controlSUITE™ software

Real-world software for real-time control.
controlSUITE™ software

controlSUITE for C2000™ microcontrollers is a cohesive software suite designed to minimize software development time. From device-specific drivers and support software to complete system examples in sophisticated system applications, controlSUITE provides solutions at every stage of development and evaluation.

**controlSUITE Benefits**

**One stop for all C2000 software**
- Single, centralized location
- Intelligent installer eliminates the search for dependencies

**Open, real-world systems**
- Compilation of 15 years of systems and applications expertise
- Unique, optimized libraries for math, filtering, DSP and specific applications with complete system examples utilizing incremental builds
- Allows developers to focus on differentiation, not basics

**Program the MCU your way**
- Significantly reduces development time with hardware abstraction and extensive libraries
- Four inter-usable levels of hardware abstraction

An embedded engineer must answer many critical questions when selecting a new MCU platform. controlSUITE turns those questions into answers.

**What development environment should I use?**
What software tools are there? What debug options do I have?

**How does the device work?**
What support do I have? What libraries do I have to work with?

**How do I take the MCU and put it into my application?**
How can I differentiate myself from competition?

---

**Application Development**

**Device Evaluation**

- Code Composer Studio v4
  - Eclipse-based IDE
  - Full C/C++ support
  - BIOS real-time Operating System

- From Partners and 3rd Party Network
  - Code generation tools
  - Modeling and simulation tools

- Three device interfacing options
  - Bit Field Headers
  - Framework
  - API Drivers
  - Libraries and Utilities
  - IQMath – virtual floating-point
  - Detailed documentation, FAQs, walkthroughs and tutorials
  - Graphical User Interfaces (GUIs)

---

**Debug Tools and Software Tools**

- Modular Application Libraries
  - Digital Motor Control
  - Digital Power
  - Fully-implemented systems
  - Hardware Development Kits
  - Software frameworks demonstrate individual technologies and system integration

---

www.ti.com/controlsuite
Device drivers, APIs, utilities and libraries are used to build technology examples and system frameworks that are demonstrated on open source hardware using TI’s professional debug tools.

**controlSUITE provides solutions**

*I never know if I am aware of every resource that’s available.* Install controlSUITE once and it delivers a comprehensive package of every resource available for your device or application.

*I am never sure I am using the latest version of software, or if something completely new has been released.* controlSUITE can automatically or manually synchronize with a central repository, keeping you up to date with the latest revisions or newest offerings.

*My current vendor provides resources that are a) good, but expensive b) free, but are not properly supported c) licensed, and will not provide source.* Everything in controlSUITE is completely free, meticulously documented, under version control, and nearly 100% open source on all software and hardware.

*I like to explore a system example, then tweak to my needs vs. I like to build my application from the ground up, bit by bit.* controlSUITE offers the complete solution, with four levels of hardware abstraction, libraries, software examples, full systems, and GUIs you can jump in at any level depending on your experience and comfort.
**Device Support** Source code and examples that enable easy device and peripheral initialization and configuration

### 4-level Hardware Abstraction Layer

**Level 1 – Registers and Addresses**
Baseline assembly or C communication to all hardware registers and addresses. Most MCU vendors stop here.

**Level 2 – Bit Fields**
At the heart of C2000 software is a set of memory-mapped bit fields and structures that make registers and register bits seem like simple variables.
- Bit fields can be manipulated without masking
- Flexibility to access a register as a whole or by bits
- Auto-complete field names in CCStudio
- View in CCStudio watch window just like variables

**Level 3 – API Drivers**
The peripheral API drivers are a high-level library of functions that wrap common tasks into parameter-based functions.
- C call-able functions that automatically set register bit fields
- Further reduces learning curve for new programmers
- Common tasks and peripheral modes supported

**Level 4 – Framework**
Built primarily upon the bit field headers, the framework represents a system template and starting point for all applications.
- Function-based device initialization, allowing users to focus on application code
- System management via state machines set to run at a specific frequency
- Ability to connect to an external GUI via SCI
- Simple switching between RAM and Flash eases early development
- Pre-configured GPIO mapping

---

**Example code provided across all levels**

**Using direct register access**
```
interrupt void IsrAdc( void )
{
    // Period of ePWM1 is set in init;
    // Multiply period by desired duty
    // to get CMPA value;
    EPwm1Regs.CMPA.half.CMPA = EPwm1Regs.TBPRD * duty;
}
```

**Using bit field headers**
```
interrupt void IsrAdc( void )
{
    /* set a new pwm value */
    PWM_setDutyA(PWM_MODULE_2, duty);
}
```

**Using API Driver**
```
interrupt void IsrAdc( void )
{
    // Interrupts set up elsewhere
// Set duty cycle
MOVB @9,#0x0F,UNC
// Set PWM1A on Zero Event
AND AL,#0x0FFFC
ORB AL,#0xFFFC
MOV @11,AL
// Clear PWM1A on Up-count
// CompareA event
AND AL,#0x0FFFCF
ORB AL,#0x10
MOV @11,AL
```
Libraries  Core building blocks and utilities used across all systems

DSP Library — Provides a wealth of popular DSP operation including FFTs, filters, vector math and matrix math.

Utilities – Flash API and Boot ROM Utilities

IQMath Library
A library and compiler intrinsic that allows you to select your range and resolution by choosing which bits of your binary represented number are integer (I) and which are the quotient (Q). It also allows you to write C functions in floating point format instead of dealing with fixed-point scaling, and the compiler takes care of the rest.

Math Library — Basic math operations such as trigonometry issued across three sets of libraries – optimized for fixed-point, floating-point or the Control Law Accelerator (CLA).

Application Libraries
Specialized, application specific software functions
• Modular macros with variable inputs and outputs
• At initialization all variables are defined and outputs of one block are set as inputs to the next
• Complete documentation – including source code, use and technical theory – is provided for every module

Digital Motor Control Library

Transforms and Estimators
• Clarke, Park, SMObserver, Phase Voltage, Resolver, Flux, Speed Calculators and Estimators

Control
• Signal Generation, PID, BEMF Commutation, Space Vector Generators

Peripheral Drivers
• Different modes and topology support
• ADC, PWM, Encoders, Sensor Captures

Digital Power Library

Control and Math
• 2P2Z, 3P3Z, IIR, PID, AC Rect, PFC, Constant Power/Voltage, Peak/Average/Over Current, LLC, Ramp/Sine/Slew Generators, Sequencers, PID Mapping, Data Logging, Math, Filters

Topologies
• SynchBuck, SB2IL, SB3IL, BuckBoost, PSFB, PFC+MPIL, HHB, FB, Resonant, High Res, DAC

Analog Drivers
• Different modes and configurations – ADC, COMP

Ex: Using “Park” from DMC Library
//initialization code, define macro per library
#define PARK_MACRO(v)

v.Ds = _IQmpy(v.Alpha,v.Cosine) + _IQmpy(v.Beta,v.Sine);
v.Qs = _IQmpy(v.Beta,v.Cosine) - _IQmpy(v.Alpha,v.Sine);

//incremental build code, connect outputs and inputs
park1.Alpha = clarke1.Alpha;
park1.Beta = clarke1.Beta;

//run-time code, call the function
PARK_MACRO(park1)
Applications: Motor Control  Full example systems have been built using the Digital Motor Control (DMC) library

**Incremental Build**

DMC example systems all use an Incremental Build approach, which allows an incremental section of code to be built so that the developer can verify each section of their application one step at a time. This is critical in real-time control applications with so many different variables that may affect the system and so many unique challenges across motor parameters.

For example, in the Sensorless PMSM FOC system below the following incremental builds are built into the software:

**Build Level 1:** Using a dummy signal, verify Inverse Park, Space Vector and PWM driver are producing correct waveforms

**Build Level 2:** Verify ADC conversion, Phase Voltage calculation, Clarke and Park transforms

**Build Level 3:** Closed loop PID current control verification

**Build Level 4:** Sliding Mode Observer and Speed Estimator verification

**Build Level 5:** Closed loop PID speed control

---

**Documentation**

**All systems come with:**
- Incremental build software
- Step-by-step user’s guide
- DMC control theory
- Screenshots
- Hardware hook-up
- Debug tips

**All hardware example systems are open source, ready for you to insert into your design and include:**
- Gerber files
- Schematics
- BOM
- Feedback and sensing circuitry
- Isolated JTAG

<table>
<thead>
<tr>
<th>Motor</th>
<th>Technique</th>
<th>Type</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACI</td>
<td>FOC</td>
<td>Speed &amp; Torque</td>
<td>Tachometer</td>
</tr>
<tr>
<td></td>
<td>FOC</td>
<td>Speed &amp; Torque</td>
<td>Sensorless</td>
</tr>
<tr>
<td>BLDC</td>
<td>Trapezoidal</td>
<td>Speed</td>
<td>Half Effect</td>
</tr>
<tr>
<td></td>
<td>Trapezoidal</td>
<td>Speed</td>
<td>Sensorless</td>
</tr>
<tr>
<td></td>
<td>Sinusoidal</td>
<td>Speed</td>
<td>Half Effect</td>
</tr>
<tr>
<td></td>
<td>Sinusoidal</td>
<td>Speed</td>
<td>Sensorless</td>
</tr>
<tr>
<td>PMSM</td>
<td>FOC</td>
<td>Speed &amp; Torque</td>
<td>Encoder</td>
</tr>
<tr>
<td></td>
<td>FOC</td>
<td>Position</td>
<td>Encoder</td>
</tr>
<tr>
<td></td>
<td>FOC</td>
<td>Speed &amp; Torque</td>
<td>Sensorless</td>
</tr>
<tr>
<td>Stepper</td>
<td>Microstep</td>
<td>Position</td>
<td>Sensorless</td>
</tr>
<tr>
<td>Brushed</td>
<td>DirectDrive</td>
<td>Speed &amp; Position</td>
<td>Encoder</td>
</tr>
</tbody>
</table>

**Systems included or planned**

**Motor types, control and feedback techniques**

These system examples have been created across different motor types, control techniques and feedback methods.

**C2000 Digital Motor Control Gives You MORE**

C2000 based motor control systems enable energy and cost savings throughout products

- Variable speed control    ➔ MORE efficient motors
- Field-oriented control   ➔ MORE efficient control
- Space vector PWM         ➔ MORE efficient power stage
- Sensor-less control      ➔ MORE cost effective
- Multi-axis control       ➔ MORE motors per controller
- Integrated digital PFC   ➔ MORE system functions
- Meeting IEC standards    ➔ MORE reliable and robust
- Broadest MCU portfolio   ➔ MORE products, one platform
Applications: Digital Power  Full example systems have been built using the DPS library

**Multiple control methods and multiple topologies through modular software**

**Why go for digital power?**

**Reduces costs**
- Tunable platforms lead to new products quicker
- Calibration, better noise and temperature immunity
- Reduced board area and parts count

**Higher quality**
- Adaptive; efficiency across load range
- Flexibility through programmability
- Calibration at final functional test
- Less sensitive to drift and better noise immunity
- Parameter monitoring for continual quality improvement
- Proven concept in mature digital motor control market

**Higher reliability**
- Built-in supervision
- Intelligent diagnostics, failure prediction, reporting capability

**Control Topologies**
Flexible peripheral support, with DPS library of drivers for any power stage topology, including
- Synchronous buck: 1, 2, 3 phase interleaved
- Synchronous buck boost
- Power factor correction: 1, 2, 3 phase interleaved
- Half and full bridge, phase shifted and synch rectification
- Resonant

**Control Techniques**
Programmable control techniques with performance and flexibility for personalized system management and product differentiation
- 2 Pole, 2 Zero
- 3 Pole, 3 Zero
- PI, PID
- Non-linear techniques
- Adaptive co-efficient sets
- Phase management
- Fault monitoring, clamping, dead-band adjustment
- Tunable efficiency across load and operating range

www.ti.com/controlsuite
Development Kits

C2000 development kits are designed to be modular and robust. As completely open source evaluation and development tools, users can modify both the hardware and software to best fit their needs.

**Hardware and software packages for every kit available in controlSUITE**

Each C2000 development kit comes with complete software and hardware support. Regardless of whether you have a device evaluation kit (such as the controlSTICK or experimenter’s kits) or a full application kit, controlSUITE contains everything you need to maximize the benefits of these kits.


---

**C2000 Tools Platforms**

**controlSTICK Platform**

As an entry-level evaluation kit, the controlSTICK represents a simple, stand-alone device that allows users to simply plug in the tool to learn the device and software. Features include:

- Simple stand-alone USB memory stick form factor evaluation tool
  - Piccolo F28027
  - Onboard USB JTAG emulation
  - Header pins provide access to most Piccolo pins
- 11 example projects explain Piccolo peripherals
- Jumpers and patch cords included for easy connectivity
- USB extension cable

---

**controlCARD Platform**

With a standard DIM100 interface and identical pin-outs, evaluating or migrating between devices has never been easier. controlCARDs bring simplicity and compatibility, adding extra usability to C2000 tools.

- Pin-compatible daughter cards provide an easy method to evaluate multiple C2000 devices
- Standard DIM interface allows access to analog I/O, digital I/O, and JTAG signals
- Robust design
  - Noise filter at ADC input pins
  - Ground plane
  - Isolated UART communication
  - Supply pin decoupling
- All C2000 MCU life support included on card
- controlCARD can be added to any design by adding a DIM100 connector and a 5V supply
- Simple, small form factor for small prototype or production runs
- Multiple versions available starting at $49

---

[www.ti.com/controlsuite](http://www.ti.com/controlsuite)
Development Kits

**controlCARD Based Device Evaluation Kits**
Evaluation kits assist the users in learning C2000 devices

**Experimenter’s Kits**
This full-featured kit brings access to all C2000 device pins. With an integrated prototyping area, the Experimenter’s kit is a great tool for initial development or experimentation.
• Docking Station
  - Prototyping areas
  - Access to most controlCARD signals
  - 5V and 3.3V rails
  - On-board USB JTAG emulator
    – Excludes C28345 and C28346
  - Kits available for all controlCARDs

**Peripheral Explorer Kit**
The Peripheral Explorer Kit provides a simple way to learn and interact with all F28335 peripherals.
• Example projects to explain GPIO, ADC, PWM, Delfino DMA, SPI, I2C, and more
• EEPROM for SPI communication example
• On-board audio codec for high speed serial
• On-board USB JTAG emulator

**controlCARD Based Application Kits**
Take full advantage of C2000’s powerful core and peripherals with the application development kits. For beginners and experts alike, C2000 application kits demonstrate the full capabilities of digital control.

All application kits include:
• Open source, application example software
• Detailed lab style documentation
• Complete hardware documentation
• controlCARD and application baseboard included

- **High Voltage, DMC + PFC Developer’s Kit**
  TMDSHVMTRPFCKIT – $599

- **Sensorless FOC DMC + PFC Developer’s Kit**
  TMDS1MTRPFCKIT 1 Motor – $369
  TMDS2MTRPFCKIT 2 Motor – $399

- **Digital Power Experimenter’s Kit**
  TMDSDCDC2KIT – $229

- **Digital Power Developer’s Kit**
  TMDSDCDC8KIT – $325

- **AC/DC Developer’s Kit**
  TMDSACDCKIT – $695

- **Resonant DC/DC Developer’s Kit**
  TMDSRESDCKIT – $229

- **Renewable Energy Developer’s Kit**
  TMDSENRGYKIT – $349

www.ti.com/controlsuite
Debug and Development Tools

The right software development tools are essential in any project. C2000’s controlSUITE, along with C2000 partners, provide a complete ecosystem of debug and software tools.

**Code Composer Studio™ (CCS v4)**

TI’s Code Composer Studio Version 4 is a complete development environment. Based on the popular Eclipse IDE, CCSv4 represents a familiar and friendly interface with hundreds of plug-ins to support it.

controlSUITE includes a FREE, fully-featured version of Code Composer Studio v4 (for use with XDS100 emulators)

- Takes advantage of the real-time debug circuitry on C2000 devices and provides a powerful debugging environment.
- Real-time watch windows and graphs aid in software verification and debugging.
- Interrupt debugging is made possible by interrupt servicing even when the main program is halted.
- Includes many additional tools for C2000, such as the BIOS real-time operating system, an emulator, flash programmer, and more.

**Real-time Debug**

Traditional debugging approaches (Stop Mode) require that programmers completely halt their system which stops all threads and prevents interrupts from being handled, which makes debugging extremely difficult if the system/application has real-time constraints. Real Time Mode debug support provides a better gauge of real-world system behavior by enabling programmers to halt and examine the application while allowing user specified time critical interrupts to be handled.

- Enables real-time, non-intrusive, continuous visibility into the way target applications operate in the real world
- Does not require use of target memory, special interrupts, or integration with the application
- Allows time critical interrupts to be marked for special treatment (high priority)
- Allows time critical interrupts to be serviced while background program execution is suspended
- Built into C2000 devices and Code Composer Studio
- Real-time refresh options

**BIOS Real-time Operating System (RTOS) v6**

BIOS is TI’s royalty-free real-time operating system for DSP processors, including C2000 and the C28x core. BIOS is a complete and powerful solution to OS and scheduling needs. BIOS requires no runtime license fees and is backed by Texas Instruments’ worldwide training and support organizations. BIOS includes:

- Deterministic kernel with a preemptive scheduler
- Graphical or script-based OS configuration
- Graphical analysis and debug tools
- Interrupt dispatcher and interrupt management macros
- Multiple intertask communication services including semaphores, mailboxes and queues

Screenshots from BIOS v6 Code Composer Studio, showing task CPU load and task priorities

**Emulation Tools**

JTAG emulation is a critical part of debugging and C2000 offers a variety of options to satisfy your emulation needs.

- Many evaluation kits have an on-board XDS100 emulator
- Stand-alone XDS100 emulators available for as low as $79
- Faster, more capable XDS510 emulators available for as low as $249
- Isolated emulators also available

**Partners and Developer Network**

C2000’s developer network includes partners that specialize in debug and software development tools. Vissim and Matlab are powerful tools that reduce programming to a visual process. Visit our Partners website for more information on these programs and more. [www.ti.com/c2000partners](http://www.ti.com/c2000partners)
Getting Started

Depending on your experience and needs there are two options for getting started with controlSUITE. New users or users who want to learn about developing a system should start with an evaluation kit or application kit. Users who have used MCUs before, have extensive expertise in their application space, and are looking to take advantage of C2000’s powerful C28x core and unique advanced peripherals should start from a device they have selected for their application.

Start from the device

1. Start by downloading controlSUITE and selecting the device support package of your choice.
2. Find the corresponding directory holding the device support package. Inside, you’ll find the actual software along with examples for CCS v4.
3. Using the quick start guide found in the “–docs” folder, you can quickly learn how to set up CCS v4, how to execute the examples, and how to incorporate the headers and sample code into a real project.
4. Import the CCSv4 example projects to see simple demonstrations of the device and peripherals, which can be used as a starting point. An example project exists for every aspect of the device.
5. Before moving to your application, take a look at the development tools that we have. Each comes with complete hardware and software packages, allowing you to reduce design or programming time.

Start with a kit

1. Start by downloading controlSUITE and selecting the kit that you would like to use. The appropriate device support package and baseline framework software will automatically install as well.
2. Find the corresponding directory for the kit. Inside, you’ll find the baseline framework along with specific examples and a hardware package including everything you need to understand the hardware — schematics, layouts, bill of materials, and more.
3. Run the GUI, connect to the board, and start playing with the variables to see how the system reacts to inputs.
4. When you’re ready to dig into code, install and open Code Composer Studio v4. Import the example projects. Through experimentation and documentation, learn about the device and framework.
5. Start developing your own application right away. Use the framework and examples as an instant starting point.
6. Or, for users interested in specific applications, install the kit software and learn how the system is put together.
Other Resources

C2000 software and support does not stop at controlSUITE. From online communities to hands-on training, C2000 offers a variety of resources for learning and development acceleration.

**Online Resources**
- E2E community: Join fellow engineers at the TI E2E Community website, where you can find training videos, blogs, and an active forum to find answers to your questions. With a rapidly growing user base, the E2E community will serve as a nexus of all things TI. [community.ti.com](http://community.ti.com)
- Wiki: With large amounts of information and FAQs, the wiki is a great place to start finding answers to your questions. [www.tiexpressdsp.com](http://www.tiexpressdsp.com)

**Training**
- Hands-on training: Attend hands-on training labs and get face time with an instructor on C2000 devices and applications.
- Online training: Learn at your own pace. Watch online training videos and presentations from the comfort of your desk or bench. [www.ti.com/training](http://www.ti.com/training)

**Events**
- Come join our MCU Days and TI Technology Days in a location near you. Join lab sessions, lectures, and visit booths and exhibits to see C2000 in action.

---

**TI Worldwide Technical Support**

**Internet**

**TI Semiconductor Product Information Center Home Page**
support.ti.com

**TI Semiconductor KnowledgeBase Home Page**
support.ti.com/sc/knowledgebase

**Product Information Centers**

**Americas**
- Phone: +1(972) 644-5580
- Fax: +1(972) 927-6377
- Internet/Email: support.ti.com/sc/pic/americas.htm

**Europe, Middle East, and Africa**
- Phone: +49 (0) 8161 80 2121
- Fax: +49 (0) 95 98 10 701
- Internet: support.ti.com/sc/pic/euro.htm

**Asia**
- Phone: +81-3-3344-5317
- Fax: +81-3-3344-5317
- Internet/Email: support.ti.com/sc/pic/japan.htm

**Japan**
- Phone: +81-3-3344-5317
- Fax: +81-3-3344-5317
- Internet/Email: support.ti.com/sc/pic/japan.htm

**Note:** The European Free Call (Toll Free) number is not active in all countries. If you have technical difficulty calling the free call number, please use the international number above.

**Important Notice:** The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI’s standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer’s applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company’s products or services does not constitute TI’s approval, warranty or endorsement thereof.

© 2010 Texas Instruments Incorporated

The platform bar, controlSUITE, C2000 and Code Composer Studio are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.

SPRB199