

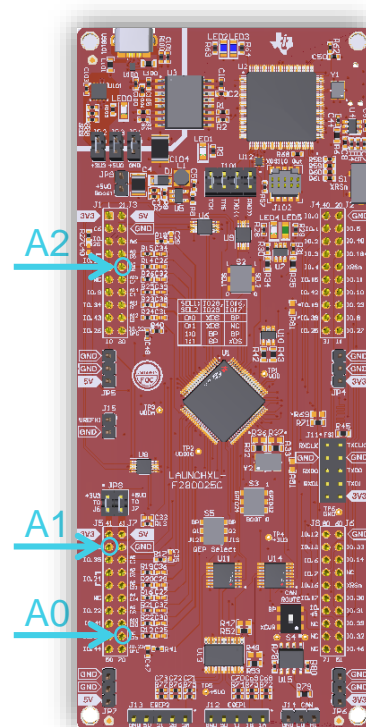
C2000 SysConfig Demonstration

GUI-Based Configuration Tool for C2000 MCUs

Walkthrough of Initializing and C2000 Configuring Device Using SysConfig

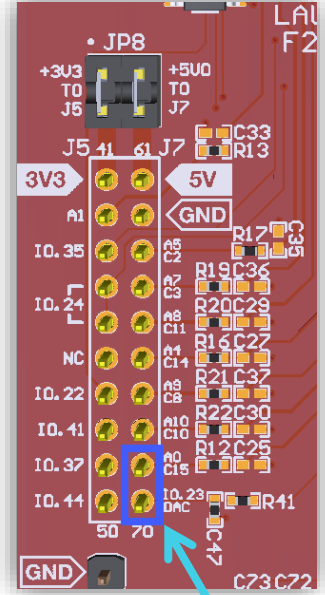
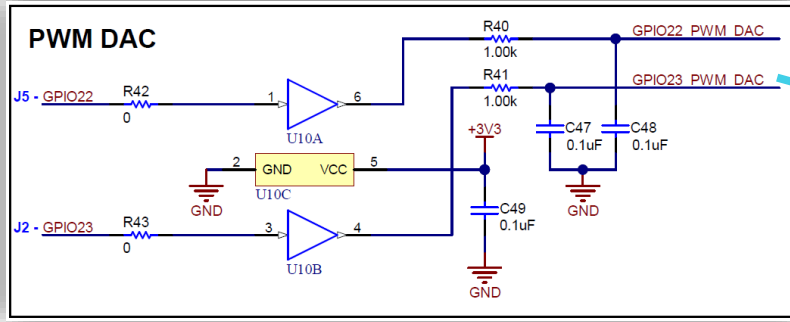
SysConfig Demo Overview and Objective

- 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



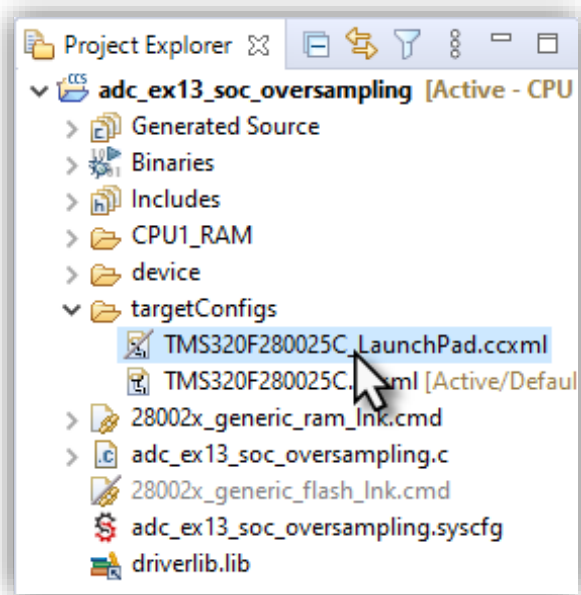
LaunchPad Hardware Overview

- 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

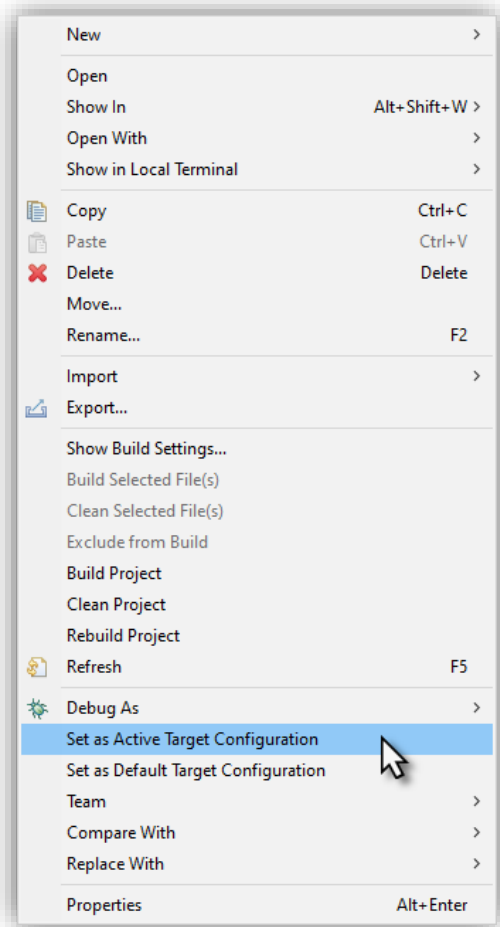


ADCINA0/C15	69
GPIO23 PWM DAC	70

Set Active Target Configuration



Right click on desired target



Change SOC0 to Channel A0 (1)

The screenshot shows the TI Configurator interface for an ADC configuration. The left sidebar lists various system components, with 'ADC' selected under 'ANALOG (4)'. The main window displays the configuration for 'myADC0'. The 'SOC Configurations' section is expanded, showing 'SOC0 Start of Conversion 0'. A dropdown menu is open, showing a list of 'SOC/EOC number' options from 0 to 12. The 'Enable SOC0' checkbox is checked, and the 'SOC/EOC number 0' option is selected. The 'SOC0 Channel' is set to 'Start of Conversion 0'.

- Expand *SOC Configurations* to confirm the enabled SOC0s

Change SOC0 to Channel A0 (2)

SOC0 Start of Conversion 0

SOC0 Channel ADCIN0 is converted

SOC0 Module Channel Name A0

SOC0 Device Pin Name 19: A0/C15

SOC Triggers ^

Sample Time Calculator ^

SOC0 Sample Window [SYSCLK counts] 8

SOC0 Sample Time [ns] 80

SOC1 Start of Conversion 1

SOC1 Channel ADCIN1 is converted

SOC1 Module Channel Name A1

SOC1 Device Pin Name 18: A1

SOC Triggers ^

SOC1 Start of Conversion 1

SOC1 Channel ADCIN1 is converted

SOC1 Module Channel Name

SOC1 Device Pin Name

SOC Triggers ^

Sample Time Calculator ^

SOC1 Sample Window [SYSCLK counts]

SOC1 Sample Time [ns]

SOC2 Start of Conversion 2

ADCIN0 is converted

ADCIN1 is converted

ADCIN2 is converted

ADCIN3 is converted

ADCIN4 is converted

ADCIN5 is converted

ADCIN6 is converted

ADCIN7 is converted

ADCIN8 is converted

ADCIN9 is converted

ADCIN10 is converted

ADCIN11 is converted

ADCIN12 is converted

- Expand SOC0 configuration and change the Channel to ADCIN0

Change SOC0 to Channel A0 (3)

SOC0 Start of Conversion 0	▼
SOC0 Channel	ADCIN0 is converted ▼
SOC0 Module Channel Name	A0
SOC0 Device Pin Name	19: A0/C15
SOC Triggers ▲	
Sample Time Calculator	✕
Use Sample Time Calculator	<input type="checkbox"/>

- Expand Sample Time Calculator

Sample Time Calculator ⓘ	▼
Use Sample Time Calculator	<input checked="" type="checkbox"/> ←
Ron ADC Sampling Switch Resistance [Ohms]	500
Ch ADC Sampling Capacitance [pF]	12.5
Cp ADC Channel Parasitic Capacitance [pF]	15.8
Cs Input Capacitance [pF]	100000 ←
Rs Input Resistance [Ohms]	1000 ←
Settling Error [LSB]	0.5
Calculated Sample Time [ns]	2356.67
Use Calculated Sample Time for this SOC	<input checked="" type="checkbox"/> ←
SOC0 Sample Window [SYSCLK counts]	236
SOC0 Sample Time [ns]	2356.67

- Check *Use Sample Time Calculator*
- Input $C_s = 0.1\mu\text{F}$ and $R_s = 1\text{k}\Omega$
- Check *Use Calculated Sample Time for this SOC*

Change SOC1-5 to Channel A0

- Repeat SOC0 steps on SOC1 through SOC5

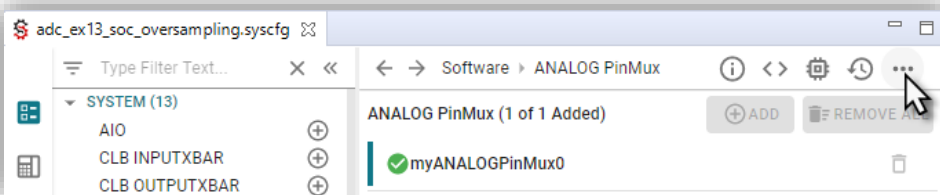
Configure MUX for PWM DAC (1)

- GPIO23 is MUX'd with EPWM4_B (Pin 65 of 80-pin device on LP)

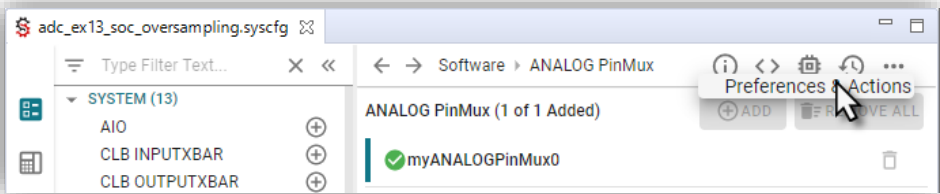
Table 6-3. Digital Signals

SIGNAL NAME	PIN TYPE	DESCRIPTION	GPIO	80 QFP	64 QFP	48 QFP
ADCSOCAA0	O	ADC Start of Conversion A for ePWM4-D	33, 8	38, 58	32, 47	25
EPWM4_B	O	ePWM4 Output B	15, 3	14, 70	61	17
EPWM4_A	O	ePWM4 Output A	22, 6	67, 80	56, 64	48
EPWM4_B	O	ePWM4 Output B	23, 7	65, 68	54, 57	43
EPWM5_A	O	ePWM5 Output A	16, 8	39, 58	33, 47	26

- Click on 3-dot settings shortcut

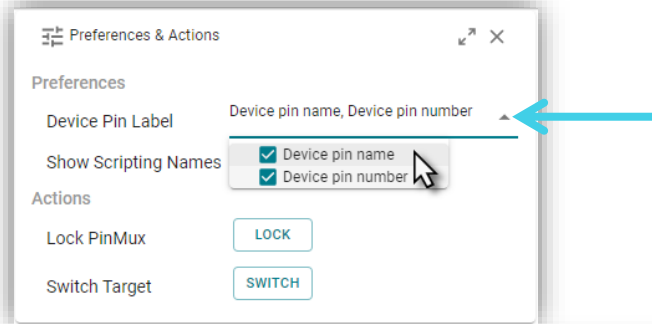


- Click on *Preferences & Actions*



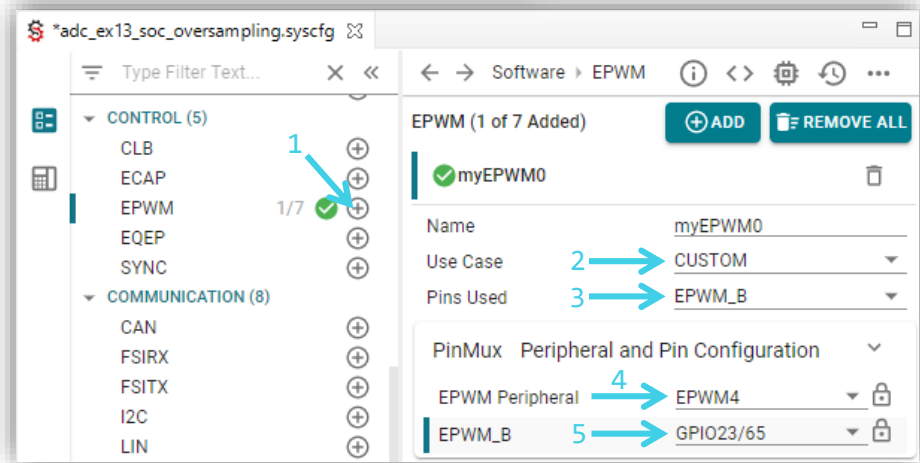
Configure MUX for PWM DAC (2)

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



Configure EPWM PinMux:

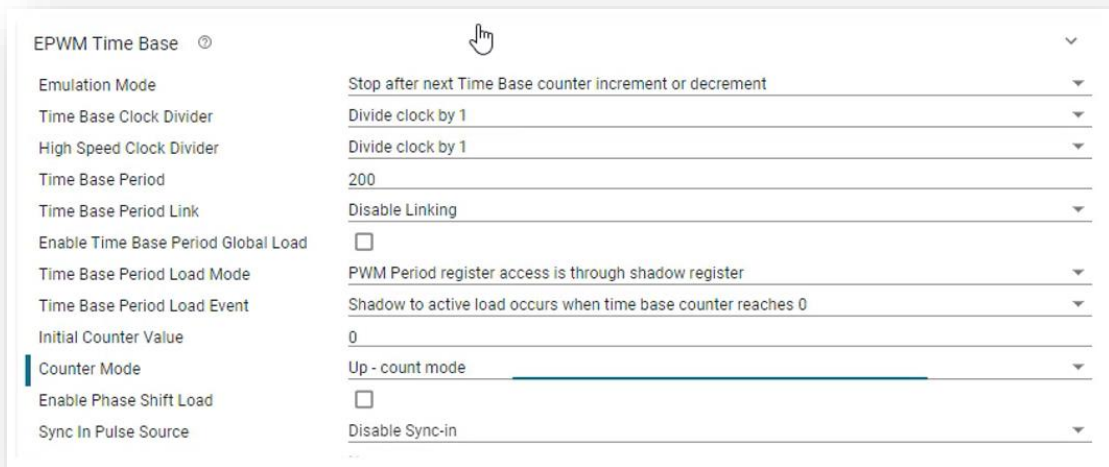
1. Add an EPWM configuration under Control >> EPWM >> +
2. CUSTOM
3. EPWM_B
4. EPWM4
5. GPIO23/65



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

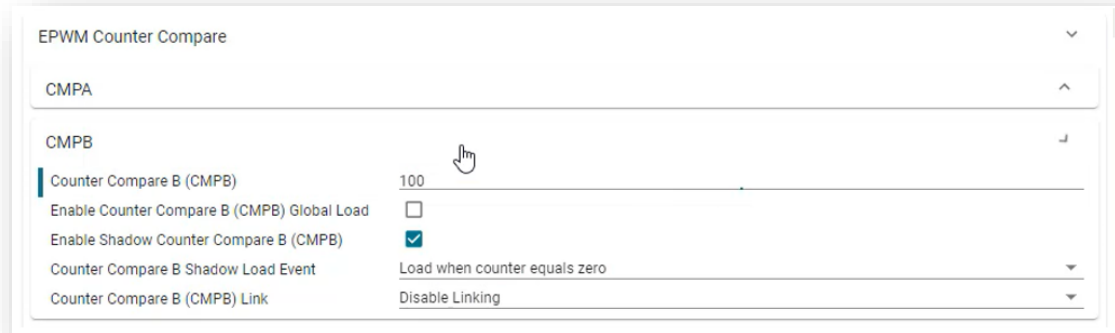


EPWM Time Base ⓘ

Emulation Mode	Stop after next Time Base counter increment or decrement
Time Base Clock Divider	Divide clock by 1
High Speed Clock Divider	Divide clock by 1
Time Base Period	200
Time Base Period Link	Disable Linking
Enable Time Base Period Global Load	<input type="checkbox"/>
Time Base Period Load Mode	PWM Period register access is through shadow register
Time Base Period Load Event	Shadow to active load occurs when time base counter reaches 0
Initial Counter Value	0
Counter Mode	Up - count mode
Enable Phase Shift Load	<input type="checkbox"/>
Sync In Pulse Source	Disable Sync-in

Configure Counter Compare for PWM DAC

Configure Counter Compare
– CMPB Value of 100



The screenshot shows the configuration interface for the EPWM Counter Compare. The main title is "EPWM Counter Compare". Below it, there are sections for "CMPA" and "CMPB". The "CMPB" section is expanded, showing the following configuration:

Parameter	Value
Counter Compare B (CMPB)	100
Enable Counter Compare B (CMPB) Global Load	<input type="checkbox"/>
Enable Shadow Counter Compare B (CMPB)	<input checked="" type="checkbox"/>
Counter Compare B Shadow Load Event	Load when counter equals zero
Counter Compare B (CMPB) Link	Disable Linking

Configure Action Qualifier for PWM DAC

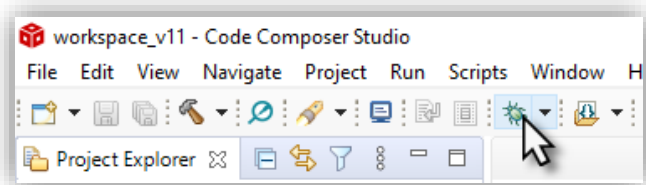
Configure Action Qualifier

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

The screenshot shows the configuration interface for the EPWM Action Qualifier. It is organized into several sections:

- EPWM Action Qualifier**:
 - Enable Continuous SW Force Global Load:
 - Continuous SW Force Shadow Mode: Shadow mode load when counter equals zero
- ePWMxA Output Configuration**
- ePWMxB Output Configuration**:
 - ePWMxB Global Load Enable:
 - ePWMxB Shadow Mode Enable:
 - ePWMxB Shadow Load Event: Load when counter equals zero
 - ePWMxB T1 Trigger Source: Digital compare event A 1
 - ePWMxB T2 Trigger Source: Digital compare event A 1
 - ePWMxB One-Time SW Force Action: No change in the output pins
 - ePWMxB Continuous SW Force Action: Software forcing disabled
 - Events to Configure for ePWMxB output: Time base counter equals zero, Time base counter up equals CMPB
- ePWMxB Event Output Configuration**

Debug Project



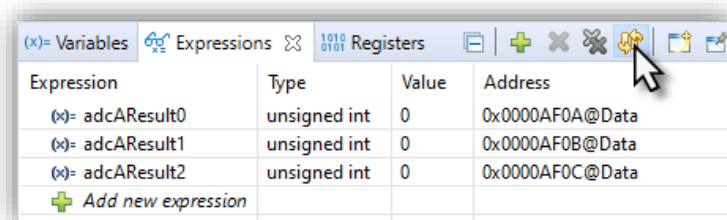
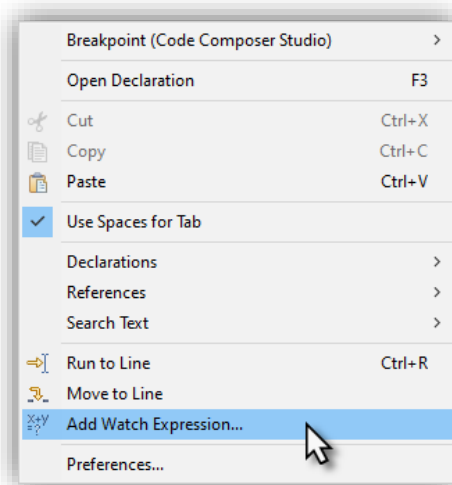
Clicking Debug Icon Will Do the Following:

- 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

```
62 //  
63 // Included Files  
64 //  
65 #include "driverlib.h"  
66 #include "device.h"  
67 #include "board.h"  
68  
69 //  
70 // Globals  
71 //  
72 uint16_t adcResult0;  
73 uint16_t adcResult1;  
74 uint16_t adcResult2;  
75
```

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