate), BCLK = 12.288 MHz (BCLK/FSYNC = 256)
###################################################################

#Continuous pulse mode script
#adsr note at the start needed
#no asdr note at the end needed

w a0 01 01    # device reset

w a0 00 00    # locate page 0x00
w a0 02 09    # come out of sleep mode with VREF and DREG up

w a0 00 01    # locate page 0x01
w a0 2d 04    # enable chirp only

w a0 00 17    # locate page 0x17
w a0 7c 14 f1 a6 c6 # 10kHz chirp start frequency


w a0 00 18    # locate page 0x18
w a0 08 00 00 00 00    # chirp delta frequency of 0 Hz

w a0 00 1c    # locate page 0x1c
w a0 40 00 00 00 00    # adsr_note

#restart
#w a0 50 00 00 bb 80     #48k samples for 1s
w a0 50 00 00 12 c0 # 100ms default

#sustain
#w a0 54 00 00 5d c0     #24000 samples at 500ms
w a0 54 00 00 01 e0 #10ms default

#attack
w a0 58 00 44 52 3f    # default with k=0.00417 @10k.
#w a0 58 00 03 6a 83    # 100ms with k=0.0002085 @10k.

#sustian level
#w a0 64 20 00 00 00 # sustain lvl at k=0.5

#decay timer
#w a0 60 ff bb ad c1 #5ms

#channel selection
w a0 00 11    #locate page 0x11
w a0 70 40 00 40 00    # OUT1
#w a0 74 40 00 40 00    # OUT2

w a0 00 00    # locate page 0x00
w a0 76 0c    # enable 2 DAC channels
w a0 78 40    # enable DAC
```



| Measure | P1:freq(C2) | P2:pkpk(C2) | P3:freq(C3) | P4:pkpk(C3) | P5:freq(Math) | P6:pkpk(Math) |
|---|---|---|---|---|---|---|
| value | 10.8380 kHz | 2.72 V | 10.0145 kHz | 3.00 V | 4.9716 kHz | 5.47 V |
| status | ☒ | ✓ | ☒ | ✓ | ☒ | ✓ |

```
C2          DC1M      Z2       zoom(C2)
     1.00 V/div           1.00 V/div
  -1.170 V ofst          20.0 ms/div
----     3.45 V      ----     3.45 V
------  -830 mV      ------  -830 mV
Δy      -4.28 V      Δy      -4.28 V
```

```
Timebase           0 ms   Trigger        C2 AC
          50.0 ms/div     Single        6.17 V
2.00 MS      4.0 MS/s     Edge        Positive
X1=   102.1452 ms    ΔX=   -99.9770 ms
X2=     2.1683 ms  1/ΔX=  -10.00230 Hz
```

Waiting for Trigger 8/20/2024 4:20:59 PM

**Figure 4-5. SG2 Continuous Pulse Mode Scope Capture: 10KHz Signal at 2Vrms, Full-Scale**

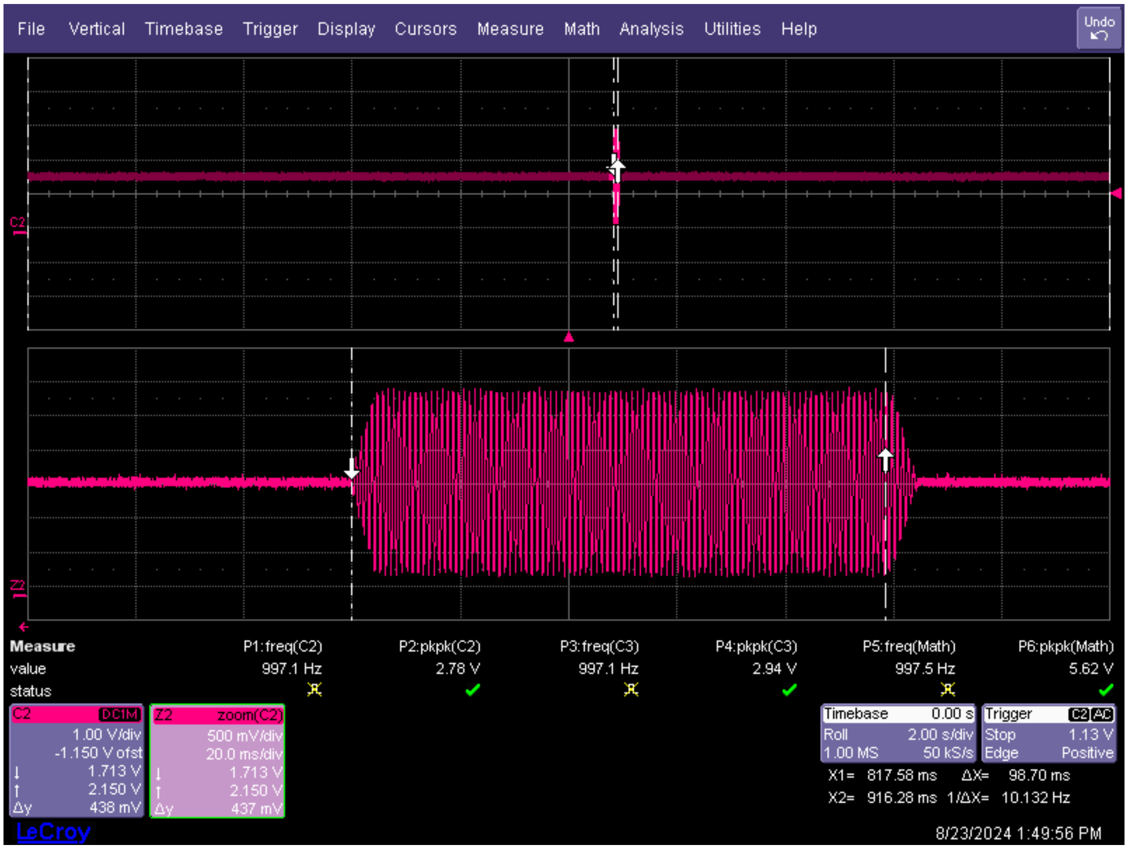
### *4.4.3 One Shot Mode*

SG2 in One Shot Mode generates a single pulse with a defined sustain time interval. Refer to Section 3.2. To initiate the pulse again, users must execute the entire script as is. This mode mostly requires the sustain timer configuration.

There is no restart time configuration in One Shot Mode. Other ADSR parameters applicable to this mode are mentioned in Table 3-4.

```
Key: w a0 XX YY ==> write to I2C address 0xa0, to register 0xXX, data 0xYY
# # ==> comment delimiter
#
#The following list gives an example sequence of items that must be #executed in the time between
powering the device up and reading data #from the device. Note that there are other valid sequences
depending #on which features are used.
#See the corresponding EVM user guide for jumper settings and audio #connections.
#
# Line-Out Fully-Differential 2-channel : OUT1P_M- Ch1, OUT2P_M- Ch2.
# FSYNC = 48 kHz (Output Data Sample Rate), BCLK = 12.288 MHz (BCLK/FSYNC = 256)
###################################################################
#one shot mode
#repeat script to generate pulse again

w a0 01 01    # device reset

w a0 00 00    # locate page 0x00
w a0 02 09    # come out of sleep mode
                # with VREF and DREG up
```

```
w a0 00 01    # locate page 0x01
w a0 2d 04    # enable the chirp only

w a0 00 17    # locate page 0x17
w a0 7c 02 18 2a 47    # chirp start frequency, 1khz

w a0 00 18    #locate page 0x18
w a0 08 00 35 9d d3    # chirp delta frequency of 100 Hz

w a0 08 00 00 00 00    # chirp delta frequency is set to zero

w a0 00 1c    #locate page 0x1c
w a0 40 ff ff ff ff # adsr_note; write this code to generate a chirp
                    #and initiate the signal again with 0x01,
                    #the signal will last based on the sustain timer set

#restart
w a0 50 ff ff ff ff  # restart_timer

#sustain
#w a0 54 00 00 5d c0    #24000 samples at 500ms
w a0 54 00 00 12 c0 # 480 samples 100ms default


w a0 00 17    # locate page 0x17
w a0 74 FF FF FF FF # power up delay continued

w a0 00 11 # locate page 0x11
w a0 70 3F FF 3F FF    # Please refer the table to set volumes #accordingly
w a0 74 0F FF 0F FF    # Please refer the table to set volumes        #accordingly

w a0 00 00    # locate page 0x00
w a0 76 0c    # enable 2 DAC channels
w a0 78 40    # enable DAC
w a0 40 00 00 00 01    #adsr enable note.
```



| Measure | P1:freq(C2) | P2:pkpk(C2) | P3:freq(C3) | P4:pkpk(C3) | P5:freq(Math) | P6:pkpk(Math) |
|---|---|---|---|---|---|---|
| value | 997.1 Hz | 2.78 V | 997.1 Hz | 2.94 V | 997.5 Hz | 5.62 V |
| status | ☒ | ✓ | ☒ | ✓ | ☒ | ✓ |

C2 DC1M
1.00 V/div
-1.150 V ofst
↓ 1.713 V
↑ 2.150 V
Δy 438 mV

Z2 zoom(C2)
500 mV/div
20.0 ms/div
↓ 1.713 V
↑ 2.150 V
Δy 437 mV

Timebase 0.00 s
Roll 2.00 s/div
1.00 MS 50 kS/s

Trigger C2 AC
Stop 1.13 V
Edge Positive

X1= 817.58 ms ΔX= 98.70 ms
X2= 916.28 ms 1/ΔX= 10.132 Hz

8/23/2024 1:49:56 PM

**Figure 4-6. SG2 One Shot Mode Scope Capture: 1KHz Signal at 2Vrms, Full-Scale**

## 5 Summary

Signal Generator 1 (SG1) and Signal Generator 2 (SG2) are both powerful generators with distinct capabilities. SG1 produces simple sine wave pulses while SG2 is the more flexible generator that uses the ADSR envelope. SG2 can produce signals that undergo frequency and amplitude sweeps but not SG1. SG1 is limited to simple sine waves. Navigation between the SG2 modes through the script configuration is easy, and SG2 has a added ultrasonic operation mode named Ultrasonic Activity Detection for audio applications.

# IMPORTANT NOTICE AND DISCLAIMER

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

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

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

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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

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