

TI-RSLK

Texas Instruments Robotics System Learning Kit



Module 1

Introduction: Running code on the LaunchPad using CCS



Introduction: Running code on the LaunchPad using CCS

Educational Objectives:

REVIEW Software development methodology

UNDERSTAND How to set up an Integrated Development Environment

EXPLORE The out of box examples

LEARN How to import and export CCS projects

DESIGN, BUILD & TEST A SYSTEM

Understand the debug tools and plug-ins

Prerequisites (None)

- None

Recommended reading materials for students:

- MSP432P401R SimpleLink™ Microcontroller LaunchPad™ Development Kit (MSP-EXP432P401R) User Guide (SLAU597)
- MSP-EXP432P401R Quick Start Guide (SLAU596)
- MSP432P4xx Technical Reference Manual (SLAU356)
- MSP432P401Rx Datasheet (SLAS826)
- TI Resource Explorer (MSP432 SimpleLink SDK)
- TI SimpleLink Academy (<http://dev.ti.com/MSP432-Simplelink-Academy>)
- Chapter 1, **Embedded Systems: Introduction to Robotics**, Jonathan W. Valvano, ISBN: 9781074544300, copyright © 2019

Introduction to the curriculum

In the following modules you will learn about the concepts of robotics in the context of embedded systems. The most important part of the robot will be the main processor or “brain” of the system. The processor will manage the programmable logic of the system and interface with the peripherals for inputs such as sensors and outputs such as motors.

To prepare us to build the robotic system, we will first learn how to master the processor by setting up our hardware development kit and the software development environment used to write the software to control our system.

Software Development

The first step to any embedded development is to set up the software development environment we plan to use once the hardware has been chosen. It is often popular and wise to choose an Integrated Development Environment (IDE). An IDE can have a list of features that aid in the ease or speed of software

development. In the hardware context, this could include providing critical debugging information needed to understand the memory usage and performance of the software on the processor.

Code Composer Studio (CCS) is an industry-ready IDE option that is provided by Texas Instruments for use with TI microcontrollers and embedded processors. CCS has many features that make it very capable for professional engineers to develop firmware for real products. It comprises a suite of tools (optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler) used to develop and debug embedded applications. Because it can do so much, it can also be a lot to learn for beginners, but don't get discouraged as this module will direct you on how to set up CCS so you can go through exercises smoothly as you build your robotic system.

Your code is stored inside of a CCS project. A project can contain many items including your code files, configurations, and other relevant files.

The Project Explorer in CCS shows us the various components used for each project. A **linker** builds a single software system by connecting (linking) software components. In CCS, the **build** command performs both a compilation and a linking.

In an embedded system, the **loader** will program object code into flash on the microcontroller. We place object code in flash ROM because flash retains its information if power is removed and restored. In CCS, the **Debug** command performs a load operation and starts the debugger.

A **debugger** is a set of hardware and software tools we use to verify system is operating correctly. The two important aspects of a good debugger are control and observability.

A logic analyzer is a tool that will help you debug your circuit. You can view in real time the signals that are being generated on the pins. In this course we will make use of the TExaSdisplay logic analyzer. This is a free tool that works within the MSP432 LaunchPad and uses your PC for display.

In the lab associated with this module, you will install your copy of Code Composer Studio and test some code examples that are provided for your LaunchPad as a getting started exercise. This will be a good starting point as we familiarize ourselves with the main digital control unit of the explorer robot.

ti.com/rslk



IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2019, Texas Instruments Incorporated