Using Position Manager SinCos Library on IDDK

User's Guide

Literature Number: SPRUI57
January 2016
# Contents

1 Introduction to IDDK ................................................................. 4  
2 Hardware Configuration ............................................................. 4  
3 Hardware Overview ................................................................. 5  
   3.1 Functional Blocks ................................................................. 5  
   3.2 Processor Section (Control Processor Slot – H1) ...................... 6  
   3.3 Position Encoder Suite .......................................................... 7  
   3.4 Power Supply ..................................................................... 8  
4 Hardware Resource Mapping ..................................................... 8  
   4.1 Signal Mapping ................................................................. 8  
   4.2 Jumpers and Switches .......................................................... 9  
   4.3 Headers/Connectors ............................................................. 9  
5 Hardware Setup Instructions for PM_sincos Library Evaluation ........... 9  
6 Software Setup for PM SinCos Evaluation Example Project ............... 10  
   6.1 Installing Code Composer and controlSUITE ......................... 10  
   6.2 Software Flow Diagram ........................................................ 11  
7 Setup Code Composer Studio to Work With TMDXIDDK379D .............. 11  
   7.1 Configure the Project .......................................................... 13  
   7.2 Build and Load the Project .................................................... 13  
   7.3 Setup Watch Window ............................................................ 15  
   7.4 Run the Code .................................................................. 15  
   7.5 Closing the Project ............................................................... 16  
8 References .............................................................................. 16
List of Figures

1. IDDK EVM Kit ................................................................. 5
2. Layout of IDDK EVM With Its Functional Macros ......................... 6
3. Processor Block .................................................................. 6
4. Position Sensor Suite ......................................................... 7
5. SinCos Interface Header (Top View) ......................................... 7
6. Power Supply Block ............................................................ 8
7. Combined BiSS/SinCos Transducer Connected to the IDDK ........ 10
8. Software Flow Diagram for the Example Project PM_sincos_example ........................................ 11
9. Configuring a New Target ..................................................... 12
10. Adding the SinCos Example Project to the CCS Workspace .......... 13
11. Selecting the CPU(s) to Connect .......................................... 14
12. Configuring the Expressions Window ..................................... 15

List of Tables

1. Hardware Macros in IDDK Used for SinCos Evaluation ....................... 5
2. Digital Signal Mapping on Control Card H1 .................................. 8
3. Purpose of Jumpers and Switches ............................................ 9
4. Headers and Connectors ....................................................... 9
Using Position Manager SinCos Library on IDDK

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, enabling 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 co-processors, 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 8 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 user’s guide covers the kit contents and hardware details, and explains the functions and locations of various connectors present on the board. For more details, see the DesignDRIVE Development Kit IDDK v2.2 Hardware Reference Guide (SPRUI23).

2 Hardware Configuration

To evaluate and experiment with the Position Manager SinCos library, the following components are needed:
- IDDK EVM
- TMDXCNCD28379D
- Encoder and connectors (not included in the kit)
  - SinCos Encoder
  - Cable
- An external isolated 15 V power supply needed for MCU code development (preferably with a barrel connector commonly referred to as a “DC jack”).
- PC with Code Composer Studio™ (version 6 or greater) installed

The IDDK Hardware Reference Guide is available in controlSUITE at: controlSUITE\development_kits\TMDSIDDK_v2.0\~Docs\.
3 Hardware Overview

This section describes the components required for evaluation of the Position Manager SinCos Library. A complete hardware overview of the kit can be obtained from DesignDRIVE Development Kit IDDK - Hardware Reference Guide, available in controlSUITE.

Evaluation of the Position Manager SinCos Library requires usage of:

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

3.1 Functional Blocks

Table 1 illustrates the subset of functional blocks on DesignDRIVE Development Kit IDDK Hardware along with the macro names used for Position Manager SinCos library evaluation.

<table>
<thead>
<tr>
<th>Functional Block</th>
<th>Macro Reference</th>
<th>Macro Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supplies</td>
<td>M9</td>
<td>DC Power Supply – Linear Reg 15 V-5 V/3.3 V</td>
</tr>
<tr>
<td>Position Encoder Suite</td>
<td>M13</td>
<td>SinCos Encoder Interface</td>
</tr>
<tr>
<td>Processor/controlCARD</td>
<td>H1</td>
<td>All other functions</td>
</tr>
</tbody>
</table>
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 in controlSUITE 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)

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 TMDXCNCD28379D designed with a HSEC180pin edge connector.

Figure 3. Processor Block
3.3 Position Encoder Suite

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

- QEP
- Resolver
- SinCos
- EnDat / BiSS

Figure 4. Position Sensor Suite

3.3.1 SinCos Encoder

This macro, designated M13, is a common interface for SinCos encoders. H4 is the external interface header with pin connections shown in Figure 5.

![Figure 5. SinCos Interface Header (Top View)](image)
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 and can receive this through M9 or M3. When the controller processor is to be on the cold-side, M9 should be used and this is what is 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:
- [Main]-J6, J7, and J8 should be jumpered
- [Main]-J2 should NOT be jumpered
- [Main]-R9, R11, R13 should be populated
- [Main]-R8, R10, R12 should NOT be populated
- 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 IDDK documentation (available in controlSUITE).

![Figure 6. Power Supply Block](image)

4 **Hardware Resource Mapping**

Section 4.1 shows the hardware resource requirements and signal mapping for PM SinCos library evaluation on the IDDK.

## 4.1 Signal Mapping

Table 2 shows the functional mapping of signals connected to the control processor on H1 that are relevant to PM SinCos library evaluation.

<table>
<thead>
<tr>
<th>IDDK Signal Name</th>
<th>MCU GPIO</th>
<th>MCU Peripheral Associated With GPIO</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC-A-2</td>
<td>-</td>
<td>ADC-C/CMPSS-4</td>
<td>Sine input</td>
</tr>
<tr>
<td>SC-B-2</td>
<td>-</td>
<td>ADC-D/CMPSS-7</td>
<td>Cosine input</td>
</tr>
<tr>
<td>SC-R</td>
<td>-</td>
<td>ADC-D/CMPSS-8</td>
<td>Index input</td>
</tr>
<tr>
<td>CompOutSC-A</td>
<td>15-54</td>
<td>CMPSS-4/QEP-2</td>
<td>CMPSS - QEP channel A connection</td>
</tr>
<tr>
<td>CompOutSC-B</td>
<td>14-55</td>
<td>CMPSS-7/QEP-2</td>
<td>CMPSS - QEP channel B connection</td>
</tr>
<tr>
<td>CompOutSC-R</td>
<td>59-57</td>
<td>CMPSS-8/QEP-2</td>
<td>CMPSS - QEP channel I connection</td>
</tr>
</tbody>
</table>
4.2 **Jumpers and Switches**

The jumpers shown in Table 3 need to be populated for PM SinCos library evaluation.

**Table 3. Purpose of Jumpers and Switches**

<table>
<thead>
<tr>
<th>Jumpers/Switches</th>
<th>Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Main] - J6-J8</td>
<td>Populate</td>
<td>Jumper to bring out linear regulator block M9 voltages</td>
</tr>
<tr>
<td>[Main] - J12</td>
<td>Do Not Populate</td>
<td></td>
</tr>
<tr>
<td>[Main] - J15-16</td>
<td>Populate</td>
<td>Connect SinCos input signals to MCU</td>
</tr>
<tr>
<td>[Main] - J18</td>
<td>Populate (1-2)</td>
<td>Connects Encoder data out (En-Do-1 to the processor)</td>
</tr>
<tr>
<td>[Main] - J19-J21</td>
<td>GND headers for probe access</td>
<td></td>
</tr>
<tr>
<td>[M9] – SW1</td>
<td>Turn ON</td>
<td>Select 15 V supply from JP1 or onboard from M8 to generate 5 V and 3.3 V</td>
</tr>
</tbody>
</table>

4.3 **Headers/Connectors**

Table 4 shows headers and connectors that are relevant to SinCos configuration.

**Table 4. Headers and Connectors**

<table>
<thead>
<tr>
<th>Headers/Connectors</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[M9] – JP1</td>
<td>15 V DC power supply jack adapter</td>
</tr>
<tr>
<td>[Main] - H1</td>
<td>180 pin HSEC connector slot for control processor card</td>
</tr>
<tr>
<td>[Main] – H4</td>
<td>4x2 header for SinCos</td>
</tr>
</tbody>
</table>

5 **Hardware Setup Instructions for PM_sincos Library Evaluation**

1. Ensure default configuration – Make sure that jumpers [Main]-J6, [Main]-J7 and [Main]-J8 are in front of macro M9, that resistors [Main]-R9, R11, R13 are populated and that GND plane resistors R14 and R15 are mounted as shown in the Various GND Planes on the Bottom of Board and Default Connection of Various GND Planes figures in the Power Supplies section of the DesignDRIVE Development Kit IDDK v2.2 Hardware Reference Guide (SPRUI23).

2. Unpack the TMDXCNCD28379D control card and slide it down in the connector slot of [Main]-H1. Push down vertically using even pressure on both ends of the card 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 an encoder to connector H4 in accordance with the wiring description in the manufacturer’s data sheet.

For more information, see the encoder/cable specifications and Hardware schematics, see the Control Processor Slot - H1 section of the DesignDRIVE Development Kit IDDK v2.2 Hardware Reference Guide (SPRUI23).

Insert the transducer cable into [Main] – H4 4x2 header of IDDK. Figure 7 shows a combined BiSS/SinCos transducer connected to the IDDK.
5. Ensure that toggle switch [M9]-SW1 is in “Int” position. Connect an isolated 15 V DC power supply to [M9]-JP1.

6. Turn on toggle switch [M9]-SW1. Then, [M9]-LD1 should turn on. Note that 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 SinCos Evaluation Example Project

#### 6.1 Installing Code Composer and controlSUITE

1. Install Code Composer v6.x or later (if not already installed) from [http://www.ti.com/tool/CCSTUDIO](http://www.ti.com/tool/CCSTUDIO).


3. The SinCos Library is available at the following location:

   \[
   \text{<base> install directory is:}
   \]

   \[
   C:\ti\controlSUITE\libs\app_libs\position_manager\sincos\v01_00_00_00\sincos
   \]

   The following sub-directory structure is used:

   \[
   \text{<base>\Doc}
   \text{\quad Documentation}
   \]

   \[
   \text{<base>\Float}
   \text{\quad Implementation of the library and corresponding include file}
   \]

   \[
   \text{<base>\examples}
   \text{\quad Example using SinCos library}
   \]
6.2 Software Flow Diagram

An outline software flow of the example project PM_sincos_example is shown in Figure 8.

Figure 8. Software Flow Diagram for the Example Project PM_sincos_example

7 Setup Code Composer Studio to Work With TMDXIDDK379D

1. Open “Code Composer Studio” (note that this document assumes version 6 or later).
2. Once Code Composer Studio opens, the workspace launcher may appear prompting the selection of a workspace location; the workspace is a location on the hard drive where all the user settings for the ID, which projects are open, what configuration is selected, and so forth are saved. This can be anywhere on the disk, the location mentioned below is just for reference. Also, note that if this is not your first-time running Code Composer Studio, the dialog below may not appear:
   (a) Click the “Browse...” button
   (b) Create the path below by making new folders as necessary:
      C:\c2000 projects\CCSv6_workspaces\PM_sincos_eval_workspace
   (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, such as creating a new project, importing an existing project, to watching a Tutorial on CCS. Click the ‘Import Project’ icon that skips the procedure to step 1 in Section 7.1, or close the ‘Getting Started’ Tab and go to the next step.
4. Configure Code Composer to know which MCU it will connect to. This is done by setting up the ‘Target Configuration’. All these are already set up and configured in “xds100v2_F2837x.cxml” provided as part of the files in project, and you can skip to step . However, for general information regarding setting up this configuration file, steps 6, 7 and 8 can be used.
5. A new configuration file can be set by clicking “View → Target Configuration. This opens the Target Configuration window, click on . Give a name to the new configuration file depending on the target device. If “Use shared location” checkbox is checked, then this configuration file can be stored in a common location by CCS for use by other projects as well. Then click Finish.
6. This should open up a new tab as shown in Figure 9. Select and enter the options as shown:
   (a) Connection – Texas Instruments XDS100v2 USB Emulator or Texas Instruments XDS100v2 USB Debug Probe
   (b) Device – the C2000 MCU on the control card, TMS320F28379D, for example
   (c) Click Save and close

![Figure 9. Configuring a New Target](image)

7. Click “View → Target Configurations”. In the “User Defined” section, find the file that was created in step 6 and 7. 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_sincos_example and right-click on the file “xds100v2_F2837x.cxml” and “Set as Default”. This tab also allows you to reuse existing target configurations and link them to specific projects.

8. Add the PM SinCos evaluation example project into the current workspace by clicking “Project → Import CCS Project”.
   (a) Select the project by browsing to:

```
C:ti\controlSUITE\libs\app_libs\position_manager\v01_00_00_00\sincos\examples\PM_sincos_example
```
7. Configure the Project
1. Assuming this is your first time using Code Composer Studio, 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” you can edit existing target configurations or change the default or active configuration. You 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”.

7.2 Build and Load the Project
1. Open the file PM_sincos_lib.h and ensure the definition “LINECOUNT” matches the number of electrical cycles per mechanical rotation for the chosen transducer. Save the file.
2. Right Click on the Project Name and click on “Rebuild Project” and watch the Console window. Any errors in the project will be displayed in the Console window.
3. On successful completion of the build, click the 🛠 “Debug” button located in the top-left side of the screen. The window shown in Figure 11 may pop up if being used for the first time or requesting the user to select which of the two CPUs to connect to. Choose CPU1 by clicking the box next to CPU1.
4. The IDE will now automatically connect to the target, load the output file into the device and change to the Debug perspective.

5. Click “Tools → Debugger Options → Program/Memory Load Options”. You can enable the debugger to reset the processor each time it reloads program by checking “Reset the target on program load or restart” and click “Remember My Settings” to make this setting permanent.

6. Click on the “Enable silicon real-time mode” button that auto selects the “Enable polite real-time mode” button. This allows you 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 will set bit 1 (DGBM bit) of status register 1 (ST1) to a “0”. The 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.
7.3 Setup Watch Window

1. Click: View → Expressions on the menu bar to open a watch window to view the variables being used in the project. Add the "mySincos" structure to the watch window and expand it as shown in Figure 12.

![Figure 12. Configuring the Expressions Window](image)

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, you can select “Customize Continuous Refresh Interval” and edit the refresh rate of the watch window. Note that choosing too fast an interval may affect performance.

7.4 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. If encoder is not mounted on a spinning Motor, users can manually rotate the encoder shaft and observe the position changing in the watch window.
3. Once complete, reset the processor (Run → Reset → CPU Reset) and then terminate the debug session by clicking (Run → Terminate). This will halt the program and disconnect Code Composer from the MCU.
7.5 Closing the Project

It is not necessary to terminate the debug session each time you change or run the code again. Instead, the following procedure can be followed.

1. After rebuilding the project, (Run → Reset → CPU Reset) , (Run → Restart ) , enable the real-time options.
2. Once complete, disable real-time options and reset the CPU.
3. Terminate the project if the target device or the configuration is changed, and prior to shutting down CCS.

8 References

DesignDRIVE Development Kit IDDK v2.2 Hardware Reference Guide (SPRUI23)
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.

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.

<table>
<thead>
<tr>
<th>Products</th>
<th>Applications</th>
<th><a href="http://www.ti.com">www.ti.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Automotive and Transportation</td>
<td><a href="www.ti.com/audio">audio</a></td>
</tr>
<tr>
<td>Amplifiers</td>
<td>Communications and Telecom</td>
<td><a href="www.amplifier.ti.com">amplifier.ti.com</a></td>
</tr>
<tr>
<td>Data Converters</td>
<td>Computers and Peripherals</td>
<td><a href="www.dataconverter.ti.com">dataconverter.ti.com</a></td>
</tr>
<tr>
<td>DLP® Products</td>
<td>Consumer Electronics</td>
<td><a href="www.dlp.com">www.dlp.com</a></td>
</tr>
<tr>
<td>DSP</td>
<td>Energy and Lighting</td>
<td><a href="dsp.ti.com">dsp.ti.com</a></td>
</tr>
<tr>
<td>Clocks and Timers</td>
<td>Industrial</td>
<td><a href="www.ti.com/clocks">www.ti.com/clocks</a></td>
</tr>
<tr>
<td>Interface</td>
<td>Medical</td>
<td><a href="interface.ti.com">interface.ti.com</a></td>
</tr>
<tr>
<td>Logic</td>
<td>Security</td>
<td><a href="logic.ti.com">logic.ti.com</a></td>
</tr>
<tr>
<td>Power Mgmt</td>
<td>Space, Avionics and Defense</td>
<td><a href="power.ti.com">power.ti.com</a></td>
</tr>
<tr>
<td>Microcontrollers</td>
<td>Video and Imaging</td>
<td><a href="www.microcontroller.ti.com">microcontroller.ti.com</a></td>
</tr>
<tr>
<td>RFID</td>
<td></td>
<td><a href="www.ti-rfid.com">www.ti-rfid.com</a></td>
</tr>
<tr>
<td>OMAP Applications Processors</td>
<td>TI E2E Community</td>
<td><a href="www.ti.com/omap">www.ti.com/omap</a></td>
</tr>
<tr>
<td>Wireless Connectivity</td>
<td></td>
<td><a href="www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a></td>
</tr>
</tbody>
</table>

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2016, Texas Instruments Incorporated