Lab 4 – Introduction to Using SYS/BIOS Real-Time Operating System

on

TI TMS320F28027 Piccolo Microcontroller

NOTE: THIS DOC IS WRITTEN ASSUMING A 28027 LAUNCHPAD IS BEING USED

NOTE: THIS DOC IS WRITTEN ASSUMING CODE COMPOSER STUDIO VERSION 5 IS BEING USED

Code Composer Studio versions 6 and 7 do not easily support the "Raw Log" functions that appear in the code in this lab, therefore you need to install CCSv5 to do this lab

In this lab you will:

- experiment with code that uses SYS/BIOS RTOS
- explore the graphical configuration environment for setting up a project that uses SYS/BIOS
- explore C code structure for the various thread-types
- use existing code templates there is no need to use your files from previous labs
 - 1. experiment with "Idle Example" code
 - 2. experiment with "Swi Example" code
 - 3. experiment with "Task Example" code
 - 4. modify the Task Example code by adding a new task and additional semaphore
- observe print messages in Raw Logs window

What to submit to D2L (use .zip file):

- answers to any questions posed in this handout or on the whiteboard
- any snapshot(s) you feel would help make your lab report better
- code you write or modify for Item 4 above, i.e., the .c and .cfg files
- snapshot(s) of information on the new task you add for Item 4
- snapshot of the Raw Logs display for Item 4

Some Notes on Tsk Threads in This Lab

- In this lab, Tsk threads are created statically, i.e., they are created by the RTOS once only at the time that the system starts up.
- In this lab, Tsk threads are always in one of three states: READY, BLOCKED, or RUNNING. They are never TERMINATED.
- In SYS/BIOS, the state of a semaphore is defined as follows:
 - = 0 \rightarrow not available (blocking)
 - = 1 \rightarrow available

Some Other Notes

- In order to use SYS/BIOS in a project, you must pick a SYS/BIOS template to start with to get the appropriate Include files for SYS/BIOS.
- The xds100 Emulator on the Piccolo is inexpensive, but a bit slow. It can take some time for the Build and Load processes to be performed. Be patient.
- When you want to re-Build your code, I recommend that you explicitly **Terminate** the Debug session first, otherwise I noticed the Raw Logs might not update properly thereafter.

0. Install Code Composer Studio v5 (including SYS/BIOS)

Copy the .exe install file from Share Out\ELEX\7820 to your desktop and run the installer.

Select Custom install. You only need to select the c2000 tools and the XDS emulator packages.

The install takes about 15 min on the lab PC (and you may need to say yes to a firewall message part way through).

Note that, when you set up your CCS project later in this lab, it will need to use a pre 3.30 version of XDC tools.

You can check the packages that get installed in CCS by doing:

Window – Preferences – Code Composer Studio – RTSC – Products



• To see Help Info on SYS/BIOS, e.g. API functions:

In Code Composer: Help – Help Contents





1. Create New Project Based on SYS/BIOS Template "Idle Example"

Create a new project called "Lab4Idle" according to the following steps:

View – Resource Explorer (Examples) or similar

And navigate to the 28x Idle Example



And Import the example project into CCS.

And **Build** it.

Set up the **Debugger Configuration** as necessary.

Create a new CCS Project. Iarget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator Compiler version: T1 46.27 Advanced settings Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buf Compiler version: T1 46.27 New CCS Project CCC Project CCC Onfiguration Settings elect the RTSC Configuration project settings. arget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator CCC Project CCC Project CCC Onfiguration Settings elect the RTSC Configuration project settings. arget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator CCC Project CCC Project CCC Onfiguration Settings elect the RTSC Configuration project settings. arget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator CCC 28XX (C2000) XDCtools versice CCC 28XX (C2000	CS Broject		
Iarget TM5320F28027 Connection: Texas Instruments XDS100v2 USB Emulator C 28XX [C2000] Project name: idle_TMS320F28037 U Se glefault location Location: DNWork/CodeComposerviale_TMS320F28027 Location: DNWork/CodeComposerviale_TMS320F28027 Project templates and examples A dvanced settings Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buff Rew CCS Project SC Configuration Settings elect the RTSC Configuration project settings. [arget: TM5320F28027 connection: Texas Instruments XDS100v2 USB Emulator C 28XX [C2000] XDCtools versite: 325.3.72 Products and Repositorice Order S 262.027 connection: Texas Instruments XDS100v2 USB Emulator C 28XX [C2000] XDCtools versite: S 25.4.50 S \$VS/BIOS	Create a new CCS Project.		V
arget: TMS20F28027 Connection: Tecas Instruments XDS100v2 USB Emulator C 28XX [C2000] Project name: idle_TMS320F28027 W Use glefault location: D:Work/CodeComposer/idle_TMS320F28027 Compiler version: TMv6.2.7 A dvanced settings Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buf Rev CCS Project SC Configuration Settings releast the RTSC Configuration project settings. aget: TMS320F28027 connection: Texas Instruments XDS100v2 USB Emulator Castax (C2000] XDCtools versice: 225.3.72 Products and Repositories Q System Analyzer (UA Target) Q System Analyzer (UA Target) Setem System Analyzer (UA Target) Q System Analyzer (UA Target) Q System Analyzer (UA Target)			
Connection: [Itexas instruments XUSIOV2 USB Emulator Project name: idle_TMS320F28027 Compiler version: Itwork/CodeComposer/idle_TMS320F28027 Compiler version: Itwork/CodeComposer/idle_TMS320F28027 Compiler version: Itwork/CodeComposer/idle_TMS320F28027 Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buff New CCS Project SC Configuration Settings rease Instruments XDS100v2 USB Emulator connection: Teras Instruments XDS100v2 USB Emulator Cast (2000) XDCtools versice 325.3.72 Products and Benositories Cast (2000) XDCtools versice System Analyzer (UA Target) Stemplation Stemplation System	Target: TMS320F28		
C28XX [C2000] Project name: idle_TMS320F28027 Isource in the intervention of the intervention	Connection: Lexas Instru	Iments XDS100v2 USB Emulator	Veri
Project name idle_TMS320F2802 I Use gefault location Location: Location: DeWork/CodeComposervidle_TMS320F28027 Compiler version: Tito 62.7 N Advanced settings Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buff Rew CCS Project Image: StC Configuration Settings Image: Idle TMS320F28027 Image: Connection: Texas Instruments XDS100v2 USB Emulator Image: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator Image: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator Image: Texas Instruments XDS100v2 USB Emulator Image: System Analyzer (UIA Target) Image: System Analyzer (UIA Target) Image: System Analyzer (UIA Target) Image: Target: Target: titargets.C28_large Platform: tiplatforms.tms320r28027 Build-profile: debug	🖆 C28XX [C2000]		
☑ Use gefault location Location: D:Work/CodeComposer/idle_TMS320F28027 Pr Compiler version: T1x62.7 N A dvanced settings Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buf Image: Configuration Settings elect the RTSC Configuration project settings. lett the RTSC Configuration project settings. arget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator Image: C28XX (C200) XDC tools verside: 325.3.72 Image: SYS/BIOS Image: SyStem Analyzer (UIA Target) Image: Titargets.C28_large Platform: ti.platforms.tms3205/28.77 Build-profile: debug	Project name: idle_	_TMS320F28027	
Compiler version: TI V6.2.7 Advanced settings Project templates and examples A 20x specific example which uses an Idle function to print to standard out and a log buf Compiler version: Carlos and the standard out and the standard out and a log buf Compiler version: Carlos and the standard out and the standard out and a log buf Compiler version: Carlos and the standard out and the standard	Use <u>d</u> efault location		
Compiler version: ITVOL2 IN Advanced settings: Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buff Image: The standard	Location: D:(V	Vork\CodeComposer\idle_1MIS320F28027	Browse
 Advanced settings Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buff 2 < Back Next > Emish New CCS Project SC Configuration Settings elect the RTSC Configuration project settings. [arget: TMS320F28027 connection: Texas Instruments XDS100v2 USB Emulator C28XX [C2000] XDCtools verside: 325.3.72 Products and Repositories & Order SYS/BIOS SyStem Analyzer (UIA Target) System Analyzer (UIA Target) System Analyzer (UIA Target) Farget: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile debug 	Complier version.		- More
 Project templates and examples A 28x specific example which uses an Idle function to print to standard out and a log buf 28x specific example which uses an Idle function to print to standard out and a log buf New CCS Project ISC Configuration Settings ielect the RTSC Configuration project settings. [arget: TM5320F28027 connection: Texas Instruments XDS100v2 USB Emulator C28xX [C2000] XDCtools verside: 325.3.72 Products and Repositories order I SYS/BIOS I System Analyzer (UIA Target) I stargets. C28_large Platform: ti.platforms.tms320s28:TM5320F28027 Build-profile: debug 	Advanced settings		
A 20x specific example which does an use function to print to standard out and a rog built ? < Back	Project templates and A 28 specific sympole	examples	ut and a log buffer
Image: Project Image: TMS320F28027 connection: Texas Instruments XDS100v2 USB Emulator Image: Texas Instruments Image: Texas Image:			
? < Back			
? < Back			
Image: Next > Finish Image: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator Image: Texas Instruments XDS100v2 USB Emulator Image: C28XX [C2000] XDCtools verside: 3.25.3.72 Image: 0.76 Image: 0.75.4.50 Image: 0.76 Im			
New CCS Project	?	< Back Next >	Einish Can
New CCS Project ISC Configuration Settings ielect the RTSC Configuration project settings. Iarget: TMS320F28027 connection: Texas Instruments XDS100v2 USB Emulator C28XX [C2000] XDCtools version: 3.25.3.72 Products and Renositories & Order Setting & 6.35.4.50 Setting & 5.95.4.50 Setting & 1.31.08 Rer Setting & 1.31.08 Rer Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile debug			
TSC Configuration Settings Select the RTSC Configuration project settings. Larget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator C28XX [C2000] XDCtools verside: 3.25.3.72 ● Products and Repositories Image: [3.25.3.72] Image: [1.3.1.08] Image: [1.3.1.08] Image: [1.1.1.108]	New CCS Project		
ielect the RTSC Configuration project settings. [arget: TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator C28XX [C2000] XDCtools versick: 3.25.3.72 ■ Products and Banositories & Order ■ Ø SYS/BIOS Ø System Analyzer (UIA Target) Ø System Analyzer (UIA Target) Ø System Analyzer (UIA Target) Ø Selec Desee Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile debug	ISC Configuration Set	tings	
TMS320F28027 Connection: Texas Instruments XDS100v2 USB Emulator C28XX [C2000] XDCtools versick: 3.25.3.72 Products and Repositories Order Image: System Analyzer (UIA Target) Astronomic Repositories Image: Vitagets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile Build-profile debug	Select the RTSC Configura	ation project settings.	1
Connection: Texas Instruments XDS100v2 USB Emulator Connection: Texas Instruments XDS100v2 USB Emulator XDCtools versice: 3.25.3.72 Products and Repositories ♦ Order Sys/BIOS Sys/BIOS System Analyzer (UIA Target) Sele Dese Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	Target: TMS320F28(027	
C28XX [C2000] XDCtools version: 3.25.3.72 ■ Products and Repositories Image: Cash of the second		iments XDS100v2 USB Emulator	
C28XX [C2000] XDCtools verside: 3.25.3.72 ■ Products and Repositories Image: Image: Image: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	CONTICUTORS 1 CAUS INSUL		
XDCtools verside: 3.25.3.72			
▶ Products and Banositories ♦ Order ▲ Ø ♥ ● SYS/BIOS ♥ ● 6.35.4.50 ▲ Ø ♥ ● System Analyzer (UIA Target) ▲ Ø ♥ ● 1.3.1.08 €ee Desee Desee Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile debug	C28XX [C2000]		
▲ W ▲ SYS/BIOS ▲ Add ● W ▲ System Analyzer (UIA Target) ▲ Ed ■ W ▲ System Analyzer (UIA Target) ■ Ed ■ W ▲ System Analyzer (UIA Target) ■ Ed ■ Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	C28XX [C2000] XDCtools version: 3.2	5.3.72	
Image: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile debug	C28XX [C2000] XDCtools version: 3.2	5.3.72 Sitories & Order	
Image: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	C28XX [C2000] XDCtools version: 32 Products and Repo	5.3.72 ssitories & Order	<u>A</u> dd
Sele Dese Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	C28XX [C2000] XDCtools version: 3.2 Products and Report V System An System An	5.3.72 Pasitories & Order 50 Jalyzer (UIA Target)	Add Edit
Dese Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	C28XX [C2000] XDCtools version: 32 Products and Renor ♥ SYS/BIOS ♥ System An ♥ \$ 1.31.06	5.3.72 ssitories & Order 50 Ialyzer (UIA Target) 3	Add Edit Remove
Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	C28XX [C2000] XDCtools version: 32 Products and Repo V & SYS/BIOS V & 6.35.4.5 V & System An V & System An	5.3.72 asitories & Order 50 halyzer (UIA Target) 3	Add Edit Remove
Target: ti.targets.C28_large Platform: ti.platforms.tms320x28:TMS320F28027 Build-profile: debug	C28XX [C2000] XDCtools verside: 32 Products and Report V S/BIOS V S/S/BIOS V S/S/BIOS V S/S/BIOS V S/S/BIOS V S/S/BIOS V S/S/BIOS V S/S/BIOS V S/S/BIOS	asitories & Order 50 alyzer (UIA Target) 3	Add Edit Remove Select Al Deselect A
Platform: ti.platforms.tms320x28:1MS320F28027 Build-profile: debug	C28XX [C2000] XDCtools version: SXDCtools version: SXS/BIOS SXS/SXS/SXS/SXS/SXS/SXS/SXS/SXS/SXS/SX	25.3.72 asitories & Order 50 Jalyzer (UIA Target) 3	Add Edit Remove Select Al Deselect A
	C28XX [C2000] XDCtools version: 32 Products and Report SYS/BIOS SYS/SIS SYS/BIOS SYS/SIS SYS	15.3.72 sitories & Order 50 Ialyzer (UIA Target) 8 .C28_large	Add Edit Remove Select Al Deselect A
	C28XX [C2000] XDCtools version: 32 Products and Repo ♥ SYS/BIOS ♥ ♣ System Ann ♥ ♣ System S	15.3.72 asitories & Order 50 Islyzer (UIA Target) 3 .C28_large ms.tms320x28:TMS320F28027	Add Edit Remove Select Al Deselect A
	C28XX [C2000] XDCtools verside: Products and Report Image: Image: Target: ti.targets Platform: ti.platform: Build-profile: debug	15.3.72 asitories & Order 50 Ialyzer (UIA Target) 3 .C28_large ms.tms320x28:TMS320F28027	Add Edit Remove Select Al Deselect A
	C28XX [C2000] XDCtools verside: 32 Products and Pape Products and Pap	15.3.72 petitories & Order 50 1alyzer (UIA Target) 3 .C28_large ms.tms320x28:TMS320F28027	Add Edit Remove Select Al Deselect A
A Back Next > Finish	C28XX [C2000] XDCtools version: 32 Products and Repo V & SYS/BIOS V & 6.35.4.5 V & 7.35.4.5 V & 7.35.4.5V V & 7.35	15.3.72 Disitories & Order 50 Ialyzer (UIA Target) 8 .C28_large ms.tms320x28:TMS320F28027	Add Edit Remove Select Al Deselect A

Rename the project from idle_TMS320F28027 to Lab4Idle

.cfg File

.cfg file can be viewed via a graphical interface - this makes it easier to develop code that uses SYS/BIOS

Observe:

- packages to support SYS/BIOS and UIA have been installed
- your two files: idle.c and idle.cfg • die.cfg - Code Composer Studio igate Project Scripts Bun <u>W</u>indow <u>H</u>elp | ∲ + | ∲ + | ∦ | 100 € + € + CCS Edit - Lab-4idik 10.0014 ET CCS Ede Quick Access Project Explorer edg 🛙 E = v = o + SYS/BIOS - Welcome + 0 Error Handling Device Support Advar BELOS ler/c2000_6.2.7/in /tools/compiler/c2000_6.2.7 s/bios_6_35_04_50/packages s/uia_1_03_01_08/packages s/wdctools_3_25_03_72/pack Add SYS/BIOS to my configuration Default Diags Error Hwi Idle Log + Introduction SYS/BIOS is a scalable real-time kernel. It is designed to instrumentation. SYS/BIOS provides preemptive multi-ti minimize memory and CPU requirements on the target. Getting Started More Information Navigate to the <u>System Overview</u> screen that depicts all of the core modules in SYS/BIOS. You can add any of these modules to your configuration by right-clicking on it and selecting "Use". You can co any module by clicking on it. At minimum, two different views are tl available to choose from: User Guide, You can also access this guide and the release note Code Composer Studio's menu Help->Help Contents and then ing the SYS/BOS book documentation item. API Reference, You can also access this documentation f composer Studio's menu Help->Help Contents and then teference" underneath SYS/BIOS. Basic. This view includes all of the configureers will need. Tooling & Contention Help, Each check box, drop list, etc. in the SYS Grace pages corresponds to a single configuration parameter. Hos mouse over a field to see a summary of the parameter. Or ight of the field and select "Help" to be taken to the documentation for th parameter in the "API Reference" described above. Advanced. This view always includes every configuration se generic table format. 2 er. Or, right click Note that you may freely move between the different views, and se phons in one view will immediately update the entry view, and te say return to the initial home view at any time by clicking on the H ron in the upper right comer of the config tool window. Note that varnings about your configuration may appear in the CCS Problem indow. If this under the configuration may appear in the CCS Problem 105 wiki, Online articles addre ues and tasks ms. For further support, you can post qu - -SYS/BIOS cfg Script 0 3 Target Spi 🖹 Problems 🕸 C Realtime Ana Type : 14 Loggers : 14 Services @ LoggingSetup ecsv6/ccs base Idle/idle.cfg Full License

Use this tree to select ("use") items (e.g. Hwi, Swi, Tsk) to add to your code

OR USE THE "SYSTEM OVERVIEW" GRAPHICAL ENVIRONMENT TO USE/INSTANTIATE NEW ITEMS

Use this tree to use ("instantiate") new items (e.g. Hwi, Swi, Tsk) into your code

Question 1A: How can you see the non-graphical, XDC-script-based version of the .cfg file?

After creating your project and observing the .cfg file...

1. Peruse the idle.c file

Question 1B: What headers are included at the top of the code and what are they used for?

Log_info() is a function that is much more efficient than printf() and can be used to print
information to the Raw Log buffers.

Observe that there is a fundamental difference between the **main** code from previous labs and the **main** code when SYS/BIOS is invoked:

Without SYS/BIOS, the main code stayed in an infinite loop.

With SYS/BIOS, the last thing the main code does is to branch to the RTOS, never to return unless the system is re-booted.

2. Launch the Debugger.

Observe that SYS/BIOS gets programmed into the Flash - this might take some time...

🐨 Erasing Flash Sectors.
Erasing Flash Sectors.
Erasing Sector C(Texas Instruments XDS100v2 USB Emulator/C28xx)(Cannot be canceled)
Always run in background
Run in Background Cancel Details >>

3. Open the Raw Logs window:

Tools – RTOS Analyzer – RTA (Legacy) – Raw Logs

or Tools – RTA – Raw Logs

then Resume (Run) the code for a few seconds and then Suspend (Halt) it, and observe the information in the **formattedMsg** column of Raw Logs.

Question 1C: What does the time column in Raw Logs represent?

When you Suspend the code execution in the Debug perspective, notice that a window will open where the code has temporarily stopped. It might be in idle.c or Idle.c or other:

idle.c \rightarrow your user code

Idle.c \rightarrow SYS/BIOS kernel code

Observe the graphical representation of your system showing what parts of the RTOS are used in this application:



Question 1F: Navigate in the graphical environment to find where you can **change the rate** at which the timer interrupt occurs. Change it, re-build the project, and verify that your change worked.

2. Create New Project Based on SYS/BIOS Template "Swi Example"

Close your Idle project and create a new project called "Lab4Swi" based on the 'Swi Example" template.

Peruse the swi.c file. Observe that the Swi function starts and ends (i.e., it does not have an infinite loop cf. Tsk function).

(Remember to rename the project.)

Build the code.

Set up the Debugger Configuration as necessary.

Launch the Debugger.

Run the code for a while and then observe the Raw Logs buffer.

Question 2A: What two functions are in the Idle thread?

Question 2B: Draw a processor timeline showing the main, idle, myTickFxn, *and* mySwi *threads.* (for simplicity, you do not need to show the divide-by-10)

3. Create New Project Based on SYS/BIOS Template "Task Example"

Close your Swi project and create a new project called "Lab4Task" based on the 'Task Example" template.

Peruse the task.c file. Observe that the Task function has an infinite **for** or **while** loop.

(Remember to rename the project.)

Build the code.

Set up the Debugger Configuration as necessary.

Launch the Debugger.

Run the code for a while and then observe the Raw Logs buffer.

Recall from a previous lecture that each task in a multi-tasking system has a Task Control Block (TCB) associated with it. The TCB contains information like the task priority, task state, and other info. (In SYS/BIOS, it is not called TCB.)

Question 3A: In the SYS/BIOS kernel code, find the structure definition equivalent to the

Task Control Block and describe what each member represents.

4. Modify the Task Example Project

Close your Task project and create a new project called "Lab4Taskmod" based on the 'Task Example" template.

In the following, replace "Dave" with your first name.

- Graphically modify the task.cfg file by adding an additional task instance called "newTask" that will correspond to the function "DaveTaskFxn" that you will add to your code. Set the Priority of the new task to 2.
- 2. Graphically modify the task.cfg file by adding an additional binary semaphore called "DaveSem".
- 3. Modify the task.c file by adding code for the function "DaveTaskFxn" and modifying myTaskFxn. The task is to print to the Raw Logs buffer the message shown below. DaveTaskFxn is to pend the semaphore DaveSem. The semaphore DaveSem is to be posted from the function "myTaskFxn" once every three times that myTaskFxn logs its info.

🌐 Raw Logs 🛛		- 4 4	🝷 🎫 🁬 🌩 🔻 🕕 🛛 All Log Buffers 🛛 💌 🖽 🖓 🗞	¢ ∨ ⊓ □
time	s	m	formattedMsg c. logger	
5,700,037,477	26	Main	"/task.c", line 89: 10 ticks. Tick Count = 190 xdc.runtime.LoggerBuf.Instances	#0
6,000,037,419	27	Main	"/task.c", line 89: 10 ticks. Tick Count = 200 xdc.runtime.LoggerBuf.Instances	#0
6,300,037,696	28	Main	"/task.c", line 89: 10 ticks. Tick Count = 210 xdc.runtime.LoggerBuf.Instances	#0
6,300,044,326	29	Main	"/task.c", line 119: *** DAVE HERE!!! *** 🔶 📩 xdc.runtime.LoggerBuf.Instance;	#0
6,600,037,708	30	Main	"/task.c", line 89: 10 ticks. Tick Count = 220 xdc.runtime.LoggerBuf.Instances	#0
6,900,037,665	31	Main	"/task.c", line 89: 10 ticks. Tick Count = 230 xdc.runtime.LoggerBuf.Instances	#0
7,200,037,655	32	Main	"/task.c", line 89: 10 ticks. Tick Count = 240 xdc.runtime.LoggerBuf.Instances	#0
7,200,044,285	33	Main	"/task.c", line 119: *** DAVE HERE!!! *** 🔶 xdc.runtime.LoggerBuf.Instance	#0
7,500,037,668	34	Main	"/task.c", line 89: 10 ticks. Tick Count = 250 xdc.runtime.LoggerBuf.Instance	#0
7,800,037,611	35	Main	"/task.c", line 89: 10 ticks. Tick Count = 260 xdc.runtime.LoggerBuf.Instances	#0
8,100,037,554	36	Main	"/task.c", line 89: 10 ticks. Tick Count = 270 xdc.runtime.LoggerBuf.Instance	#0
8,100,044,184	37	Main	"/task.c", line 119: *** DAVE HERE!!! *** 🔶 🚽 xdc.runtime.LoggerBuf.Instance:	#0
8,400,037,565	38	Main	"/task.c", line 89: 10 ticks. Tick Count = 280 xdc.runtime.LoggerBuf.Instance	#0
8,700,037,738	39	Main	"/task.c", line 89: 10 ticks. Tick Count = 290 xdc.runtime.LoggerBuf.Instance	#0
9,000,037,679	40	Main	"/task.c", line 89: 10 ticks. Tick Count = 300 xdc.runtime.LoggerBuf.Instances	#0
9,000,044,309	41	Main	"/task.c", line 119: *** DAVE HERE!!! *** 🔶 xdc.runtime.LoggerBuf.Instance:	#0