C2000™ Software Guide

Release v1.5

Texas Instruments Incorporated

Dec 14, 2023

CONTENTS

CHAPTER

ONE

INTRODUCTION

C2000™ real-time controllers are a portfolio of high-performance microcontrollers that are purpose-built to control power electronics and provide advanced digital signal processing for industrial and automotive applications. Software components to program various modules in C2000 MCUs are released as part of C2000 software releases. This guide provides an overview of various software components and available functionality.

1.1 Getting started with C2000 Software

There are 3 software packages associated with C2000 software development:

- *C2000Ware*
- *Digital Power Software Development Kit* (DPSDK)
- *Motor Control Software Development Kit* (MCSDK)

There are multiple ways to get started with software on C2000. If you would like to start working online on ti.com with the SDKs, use TI Resource Explorer. Another option is to download the complete package onto your desktop and start development. Download information for the SDKs are available in the respective sections of the guide.

If you are new to C2000, a good starting point is basic examples of peripherals or libraries available in *C2000Ware*. C2000Ware provides all of the basic software to get started with programming C2000 MCUs - drivers, libraries, tools and examples. C2000Ware is the Core SDK for C2000 MCUs and is an application agnostic SW Development Kit.

If you are already familiar with C2000 software development and want to evaluate a complete system solution for a specific application, you can start with an existing TI Reference design in the appropriate SDK. There are multiple designs showcasing various design choices and reference SW in SDK to get started with such an evaluation. Refer to *Digital Power Software Development Kit* and *Motor Control Software Development Kit* for details on supported reference designs.

The Release notes for C2000Ware and SDKs lists all components and tools needed to run the examples or reference applications.

1.2 Development Tools

- Development Toolchain - CCS IDE

1.3 Development Kits

Figure below shows the various hardware options: LaunchPads , controlCARDS and Application Kits listed in this link

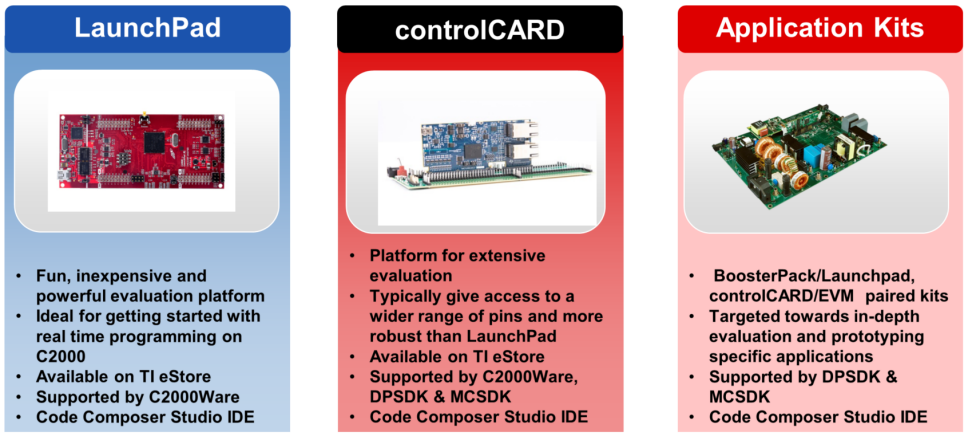


Fig. 1.1: Hardware - LaunchPads, controlCARDs or Application Kits

CHAPTER

TWO

C2000WARE

C2000Ware is the Core SDK for C2000™ MCUs. It is a collection of software components which enable easy use of the various hardware modules in the C2000. It consists of cohesive set of development tools including device-specific drivers, bit-fields, libraries (Math, DSP, Control), peripheral examples, utilities, hardware files, and documentation. Some of these software components can be directly used in customer code as building blocks – drivers, libraries. It also contains a rich set of examples to help unlock the full potential of the core and its peripherals. Fig. 2.1 shows the various software components which are part of C2000Ware. C2000Ware is supported across the following platforms - F28x7x, F28004x, F2838x, F28002x, F28003x, F280013x, F280015x and F28P65x.

C2000Ware availability:

- Download from C2000Ware for C2000 MCUs.
- View the online TI Resource Explorer for C2000Ware TIREX
- **C2000Ware on GitHub**
  - C2000Ware core SDK repo
  - c2000ware-FreeRTOS
  - c2000ware-c2000-academy

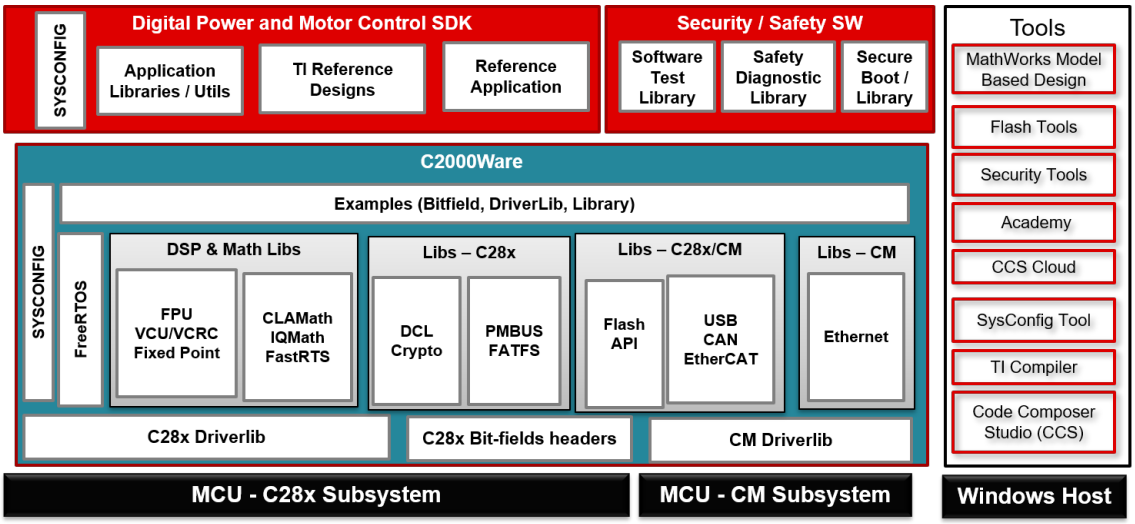


Fig. 2.1: C2000Ware Software Stack

2.1 Drivers

C2000Ware supports multiple ways of accessing peripheral registers - direct register access, driver library (DriverLib), and bitfields. Details of these are mentioned in the sections below. DriverLib is the preferred model of accessing peripherals for the new generation MCUs. Only Bitfield is supported in the older generation of MCUs. But Bitfield is supported in the new generation MCUs also for compatibility and easy migration.

Fig. 2.2 shows the comparison between DriverLib and Bitfield.

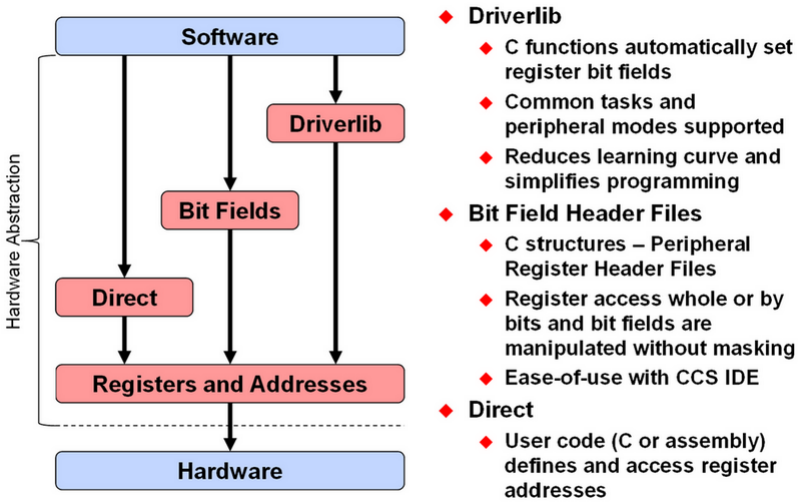


Fig. 2.2: DriverLib, Bit field and direct regiser access

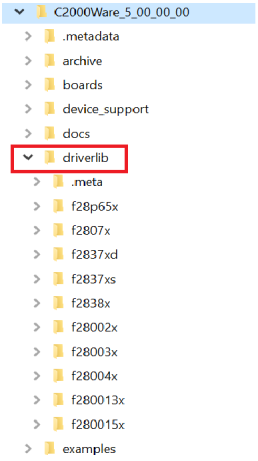
Differences are summarized below:

- Direct register access
  - Register address #define individually
  - User must compute bit-field masks
  - Not easy to read
  - eg. `*CMPR1 = 0x1234;`
- Bit field header files
  - Header files define all registers as structures
  - Bit fields directly accessible
  - Easy to read
  - eg. `EPwm1Regs.CMPA.bit.CMPA = EPwm1Regs.TBPRD * duty;`
- DriverLib
  - DriverLib performs low-level register manipulation
  - Easy to read
  - Highest abstraction level
  - eg. `EPWM_setCounterCompareValue(EPWM1_BASE, EPWM_COUNTER_COMPARE_A, duty);`

2.1.1 DriverLib

The C2000 Peripheral Driver Library (Driverlib) is a set of low-level drivers for configuring memory mapped peripheral registers. It provides drivers for all peripherals and provides access to almost all functionality. The driverlib is written in C language and comprises of data structures, macro definitions and functions. Full source code for driverlib is provided within C2000Ware.

At the top-level, driverlib is organized based on devices and there is a device specific driverlib folder under each device. Within this device specific driverlib folder, source code for peripheral drivers applicable to that device is provided.



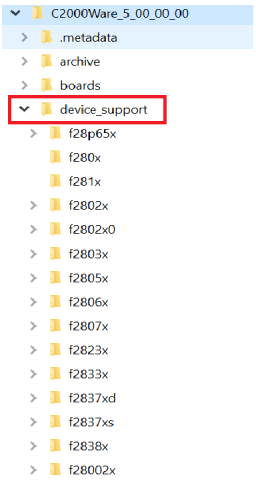
The source code and header files for the driverlib are located in the driverlib folder. The header (.h files) contain:

- #defines – usually meant to be used for parameters
- typedefs – both struct and enum, usually meant to be used for parameters
- static inline functions – all relatively short functions are inlined for performance
- extern function prototypes

The implementation (.c files) contains the implementations of the extern functions. Peripheral drivers make use of a hardware header file to obtain information on register definitions for peripherals. These hardware headers are provided in driverlib/inc folder and in most cases there is one header file per peripheral. There is a CCS project provided to build the driverlib library under the driverlib/ccs folder.

2.1.2 Bitfield

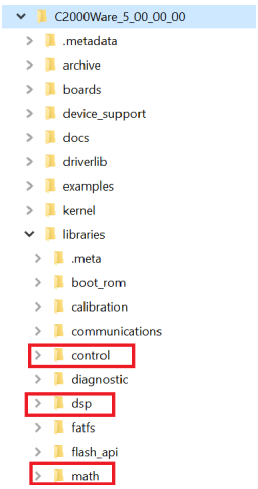
Using register files and bitfields is another approach to access hardware or peripheral registers. This approach was used to access peripherals and write applications in controlSUITE. Starting with C2000Ware, the recommended approach to access peripherals is using driverlibs. However, to assist in a smooth transition from bitfield to driverlib, bitfield software continues to be packaged as part of C2000Ware.



The bitfield source code is organized within the device_support folder within C2000Ware. Under device_support, there are device specific folders which contain the bitfield headers, common code used for initializations and examples for each peripheral. Users can refer to application report SPRAAA85E to get further details on the driverlib and bitfield implementation and how these approaches can be leveraged in building applications.

2.2 Libraries

C2000Ware supports multiple libraries for different category of processing. Details are mentioned in the sections below. The figure below shows the directory structure for Libraries.



2.2.1 DSP Libraries

To enable users who want to optimize performance of their real-time control applications while integrating DSP functionality into their system, C2000Ware provides the multiple optimized DSP libraries.

The following libraries are part of DSP Lib:

1. Fixed Point DSP Software Library – A collection of C28x assembly optimized fixed-point DSP functions (e.g. 16-bit and 32-bit Real and Complex FFTs, 32-bit FIR and IIR filters). Assembly source, benchmarking information, as well as examples to quickly get started are provided.
2. FPU DSP Software Library – A collection of assembly optimized floating-point DSP functions written for C2000 devices that contain either a single precision Floating Point Unit (FPU), an FPU with Trigonometric Math Unit (TMU type 0), or a double precision FPU (FPU64). Functions supported include single-precision and double-precision Real and Complex FFTs, FIR and IIR filters, Vector Math operations, as well as a single-precision fast square root math function. Assembly source, benchmarking information, as well as examples to quickly get started are provided.
3. Viterbi, Complex Math, CRC Libraries – A collection of assembly optimized fixed-point DSP functions written specifically for devices containing a VCU accelerator. These functions include Real and Complex FFTs, CRC, Viterbi decoding, Reed-Solomon decoding and De-interleaving. Please note that on newer devices, the VCU accelerator is replaced with an accelerator containing only CRC functionality, hence termed the VCRC. Assembly source, benchmarking information, as well as examples to quickly get started are provided.

2.2.2 Math Libraries

Similar to the DSP functionality provided above, Math focused operations are supported through the optimized Math libraries.

The following libraries are part of the Math Lib:

1. CLA math library – A collection of assembly optimized floating-point Math functions for devices containing a CLA. The functions include single-precision Trigonometric functions, and intensive math operations such as Division, Square root (and Inverse), Logarithm (and inverse i.e. Exponent). Assembly source, benchmarking information, as well as examples to quickly get started are provided.
2. Fast Integer Division – The TI C28x Compiler supports C-callable intrinsics that execute on the HWINTDIV, an accelerator that supports specialized instructions to perform fast 16-bit, 32-bit and 64-bit fixed-point integer divisions. Examples of intrinsics usage are provided, but also note that additional methods of invoking the accelerator supported instructions include using the standard division operators as well as the standard library functions.
3. FPUFastRTS library – A collection of assembly optimized floating-point Math functions written for C2000 devices that contain either a single precision Floating Point Unit (FPU), an FPU with Trigonometric Math Unit (TMU type 0), or a double precision FPU (FPU64). Functions supported include single-precision and double-precision Trigonometric functions, and intensive math operations such as Division, Square root (and Inverse), Logarithm (and inverse i.e. Exponent). Some of these are functions that are natively supported by the C standard (and implemented in the TI compiler provided RTS library), whereas some are not. In both cases, this library provides faster implementations of the respective math functions, at the expense of accuracy. Assembly source, benchmarking information, as well as examples to quickly get started are provided.
4. IQMath library – A collection of assembly optimized fixed-point mathematical functions for C/C++ programmers to seamlessly port a floating-point algorithm into fixed point code on fixed-point devices. By enabling users to write "floating-point like" code on fixed-point devices, the IQMath library appears to the user to be a virtual floating-point engine. This enables easy to write code (avoiding difficult fixed-point code development involving keeping track of scaling). It also achieves execution speeds considerably faster than equivalent code written in standard ANSI C language. An additional benefit is that code written in IQMath can be ported over

seamlessly (through just a Macro switch) from a fixed-point device (where the fixed-point version of the algorithm runs) to a floating-point device (where the native floating-point version of the algorithm runs). Examples are provided to quickly get started.

2.2.3 Digital Control Library

The Digital Control Library (DCL) is a suite of over two hundred functions which enable high performance closed loop control on C2000. The library can be applied to any closed loop control application in which C2000 is used, including digital power supplies, motor control, motion control, and solar inverters. DCL functions support the fixed point C28x, FPU, FPU64, and the CLA. The library comprises two components: controllers and utilities.

Among the controllers are linear PID, PI, double integrating PI, and direct form compensators such as the 2-pole, 2-zero type. The library also contains nonlinear PID and PI controllers which offer superior transient performance. Each controller is supported by functions to load and update parameters safely. Functions to load controller parameters from a pole-zero description are also included.

Utilities include data loggers, a transient capture module, and signal clamps. Data loggers allow users to conveniently capture important signals in the control loop, such as the loop error or control effort. The Transient Capture Module is a triggered data logger, which can capture a transient event such as a step response together with the conditions immediately preceding the event. Transient response data can be graded using one of three performance index functions in the library. Clamps are useful in situations where loop saturation must be managed to avoid integrator wind-up.

DCL functions are supplied entirely in the form of source code. Most time critical controllers are coded in both inline C and in either assembly for best efficiency and determinism. The library also includes a set of code examples illustrating how to apply the controllers, a small block-set of Simulink models, a library user's manual, and a PID tuning guide.

The Digital Control Library can be found in the `/libraries/control` sub-directory of the C2000Ware installation. A series of training videos on the DCL is available at Digital Control Library - Training series page.

A 4-part technical seminar offering an introduction to control theory can be found on the C2000 MCU workshops page.

2.2.4 Crypto Library

Cryptography is set of algorithms used in protecting data at rest and transit in computer systems, communication equipments and IoT devices. Cryptography has become indispensable tool in protection information in modern day computer/electronic systems. The previous AES library in the C2KWare has now been renamed as Crypto library, it can be found in the `/libraries/security`. The Crypto library has support for algorithms - AES and SHA256. The library also provides examples for usage of different Crypto API supported.

The Advance Encryption Standard(AES) algorithm supports Electronic Code Book (ECB), Cipher Block Chaining (CBC), Counter mode (CTR), Cipher based Message Authentication Code (CMAC) modes. The AES CMAC mode specifically can be used for assurance of data integrity, other modes can be used for data confidentiality. There are space optimized and speed optimized implementation of the algorithm available for all modes. The key sizes of 128 and 256 bit are currently supported for AES.

The Secure Hashing Algorithm (SHA) is one type of cryptographic hashing function. It maps data of an arbitrary size to a bit array of fixed size. It is a one way function, that is, a function for which is practically infeasible to invert or reverse the computation. Some of the applications of it are verifying the integrity of messages, signature generation and verification, password verification etc. The crypto library provides support for SHA256 algorithm.

The C2000_Crypto_Library_User_Guide in `/libraries/security/crypto/c28/docs` provides more information on Application Programming Interface(API) supported for different algorithms. The user guide also provides benchmarking information of the API.

2.3 FreeRTOS support(FreeRTOS)

C2000Ware SDK supports FreeRTOS on C28x and Cortex-M4(CM) cores.

FreeRTOS is a market-leading real-time operating system (RTOS) for microcontrollers and small microprocessors. Distributed freely under the MIT open source license, FreeRTOS includes a kernel and a growing set of libraries suitable for use across all industry sectors. FreeRTOS is built with an emphasis on reliability and ease of use.

C2000Ware includes the kernel, C28x and CM ports and demos. Currently the demos are available for F2838x, F28002x, F28003x ,F280013x ,F280015x and F28P65x.

For More details , pls refer this FreeRTOS documenation

2.4 C2000 TI-RTOS

Version 2.16.01.14 of C2000 TI-RTOS can be downloaded from this link More information on TI-RTOS is available at https://www.ti.com/tool/TI-RTOS-MCU

Note: C2000 TI-RTOS is in long-term maintenance, and as such there is no plan to provide new updates or add support for any devices. It is recommended to migrate to FreeRTOS for any new development. Refer to the FreeRTOS section in this guide for more information on FreeRTOS support.

2.5 Calibration software

Several C28x based devices support high resolution pulse width modulation (HRPWM) and capture (HRCAP) along with enhanced PWM and capture modules. The calibration libraries for these can be found at /libraries/calibration location in C2000Ware installation.

2.5.1 HRPWM Calibration Library

The high resolution PWMs are recommended to be used when PWM resolution falls below ~9-10 bits. Although each application may differ, typical low frequency PWM operation (below 250 kHz) may not require HRPWM. HRPWM capability is most useful for high frequency PWM requirements of various power conversion topologies.

HRPWM is based on micro-edge positioner (MEP) technology. MEP logic is capable of positioning an edge very finely by sub-dividing one coarse system clock of a conventional PWM generator. The time step accuracy is on the order of 150 ps. See the device-specific data sheet for the typical MEP step size on a particular device.

The MEP step size varies based on worst-case process parameters, operating temperature, and voltage. The HRPWM calibration library consists of a self-check software diagnostics mode to check if the MEP logic is running optimally, under all operating conditions. Applications that use the HRPWM feature should use the TI-supplied MEP scale factor optimization (SFO) libraries. The SFO library helps to dynamically determine the number of MEP steps per EPWMCLK period while the HRPWM is in operation.

SFO library is available for both bit-field and Driverlib based code base. Refer to readme.txt available at /libraries/calibration/hrpwm/<device>/lib location for usage specific details.

2.5.2 HRCAP Calibration Library

The HRCAP calibration library is only required for devices which have type-0 HRCAP. Later versions of HRCAP include this calibration logic in hardware itself.

The HRCAP calibration (HCCal) logic is capable of capturing an edge in discrete time steps which subdivides an HRCAP clock cycle. The HRCAP step size is of the order of 300 ps and varies based on worst-case process parameters, operating temperature, and voltage. See device-specific data sheet for typical HRCAP step size on your device.

Applications that use the HRCAP type-0 in high-resolution capture mode should use the TI-supplied HRCAP calibration library to dynamically scale the HRCAP step size to a fraction of the HCCAPCLK cycle while the HRCAP is in high-resolution mode. The library also provides functions to measure the high resolution high pulse width, low pulse width, and period. Refer to device TRM for usage specific details.

2.6 USB

The USB Library provides a set of data types and functions for creating USB device, host, or dual mode applications. USB library can be found in the /libraries/communications/usb directory of the C2000Ware installation. Pre-built library and source code for the USB library are provided. Reference examples for host, device and dual mode operation are also included in the Driverlib examples directory of C2000Ware installation. The USB library can be categorized into following main groups: device mode specific functions, host mode specific functions and general purpose functions.

USB library's device mode functions provide four API layers relevant to the development of USB device applications:

- Device Class APIs
- Device Class Driver APIs
- The USB Device API
- The USB DriverLib API

Source code and headers for the device mode USB functions can be found in the device directory of the USB library tree, typically libraries/communications/usb/<device>/include/device and libraries/communications/usb/<device>/source/device.

USB DriverLib API is the lowest layer in the USB device stack is the USB driver. This is typically not used for any application development but can be used for development of USB stack. USB Device API provides a group of functions specifically intended to allow development of fully featured USB device applications. These APIs can be used in the development of USB device class drivers and can also be used directly.

Device Class Driver APIs provide high level USB function to applications wishing to offer particular USB features with minimal application overhead. Device class drivers are currently provided to allow creation of a generic bulk device, a Communication Device Class (virtual serial port) device and a Human Interface Device class device (mouse, keyboard, etc.)

Device Class APIs provided an additional layer to specialize the device operation and simplify the interface to the application.

USB library's host mode functions provide the following four layers:

- Device APIs (Mouse, Keyboard, Filesystem)
- Host Class Driver APIs (HID, Mass Storage, Hub)
- Host Controller APIs
- DriverLib USB Driver APIs

Source code and headers for the device host mode USB functions can be found in the device directory of the USB library tree, typically libraries/communications/usb/<device>/include/host and libraries/communications/usb/<device>/source/host

USB DriverLib API is the lowest layer in the USB device stack is the USB driver. This is typically not used for any application development but can be used for development of USB stack.

The host controller APIs provide all of the functionality necessary to provide enumeration of devices. This is the lowest level of the driver interface which interacts directly with the DriverLib USB APIs.

The host class drivers provide access to devices that use a common USB class interface. The USB library currently supports the following USB class drivers: Mass Storage Class (MSC) and Human Interface Device (HID) and Hub.

The Host Device APIs provide interfaces for a HID mouse, a HID keyboard and a mass storage device

General purpose functions included in the USB library are not specific to USB host or device operation and are provided for general use in the development of USB applications.

Source code and headers for the general-purpose USB functions can be found in the top level directory of the USB library tree, typically `libraries/communications/usb/<device>/include` and `libraries/communications/usb/<device>/source`.

2.7 PMBus

The PMBus software library is a C28x target mode only communications software stack. The PMBus target communication stack is based on PMBus specification (Part I, II) v1.2.

Within the communications software stack, the PMBus target state machine is implemented and is provided as pre-built binary as well as source code. Additionally, device-specific controller and target examples are provided to demonstrate usage of the software stack. PMBus library can be found in the folder `libraries/communications/PMBus/c28` of the C2000Ware installation.

2.8 C2000 MCU Software Diagnostic Library (SDL)

The C2000 MCU Software Diagnostic Library (SDL) includes software that helps make designing for functional safety applications with TI C2000 real-time control microcontrollers easier and faster. The safety mechanisms employed are intended to aid the customer in achieving their specific safety goals.

The C2000 SDL include specific functional safety software mechanisms to enable customers to meet their functional safety goals. The functional safety software mechanisms provided are included and detailed in the device safety manual.

C2000 has selected a set of functional safety software mechanisms as reference according to the following criteria:

- Valid mechanism according to the device safety manual
- Easily integrated into various systems
- Leverages specific C2000 device features for functional safety purposes
- Reduces customer software development and time to market

Where valid functional safety software mechanisms are omitted from the libraries, this is due to one or more of the following reasons:

- Functional safety mechanism is simple and does not require a software library or API provided by C2000
- Functional safety mechanism is tightly coupled with the system implementation and it is not feasible to provide a generic solution in a software library from C2000

Additionally, functional safety software mechanisms which are not provided in software libraries are detailed in the device safety manual. Sufficient information is provided to enable customers to implement and integrate such safety mechanisms.

Software Diagnostic Library for F2807x, F2837xD and F2837xS are available for download from C2000-SAFETI-DIAGNOSTICS-LIB.

As of *C2000Ware* v5.00.00.00, the C2000Ware Software Diagnostic Library includes support for F28002x, F28003x, F28004x, F2838x, F280013x and F280015x devices. Find the diagnostic library and examples under `/libraries/diagnostic`.

For more information on functional safety with C2000 MCUs, refer to Industrial Functional Safety for C2000 Real-Time Microcontrollers or Automotive Functional Safety for C2000 Real-Time Microcontrollers. For more information on safety mechanisms, refer to C2000 real-time MCU Safety Mechanisms

2.9 CAN

The CAN software support includes driverlib and the reference example applications for both CAN and CAN-FD (F2838x). The references examples include internal/external loop back, transmit, receive, and DMA usage.

Different CAN communication protocol stacks is supported by 3rd party Software providers on many C2000 devices. Some of the supported stacks are J1939, CANOPEN, and ISO 15765. For more details on third party offering refer to third party software stacks.

2.10 Ethernet

The Ethernet software for new generation F2838x devices runs on the Communication Manager (CM). The C2000Ware package contains the necessary software to get started with Ethernet. The package contains the low level driver Ethernet.c and Ethernet.h. There are also ports of lwIP, ptpd (third party stacks). Even though the CM side runs the stack and low level driver, the clock configurations and Pin Mux configurations should be done from the CPU1 side of F2838x before running the code on CM side.

Ethernet examples in C2000Ware:

1. Low Level Ethernet Driver Examples - Present in <C2000Ware Installation Directory>/driverlib/f2838x/examples/cm/ethernet. These are low level Driver examples, which show how to configure the Low level driver to use different features of the driver.
2. LWIP HTTP webserver Example - This uses the LWIP open source community stack and creates a webserver on F2838x device present in <C2000Ware Installation Directory>/libraries/communications/Ethernet/third_party/lwip/examples/enet_lwip
3. PTPD example - This example uses ptpd community stack for the precision time Control and runs a PTP slave, and synchronizes the Ethernet PTP clock to the Master Clock based on PTP protocol. It is available at <C2000Ware Installation Directory>/libraries/communications/Ethernet/third_party/ptpd/examples/enet_ptpd
2. LWIP UDP Example - This example application demonstrates the operation of the F2838x microcontroller Ethernet controller using the lwIP TCP/IP Stack. This example sets up the Ethernet MAC. F2838x device receives and sends data over UDP. This example is located in <C2000Ware Installation Directory>/libraries/communications/Ethernet/third_party/lwip/examples/enet_lwip

2.11 EtherCAT

The EtherCAT software package includes all the necessary software (except EtherCAT target stack code) to setup F2838x (CPU1 or CM) as an EtherCAT Target Controller (ESC). This includes usage documentation, examples, EtherCAT hardware abstraction layer (HAL) drivers, and Beckhoff Target Stack Code (SSC) Tool configuration file.

The package details are as follows:

- **docs** Getting started guide on using the software, setting up an EtherCAT Controller, obtaining the EtherCAT target stack code, and suggestions for troubleshooting
- **examples** CPU1 and CM examples showcasing a sample ESC echoback application. This includes two pre-built image demos (which include EtherCAT target stack code) that can be flashed onto the device for quick evaluation of the device as an ESC.
- **include, source** -HAL drivers required by the EtherCAT target stack code which provide the necessary read/write, initialization, and interrupt handler APIs.
- **ssc_configuration** This folder is created once the secondary installer "ethercat_slave_ssc_and_demo_setup.exe" is run. It contains the echoback example application code, required system source code, and the Beckhoff SSC Tool configuration file. This configuration file gets imported into the SSC tool to configure all the stack code settings so that the target stack code can be generated for the example application.

2.12 HIC

Host Interface Controller (HIC) allows the external host processor (controller) to access the peripherals of the device (responder). The host just needs to have an asram interface to access HIC on responder. The software support in C2000Ware includes the driverlib which can be used to configure the responder in different modes such as mailbox mode or direct access mode based on the latency requirements or pin overhead considerations. It also has reference examples to demonstrate multiple applications which have been test on TI Internal Validation Platform. User reference guide for examples can be found at SPRACR2

HIC examples in C2000Ware:

1. HIC 16-bit Memory Access FSI Example - Example demonstrates the configuration required to use the FSI peripheral on the device by the host. The host side example at <C2000Ware Installation Directory>/driverlib/f2838x/examples/c28x/emif/emif_ex7_16bit_asram_hic_fsi. The device side example at <C2000Ware Installation Directory>/driverlib/f28002x/examples/hic/hic_ex3_config_16bit_fsi
2. HIC 8-bit Memory Access ADC Example - Example demonstrates the configuration required to use the ADC peripheral on the device by the host. The host side example at <C2000Ware Installation Directory>/driverlib/f2838x/examples/c28x/emif/emif_ex8_8bit_asram_hic_adc. The device side example at <C2000Ware Installation Directory>/driverlib/f28002x/examples/hic/hic_ex2_config_8bit_adc

2.13 FSI

Fast Serial Interface (FSI) is the TI properitery communication peripheral which enables high-speed, reliable, serial communication across devices having isolation between them. C2000Ware software support contains device drivers and examples for FSI which help to understand and implement the functionalities such as communication between multiple devices connected in daisy chain topology, triggering the transmit using PWMs, communicating from the lead device with the node devices not having FSI, etc. The device technical reference manual contains additional details for the usage of FSI.

The FSI device drivers can be found in <C2000Ware_XX_XX>/driverlib/<device>/driverlib, and the examples can be found in <C2000Ware_XX_XX>/driverlib/<device>/examples/fsi

Additional example showcasing the event synchronization feature for all devices connected in daisy chain topology is available with the <C2000Ware_XX_XX>/examples/demos folder.

2.14 Flash

Non-volatile flash memory on C2000 devices is primarily used as program memory for the core and also as static data memory. During application development, application executable can be programmed in to the flash memory using Code Composer Studio (CCS). When CCS identifies that the code is mapped in to the flash memory for a given application, it will automatically invoke CCS On-Chip Flash Plugin to load the executable in to the flash memory. Flash Plugin GUI is available at CCS Debug view -> Tools -> On-chip Flash. By default, the on-chip Flash Plugin erases the flash before programming, generates ECC for the executable and programs it along with the main array flash content and verifies the programmed content. If needed, users can enable blank check - CPU verification to confirm that flash is erased before programming. All of the USER OTP fields (DCSM and boot settings) can be programmed as well using the CCS On-chip Flash Plugin. Checksum calculation feature is also available within the Plugin GUI. Performance of the CCS flash programming is best when using high performance debug probes (XDS200 is suggested).

TI also provides UniFlash, a standalone JTAG based flash programming tool. This has a smaller footprint compared to that of CCS, since it does not include debug support. Users can create device-specific UniFlash package to further reduce the tool size (See Standalone Command Line window in UniFlash after installation). UniFlash offers all the GUI operations that CCS Flash Plugin does.

If the application safety standard requires the ECC generator and the verifier unit to be different, users can use the –ecc options provided in the link step in CCS. This when enabled, linker will generate the ECC image for the flash application and append it at the end of the executable. TI flash tools (CCS and UniFlash) are capable of programming this appended ECC image instead of the default generated Auto ECC. When using the linker ECC option, Auto ECC Generation provided in the TI flash tools should be disabled. Please refer to TMS320C28x Assembly Language Tools guide for more information. Note that TI flash tools do not support programming linker generated ECC for TMS320F28M3x, TMS320F2837x and TMS320F2807x devices.

Customers may also refer to 3rd party programmers like CodeSkin, DATA IO etc. to get flash programming support via serial peripherals like SCI, CAN etc. Elprotronic offers JTAG based C2000 GANG programmers.

TI flash tools and 3rd party flash programmers use Flash API library to perform flash erase and program operations. Flash API library and reference guide for each device are available in C2000Ware at `/libraries/flash_api`. Flash API library has functions to do flash operations like erase, blank check, program flash and corresponding ECC, program verify, suspend an erase or program operation, resume an erase or program operation, ECC calculation, checksum calculation, remap an ECC address to corresponding main array address etc. Note that ECC is not available for some C2000 devices. Please check Datasheet and TRM for the details. Applications that require erase or program flash at runtime can embed Flash API library to accomplish the same. An example to use Flash API library is provided in C2000Ware (e.g. `C2000Ware_x_xx_xx_xx\driverlib\f2838x\examples\c28x\flash\flashapi_ex1_programming`).

C2000Ware contains examples to illustrate flash kernels. Flash kernels are used by C2000 devices to provide firmware updates to a system, with a mechanism to switchover to the new firmware through a device reset. C2000Ware examples for flash kernels illustrate how to provide firmware updates over SCI, CAN [DCAN] and MCAN [CAN-FD] peripherals. For example, the C2000Ware flash kernels for the F28003x device can be found in the `/driverlib/f28003x/examples/flash` sub-directory of the C2000Ware installation. For example, at this path the flash_kernel_ex3_boot.c and flash_kernel_ex3_sci_flash_kernel.c refer to the SCI flash kernel example, flash_kernel_ex4_boot.c and flash_kernel_ex4_can_flash_kernel.c refer to the MCAN (CAN-FD) flash kernel example, flash_kernel_ex5_boot.c and flash_kernel_ex5_dcan_flash_kernel.c refer to the CAN (DCAN) flash kernel example. For more device examples, refer to the device specific paths given in the application note.

Flash kernel user guides centered around these examples are available at SPRABV4, and SPRAD51.

C2000Ware also contains an example to demonstrate EEPROM Emulation using the On-Chip Flash of some Gen 3 C2000 MCUs (F28P65x, F280013x, F280015x). This example allows for EEPROM Emulation, including Read, Write, and Erase. The example provides both Single-Unit and Ping-Pong (between two EEPROM units) implementation models with several user-configurable aspects. For example, the EEPROM Emulation Example for F28P65x can be found in the `/driverlib/f28p65x/examples/c28x/flash` sub-directory of the C2000Ware installation. There is a separate project for the Single-Unit and Ping-Pong implementation.

An application note detailing the EEPROM Emulation driver is also available at SPRADE8.

Note: TI does not normally distribute the flash API source code nor support development, or modifications, of the flash API by customers. Programming and erase of the flash must be done in a very specific manner. Testing the flash API library involves special test modes accessed by the device during automatic test equipment (ATE). Testing of the library is accomplished by using it in TI development, device testing, write erase cycle qualification, and by all flash programming tools. Modification or rebuilding the API library will not benefit from this testing and thus the restrictions.

In situations where the application is military, aviation, or medical in nature and the user has a justifiable and legitimate need for the source code for certification purposes, it can be provided under a signed license agreement with TI. If users fall into this category, contact your TI sales or distribution representative or post in the TI E2E forum for assistance.

2.15 Live Firmware Update (LFU)

C2000Ware contains examples to illustrate Live Firmware Update (LFU). LFU refers to firmware updates to a system while it is operating, with a mechanism to switchover to the new firmware either through a device reset, or without a device reset. The latter approach is more complex, but enables the system to seamlessly switchover from old to new firmware, even while running a real-time control loop. LFU is important for high availability systems like Server Power Supplies (PSUs) that need to remain ON even during updates, such as firmware updates. C2000Ware examples for LFU illustrate how to implement LFU either with device reset, or without device reset.

C2000Ware examples can be found in the `/driverlib/f28004x/examples/flash` sub-directory of the C2000Ware installation. Refer to Examples 2, 3, and 5.

LFU user guides centered around these examples are available at SPRUIU8, and SPRUIU9.

An LFU centric reference design is available at TIDM-02011.

2.16 Examples

C2000Ware includes several examples that demonstrate the capabilities of device and peripherals utilizing the various software components.

Examples are broadly categorized as follows:

- **Driverlib based examples:** These examples demonstrate features and functionalities of peripherals using driverlib. Examples are provided for each device and for every main peripheral. This can be located under driverlib/<device>/examples. For instance, the adc examples for F28004x device can be found under `driverlib/f28004x/examples/adc`. CCS projects for each example are provided in projectspec format within the CCS folder contained within the example which can be imported in to CCS and build. A user guide is available for driverlib examples which can be located under device_support/<device>/docs For instance, the driverlib example user guide for F2838x is located at device_support/f2838x/docs/F2838x_FRM_EX_UG.pdf

- **Bitfield based examples:** These examples demonstrate features and functionalities of peripherals using the bitfield headers. Similar to driverlib examples, these are provided for each device and for every main peripheral. Bitfield examples can be located under `device_support/<device>/examples`.
- **Examples for libraries:** These examples demonstrate the features and functionalities of the various software libraries like DSP, Math and Digital Control etc. which have been explained in the previous sections. The examples for libraries are contained in an examples folder within each library. For instance, the examples for Digital Control Library (DCL) are located under `libraries/control/DCL/c28/examples`.

CHAPTER

THREE

DIGITAL POWER SOFTWARE DEVELOPMENT KIT

The Digital Power Software Development Kit (DP SDK) is a cohesive set of software infrastructure, tools, and documentation designed to minimize C2000 MCU based digital power system development time targeted for various AC-DC, DC-DC and DC-AC power supply applications. The software includes firmware that runs on C2000 digital power evaluation modules (EVMs) and TI designs (TIDs) which are targeted for solar, telecom, server, electric vehicle chargers and industrial power delivery applications. DigitalPower SDK provides all the needed resources at every stage of development and evaluation in a digital power application.

The DP SDK is built on top of *C2000Ware* and is available for download from http://www.ti.com/tool/C2000WARE-DIGITALPOWER-SDK. The SDK is also available online via the TI Resource Explorer for C2000 DP SDK.

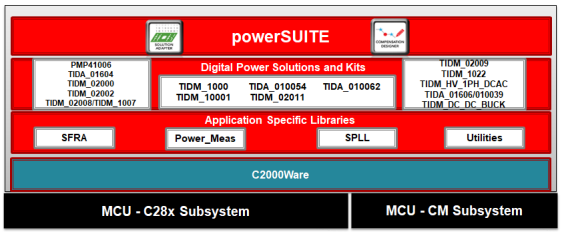


Fig. 3.1: Digital Power SDK

Table 3.1 illustrates the top-level directory structure and contents of the DP SDK.

Table 3.1: DP SDK directory structure

Directory Name	Description
.metadata	Contains the Digital Power SDK resource explorer GUI files (Do not modify)
c2000ware	Contains the C2000Ware development software and documentation
docs	Contains the Digital Power SDK package user guides and a document html page
libraries	Contains libraries with algorithm modules specific to digital power applications
solutions	Contains the development kit solutions for the C2000 devices targeting different application areas
powerSUITE	Contains set of digital power supply design software tools used to traverse through different design stage of various applications

3.1 Supported Solutions

Table 3.2 summarizes key kits supported in the Digital Power SDK. Please check the latest MC SDK release notes for an up to date list of solutions.

Table 3.2: Digital Power Solutions

EVM / TI Design	C2000™ SoC	Description
TIDM-1007	F28004x (CLA Support included)	Single-Phase Totem Pole CCM PFC
TIDM-HV-1PH-DCAC	F28004x, F2837x	Voltage source inverter & Grid Connected Inverter
TIDM-1000	F28004x, F2837x, F2838x (CLA Support included)	Three Phase PFC: Vienna Rectifier
TIDM-1022	F28004x (CLA Support included)	Valley Switching Boost PFC
TIDA-01604	F28004x (CLA Support included)	6.6-kW Single-Phase Totem Pole CCM PFC
TIDM-02002	F28004x (CLA Support included)	CLLLC: Bi-Directional Resonant Dual Active Bridge
TIDM-1001	F28002x	Two Phase Interleaved LLC Resonant Converter
TIDM-DC-DC-BUCK	F28004x	Digitally Controlled Non-Isolated DC/DC Buck Converter
TIDM-02008	F28004x (CLA Support included)	Bi-directional Single Phase CCM PFC
PMP-41006	F28004x, F28003x	1kW CCM Totem Pole PFC and Current Mode LLC
TIDA-1606/10039	F2837x(CLA Support included)	10 kW Three Phase AC-DC, T Type Inverter / T Type PFC
TIDA-010054	F28004x	10 kW, Bi-directional High Voltage Dual Active Bridge DC/DC Converter
TIDA-010062	F28004x, F28002x, F28003x	1kW, GaN CCM Totem Pole bridgeless PFC and halg-bridge LLC
TIDM-02000	F28004x	Phase Shifted Full Bridge Bi-directional DC/DC Converter
TIDM-02009	F2838x (CLA Support included)	EV Traction with DC/DC Convereter
TIDM-02011	F28003x, F28004x	Live firmware update reference design with C2000™ real-time MCUs
PMP-23069	F28002x, F28004x	3-kW, 180-W/in3 single-phase totem-pole bridgeless PFC reference design with 16-A max input
PMP-23126	F28004x	3-kW phase-shifted full bridge with active clamp reference design with > 270-W/in3 power density
PMP-40988	F28004x	Variable-frequency, ZVS, 5-kW, GaN-based, two-phase totem-pole PFC reference design
TIDA-010210	F28004x	11-kW, bidirectional, three-phase ANPC based on GaN reference design
TIDA-010231	F28004x	Analog front end for arc detection in photovoltaic applications reference design
TIDM-02013	F28003x	7.4-kW on-board charger reference design with CCM totem pole PFC and CLLLC DC/DC using C2000™ MCU
TIDM-SOLAR-DCDC	F28004x	C2000™ MCU solar DC/DC converter with Maximum Power Point Tracking (MPPT) reference design

3.2 Libraries

The libraries included in Digital Power SDK range from ramp generator and power measurement libraries to specialized software phase lock loop libraries as well as the Software Frequency Response Analyzer (SFRA) library. The libraries directory includes documentation and examples when applicable. Additional information on the libraries is available in these guides:

- DigitalPower Libraries API Reference Guide
- SFRA Library User's Guide

3.3 powerSUITE

powerSUITE is a suite of digital power supply design software tools. It helps power supply engineers drastically reduce development time as they design digitally-controlled power supplies based on C2000 MCUs. More information on powerSUITE is available at http://www.ti.com/tool/POWERSUITE.

powerSUITE consists of three tools designed to help engineers more easily traverse through the different design stages of a digital power supply systems. The tools are:

- Solution Adapter allows for customization of examples in the DP SDK and for configuration for a custom digital power board design.
- Compensation Designer allows the design of different styles of compensators (ex: PID, 2p2z, 3p3z) to achieve the desired closed loop performance.
- Software Frequency Response Analyzer (SFRA) enables the measurement of the open loop gain of a closed loop digitally controller power converter using software.

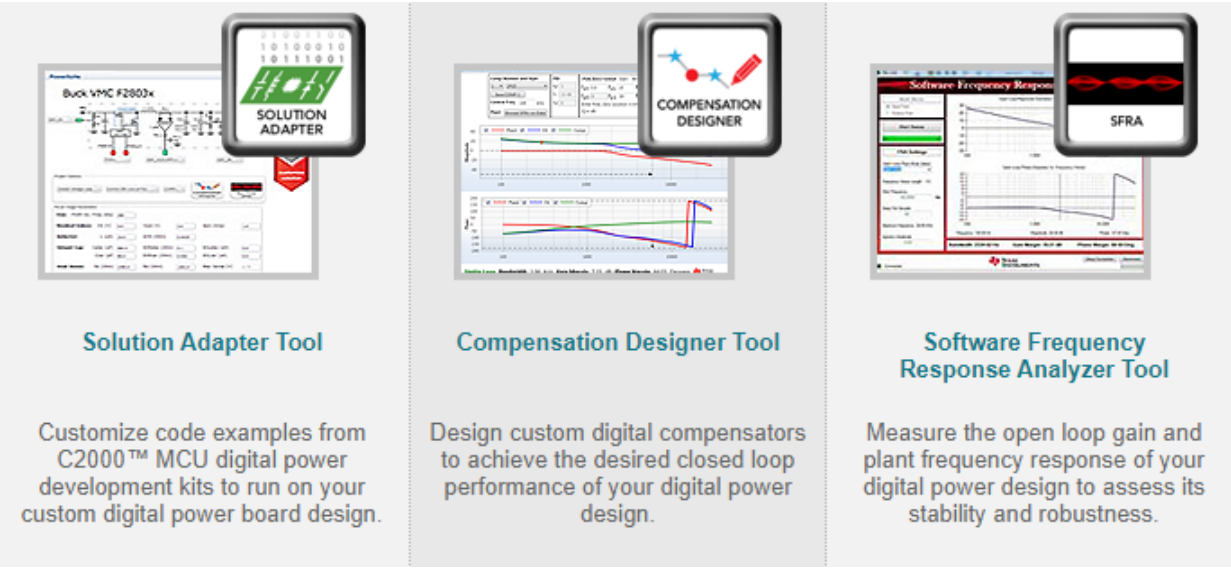


Fig. 3.2: powerSUITE components

CHAPTER

FOUR

MOTOR CONTROL SOFTWARE DEVELOPMENT KIT

The MotorControl SDK (MC SDK) is a cohesive set of software infrastructure, tools, and documentation designed to minimize C2000 real-time controller based motor control system development time targeted for various three-phase motor control applications. The software includes firmware that runs on C2000 motor control evaluation modules (EVMs) and TI designs (TIDs) which are targeted for industrial drives, robotics, appliances, and automotive applications. MotorControl SDK provides all the needed resources at every stage of development and evaluation for high performance motor control applications.

MC SDK has the latest sensored servo, sensorless InstaSPIN-FOC and sliding mode observer solutions, real-time connectivity examples, incremental and absolute encoders, and the Fast Current Loop (FCL) optimized software library.

The MC SDK is built on top of *C2000Ware* and is available for download from C2000WARE-MOTORCONTROL-SDK. The SDK is also available online via the TI Resource Explorer for C2000 MC SDK. For a list of supported devices, kits, and reference designs, see the latest MC SDK release notes.

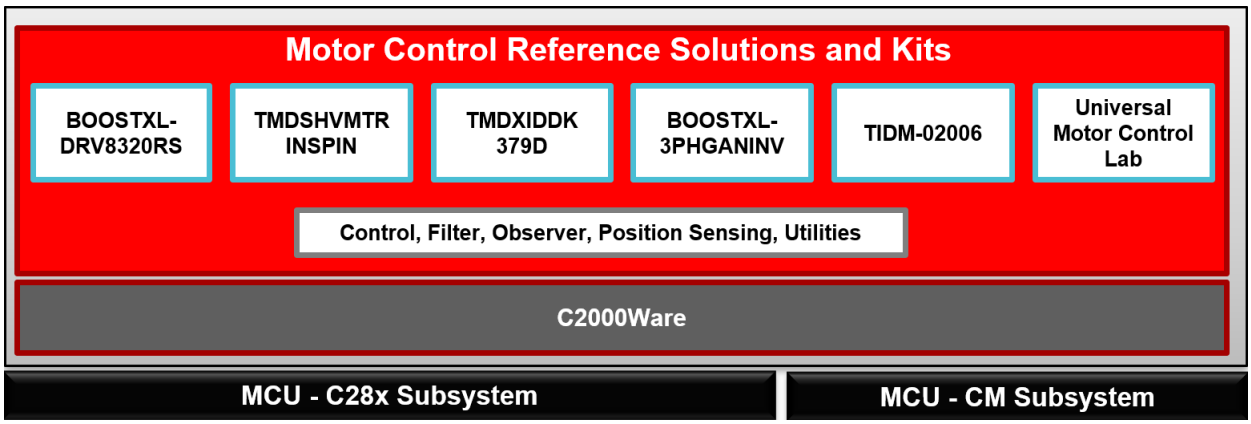


Fig. 4.1: Motor Control SDK

4.1 Features

- Complete software repository for C2000 MCU Motor Control Applications
- Sensorless solutions
  - Sensorless Torque or Velocity
  - Trapezoidal or Field Oriented Control
  - Enhanced sliding mode observer for open solution, best for high-speed applications
  - InstaSPIN-FOC

    - FAST™ software observer for premium rotor Flux, Angle, Speed, and Torque estimationsMotor Parameter Identification
    - Motor Parameter Identification
    - Observer and torque control loop automatic tuning
    - Premium performance for low-speed and highly dynamic applications
  - Reference designs may include the following examples
    - Send commands through CAN interface, potentiometer analog input, and frequency input
    - System protection
    - Flying start
    - Field Weakening
    - Stall detection and recovery
    - Start-up failure detection and recovery
    - Lost phase detection/protection
    - Motor stop with brake
    - Vibration and acoustic compensation
    - Initial Position Detection
    - Maximum Torque Per Amp (MTPA) for IPM motors
- Sensored FOC solutions
  - Sensored Velocity or Position Field Oriented Control
  - Position feedback: Resolver, Incremental and Absolute Encoders
  - Current sense techniques: Low-side shunt, in-line current sampling, and sigma-delta filter demodulation
  - Fast Current Loop (FCL): Optimized software library that takes full advantage of hardware resources to accelerate the sampling, processing, and actuation of the system to achieve the highest control bandwidth for a given PWM frequency in servo control applications.
  - Real-time connectivity examples (EtherCAT, CAN, CAN-FD, FSI)
  - Decentralized / Distributed architecture examples
  - Use of Configurable Logic Block for absolute encoders, pulse train input/output, QEP decode, and deadband compensation

4.2 Supported Solutions

Table 4.1 summarizes key kits supported in the Motor Control SDK. Please check the latest MC SDK release notes for an up to date list of solutions.

Table 4.1: Motor Control Solutions

EVM / TI Design	C2000™ SoC	Description
TIDM-2006	F2838x, F28004x, F28002x	Distributed multi-axis servo drive over fast serial interface (FSI) reference design
TIDM-2007	F2837x, F28004x	Dual-axis motor drive using fast current loop (FCL) and SFRA on a single MCU reference design
TIDM-2010	F280013x, F28002x, F28003x	Dual motor control with digital interleaved PFC for HVAC reference design
TIDM-2012	F28003x	High-voltage EV/HEV HVAC eCompressor motor control reference design
Universal Motor Control Lab	F280013x, F28002x, F28003x	Single project example for different Sensorless (FAST, eSMO, InstaSPIN-BLDC), Sensored (Incremental Encoder, Hall), and control techniques (FOC, Trapezoidal)
InstaSPIN-FOC Labs	F28004x	InstaSPIN-FOC labs for F28004x FAST ROM
Servo Drive with Fast Current Loop	F2838x, F2837x, F28004x, F28002x	Quick response control of PMSM using Fast Current Loop on the DesignDRIVE Development Kit
Position Encoders	F2838x, F2837x, F28004x, F28002x, F28003x	Implemented using Configuration Logic Block and / or other peripherals of C2000 devices T-format Master, PTO – QEP, PTO – PulseGen, PTO – QEP Div, PTO – Abs2QEP
Servo drive with CAN	F28002x, F28003x, F28004x	Sensored FOC motor drive using QEP with integrated CAN communication to control the motor

CHAPTER

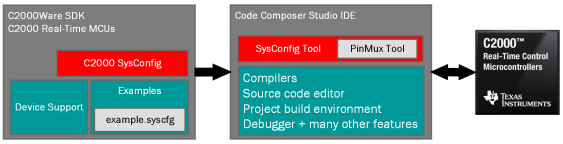
FIVE

C2000 SYSCONFIG

C2000 real-time MCUs can be initialized through C2000 SysConfig which generates reliable and pre-verified code for configuring your device. Device configuration errors are caught by the tool, and the developer is notified of the unsupported settings. C2000 SysConfig tool can also configure the device PinMux and visualize the device pins for each package. C2000 SysConfig is delivered through C2000Ware (C2000 real-time MCU software development kit) and can be used with Code Composer Studio (CCS) IDE's built-in SysConfig (System Configuration) tool or with any other supported IDE through the SysConfig tool's standalone version.

Features include:

- Peripheral configuration
- Automatic embedded code generation
- Configuration error detection
- Device level dependency identification
- Device level error detection
- PinMux tool support
- Portable configuration support



- For more information on the C2000 SysConfig features, visit SPRY341

5.1 C2000 SysConfig CCS Project Overview

Device peripherals are listed in the C2000 SysConfig tool so the designer is aware of the peripherals available in their specific device package. The configurable options for each peripheral is listed, which allows the designer to see all the different available modes. The device level inter-connects are displayed in the tool that shows the available list of signals for each MUX previously described only in the technical documentation.

More complicated peripherals such as the Configurable Logic Block (CLB), which is capable of creating custom logic inside the device, or the Dual Code Security Module (DCSM), which is used of securing the customer's intellectual property, are also included in the C2000 SysConfig ecosystem. These add-on tools will show up automatically in the tool and the designer has the options of using them in their application. The additional autogenerated artifacts from these tools will be presented to the designer seamlessly.

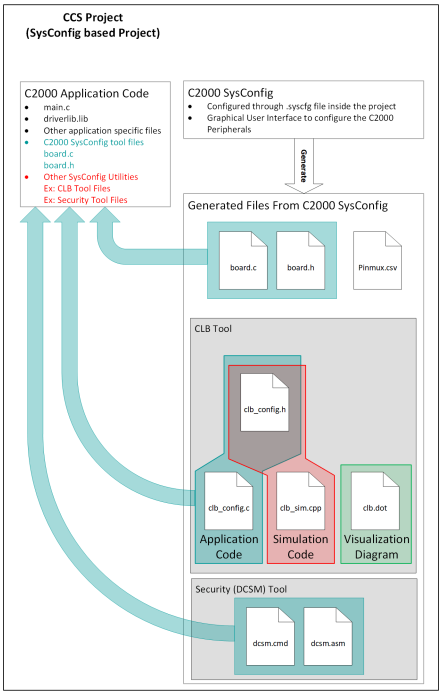


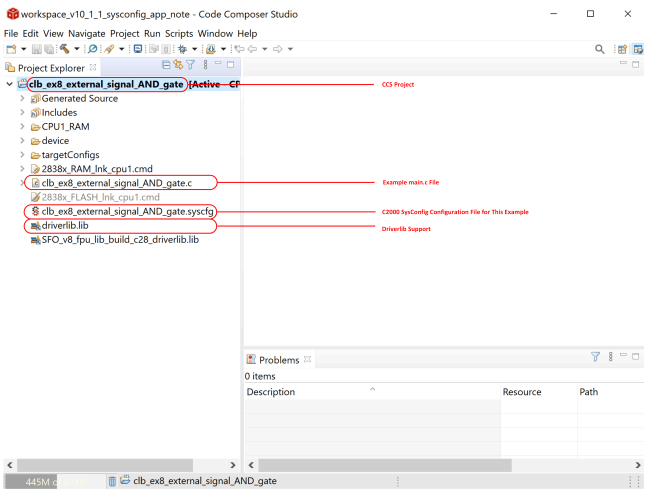
Fig. 5.1: C2000 SysConfig CCS project overview

5.2 Getting Started with C2000 SysConfig

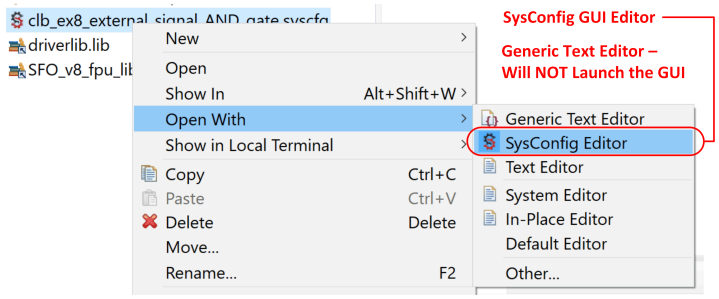
For detailed technical information on how to get started with C2000 SysConfig, visit SPRACX3 . The C2000 SysConfig support is built on top of the C2000 driverlib software layer. To get started, either start from an existing C2000 SysConfig based driverlib project or add C2000 SysConfig and driverlib support to an existing project. Most driverlib examples in C2000Ware have either an example.syscfg file or you can add a file with the .syscfg extension. Double clicking and opening the .syscfg file launches the C2000 SysConfig tool.

To get started with C2000 SysConfig, let's import an existing example with C2000 SysConfig support.

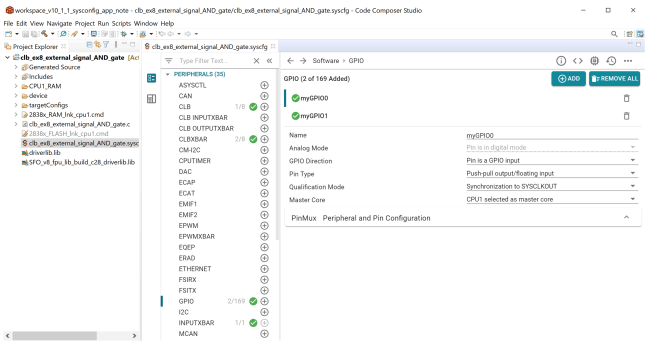
- **Launch CCS and import the example: clb_ex8_external_signal_AND_gate.projectspec**
  - Select Project → Import CCS Project
  - Browse to C2000Ware_VERSION/driverlib/f2838x/examples/c28x/clb/CCS
  - Select the clb_ex8_external_signal_AND_gate.projectspec project and import it
- **Inside your CCS project you should be able to see the syscfg file**
  - You should also see the rest of the application files
- **Double click on clb_ex8_external_signal_AND_gate.syscfg file and the C2000 SysConfig GUI will launch**
  - You can also right-click on the syscfg file, then select Open With → SysConfig Editor

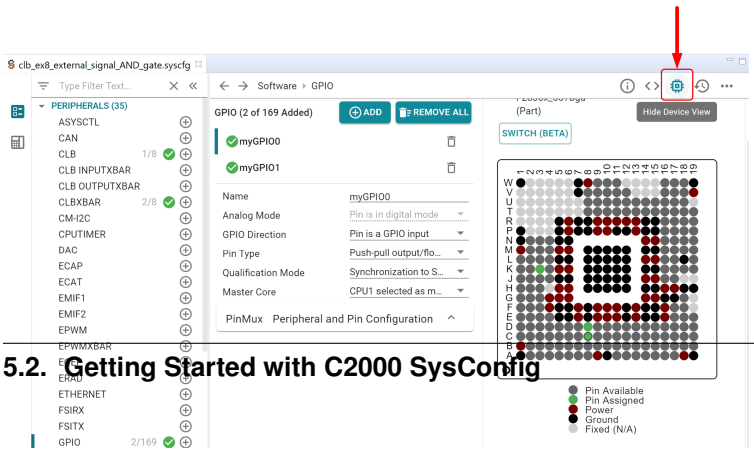


- **The C2000 SysConfig GUI should be launched inside CCS**
  - It should look similar to one shown below



- **Click the Device View button at the top right corner of the SysConfig GUI**
  - Now you can see the device and package used for your project





C2000 SysConfig support is added in the Project Properties. By default, this project was configured for F2838x family of devices and the selected device package is set to 337 BGA package. If the Project Properties for C2000 SysConfig support is not set up by default in your CCS project, the syscfg file will not launch the GUI succesfully. Most driverlib projects have the Project Properties set up by default for C2000 SysConfig. If

Project Properties are not set up correctly, SPRACX3 describes how C2000 SysConfig can be added to a CCS project.

5.3 C2000 SysConfig on dev.ti.com

C2000 SysConfig can be accessed on the web through SysConfig. The web version of SysConfig allows users to get started with their developement without downloading any tools.

5.4 C2000 SysConfig on Resource Explorere and CCS Cloud

C2000 SysConfig is available on the CCS cloud version. Users can get started with C2000 SysConfig in the context of a CCS cloud project by importing a SysConfig based example such as the one below into their CCS cloud workspace.

- F2838x clb_ex8_external_signal_AND_gate

To access all C2000 examples in resource explorer, visit:

- C2000Ware Examples on Resource Explorer

5.5 C2000 SysConfig Universal Project

C2000 Sysconfig supports migration across our expansive C2000 device portfolio with a click of a button. The universal project within the C2000WARE SDK is intended for users to start development on one device and be able to migrate to any C2000 device with one click of a button. This one click solution is integrated within SysConfig and makes it easy to select features and handle device resource/configuration.

For a detailed walkthrough on how to get started with the universal project, visit Universal C2000 Project

CHAPTER

SIX

LEGACY SOFTWARE COMPONENTS

This section describes C2000 legacy software packages.

6.1 C2000 controlSUITE™

C2000Ware is the successor to controlSUITE as the centralized, interactive, software repository for C2000. The application note "controlSUITE to C2000Ware Transition Guide" SPRUI45 highlights the differences, and similarities, between the packages to assist in getting started using C2000Ware when migrating from controlSUITE. ControlSuite is available for download from http://www.ti.com/tool/CONTROLSUITE.

The following content have been migrated to C2000Ware. Updates to this content will only be provided through C2000Ware.

- **Generic libraries (Math, DSP, Flash API, Boot ROM. . .)**
  - Multiple devices including F2802x, F2803x, F2805x, F2806x, F2833x, F2823x
- **Device Support: bit-field source and headers, examples, linker command files**
  - F2802x, F2803x, F2805x, F2806x, F2833x, F2823x

The following content will stay in controlSUITE with no plans to update.

- **TI Designs and Application examples**
  - F2802x, F2803x, F2805x, F2806x, F2833x, F2823x
  - F28M35x, F28M36x, C2834x
- **Device and driver support**
  - F28M35x, F28M36x, C2834x
- **Note: F2837xD, F2837xS, and F2807x**
  - An early version of controlSUITE was released for these devices. But this content is not supported and has been migrated to C2000Ware and the SDKs. Users should migrate to C2000Ware and the SDKs for these devices.

TI Designs and Application examples for newer devices, not listed above, are supported through the DigitalPower SDK and MotorControl SDK.

6.2 MotorWare™

C2000 MotorWare is the software and documentation package for developing InstaSPIN-FOC™ and InstaSPIN-MOTION™ based applications for F2802x, F2805x, and F2806x series. No further revisions to MotorWare is expected. MotorWare can be downloaded from http://www.ti.com/tool/MOTORWARE

For newer F28x-based devices users should download the C2000Ware MotorControl SDK.

Warning: C2000 controlSUITE and MotorWare are no longer being updated, but will continue to be available for download. The latest software releases for C2000 are now provided through C2000Ware with application examples through the DigitalPower SDK and MotorControl SDK.

CHAPTER

SEVEN

ADDITIONAL RESOURCES

The following lists some of the foundational application notes, training materials and documents related to development or debug of C2000 software. More documentation can be found in the TI.com product folder for your specific device.

7.1 C2000Ware and SDK Documentation

Within the top level /docs directory of C2000Ware, the C2000 Digital Power SDK and the C2000 MotorControl SDK is a comprehensive list of all documentation within the package. The documentation includes hardware files, library user guides, and descriptions of examples.

- **This list can be found in the following files:**
  - C2000Ware_documentation.html in <c2000Ware_install>/docs/
  - C2000Ware_MotorControl_SDK_Documentation.hml in <MC_SDK_install>/docs/
  - C2000Ware_digitalpower_sdk_documentation.html in <DP_SDK_install>/docs/

7.2 User's Guides

- **C2000 TMS320C28x Optimization Guide**
  - Describes a systematic approach to improving the performance of applications executing on the TMS320C28x CPU in C2000™ MCUs.
- **C2000 CLA Software Development Guide**
  - Describes how to leverage the Control Law Accelerator (CLA), co-processor for C28x, along with its software development.
- **C2000 Multicore Development User Guide**
  - Describes various cores available in C2000™ devices and how other resources such as peripherals, memories, GPIOs are shared among these. It also talks about how to communicate and share data among different cores.
- **TMS320C28x Optimizing C/C++ Compiler User's Guide - SPRU514**
  - Explains how to use the following Texas Instruments Code Generation compiler tools: Compiler, Post-link optimizer, Library build utility and C++ name demangler.
- **TMS320C28x Assembly Language Tools User's Guide - SPRU513**

  - Describes the assembly language tools (assembler and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives.

7.3 Application Notes

- **Getting Started With C2000™ Real-Time Control Microcontrollers (MCUs) - SPRUIV6**
  - This guide is a valuable reference that contains all of the necessary information to get started with C2000™. This guide covers all aspects of development with C2000 devices from hardware to support resources.
- **The Essential Guide for Developing With C2000™ RealTime Microcontrollers - SPRAN0**
  - This application report provides a deeper look into the components that differentiate the C2000 MCU as it pertains to Real-time control systems.
- **Signal Chain Benchmarking - A Demonstration of- Optimized Performance of C2000™ Real-time Control MCU - SPRAC**
  - This application report describes a signal chain benchmark created around a real world control application that highlights the intricacies of real-time control and the need for this more comprehensive benchmarking approach.
- **Programming TMS320x28xx and TMS320x28xxx Peripherals in C/C++ - SPRAA85**
  - This application report explores hardware abstraction layer implementations to make programming of peripherals easy using C/C++ on TMS320x28xx and TMS320x28xxx devices. The methods of using bit field structure header files and the C2000™ Peripheral Driver Library are compared to each other and to the traditional #define macro approach. Topics of code efficiency and special case registers are also addressed.
- **controlSUITE to C2000Ware Transition Guide - SPRUI45**
  - Provides a list of differences and similarities between C2000Ware and ControlSUITE development packages
- **C2000 Real-time Control Peripherals - SPRU566**
  - This application report describes the peripherals available for various C2000 devices. Details of peripherals available in each device and description of peripherals are available in this report.
- **C2000™ Unique Device Number - SPRACD0B**
  - This application report describes how 32/64-bit value stored in each device during manufacturing can be used as a unique device identifier
- **Secure BOOT on C2000 Device - SPRACT3**
  - This application report describes how the secure flash boot feature can be used to perform an application boot from flash.
- **Updating Firmware on Security Enabled TMS320F2837xx or TMS320F2807x Devices - SPRACI5**
  - This application report describes the basic DCSM protections and how to update the firmware in the Flash when the DCSM protections are enabled.
- **Enhancing Device Security by Using JTAGLOCK Feature - SPRACS4**
  - This application report provides details on how to leverage JTAGLOCK feature on C2000 device.
- **C2000™ DCSM Security Tool - SPRACP8**
  - This application report examines the features of the C2000 DCSM Security Tool.

- **Migrating Software From 8-Bit (Byte) Addressable CPU's to C28x CPU**
  - This application note discusses common scenarios faced while migrating from a 8-bit addressable device to a 16-bit addressable device and provides a guide on how to develop application irrespective of the addressability.
- **Diagnosing Delta-Sigma Modulator Bitstream Using C2000™ Configurable Logic Block (CLB)**
  - This application note describes a technical diagnostic application involving a delta-sigma modulator interfaced with a C2000 microcontroller.

7.4 Workshops and Training

- C2000 Academy hands-on labs
- **C2000 MCU Device Hands-on Workshops**
  - The C2000 Microcontroller (MCU) Workshops have been developed to help engineers gain a full understanding and complete working knowledge of the C2000 MCU family. Learning is accomplished through a detailed workshop manual and by performing the hands-on lab exercises. Each workshop starts with the basic concepts and progresses to more advanced topics in a logical flow. The topics and lab exercises build on the previous one completed, running a common theme throughout the workshop.
- **Control Law Accelerator (CLA Hands-On Workshop)**
  - This training series covers the workings of the Control Law Accelerator (CLA). It begins with a basic introduction to the CLA, its architecture and programming paradigm. There is a hands-on workshop that can be downloaded, and worked through, as you progress through the modules.

7.5 Support Links

- **TI E2E™ support forums**
  - Receive fast and reliable technical support from our engineers throughout every step of your design
- **C2000-3P-SEARCH**
  - TI has partnered with multiple companies to offer a wide range of solutions and services for TI C2000 devices

7.6 TI.com Web Articles

- The following web-based articles are available on these specific topics:
  - Byte Accesses with the C28x CPU
  - C28x Compiler Error and Warning Messages
  - C28x Compiler Understanding Linking
  - C28x Context Save and Restore
  - C28x Pipeline Conflicts
  - C28x Interrupt Nesting

- C28x Interrupt FAQ
- C2000 Migration from COFF to EABI

CHAPTER

EIGHT

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, 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 (https://www.ti.com/legal/termsofsale.html) 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.

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