C2000 ADC

Hardware Oversampling and Undersampling

Hadi Moradi

C2000 Application Engineer

Oversampling: Overview

What is Oversampling?

- Oversampling is process of sampling the input signal at a higher rate than the Nyquist frequency
  - Increases the signal-to-noise ratio (SNR)
  - Reduces RMS value of the quantization noise
  - Increases Resolution
  - Relaxes the requirements on the antialiasing filter

System benefits

- Improves the performance in digital power and motor control applications by removing the noise.

Oversampling: Overview

Methods for implementing oversampling on MCUs

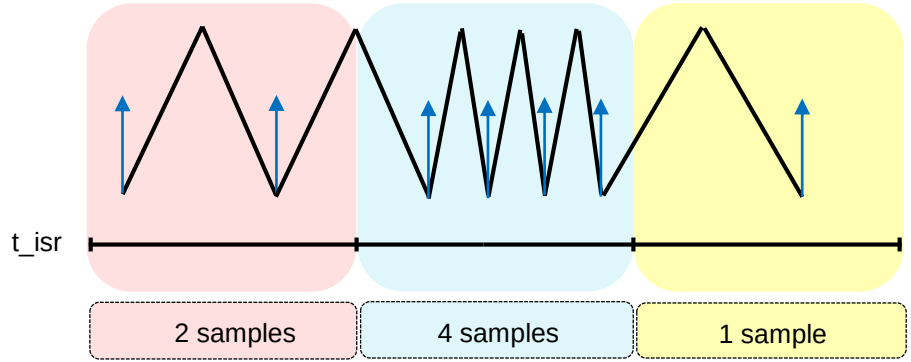
- Software Oversampling
- Hardware Oversampling

System benefits of using hardware oversampling

- Saves lots of CPU cycles
- Saves memory from not having to allocate buffers for oversamples
- Lower power consumption compared with the software-based implementation

Oversampling: Aggregation

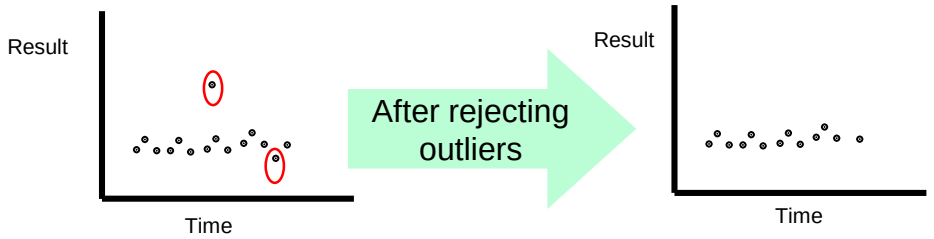
- Accumulating the successive results
  - Accumulation can be ended after specified number of samples reached
    - $Result = \frac{1}{RS}\sum_{0}^{N-1} ADCRESULT$
    - RS is right bit shift. (up to 10 bits)
    - N is oversampling factor (sample count is a power of 2, and from 2 up to 1024)
  - Accumulation can be asynchronous (for variable frequency control and fixed frequency ISR)



Oversampling: Aggregation

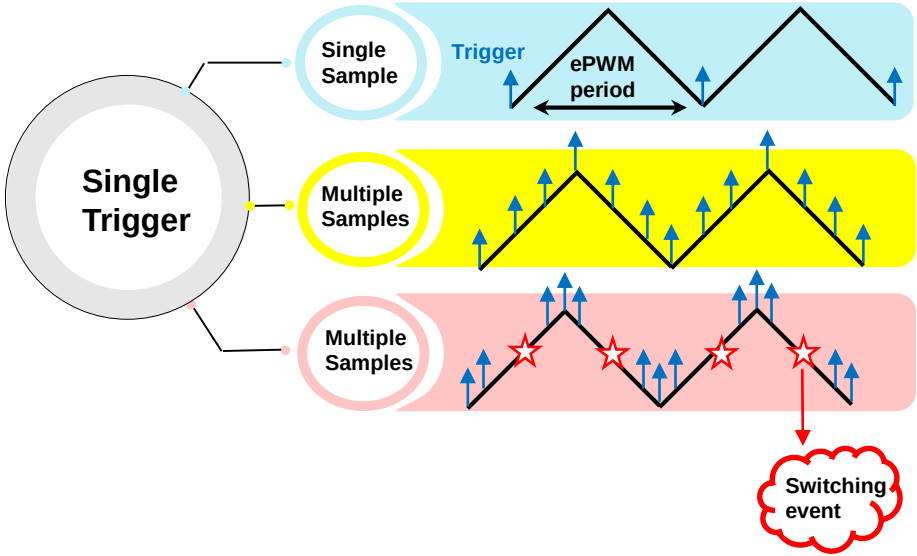
- Recording Min, Max, and Sum of successive results
  - Useful for peak-detection applications, automatic-tuning loops, and system calibration algorithms
  - Once right shifted, will be average with min/max excluded (worst outliers excluded)
  - Better noise rejection with lower sample count, leading to reduced latency

- Outlier data point rejection
  - Eliminates the largest and smallest samples



Oversampling: Complex Triggering Patterns

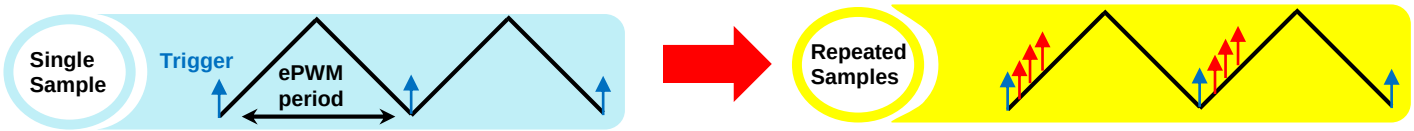
- Accumulate multiple results from multiple trigger sources
  - Get an average reading over one or more ePWM periods
  - Uneven spacing for avoiding noise due to switching



Oversampling: Complex Triggering Patterns

- Accumulate multiple results from multiple trigger sources
  - Get an average reading over one or more ePWM periods
  - Uneven spacing for avoiding noise due to switching

- Accumulate multiple results from one trigger
  - Increases resolution for current and voltage measurements



Undersampling: Overview

What is Undersampling?

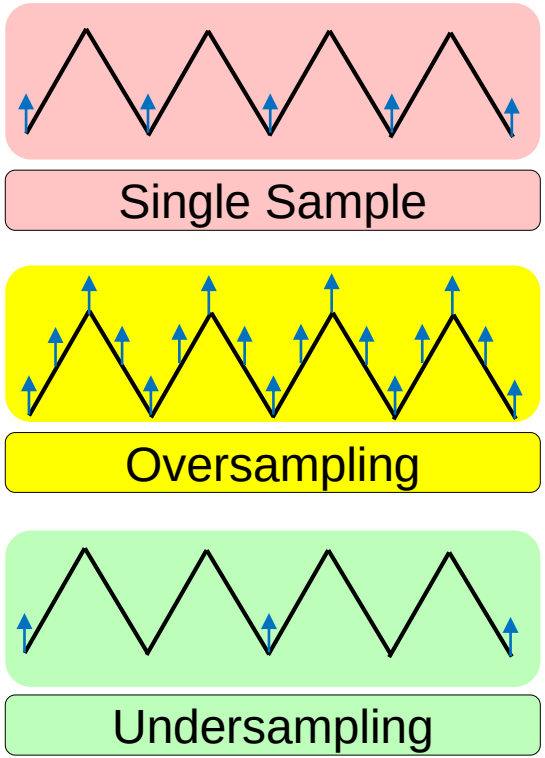
- When the sampling frequency is less than twice the maximum frequency component in the signal.
- Useful for low-frequency signals like temperature, where using the charge sharing circuit with high sampling frequency leads to higher error.

How Hardware Undersampling works?

- Ability to scale ADC triggers down for specific SOCs

System benefits of using hardware undersampling

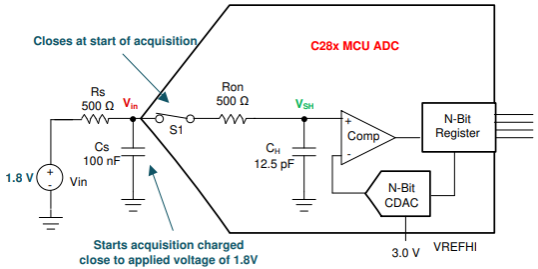
- Saves lots of CPU cycles via HW mechanism
- Allows for charge-sharing input designs to save signal condition cost



Undersampling: Overview

Charge Sharing Operation

- At the beginning of each sample, the charge between Cs and CH quickly equalizes
- In the time between samples, Cs is recharged from the source voltage through Rs



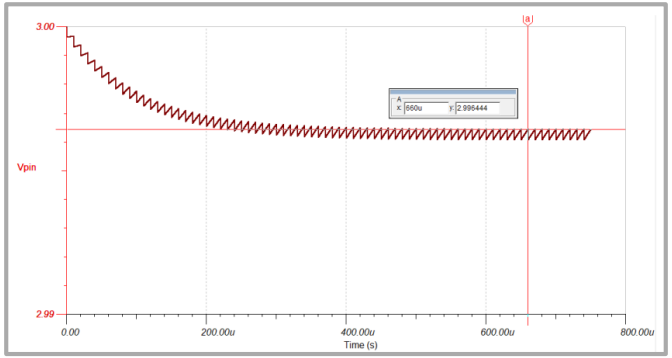
Undersampling: Overview

Charge Sharing Operation

- At the beginning of each sample, the charge between Cs and CH quickly equalizes
- In the time between samples, Cs is recharged from the source voltage through Rs

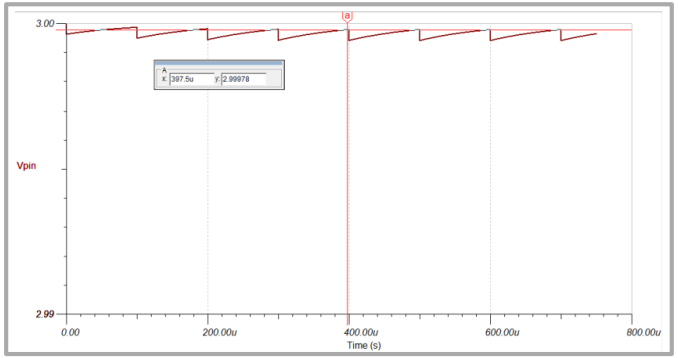
Problem Statement

- The ADC fast sampling leads to the higher error



Sample rate:100kHz
Error: ~ 3.5mV

Reduced sample rate

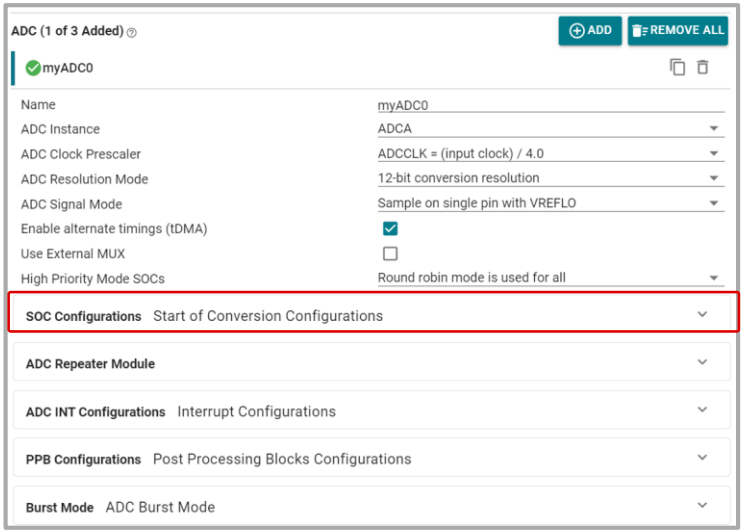


Sample rate:10kHz
Error : ~ 200uV

Oversampling and Undersampling: Using SysConfig GUI

Steps to setup the Repeater Module:

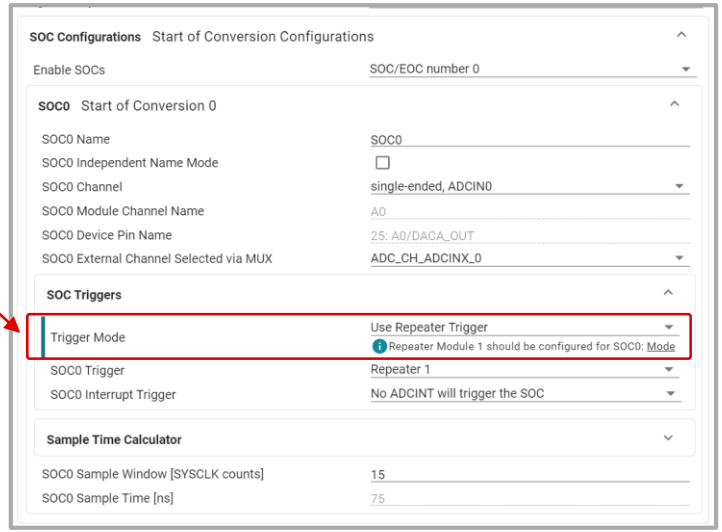
1) From ADC module, click on “SOC Configuration”



SOC Configuration

Trigger Mode

2) Choose “Use Repeater Trigger” for Trigger Mode

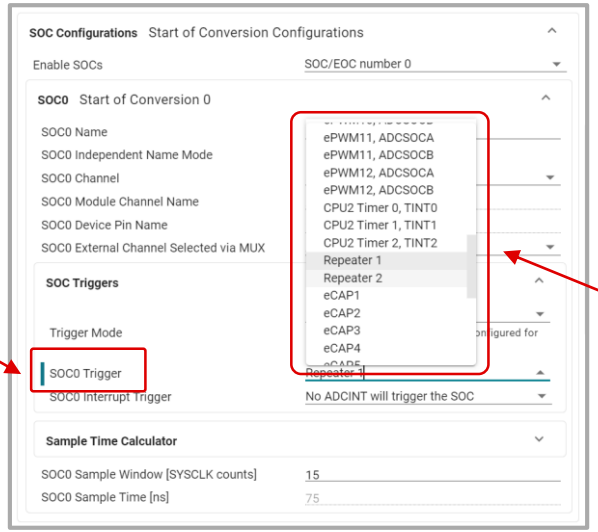


Oversampling and Undersampling: Using SysConfig GUI

Steps to setup the Repeater Module:

3) Set “Repeater 1” or “Repeater 2” from SOCx Trigger options

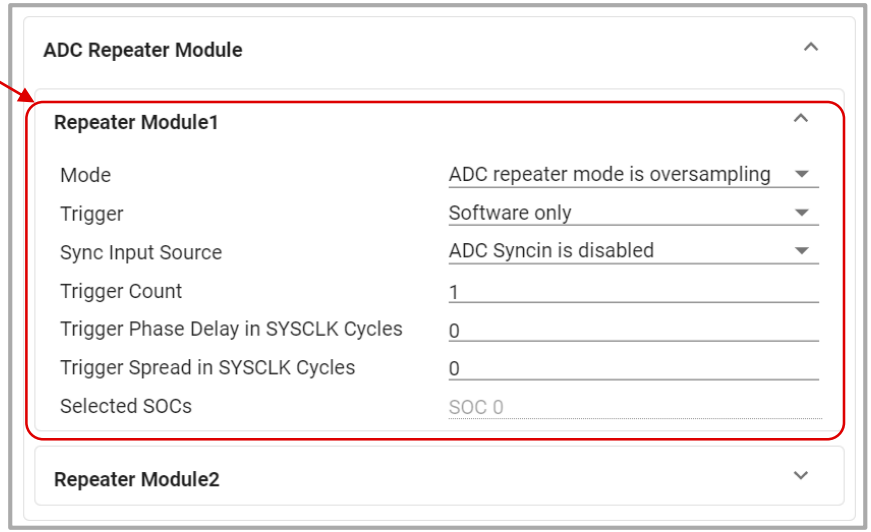
4) Go to “ADC Repeater Module” and set Mode, Trigger, …



SOCx Trigger

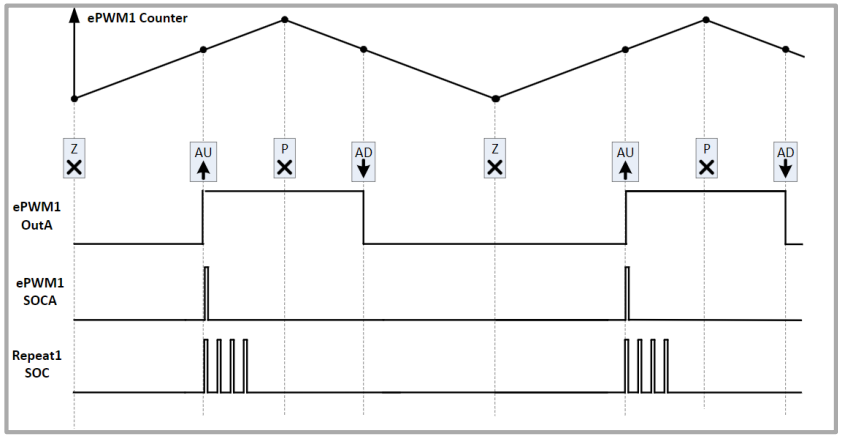
Repeater Module 1

SOCx Trigger Options

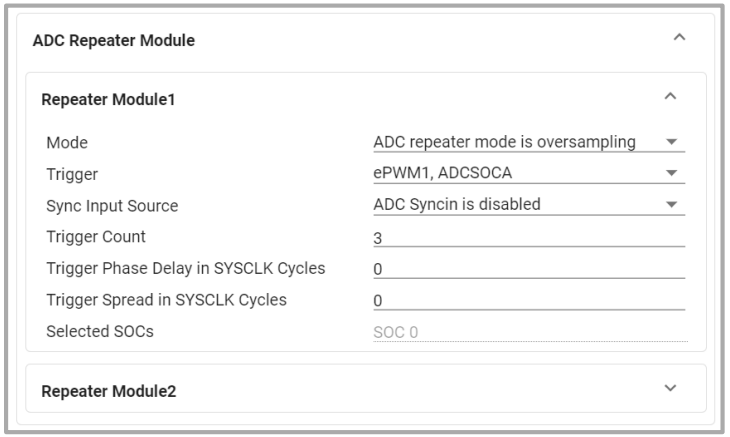


Oversampling and Undersampling: Using SysConfig GUI

- Example 1: Oversampling ADC Trigger (MODE = Oversampling, TRIGGER = ePWM SOCA, NSEL = 3, PHASE = 0, SPREAD = 0)



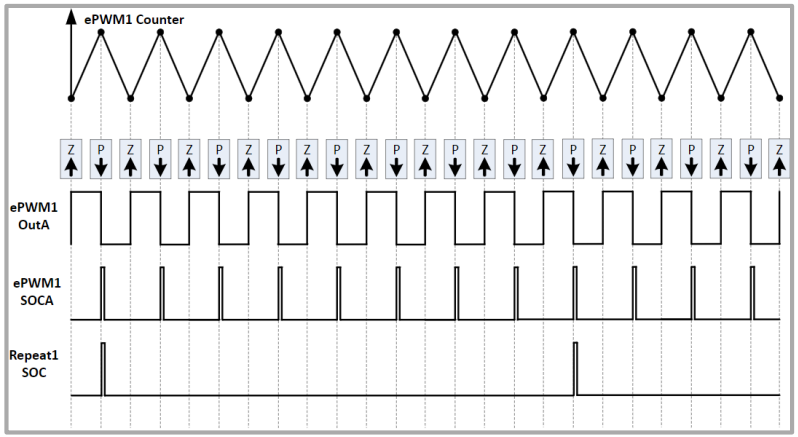
Oversampling Trigger Scheme



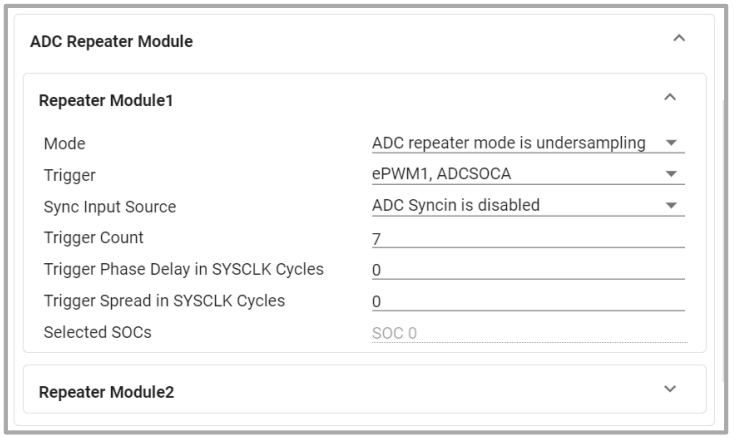
SysConfig GUI Configuration

Oversampling and Undersampling: Using SysConfig GUI

- Example 2: Underrsampling ADC Trigger (MODE = Undersampling, TRIGGER = ePWM SOCA, NSEL = 7, PHASE = 0, SPREAD = 0)



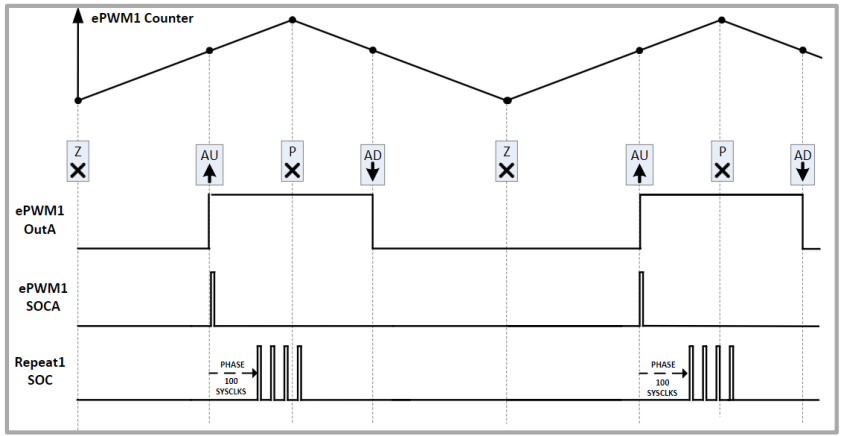
Undersampling Trigger Scheme



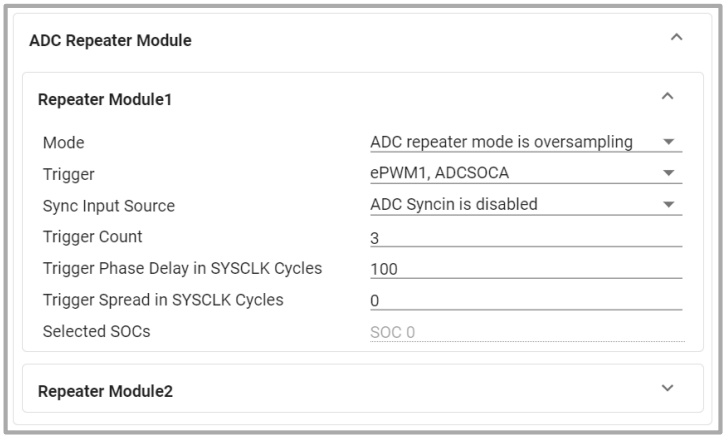
SysConfig GUI Configuration

Oversampling and Undersampling: Using SysConfig GUI

- Example 3: Oversampled ADC Trigger with Phase Delay (MODE = Oversampling, TRIGGER = ePWM SOCA, NSEL = 3, PHASE = 100, SPREAD = 0)



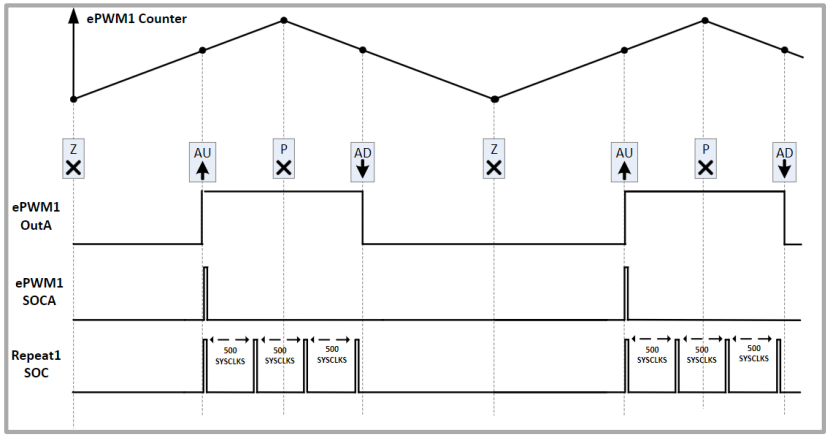
Oversampling Trigger Scheme



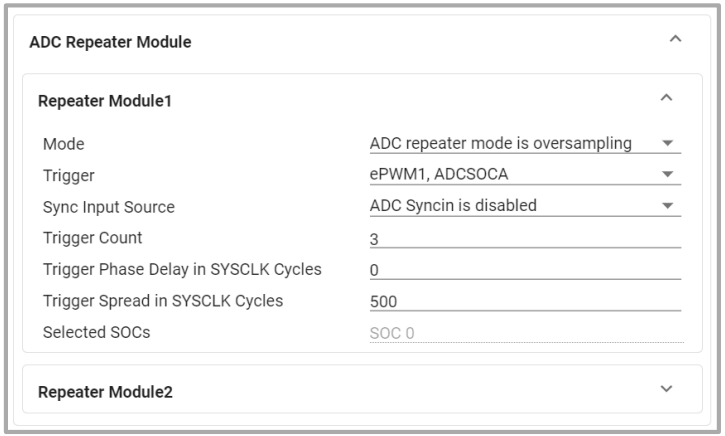
SysConfig GUI Configuration

Oversampling and Undersampling: Using SysConfig GUI

- Example 4: ADC Repeated Trigger with Sample Spread (MODE = Oversampling, TRIGGER = ePWM SOCA, NSEL = 3, PHASE = 0, SPREAD = 500)



Oversampling Trigger Scheme



SysConfig GUI Configuration

Additional Resources

- ADC Module in C2000 Academy
- ADC Oversampling Application Note
- C2000 Real-Time Control MCU Peripherals reference guide
- C2000 SysConfig Video Series
- Charge-Sharing Driving Circuits for C2000 ADCs (using PSPICE-FOR-TI simulation tool)
- TI Precision Lab ADC Series