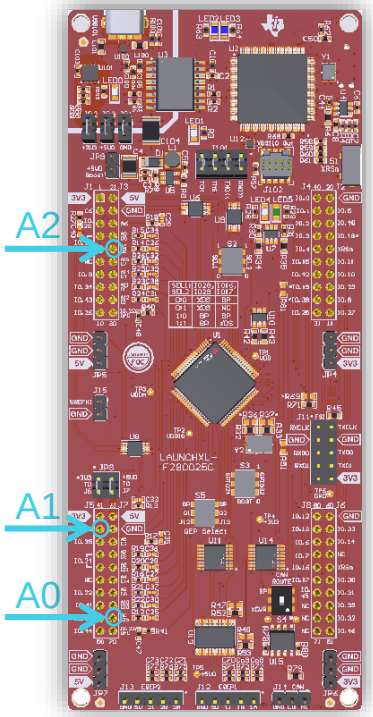
C2000 SysConfig Demonstration

GUI-Based Configuration Tool for C2000 MCUs

Walkthrough of Initializing and C2000 Configuring Device Using SysConfig

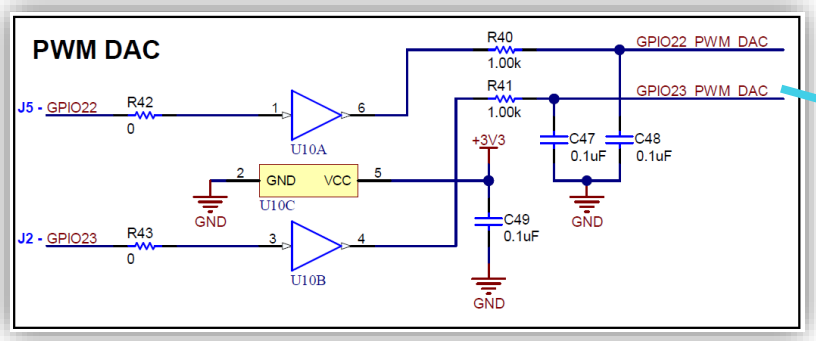
SysConfig Demo Overview and Objective

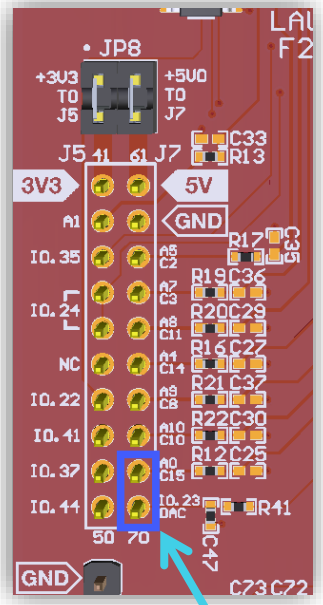
- Import an existing ADC example into Code Composer Studio
  - adc_ex13_soc_oversampling
    - Use EPWM1 to trigger 6 ADC conversions @ 50kHz
      - adcAResult0 = A0 Conversion
      - adcAResult1 = A1 Conversion
      - adcAResult2 = Average of 4 x A2 Conversions (Oversampled)
- Use SysConfig tool to easily modify the example project
  - Leverage device hardware to generate analog signals and route them to ADC pins
- Hardware: LAUNCHXL-F280025C

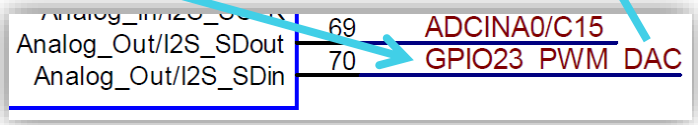


LaunchPad Hardware Overview

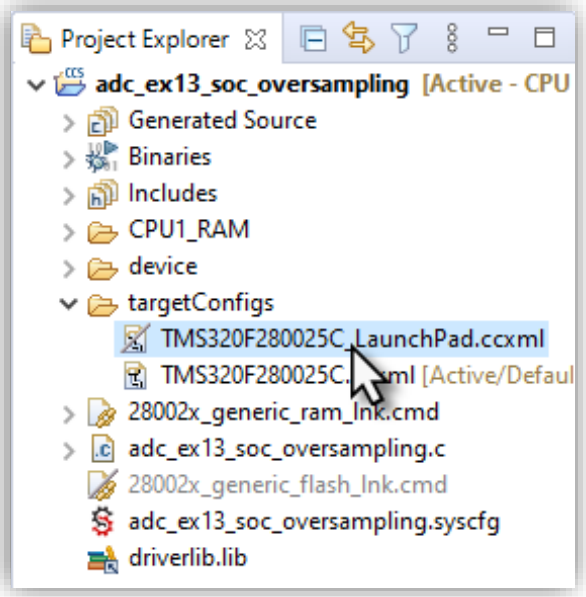
- All LaunchPads include some form of a DAC signal
  - Device DAC
  - PWM-DAC
    - Transforms an ePWM signal into a DC voltage using buffer and low-pass RC filter circuitry
- Connect DAC pin to the A0 pin using a jumper wire
  - Drive DAC and allow ADC A0 to sample it



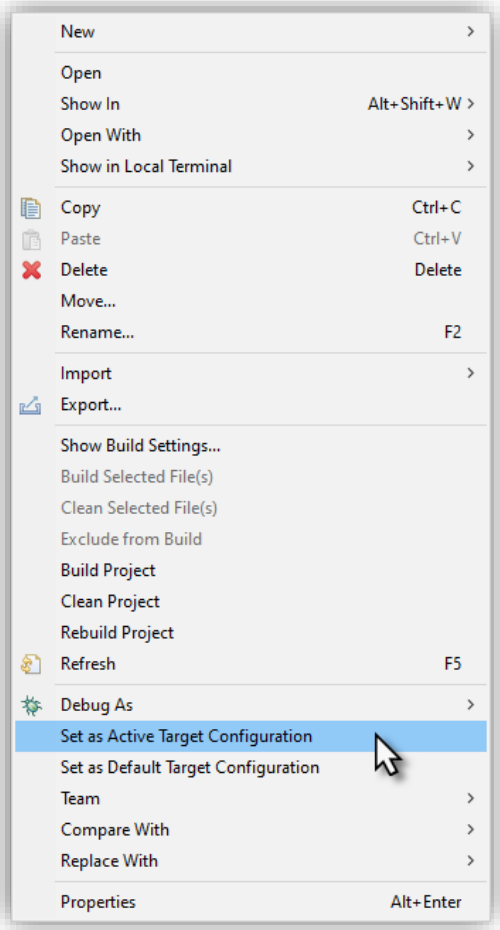




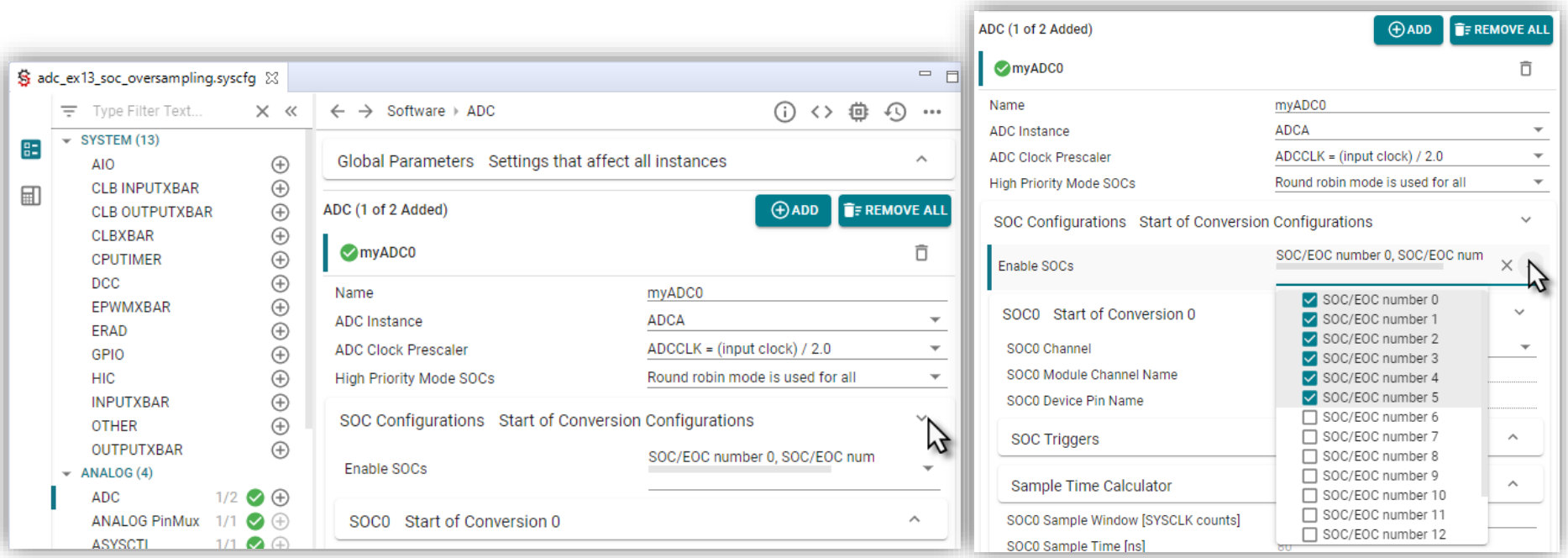
Set Active Target Configuration



Right click on desired target

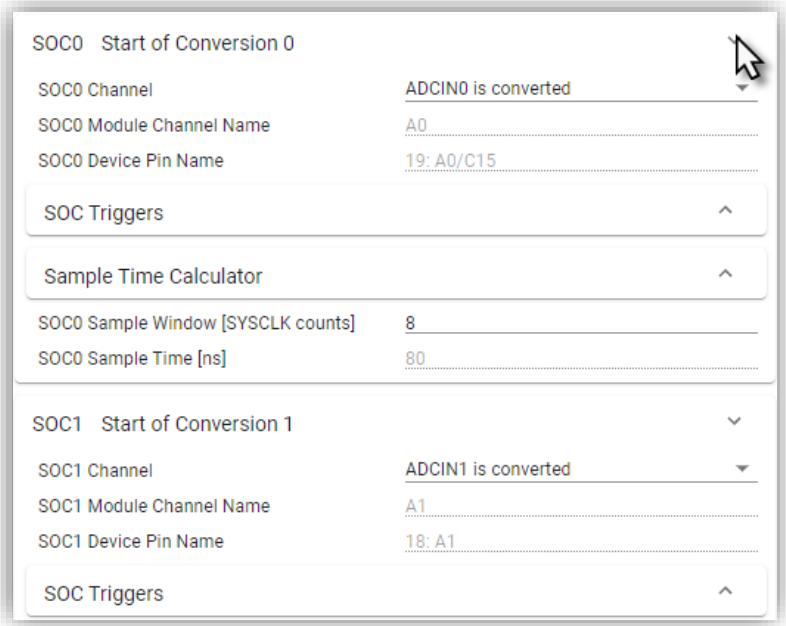


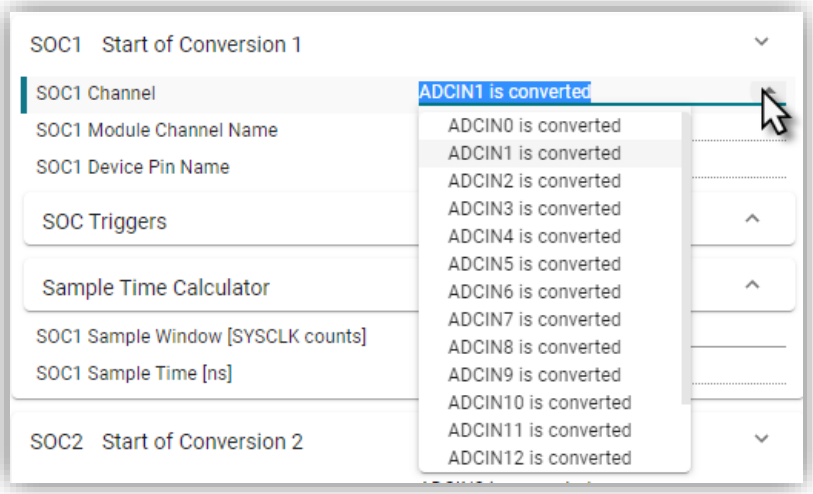
Change SOC0 to Channel A0 (1)



- Expand *SOC Configurations* to confirm the enabled SOCs

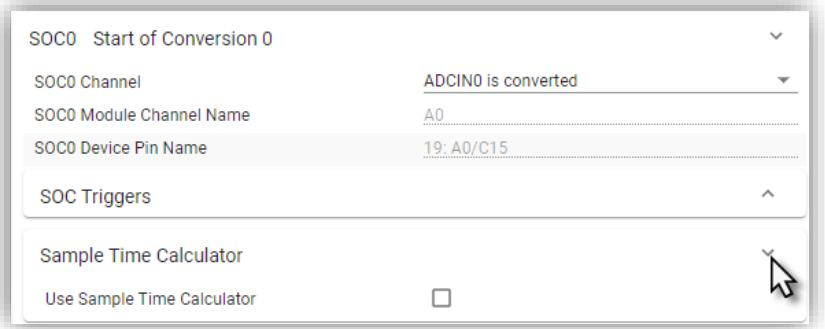
Change SOC0 to Channel A0 (2)



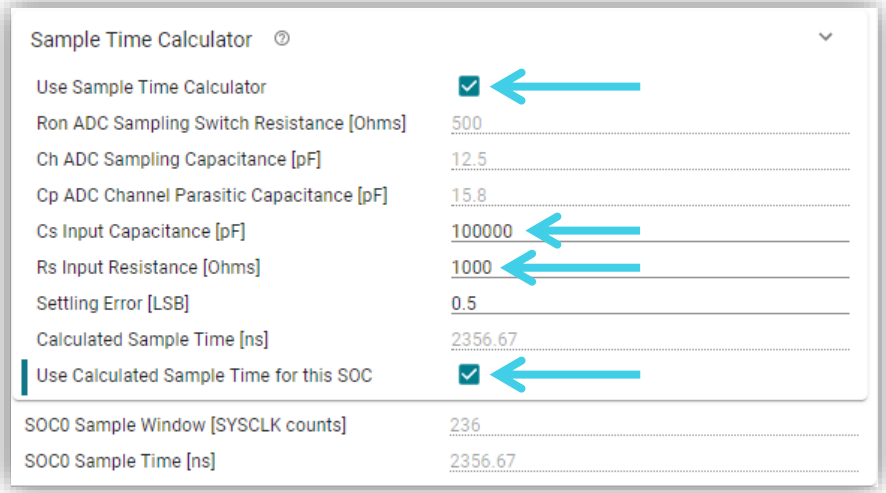


- Expand SOC0 configuration and change the Channel to ADCIN0

Change SOC0 to Channel A0 (3)



- Expand Sample Time Calculator



- Check *Use Sample Time Calculator*
- Input Cs = 0.1uF and Rs = 1kΩ
- Check *Use Calculated Sample Time for this SOC*

Change SOC1-5 to Channel A0

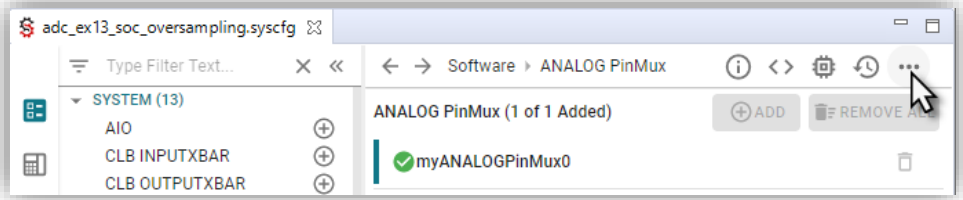
- Repeat SOC0 steps on SOC1 through SOC5

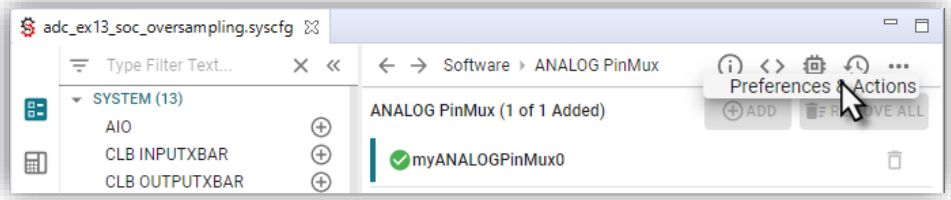
Configure MUX for PWM DAC (1)

- GPIO23 is MUX'd with EPWM4_B (Pin 65 of 80-pin device on LP)
- Click on 3-dot settings shortcut
- Click on *Preferences & Actions*

Table 6-3. Digital Signals

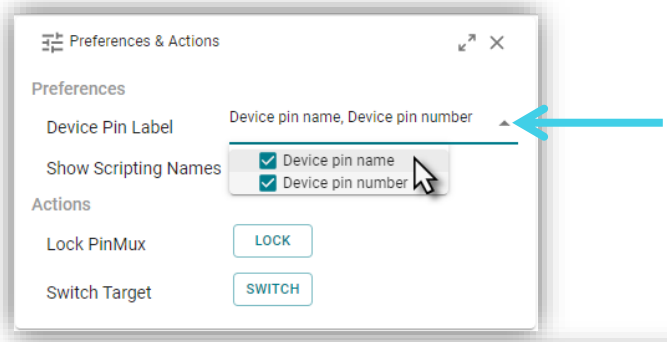
SIGNAL NAME	PIN TYPE	DESCRIPTION	GPIO	80 QFP	64 QFP	48 QFP
ADCSOCAO	O	ADC Start of Conversion A for [illegible]	33, 9	38, 58	32, 47	25
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
EPWM4_A	O	ePWM-4 Output A	22, 6	67, 80	56, 64	48
EPWM4_B	O	ePWM-4 Output B	23, 7	65, 68	54, 57	43
EPWM5_A	O	ePWM-5 Output A	16, 8	39, 58	33, 47	26





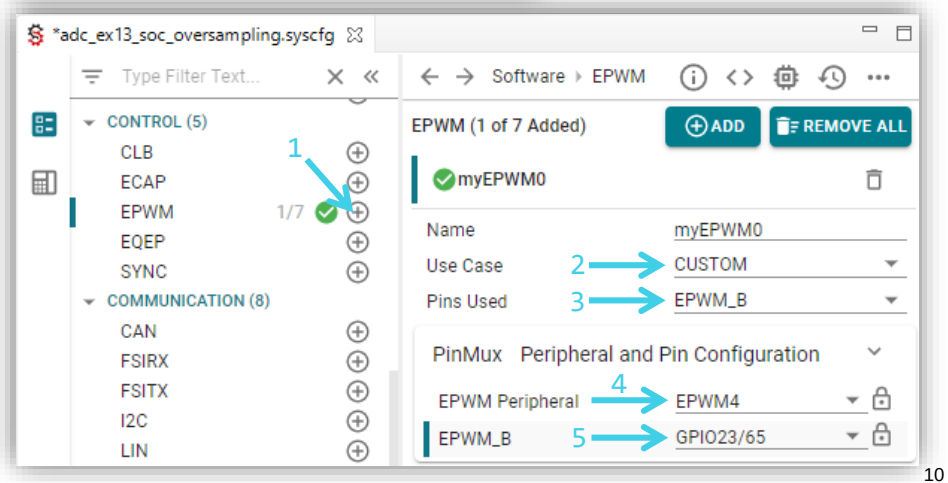
Configure MUX for PWM DAC (2)

- Expand *Device Pin Label* setting
- Check *Device pin name* option
- Close *Preferences & Actions*



Configure EPWM PinMux:

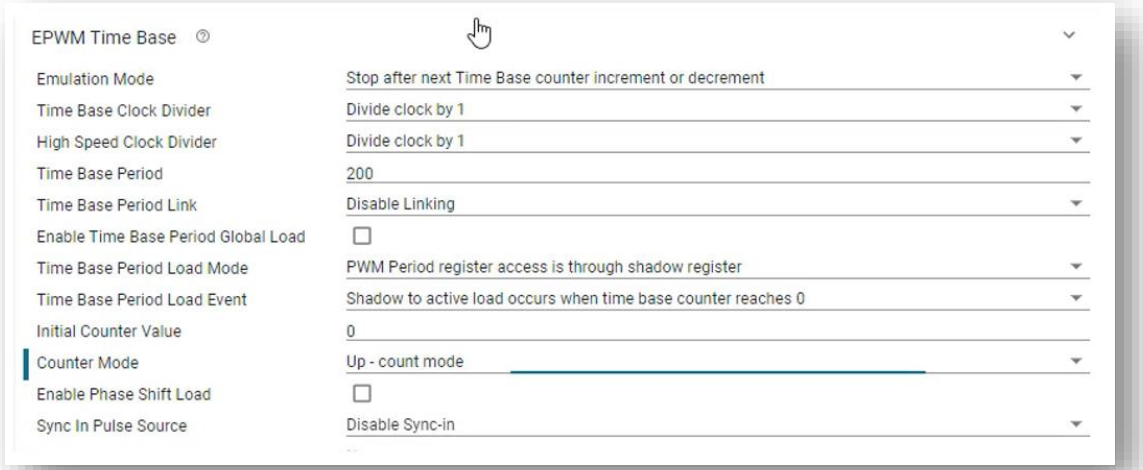
1. Add an EPWM configuration under Control >> EPWM >> +
2. CUSTOM
3. EPWM_B
4. EPWM4
5. GPIO23/65



Configure Time Base for PWM DAC

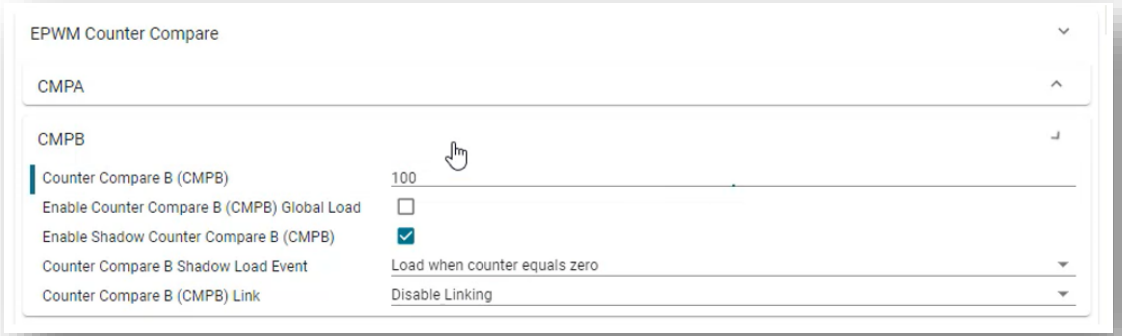
Configure EPWM Time Base

- Period of 200
- Change HS Clock Divider to Divide Clock by 1
- Up-Count Mode



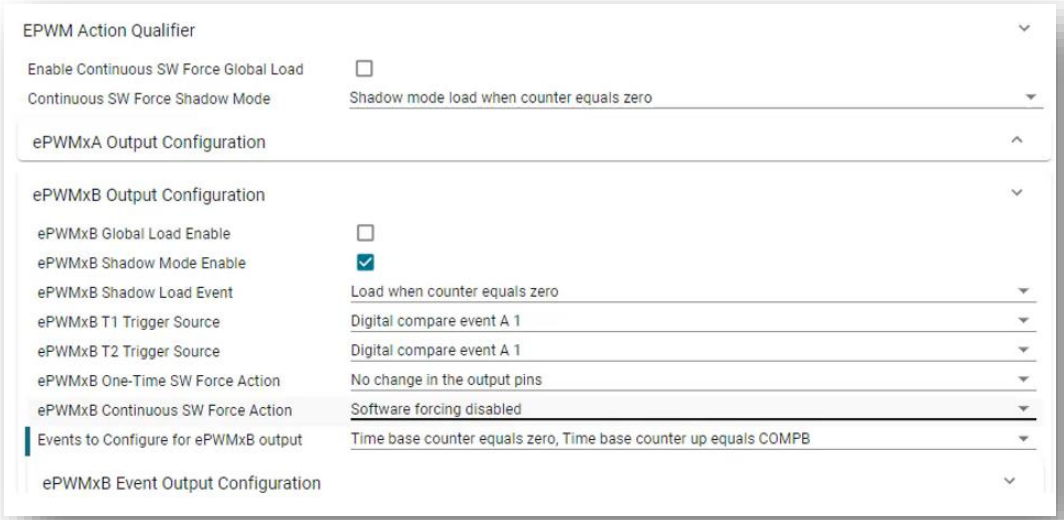
Configure Counter Compare for PWM DAC

Configure Counter Compare
- CMPB Value of 100

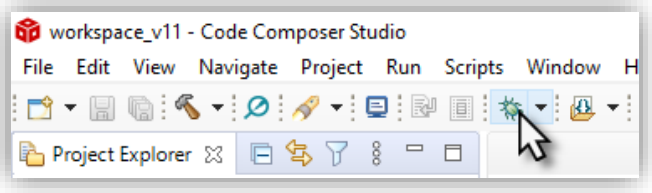


Configure Action Qualifier for PWM DAC

Configure Action Qualifier
- Enable ePWMxB Shadow Mode
- Action Events
  - Time Base Equals Zero
    - Trigger Pin High
  - Time Base Up Equals CMPB
    - Trigger Pin Low



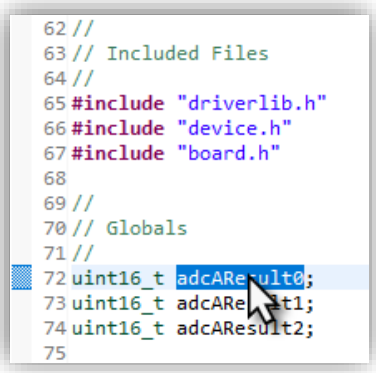
Debug Project



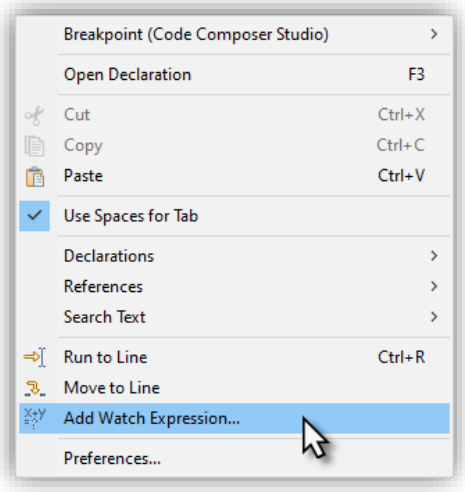
Clicking Debug Icon Will Do the Following:

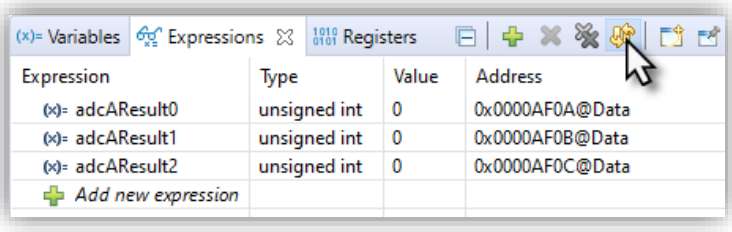
- Compile and link the active project
- Connect to the target specified in the .ccxml
- Load the generated .out file to the target
- Run to main and halt

Add Variables to Watch Expressions Window



1. Select variable
2. Right click





Enable Continuous Refresh

Helpful SysConfig Resources

- Test out SysConfig in the Cloud
- Download Standalone SysConfig Tool
- Learning Material
  - Application Report: C2000 SysConfig
  - SysConfig Training Module and Hands-On Lab in C2000 Academy
  - Speed Up Development With C2000™ Real-Time MCUs Using SysConfig
  - C2000 SysConfig Software Guide

Check Video Description for Additional Resources