Module 4

Introduction: Software Design Using MSP432

Introduction: Software Design Using MSP432

Educational Objectives:

REVIEW C programming
UNDERSTAND conditional statements and loops in C
DEVELOP logic and arithmetic functions
LEARN how to debug simple programs in C
DESIGN, BUILD & TEST A SOFTWARE COMPONENT

As the robot explores its world, it must make decisions. In the lab associated with this module, you will write software that takes input from three distance sensors and determines if one of eight possible scenarios is present, see Figure 1. The actual sensors will be interfaced in Lab 15, but in this lab you will write software to be used in the robot later.

Prerequisites (Module 1)

- Running code on the LaunchPad using CCS (Module 1)

Recommended reading materials for students:

- Volume 1 Chapter 1, Sections 2.8, 5.1, 5.2, 5.3, 5.6, and 5.8
  Embedded Systems: Introduction to the MSP432 Microcontroller ISBN: 978-1512185676, Jonathan Valvano, copyright (c) 2017

or

- Volume 2 Sections 1.4, 1.5, 3.1, 3.2, 3.3, and 3.4
  Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller, ISBN: 978-1514676585, Jonathan Valvano, copyright (c) 2017

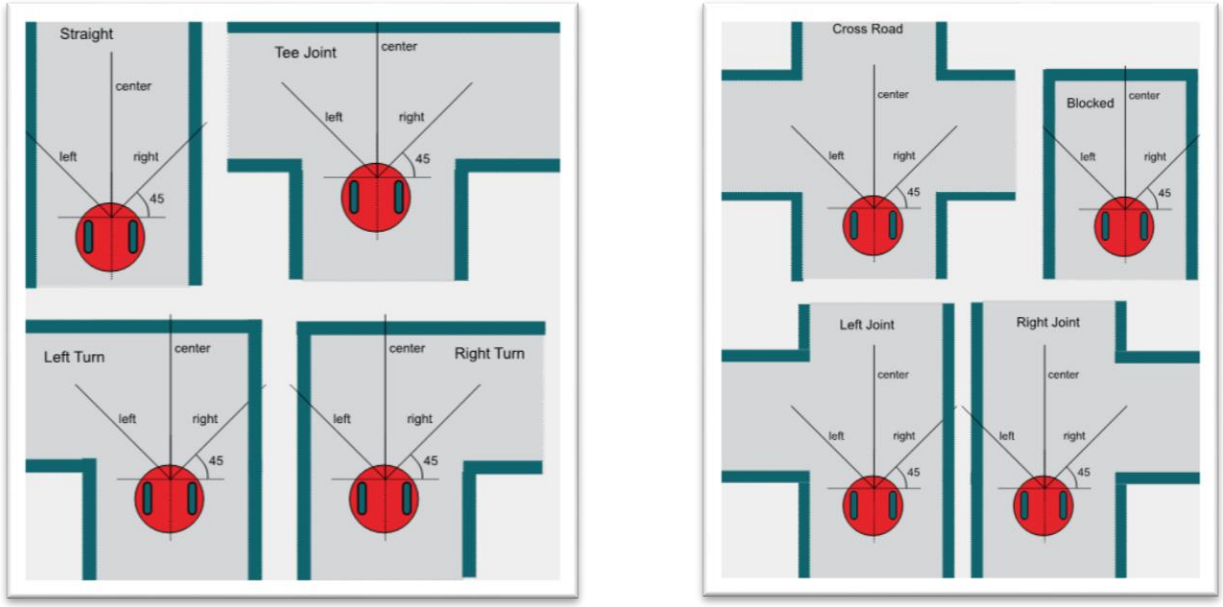

Figure 1. Eight possible scenarios as the robot explores the maze. As the robot approaches an intersection, it first determines what alternative paths exist, and then it chooses which way to go

Introduction: Software Design Using MSP432

This module serves as a brief introduction to C. C is a general-purpose programming language initially developed by Dennis Ritchie between 1969 and 1973 while at AT&T Bell Labs. In 1999, a professional standard version of C, called C99, was defined. In this class we will write our software in C99, because it is prevalent in industry.

A **compiler** is system software that converts a high-level language program (human readable format) into object code (machine readable format). It produces software that is fast but to change the software we need to edit the source code and recompile.

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 ROM. We place object code in ROM because ROM is 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.

Before we write software, we need to develop a plan. Software development is an iterative process. Even though we list steps the development process in a 1,2,3,4 order, in reality we cycle through these steps over and over. I like to begin with step 4), deciding how I will test it even before I decide what it does.

1) We begin with a list of the inputs and outputs. This usually defines what the overall system will do. We specify the range of values and their significance.
2) Next, we make a list of the required data. We must decide how the data is structured, what does it mean, how it is collected, and how it can be changed.
3) Next we develop the software algorithm, which is a sequence of operations we wish to execute. There are many approaches to describing the plan. Experienced programmers can develop the algorithm directly in C language. On the other hand, most of us need an abstractive method to document the desired sequence of actions. Flowcharts and pseudo code are two common descriptive formats. There are no formal rules regarding pseudo code, rather it is a shorthand for describing what to do and when to do it. We can place our pseudo code as documentation into the comment fields of our program. Next we write software to implement the algorithm as define in the flowchart and pseudo code.
4) The last stage is debugging. Learning debugging skills will greatly improve the quality of your software and the efficiency at which you can develop code.

In the lab associated with this module, you will develop and test some software functions that will be used later in the explorer robot. In particular, the first function will convert ADC measurements from a sensor into distance to the wall, and the second function will take three distance measurements and classify the situation into the most likely scenario.

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/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265