

# Preface

Texas Instruments Robotics System Learning Kit  
The Maze Edition

# Preface

The ultimate goal of the learning kit is to design, build, and test a robot system capable of solving complex tasks. One possible robot is shown in Figure 1. Example challenges include exploring a maze, racing autonomously, finding an object, and following a line. However, it is not the final robot that matters, but the educational journey that discovers a wide range of engineering principles along the way. Rather than just providing the robot kit and a challenge to solve, this curriculum follows an educational road map that intentionally exposes deep learning along the way.

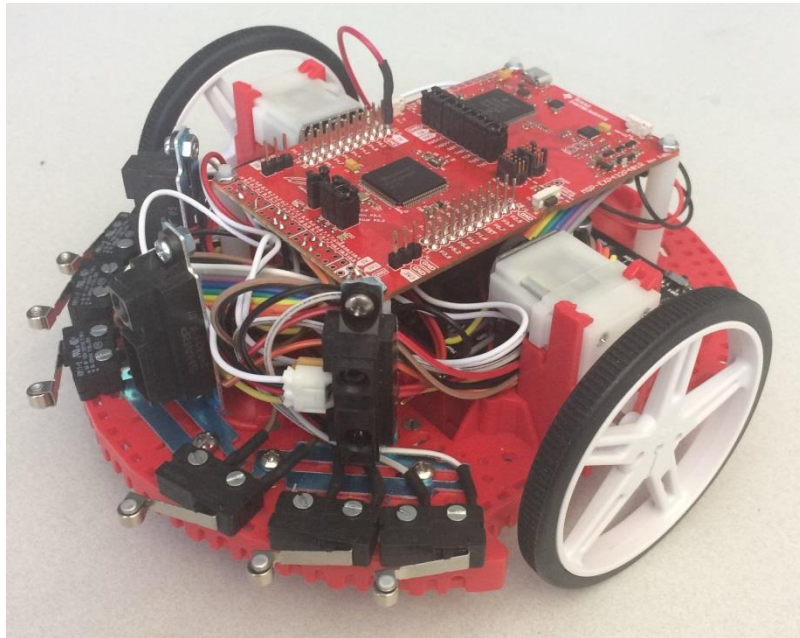


Figure 1: TI-RSLK Maze Robot

The **EE skills** you will learn include voltage, current, power, energy, batteries, resistors, capacitors, transistors, diodes, and DC motors. You will learn how to use a voltmeter, an ohmmeter, a current meter, and an oscilloscope.

This is an **embedded systems** curriculum; hence you will interface numerous devices to the MSP432 microcontroller. In particular, you will interface LEDs, switches, a line sensor, DC motors, tachometers, and an LCD. Your microcontroller hardware/software skills will include pulse-width modulation (PWM), flash read-only memory, periodic timers, edge-triggered interrupts, digital to analog converter (DAC), analog to digital converter (ADC), synchronous serial communication, and asynchronous serial communication.

A significant component of the curriculum involves software development. You will develop **software skills** in multithreading, data structures, debugging, linked lists, semaphores, and first in first out (FIFO) queues. You will learn how to use a logic analyzer for interface testing, and thread profiling.

This is a lab-based curriculum. However, there are numerous **fundamental concepts** to master, such as the Nyquist Theorem, the Central Limit Theorem, digital filtering, and Little's Theorem.

The overriding theme of this curriculum is to teach **systems design** in a bottom up fashion. We begin with simple components so that you learn fundamentals. A system is created by combining multiple components with the appropriate hardware and software interfaces. Once you master the fundamentals of one component, its operation can be abstracted into a set of high-level functions. Separating how a component works (low-level implementation) from what it does (high-level abstraction) is the key for developing complex systems. Obviously, the most important system in this curriculum will be the robot. However, there will be other systems like a security system, a traffic light control system using a finite state machine, Bluetooth Low Energy (BLE) communication system, and the Wi-Fi-based internet of things (IoT) system.

A system is comprised of subsystems connected together to solve a unified objective. An effective approach to teaching systems is to begin with very simple components. First, one completely understands how the component works. Second, one creates an abstraction that separates what it does from how it works. Third, components are interfaced together to create a new more complex system.

# Preface

The terms system, subsystem, and component are used here interchangeably.

As you can see from Figure 2, there are twenty modules in the curriculum. Each module is relatively independent, and you can thread together modules to create a particular learning experience for your students.

## Each module has:

- Introduction to module (1 page)
  - Overview
  - Educational objectives
  - Prerequisites, bullet list linking to other modules
  - References
- Class lecture PowerPoint slides (one to three files)
- Screen capture video with audio of PowerPoint (one to three videos)
- Class activity, homework exercises or practice problems.
- Lab document
- Hardware needed,
  - BOM excel file of parts
  - Circuit diagrams in CircuitMaker.
- Lab solution for faculty to access, not available to students.
- One to three videos of finished lab
- Quizzes
- Quiz solutions for faculty to access, not available to students.

The most important document is the lab manual. Performing labs results in the design, construction, and testing of the robot system. To find the circuit diagrams, create an account in Circuit maker. Launch the application, under projects select Tags, and search MSP432. You will find starter circuits for each lab that has hardware.

The robot challenge document lists some possible final projects for the course. Most users of this curriculum will pick and choose a subset of the modules, allowing the user to focus on which concepts they wish to learn

(or teach). Challenges are sorted by the set of sensors and actuators that are required.

## Robot Features (Full set, Advanced Kit) :

- Robot Chassis with 2 DC motors and wheels
- 6 AA NiMH batteries
- Motor driver and power distribution board ( MDPD) with motor drivers and voltage regulator to power your system
- 3 IR distance sensors
- 6 touch/bump sensors
- 8 line sensors
- 2 tachometers
- Tachometer
- BLE or Wi-Fi

## Course prerequisites:

- Algebra and college physics
- Basic knowledge of computers and architecture
- C programming

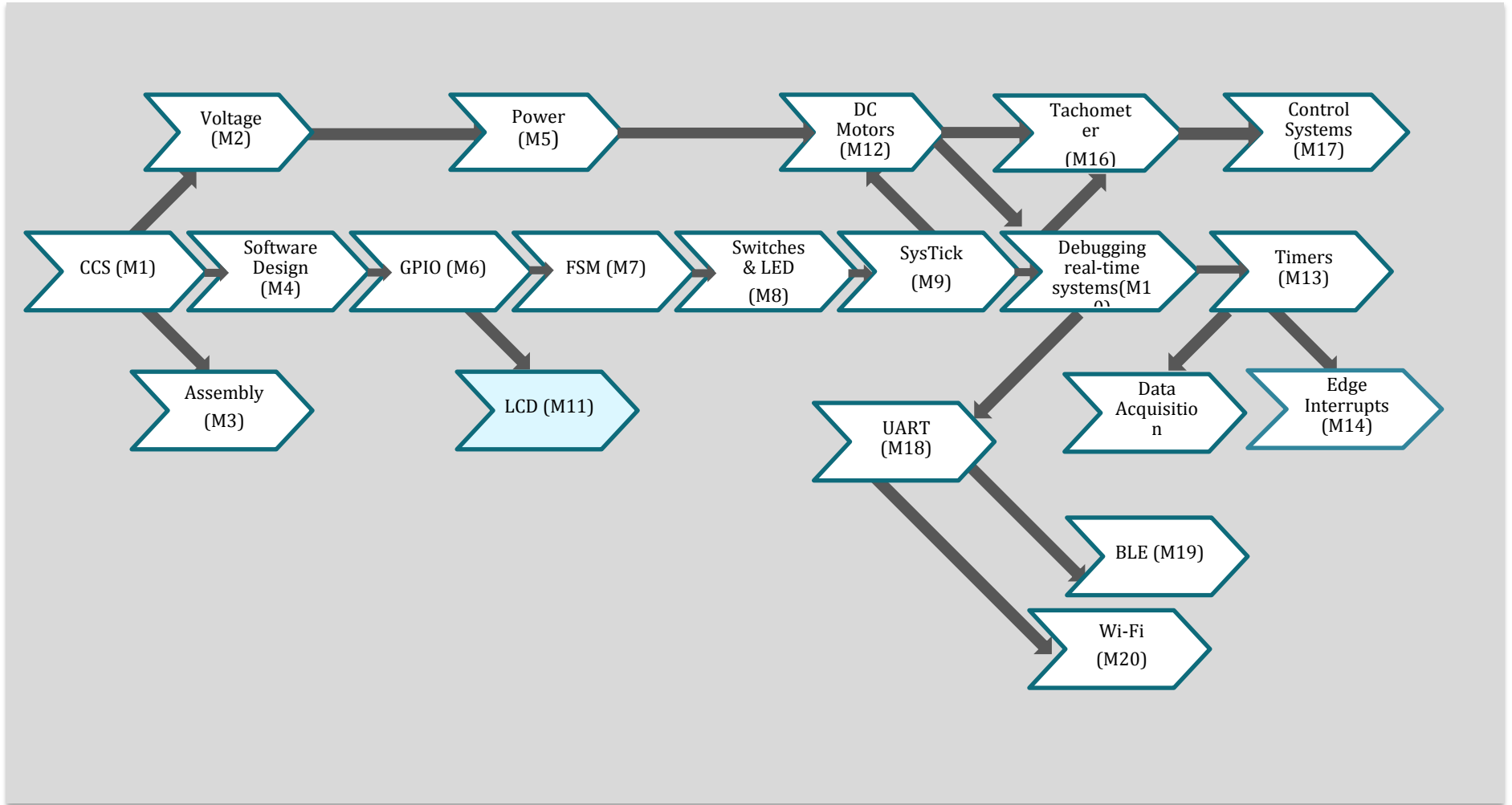


Figure 2. TI-RSLK: Learning Modules & Curriculum Pathways

**Modules:****<CCS> 1. Running Code on the LaunchPad using CCS**

Prerequisites: none

Equipment: LaunchPad

Theory: how to install and configure CCS for this class

Lab: installing CCS, MSP432 drivers, and running the

TExaSdisplay logic analyzer

**<Voltage\_Current> 2. Voltage, Current and Power**

Prerequisites: none

Motivation: The hardware interfaces for the robot involve voltage, current and power

Tools: Voltmeter, current meter, ohmmeter, signal generator, oscilloscope

Equipment: 220 470 22k 33k ohm resistors, 0.47uF 10uF capacitors, voltage supply, 10-mA LED and 2-mA LED

Theory: Resistor, Ohm's Law, LED voltage current relationship

Lab: Characterize LED, Measure Reactance

**<Assembly> 3. ARM Cortex M**

Prerequisites: &lt;CCS&gt;

Motivation: teaching assembly will help understand how it works, and how to debug

Tools: CCS

Equipment: LaunchPad (20-instruction subset of Cortex M)

Theory: machine code, registers, execution, bus, memory, simple I/O

Lab: assembly language programming

Installing and running assembly language

**<SoftwareDesign> 4. Software Design using MSP432**

Prerequisites: &lt;CCS&gt;

Motivation: most of the labs are C programmed on the MSP432, C programming is a prerequisite to the class, but an introduction to C code on the MSP432 is appropriate

Tools: CCS

Equipment: LaunchPad using built-in switches and LEDs

Theory: Using typical input parameters for the robot, perform, logical operations of AND, OR, EOR, shift, add, subtract, multiply, divide, variables, and functions

Lab: Simple C programming converting ADC-inputs to calibrated distance. Given three distance measurements, implement a classification algorithm to interpret the robot world

**<Power> 5. Battery and Voltage Regulation**

Prerequisites: &lt;Voltage\_Current&gt;

Motivation: Robot is battery powered; robot motor controller board has power regulation

Tools: Voltmeter, current meter, ohmmeter

Equipment: Two power resistors 5W 10-ohm resistor and 22-ohm 5W resistor, rechargeable battery (4.8V/10ohm) is 0.48A, 2A-hr battery lasts 4hr Robot with power regulation/motor driver board

Theory: Resistor, Ohm's Law

Theory: total energy in battery voltage current relationship while V&gt;regulator minimum

Lab: battery power, calculations, measurements

**<GPIO> 6. GPIO – MSP432**

Prerequisites: &lt;Logic&gt;

Motivation: robot line sensor is needed for line following

Tools: CCS Voltmeter, oscilloscope,

Equipment: LaunchPad, integrated line sensor

Theory: conversion light to voltage, direction registers, input, output, friendly (This connects to the maze robot).

Lab: input from line sensor, output to built-in LED

Input line sensor, detect position relative to a black line on a white field

**<FSM> 7. Finite State Machines**

Prerequisites: &lt;Logic&gt;&lt;GPIO&gt;

Motivation: FSMs are an effective solution to robotic functions

Tools: CCS

Equipment: LaunchPad using built-in switches and LEDs

Theory: loops, decisions

Lab: Very simple C programming, 2-input, 2 output FSM

**<Switches\_LED> 8. Interfacing Input and Output**

Prerequisites: <GPIO>

Motivation: robot touch sensors will be switches, LEDs provide debugging outputs for the robot

Tools: Voltmeter, current meter, ohmmeter, CCS

Equipment: switches, LEDs, resistors, LaunchPad

Theory: GPIO, LED, positive/negative logic, pullup/pulldown, input/output

Lab: Input from switches, output to LED

**<SysTick>9. SysTick Timer**

Prerequisites: <Switches\_LED>

Motivation: introduction to time, introduce the concept of PWM that will be needed to drive the DC motor later; this is also introduces the need for interrupts, because this one task will require 100% processor utilization

Tools: CCS, logic analyzer,

Equipment: LaunchPad, LED

Theory: Introduce the need for the microcontroller to manage time. Define processor utilization. Use SysTick to create time delays. Use time delays to create a PWM signal. Use PWM to control power delivered to an actuator (LED).

Lab: A GPIO port is connected to an LED and the software controls brightness of the LED using duty cycle. The software varies the duty cycle sinusoidally (table look up) to make the LED appear to be “breathing”. Add a resistor and capacitor, and then observe the sinusoidal output on the oscilloscope.

**<Debug>10. Debugging Real-time Systems**

Prerequisites: <SysTick>

Motivation: system level design/debug; students need effective debugging skills. This will mimic the process students will use to design/develop/debug their robot. Eventually, students will place the robot in the maze and hit go. The robot runs for a while autonomously. After the run,

students can reconnect the USB cable and upload parameters measured during the run. This module will introduce interrupts and use SysTick to perform the line-sensor measurements in the background.

Tools: CCS

Equipment: LaunchPad, line-sensor, bump sensors

Theory: RAM versus ROM, arrays, pointers, periodic interrupts

Lab: record bump and line sensor data into arrays in Flash

**<LCD> 11. Liquid Crystal Display (optional)**

Prerequisites: <GPIO>

Motivation: optional for robot, but makes the course accessible for other non-robotic applications; if used with the robot, the LCD can help with debugging during the stand-alone running.

Tools: Logic analyzer, CCS

Equipment: Nokia 5110, proto-board, LaunchPad

Theory: SPI interface, graphics

Lab: Display text and graphics

**<Motors> 12. DC motors**

Prerequisites: <Power><SysTick>

Tools: Voltmeter, current meter, oscilloscope, CCS

Equipment: LaunchPad, Motor driver board (MOSFET, resistors, diodes), DC motor

Theory: Brushed DC motor, PWM

Lab: Open loop DC motor output, measure speed versus duty cycle, extends the simple PWM built in <SysTick>

**<Timers> 13. PWM and Periodic interrupts using Timers**

Prerequisites: <Motors><Debug>

Motivation: periodic interrupts are a simple way to create PWM outputs to DC motors

Tools: Voltmeter, current-meter, oscilloscope, logic analyzer, CCS

Equipment: LaunchPad, DC motor on robot

Theory: timers, interrupts, frequency, PWM

Lab: software adjusts power to one DC motor

**<EdgeInterrupts> 14. Real-Time Systems**

Prerequisites: <Timers>

Motivation: edge-triggered interrupts is a good way to service bumper switches on robot

Tools: Oscilloscope, logic analyzer, CCS

Tools: CCS, logic analyzer,

Equipment: bumper switches on robot, LaunchPad

Theory: interrupt driven I/O, Input triggered interrupts

Lab: Input from four switches on robot to detect collision

**<ADC> 15. Data Acquisition Systems**

Prerequisites: <Timers>

Motivation: the robot uses IR distance sensors to detect walls or other robots

Tools: Oscilloscope, spectrum analyzer, logic analyzer, CCS

Equipment: sensor (IR distance sensor), LaunchPad

Theory: ADC conversion, sampling, periodic interrupts  
conversion distance to voltage, ADC signal averaging  
signal to noise ratio, central limit theorem, Nyquist,  
calibration

Lab: Input distance; detect distance and orientation to wall

**<Tach> 16. Tachometer**

Prerequisites: <Timers><Motors>

Motivation: The robot can have tachometers to measure wheel rotational speed

Tools: Voltmeter, oscilloscope, CCS

Equipment: Tachometer with digital inputs, DC motor,  
LaunchPad

Theory: period measurement interrupts

Lab: Measure motor speed

**<Control> 17. Control Systems**

Prerequisites: <Tach>

Comment: Assume students do not have a lot of control theory. However, they still could implement an incremental and an integral controller.

Motivation: If we have a tachometer or encoder, we can implement a digital controller.

Tools: Voltmeter, current meter, oscilloscope, CCS

Equipment: Robot with tachometer on the motors

Theory: Input capture, incremental control, integral control

Lab: Closed loop DC motor control, spin at constant speed

**<UART> 18. Serial communication**

Prerequisites: <Debug>

Motivation: Students could use a long USB cable to debug and control the robot in a tethered fashion while the robot is running.

Tools: Voltmeter, oscilloscope, logic analyzer, CCS

Equipment: LaunchPad connected with UART to a PC

Theory: Modulation, encoding, transmission, decoding, error detection, synchronization, FIFO queues

Lab: stream data from robot to PC, build an interpreter so student can manually control the robot from the laptop keyboard.

**<BLE> 19. Bluetooth Low Energy**

Prerequisites: <UART>

Motivation: Students could use a cell phone to debug and control the robot.

Tools: Logic analyzer, CCS

Equipment: LaunchPad connected with UART to CC2650BP (SNP)

Theory: characteristics, services, advertising

Lab: stream data from microcontroller to phone

**<Wi-Fi> 20. Wi-Fi**

Prerequisites: <UART>

Motivation: Students stream data from the robot onto a web page.

Tools: Logic analyzer, CCS

Equipment: LaunchPad connected with UART to CC3120  
Booster

Theory: UDP TCP DNS, wireless router, creating a web server

Lab: stream data from microcontroller to web server

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2018, Texas Instruments Incorporated