

Using Position Manager BiSS-C Library On IDDK Hardware

User's Guide



Literature Number: SPRUI36

November 2015

1	Introduction to IDDK.....	4
2	Hardware Configuration	4
3	Hardware Overview	6
	3.1 Functional Blocks	6
	3.2 Processor Section (Control Processor Slot – H1)	7
	3.3 Position Encoder Suite	8
	3.4 Power Supply	9
4	Hardware Resource Mapping	10
	4.1 Digital Signal Mapping	10
	4.2 Jumpers and Switches	10
	4.3 Headers and Connectors	10
5	Hardware Setup Instructions for PM_bissc Library Evaluation.....	11
6	Software Setup for PM BiSS-C Evaluation Example Projects	12
	6.1 Installing Code Composer and controlSUITE	12
	6.2 Software Flow Diagram	12
	6.3 Setup Code Composer Studio to Work With TMDXIDDK379D	13
	6.4 Configuring a Project.....	14
	6.5 Build and Load the Project	15
	6.6 Setup Watch Window and Graphs	17
	6.7 Run the Code	18
7	References	18

List of Figures

1	IDDK EVM Kit	6
2	Layout of IDDK EVM With its Functional Macros.....	7
3	Processor Block.....	7
4	Position Sensor Suite	8
5	EnDat/BiSS Interface Header	8
6	Power Supply Block	9
7	M12 Cable, Female End	11
8	Software Flow Diagram for the Example Project PM_bissc_SystemTest	12
9	Configuring a New Target.....	13
10	Adding PM_bissc Example Project to Workspace	14
11	Selecting the F2837x_RAM Configuration	15
12	Selecting the CPUs to Connect	16
13	Configuring the Expressions Window.....	17

List of Tables

1	Hardware Macros in IDDK Used for PM_bissc Evaluation.....	6
2	Digital Signal Mapping on Control Card H1.....	10
3	Purpose of Jumpers and Switches.....	10
4	Headers and Connectors.....	10

Using Position Manager BiSS-C Library On IDDK Hardware

1 Introduction to IDDK

The DesignDRIVE Kit (IDDK) is a single platform that makes it easy to develop and evaluate design solutions for many industrial drive and servo topologies. The IDDK offers support for a wide variety of motor types, sensing technologies, encoder standards, and communications networks, as well as easy expansion to develop with real-time Ethernet communications and functional safety topologies to enable more comprehensive, integrated system solutions. Based on the real-time control architecture of TI's C2000™ microcontrollers (MCUs), the kit is ideal for the development of industrial inverter and servo drives used in robotics, computer numerical control machinery (CNC), elevators, materials conveyance, and other industrial manufacturing applications.

The IDDK offers an integrated drive design with a full-power stage to drive a three-phase motor, easing evaluation of a diverse range of feedback sensing and control topologies. The kit includes a 180-pin HSEC controlCARD based on the TMS320F28379D C2000 Delfino™ MCU, which integrates dual C28x real-time processing cores and dual CLA real-time coprocessors, providing 800 MIPS of floating-point performance with integrated trigonometric and FFT acceleration.

The sophisticated sensing peripherals on the TMS320F28379D MCU, including sigma-delta filter modules with up to eight input channels, four high-performance 16-bit ADCs, and eight windowed comparators, enable the IDDK to support shunt, flux gate/ HALL, and sigma-delta current sensing simultaneously. For position feedback, the IDDK leverages integrated MCU support for resolver and incremental encoder interfaces. In addition, customers can also explore configuration options that allow the MCU to be placed on either side of the high voltage isolation barrier.

The kit is designed to plug into 110-V/220-V AC mains, delivers up to 8 Amps, and is rated to drive motors up to one horsepower.

This document covers the kit contents and hardware details, and explains the functions and locations of various connectors present on the board. This document supersedes all documents available for the kit.

For more details, see the *DesignDRIVE Development Kit IDDK Hardware Reference Guide* ([SPRUI23](#)).

2 Hardware Configuration

To experiment with the IDDK for Position Manager BiSS-C Library evaluation, the following components are needed:

- An IDDK EVM
- TMDXCNC28379D
- USB-B to A cable
- Encoder and connectors NOT included in the kit that are required for evaluation:
 - BiSS-C Encoder (for example, Kuebler 8.F5863.1426.C423)
 - Cable (for the above Kuebler encoder, a 8-pole M12 cable – length as needed by the application (max 100m))

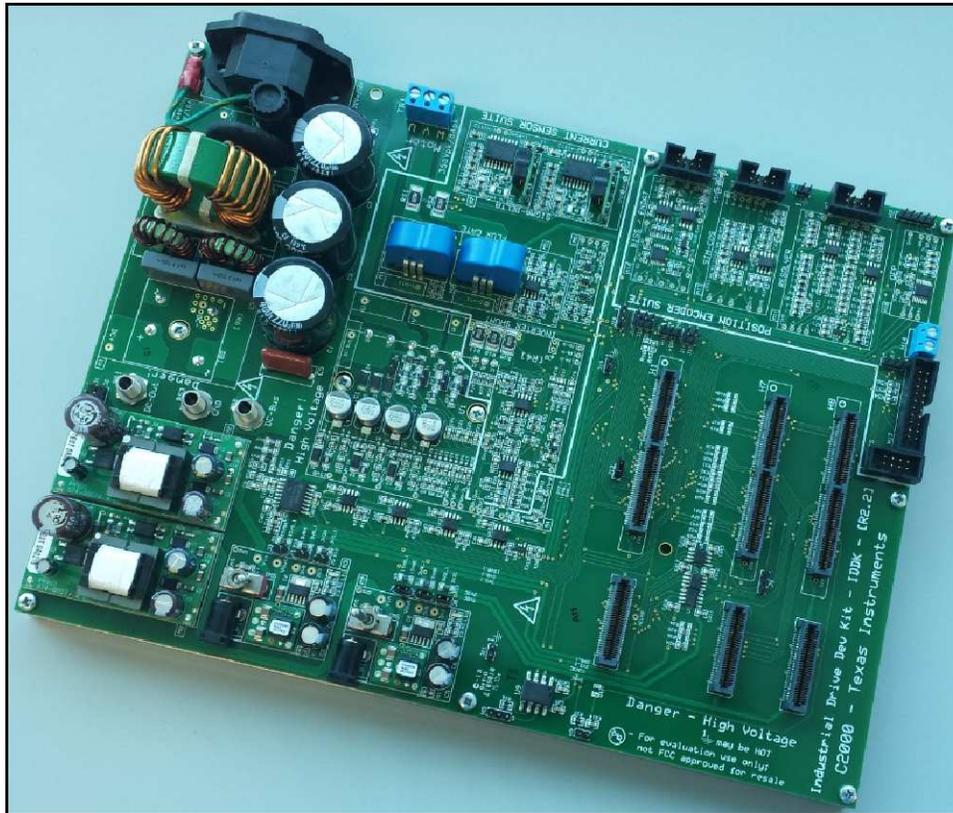


- Adapter to connect cable to IDDK (for the above Kuebler encoder, a circular 8 position female to wire leads – M12 receptacle)



- An external isolated 15-V power supply needed for MCU code development (preferably with a barrel connector commonly referred to as DC jack). Power supply EMSA150120-P5P-SZ is used for testing.
- PC with Code Composer Studio™ (CCSv6 or greater) installed
- *DesignDRIVE Development Kit IDDK Hardware Reference Guide* ([SPRUI23](#)), located at `controlSUITE\development_kits\TMDSIDDK_v2.0\~Docs\`

For the schematic details of the IDDK EVM, see the file located at `controlSUITE\development_kits\TMDSIDDK_v2.0\IDDK_HwDevPkg\IDDK_HwDevPkg_v2.2.1.`


Figure 1. IDDK EVM Kit

3 Hardware Overview

This section describes the subset of the components required for the Position Manager PM_bissc Library evaluation only. Complete hardware overview of the kit can be obtained from the *DesignDRIVE Development Kit IDDK Hardware Reference Guide (SPRU123)*, available on controlSUITE™.

Evaluation of the Position Manager PM_bissc Library requires usage of:

- Processor (CPU) block for control
- Position encoder suite
- On-board power supplies

3.1 Functional Blocks

Table 1 shows the subset of functional blocks on DesignDRIVE Development Kit IDDK Hardware, along with the macro names, used for Position Manager PM_bissc Library evaluation.

Table 1. Hardware Macros in IDDK Used for PM_bissc Evaluation

Functional Block	Macro Reference	Macro Function
Power Supplies	M9	DC Power Supply – Linear Reg 15 V to 5 V / 3.3 V
Position Encoder Suite	M12	Biss Encoder Interface
Processor/controlCARD	H1	All other functions

Each functional block, and the macros that make them, are presented in brief detail in the sections below. The layout of various macros in the board is given in [Figure 2](#). Schematic details of the individual macros are available at `controlSUITE\development_kits\TMDSIDDK_v2.0\IDDK_HwDevPkg\IDDK_HwDevPkg_v2.2.1`.

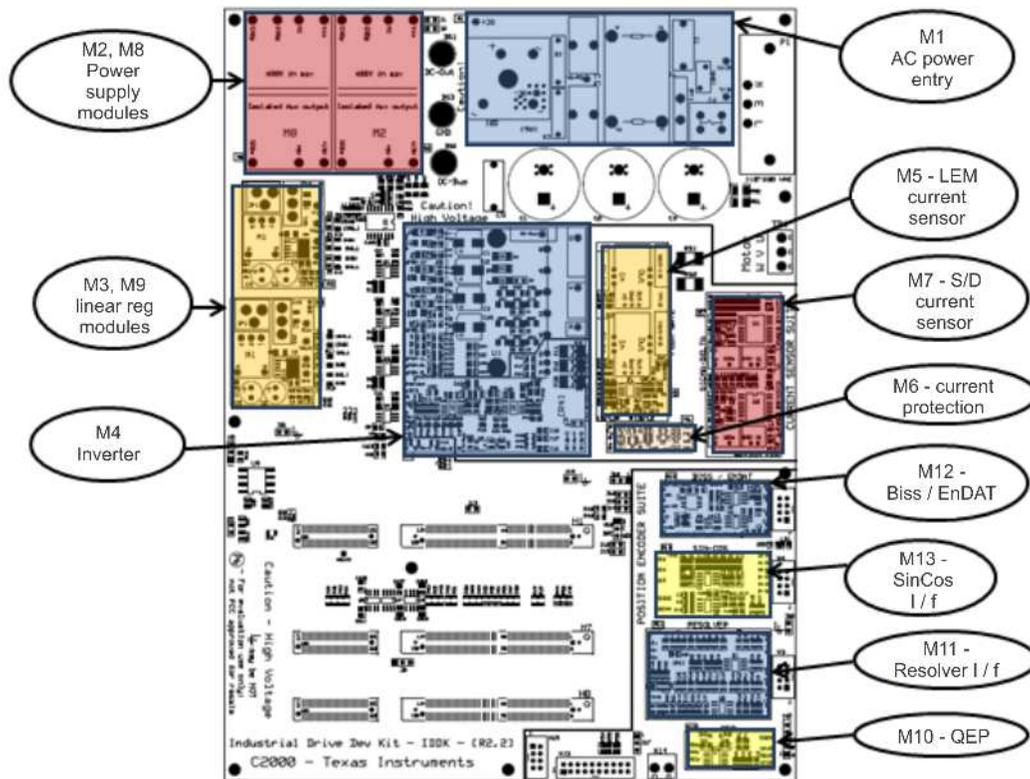


Figure 2. Layout of IDDK EVM With its Functional Macros

3.2 Processor Section (Control Processor Slot – H1)

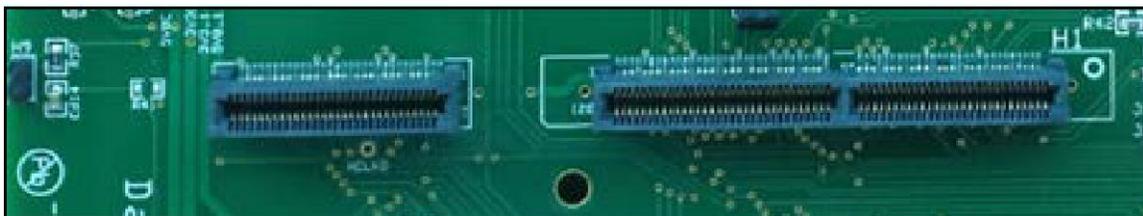


Figure 3. Processor Block

The IDDK is designed around the main control processor card in slot H1. This is designed to plug in a c2000 Delfino (TMS320F28379D) MCU control card TMDXCNC28379D designed with a HSEC180pin edge connector.

3.3 Position Encoder Suite

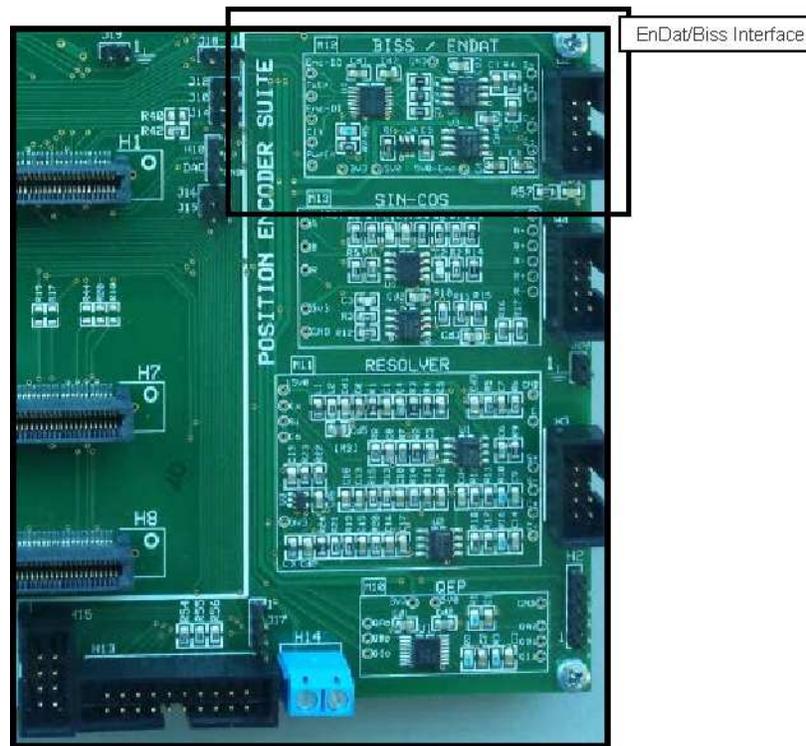


Figure 4. Position Sensor Suite

This block provides a range of position encoder and sensing interfaces, such as:

- QEP
- Resolver
- Sin-Cos
- EnDat / BiSS

PM BiSS-C Library evaluation uses this interface for communication.

3.3.1 BiSS Encoder

This is designed as a macro, referred to as M12. This is a common interface for BiSS/EnDat encoders. H6 is the external interface header and it interfaces digital signals only, as shown in Figure 5. GPIO32 is used to control on/off of 5-V supply details in Table 2.

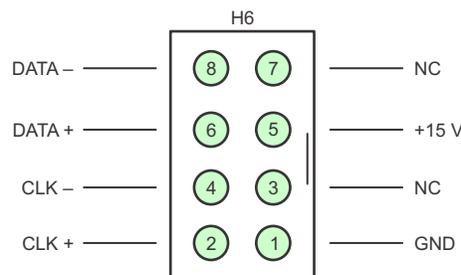


Figure 5. EnDat/BiSS Interface Header

3.4 Power Supply

An external power supply is preferred to power the controller during code development so that the board operates in low voltage. This ensures a safe environment, as there is no high-voltage node present on the board.

The kit only needs 15 V of power, which can be received through M9 or M3. When the controller processor is to be on the cold-side, M9 should be used, which is what will be assumed below.

M9 has a power supply jack ([M9]-JP1) and a toggle switch ([M9]-SW1). An external (15-V) power supply can be fed in through [M9]-JP1, while [M9]-SW1 should be turned away from the Int position when power is needed.

Settings needed to run when the controller processor is on the cold-side:

1. [Main]-J6, J7, and J8 should be jumpered
2. [Main]-J2 should NOT be jumpered
3. [Main]-R9, R11, R13 should be populated
4. [Main]-R8, R10, R12 should NOT be populated
5. Ground resistors on the secondary side of the board should be configured for the controller to be on the cold-side. For more details, see the *DesignDRIVE Development Kit IDDK - Hardware Reference Guide* ([SPRUI23](#)) and the *DesignDRIVE Development Kit IDDK - User's Guide* ([SPRUI24](#)).

Other options, such as running the PM_bissc project from a controller located on the hot-side, are possible with different settings. For more details, see the *DesignDRIVE Development Kit IDDK - Hardware Reference Guide* ([SPRUI23](#)) and the *DesignDRIVE Development Kit IDDK - User's Guide* ([SPRUI24](#)).

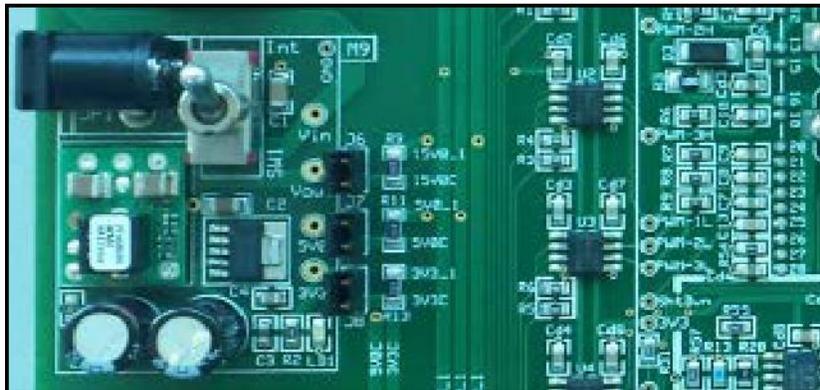


Figure 6. Power Supply Block

4 Hardware Resource Mapping

The sections below show the hardware resource requirements and signal mapping for PM_bissc Library evaluation on IDDK.

4.1 Digital Signal Mapping

Table 2 shows the functional mapping of signals, connected to the control processor on H1, that are relevant to PM_bissc Library evaluation.

Table 2. Digital Signal Mapping on Control Card H1

IDDK Signal Name	MCU GPIO	MCU Peripheral associated with GPIO	Function
En-Clk	6	ABS ENC-CLK	Absolute Encoder Clock out
C1-SPIB-CLK	7	PM CLK	Encoder SPI Clock out
En-DO	24	SPIB	SPI I/f for Abs Encoder
En-DI	25	SPIB	SPI I/f for Abs Encoder
C1-SPIB-CLK	26	SPIB	SPI I/f for Abs Encoder
C1-SPIB-STE	27	SPIB	SPI I/f for Abs Encoder
En-PwrEn	32	GPIO output	Power Enable for Abs Encoder
En-TxEn	34	PM TX En	Tx Enable for Abs Encoder

4.2 Jumpers and Switches

Table 3 lists the jumpers that must be populated for PM_bissc Library evaluation.

Table 3. Purpose of Jumpers and Switches

Jumpers / Switches	Configuration	Description
[Main] - J6- J8	Populate	Jumpers to bring out linear regulator block M9 voltages
[Main] - J10	Populate	Connects GPIO to BiSS/EnDat Clock
[Main] - J12	Don't Populate	
[Main] - J13	Populate	Connect SPISTE to ground
[Main] - J14-17	Don't Populate	
[Main] - J18	Populate (1-2)	Connects Encoder data out (En-Do-1 to the processor)
[Main] - J19-21		GND headers for probe access
[M9] - SW1	Turn ON	Selects 15-V supply from JP1 or onboard from M8 to generate 5 V and 3.3 V

4.3 Headers and Connectors

Table 4 shows the headers and connectors available on board.

Table 4. Headers and Connectors

Headers and Connectors	Description
[M9] – JP1	15-V DC power supply jack adapter
[Main] - H1	180-pin HSEC connector slot for control processor card
[Main] – H6	4x2 header for EnDat/BiSS

5 Hardware Setup Instructions for PM_bissc Library Evaluation

1. Ensure default configuration. Jumpers [Main]-J6, [Main]-J7 and [Main]-J8 must be in front of macro M9 and resistors [Main]-R9, R11, R13 must be populated. GND plane resistors R14 and R15 must be mounted as shown in figures *Various GND Planes on the Bottom of Board* and *Default Connection of Various GND Planes of the Power Supplies and GND Plane Configurations* section of the *DesignDRIVE Development Kit IDDK - Hardware Reference Guide (SPRUI23)*.
2. Unpack the TMDXCNC28379D control card and slide it down in the connector slot of [Main]-H1. Using even pressure on both ends of the card, push down vertically until it cannot slide down further. (To remove the card, spread open the retaining clips, if present, and pull the card out applying even force at the far edges).
3. Connect a USB cable to connector J1 on the control card. The control card isolates the JTAG signals between the C2000 device and the computer. LED D2 on the control card should light.
4. Connect to the encoder:
 - (a) Prepare an adapter cable to connect the encoder cable to the IDDK BiSS interface ([Main] – H6 4x2 header) using Circular 8 position female to wire leads – M12 receptacle.



For more information, see the encoder and cable specifications and hardware schematics in [Section 3.3](#).

- (b) Cable connections between the Encoder and IDDK: insert the header created using the adapter cable above into the [Main] – H6 4x2 header of IDDK. The other end of the adapter connects to the male side of the M12 cable. The female end of the M12 cable connects to the encoder as shown in [Figure 7](#).

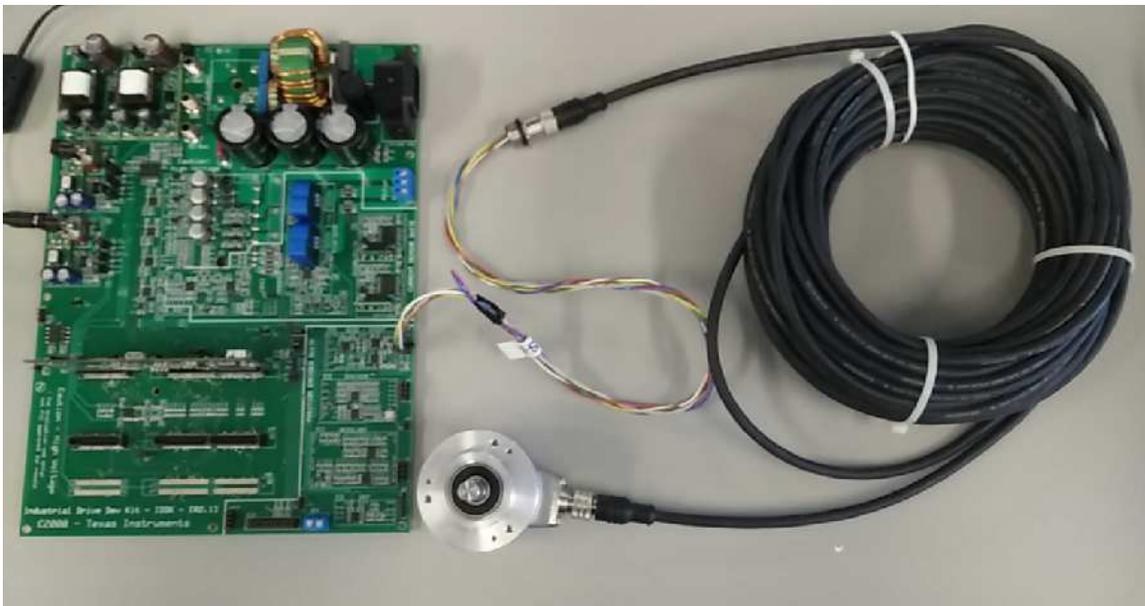


Figure 7. M12 Cable, Female End

5. Ensure that the toggle switch [M9]-SW1 is in Int position. Connect an isolated 15-V DC power supply to [M9]-JP1.
6. Turn on the toggle switch [M9]-SW1. Now, [M9]-LD1 should turn on. More LEDs on the control card should light up, indicating that the control card is receiving power from the board.

6 Software Setup for PM BiSS-C Evaluation Example Projects

6.1 Installing Code Composer and controlSUITE

1. If not already installed, install Code Composer v6.x or later from <http://www.ti.com/tool/CCSTUDIO>.
2. Go to <http://www.ti.com/controlsuite> and run the controlSUITE installer. Allow the installer to download and update any automatically checked software for C2000.
3. EnDat22 Library is available at:
 <base> install directory is C:\ti\controlSUITE\libs\app_libs\position_manager\vx.X
 The following sub-directory structure is used:
 <base>\Doc – Documentation
 <base>\Float – Contains implementation of the library and corresponding include file.
 <base>\examples – Examples using the BiSS-C library

6.2 Software Flow Diagram

The software flow used in Example Project PM_bissc_SystemTest is depicted in Figure 8.

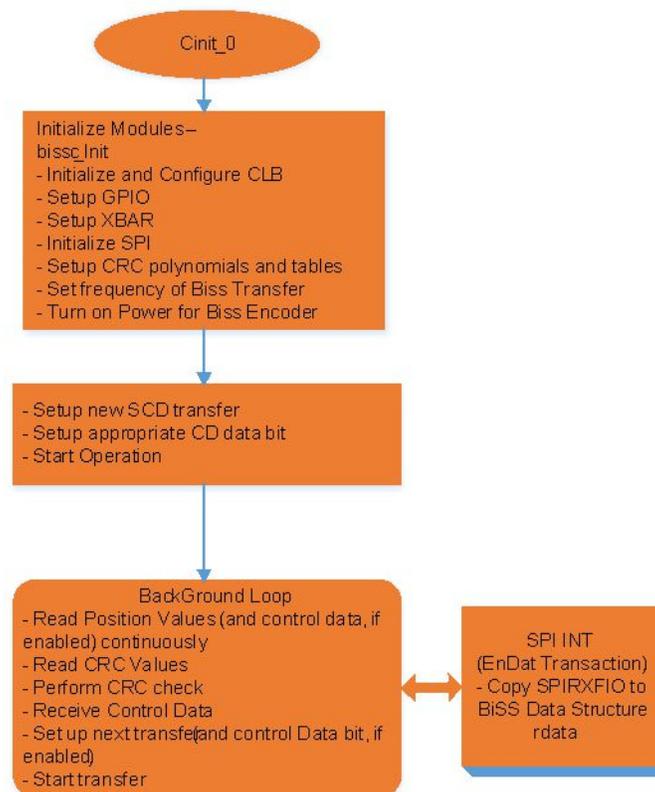


Figure 8. Software Flow Diagram for the Example Project PM_bissc_SystemTest

6.3 Setup Code Composer Studio to Work With TMDXIDDK379D

1. Open Code Composer Studio (CCSv6 or greater).
2. Once Code Composer Studio opens, the workspace launcher appears and asks to select a workspace location. Workspace is a location on the hard drive containing all the user settings for the IDE, such as which projects are open, what configuration is selected, and so forth. This can be anywhere on the disk, the location mentioned below is just for reference. If this is not the first time running Code Composer, the dialog below may not appear.
 - (a) Click the “Browse...” button
 - (b) Create the path `C:\c2000 projects\CCSv6_workspaces\PM_BiSS-C_eval_workspace` by making new folders, as necessary
 - (c) Uncheck the box that says “Use this as the default and do not ask again”.
 - (d) Click OK
3. This opens a Getting Started tab with links to various tasks including creating a new project, importing an existing project, or watching a tutorial on CCS. Click the Import Project Icon to skip the procedure to step 2 in Section 6.4. Or, close the Getting Started tab and go to the next step.
4. Configure Code Composer to know which MCU it will be connecting to by setting up the Target Configuration. This is already set up and configured in `xds100v2_F2837x.ccxml`, which is provided as part of the files in the project; the user can skip to step 1 in Section 6.4. However, for general information regarding setting up this configuration file, use steps 6, 7, and 8 in this list.
5. Set a new configuration file by clicking View → Target Configuration. This opens the Target Configuration window. In this window, click on . Give a name to the new configuration file, depending on the target device. If the Use Shared Location checkbox is checked, this configuration file can be stored in a common location by CCS for use by other projects as well. Click Finish.
6. This opens up a new tab as shown in Figure 9. Select and enter the options as shown:
 - Connection – Texas Instruments XDS100v2 USB Emulator (or) Texas Instruments XDS100v2 USB Debug Probe
 - Device – the C2000 MCU on the control card, such as the TMS320F28379D
 Click Save and close.

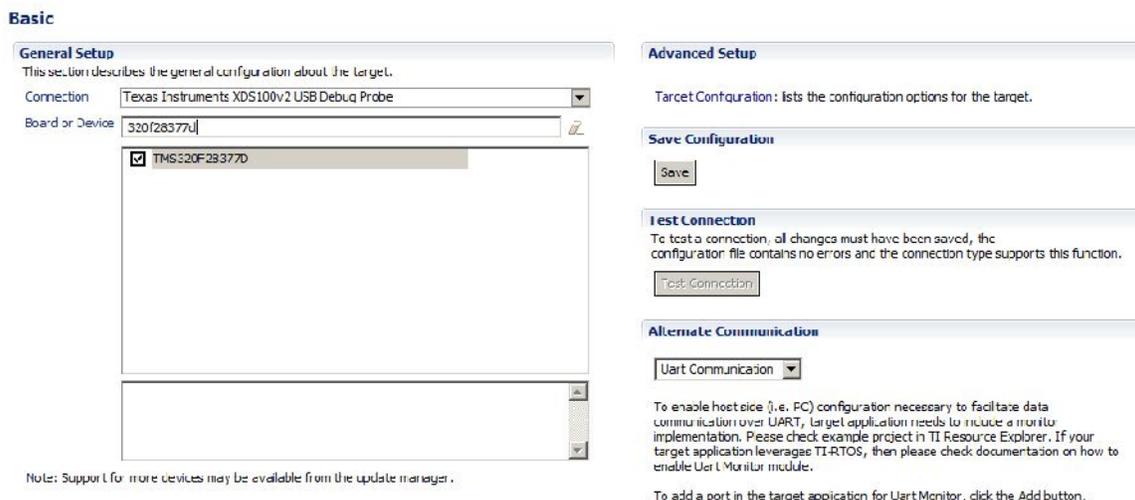


Figure 9. Configuring a New Target

7. Click View → Target Configurations. In the *User Defined* section, find the file that was created in steps 6 and 7 in this list. Right-click on this file and select Set as Default. To use the configuration file supplied with the project, click View → Target Configurations. Then, Expand Projects → PM_bissc_SystemTest, right-click on the `xds100v2_F2837x.ccxml` file and choose Set as Default. This tab also allows you to reuse existing target configurations and link them to specific projects.

8. Add the PM BiSS-C evaluation example project into the current workspace by clicking Project → Import CCS Project. Select the project by browsing to:
`C:\ti\controlSUITE\libs\app_libs\position_manager\v1_00_00_00\BiSS-C\examples\PM_bissc_SystemTest`. If there are multiple projects in this directory, click and choose the projects to import and then click Finish. This copies all the selected projects into the workspace and shows only one project; select it and click Finish.

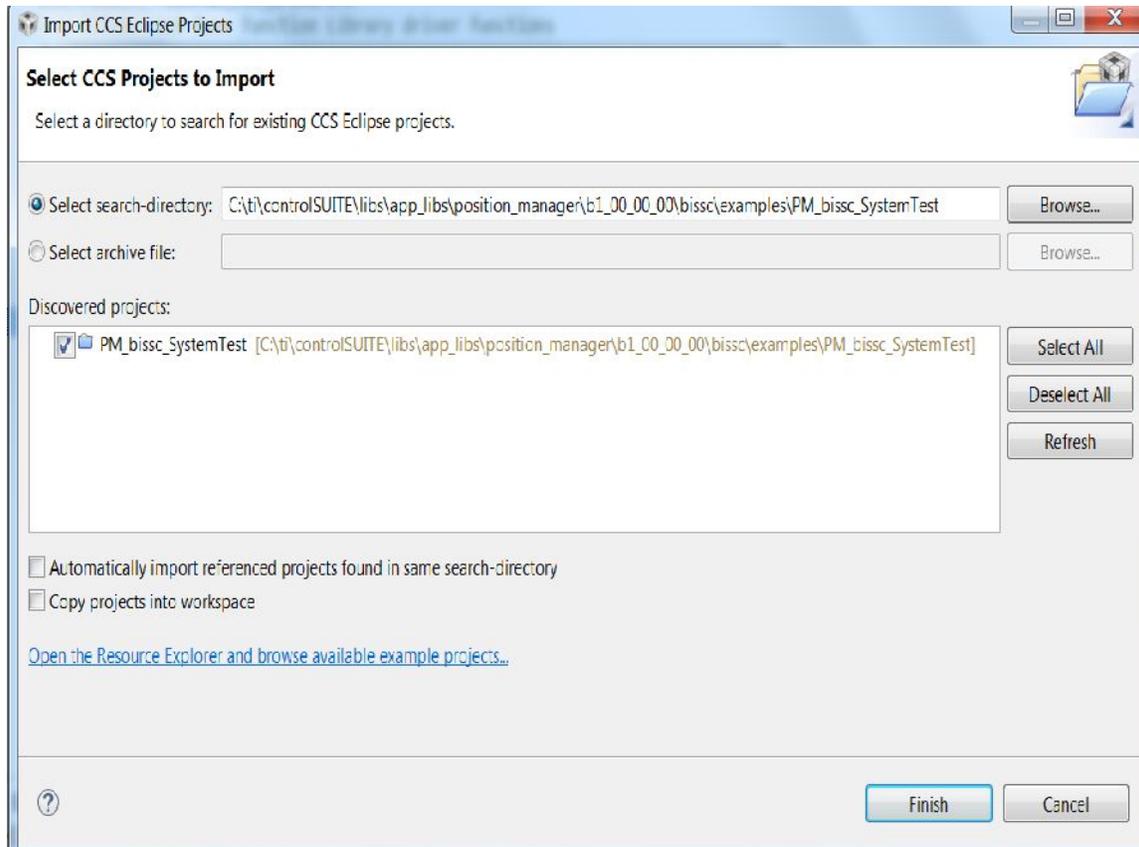


Figure 10. Adding PM_bissc Example Project to Workspace

6.4 Configuring a Project

1. Assuming this is the first time using Code Composer, the `xds100v2-F2837x` should have been set as the default target configuration. Verify this by viewing the `xds100v2-f2837x.ccxml` file in the expanded project structure and a [Active/Default] status written next to it. By going to View → Target Configurations, the user can edit existing target configurations or change the default or active configuration. The user can also link a target configuration to a project in the workspace by right-clicking on the Target Configuration name and selecting Link to Project.
2. The project can be configured to create code and run in either flash or RAM. Select either of the two; however, for lab experiments use RAM configuration most of the time and move to the FLASH configuration for production. As shown in Figure 11, right-click on an individual project and select Active Build Configuration → CPU1_RAM configuration.

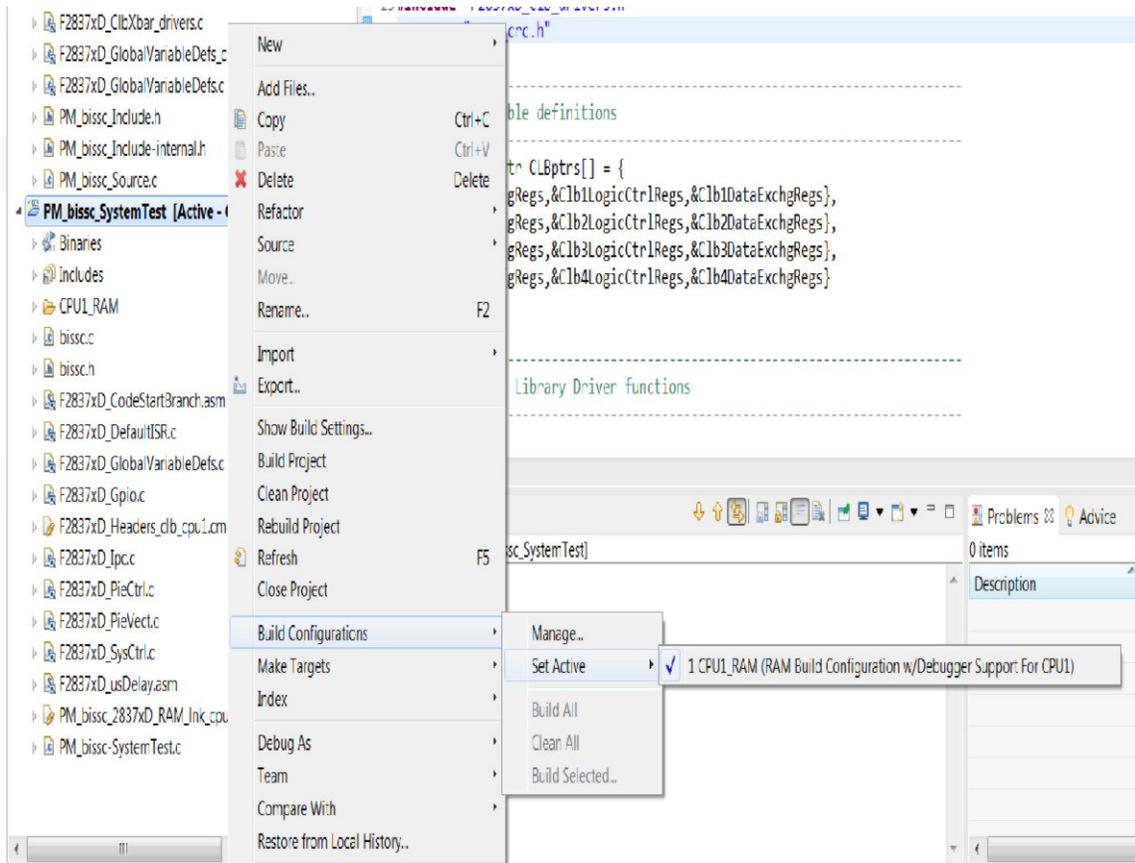


Figure 11. Selecting the F2837x_RAM Configuration

6.5 Build and Load the Project

1. Open the *bissc.h* file, ensure `BISSC_FREQ_DIVIDER` is set as needed, and save this file. For more details, see the *C2000 Position Manager BiSS-C Library User's Guide* ([SPRU137](#)) (`C:\ti\controlSUITE\libs\app_libs\position_manager\v1_00_00_00\BiSS-C\Doc\PM_bissc_Lib.pdf`).
2. Right-click on the project name, click on Rebuild Project and watch the console window. Any errors in the project are displayed in the Console window.
3. On a successful completion of the build, click the  Debug button, located in the top-left side of the screen. If it is the first time using CCS, the following window may pop up requesting the user to connect to one of the two CPUs in F28379x. Click on the box next to CPU1.

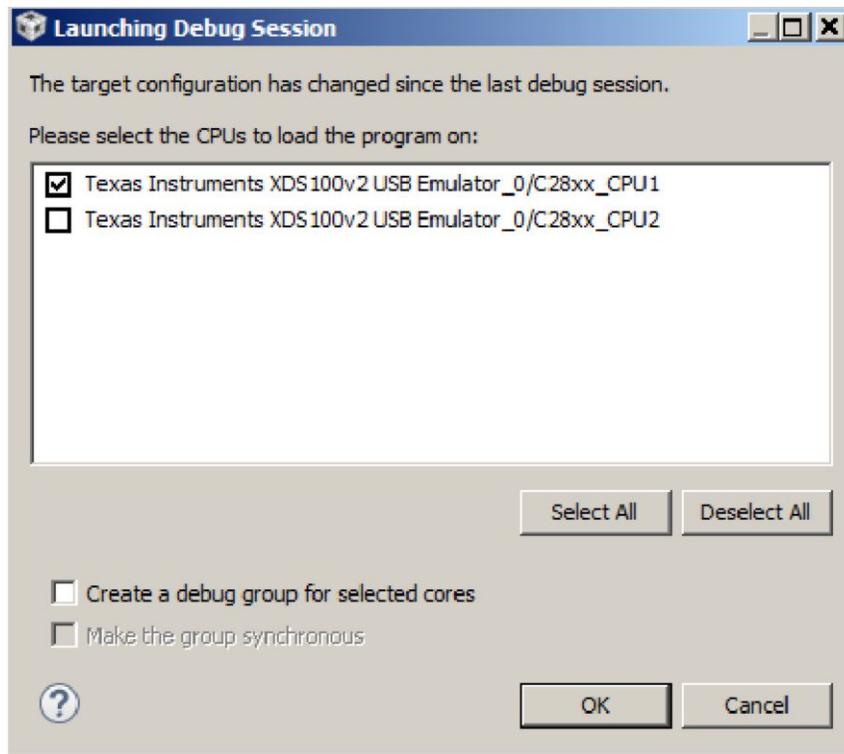


Figure 12. Selecting the CPUs to Connect

4. Now, the IDE automatically connects to the target, loads the output file into the device, and changes to the Debug perspective.
5. Click Tools → Debugger Options → Program/Memory Load Options. The user can enable the debugger to reset the processor each time it reloads the program by checking Reset the target on program load or restart and clicking Remember My Settings to make this setting permanent.
6. Click on the Enable silicon real-time mode button , which also auto-selects the Enable polite real-time mode button . This allows the user to edit and view variables in real time. Do not reset the CPU without disabling these real-time options.
7. A message box may appear. If so, select YES to enable debug events. This sets bit 1 (DGBM bit) of Status Register 1 (ST1) to a 0. DGBM is the debug enable mask bit. When the DGBM bit is set to 0, memory and register values can be passed to the host processor for updating the debugger windows.

6.6 Setup Watch Window and Graphs

1. Click View → Expressions on the menu bar to open a watch window to view the variables used in the project. Add variables to the watch window, as shown in Figure 13. It uses the number format associated with variables during declaration. Select the desired number format for the variable by right-clicking and choosing it. Figure 13 shows a typical expressions window.

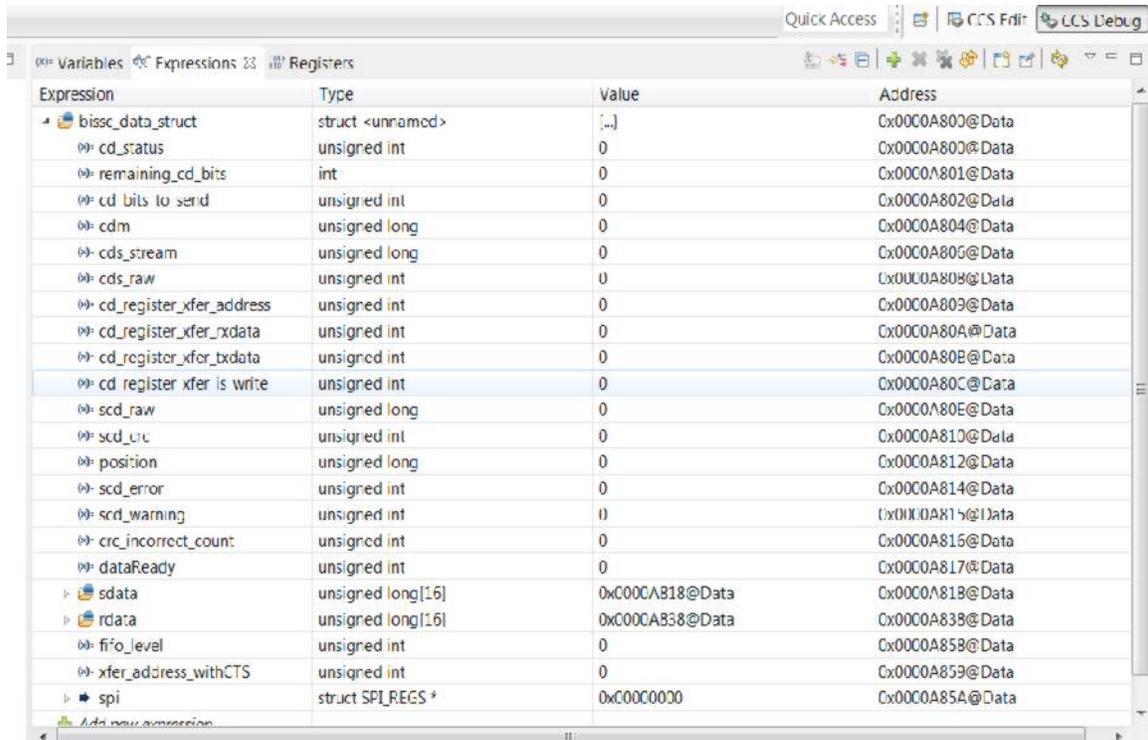


Figure 13. Configuring the Expressions Window

Alternately, a group of variables can be imported into the Expressions window; right click in the Expressions Window and select Import. Then, browse to the .txt file containing these variables. Browse to the root directory of the project and pick *PM_bissc_SystemTest_VAR.txt*, then click OK to import the variables.

2. Click on the Continuous Refresh button  in the watch window. This enables the window to run with real-time mode. By clicking the down arrow in this watch window, the user can select Customize Continuous Refresh Interval and edit the refresh rate of the watch window. Note that choosing an interval that is too fast can affect performance.

6.7 Run the Code

1. Run the code by pressing the Run Button  in the Debug tab.
2. The project should now run and the values in the watch window should continuously update.
3. Once complete, reset the processor  (Run → Reset → CPU Reset), then terminate the debug session by clicking  (Run → Terminate). This halts the program and disconnects Code Composer from the MCU.
4. As the encoder sends the position information, observe the same in the watch window as the `biss_data_struct.position` variable. The variable `biss_data_struct.position` reflects the latest position received by the encoder.
5. If the encoder is not mounted on a spinning motor, the user can manually rotate the encoder shaft and observe the position changing in the watch window.
6. It is not necessary to terminate the debug session each time the user changes or runs the code. Instead, use the following procedure. After rebuilding the project, (Run → Reset → CPU Reset) , (Run → Restart) , and enable real-time options. Once complete, disable real-time options and reset the CPU. Terminate the project if the target device or the configuration is changed (RAM to Flash or Flash to RAM) and prior to shutting down CCS.
7. Customize the project to meet your encoder and application requirements. Change the encoder type in `bissc.h`. Users can also change the BiSS-C Clock frequency as instructed in the *C2000 Position Manager BiSS-C Library User's Guide* ([SPRUI37](#)).
8. Modify the example code as needed, and perform tests with different cable lengths.

7 References

- *DesignDRIVE Development Kit IDDK Hardware Reference Guide* ([SPRUI23](#))
- *DesignDRIVE Development Kit IDDK - User's Guide* ([SPRUI24](#))
- *C2000 Position Manager BiSS-C Library User's Guide* ([SPRUI37](#))

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com