Lab 1 – Introduction (Continued) to TI Code Composer Studio v7 IDE and

TI TMS320F28027 Piccolo Microcontroller

NOTE: THIS DOC IS WRITTEN ASSUMING A LAUNCHPAD IS BEING USED; IF YOU HAVE ANOTHER PLATFORM, THERE SHOULD BE ONLY MINOR DIFFERENCES TO WATCH OUT FOR

NOTE: SOME OF THE PICS IN THIS DOC WERE TAKEN WITH CCS VERSION 5; VERSION 7 SHOULD BE SIMILAR

In this lab you will learn how to:

- Add existing files to your new Project, including TI's header files
- Add an existing main program that flashes an LED on the LaunchPad
- Observe the hierarchical structure of the Peripheral information for the CPU Timers and GPIO pins in their associated header files
- Observe and modify variables in the Variables window
- Observe the hierarchical structure of the Peripheral information for the CPU Timers and GPIO pins in the Registers window
- Observe the" look-ahead" context feature when entering a Peripheral in the C code
- Modify the main code to configure an additional GPIO pin as an output (not GPIOO) and use this line to indicate on a scope the utilization of a simple FIR dot product that you shall add to the C code

See last page for details on what to submit.

Create a New Project

Open CCS and create a new empty CCS project called "ELEX7820-Lab1" in the workspace and configure it appropriately for the Piccolo target. The project should be configured for Debug (not Release) and set as Active.

Add Existing Files (from D2L) to New Project (right-click name of project to access menu)

CCS Edit - Code Co	mpo	ser Studio	
ile Edit View N	aviga	te Project Run Scripts Window Help	
1 · 🛛 🖬 🕴	6 -	1 to	🖹 🎭 CCS Debug 🕞 ("
Project Explorer	3		
FLEX7820-La	h1 I	Active - Debuni	
a 🔊 Includes		New	1
> 🙆 C:/ti/c	-	6	
# 🗁 targetCon	-	сору сті+с	
😭 contro		Paste Ctri+V	
i readm	×	Delete Delete	
28027_RA		Source ,	
		Move	
		Rename F2	
	i.	Import	
	-	Emot	
	ш	Export	
		Show Build Settings	
		Build Project	
		Clean Project	
		Rebuild Project	
	5	Refresh FS	
	~	Close Project	
		close Project	
		Build Configurations	
		Make Targets	
		Index >	
		Add Files	
		Debug As	posele 22 🗹 🖓 🕈 🖸
		Team	nsoles to display at this time.
		Compute With	
		Restore from Local Mistory	
		Reference Court History	
		Relactor ,	
		Source	
a harrie		Source	
Licens		Properties Alt+Enter	

rganize 🔻 🛛 New fol	der				-	
COMP 🖌	•	Name	Date modified	Туре	Size	
📗 стмх		DSP2802x Adc.h	1/4/2011 10:34 AM	C/C++ Header		17 KB
📗 СТУМ		DSP2802x_BootVars.h	1/4/2011 10:34 AM	C/C++ Header		2 KB
DELX		h DSP2802x_Comp.h	1/4/2011 10:34 AM	C/C++ Header		3 KB
DEMX		h DSP2802x_CpuTimers.h	1/4/2011 10:34 AM	C/C++ Header		6 KE
Document Preg	1	DSP2802x_DevEmu.h	1/4/2011 10:34 AM	C/C++ Header		3 KB
		h DSP2802x_Device.h	1/4/2011 10:34 AM	C/C++ Header		9 KB
EASA		h DSP2802x_ECap.h	1/4/2011 10:34 AM	C/C++ Header		6 KB
ECHO		h DSP2802x_EPwm.h	1/4/2011 10:34 AM	C/C++ Header		23 KB
ECON		DSP2802x_GlobalVariableDefs.c	1/4/2011 10:34 AM	C File		8 KB
EENG		b] DSP2802x_Gpio.h	1/4/2011 10:34 AM	C/C++ Header		12 KB
ELEX		DSP2802x_Headers_nonBIOS.cmd	1/4/2011 10:34 AM	Windows Comma		6 KB
ELIS		h DSP2802x_12c.h	1/4/2011 10:34 AM	C/C++ Header		8 KB
· · · · · · · · · · · · · · · · · · ·	-	h DSP2802x_NmiIntrupt.h	1/4/2011 10:34 AM	C/C++ Header		4 KE

CCS Edit - Code Composer Studio	· It is assume that the	per sume Alab sume	
Eile Edit View Navigate Project Run Scripts Window	<u>d</u> elp		
📑 • 🖃 🗟 • 🚸 • 🛷 • 🔲 🍤 🗇 •	<> ▼		😭 🎭 CCS Debug 🕞 (🎽
Project Explorer 🕱 🔲 🖨 😓 🖓 🖓			- 8
 Project Explorer 33 ELEX7820-Lab1 [Active - Debug] ELEX7820-Lab1 [Active - Debug] ELEX7820-Lab1 [Active - Debug] Criti/ccsv5/tools/compiler/c2000_6.1.0/include DSP2802_Compiler DSP2802_	Console ⊠ No consoles to display at this time.		
🕈 Licensed : 73 E 🛱 ELEX7820-Lab1	۱۱		

These files comprise all the files needed to develop a program on the c2000 target (with the exception that there is no RTOS yet). For example, there is a main program file and many of the .h files contain structures to support use of the c2000's peripherals.

Build (i.e., Compile & Link) Project (right-click name of project to access menu)







Look at main.c file (double-click on file name)

Look at DSP2802x_Device.h file

<u>File Edit View Navigate Project Run Scripts Window H</u>	dp	
📑 🖬 🕼 🔦 🔹 🕸 🕶 🛷 🔳 👳 🔶 🖛	> •	😭 🎭 CCS Debug 🕞 🔅
🏠 Project Explorer 🛛 📄 😫 🏹 " 🗆	E ELEX7820-Lab1-main.c In DSP2802x_Device.h ⊠	
Project Explorer Image Image Image Image ■	CE ELD2820-Labi-main. Ib DSP2802x,Device.h X3 97// For portability, User Is Recommended To Use Following Data Type Size 98// Vorfinitions For Isbeit and 32-Bit Signed/Unsigned Integers: 99// 108 108 108 109 109 101 102 103 103 104 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 105 106 117 118 117 118 118 119 119 1117 1117 1118 1119 11117 1118 1118	8027PT)
□ [◆] Licensed : 73 E	Writable Smart Insert 29 : 28	

CCS Edit - ELEX7820-Lab1/DSP2802x_Headers_nonBIOS.cmd - Co	Je Composer Studio	
<u>File Edit View Navigate Project Run Scripts Window</u>	lelp	
📫 • 🔛 🔞 • 🐐 • 🛷 • 🔲 👳 🔶 •	-⇔ ▼	😭 🎭 CCS Debug 🕞 🔅 👋
🎦 Project Explorer 🛛 📄 🖕 🍸 🗖 🗖	🗈 ELEX7820-Lab1-main.c 🚯 DSP2802x_Device.h 🛛 😡 DSP2802x_Headers_nonBIOS.cmd 🔀	- 8
Image: Section of the section of t	■ ELEX7820-Labl-main. ▲ DSP2802x_Device.h ▲ DSP2802x_Headers_nonBIOS.cmd >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	rs */ egisters */ isters */ U Timer1 & Timer2 reserve U Timer1 & Timer2 reserve
	67	
D [♦] Licensed : 73 E	Writable Smart Insert 1:1	

Look at DSP2802x_Headers_nonBIOS.cmd file

What is the basic function of the "DSP2802x_Headers_nonBIOS.cmd" file?

Note the Structure Hierarchy in the CPU Timer and GPIO Header Files

DSP2802x_CpuTimers.h file

Observe the structure of structs and unions in the "DSP2802x_CpuTimers.h" file.



Launch the Debugger...

CCS Debug - CCX7820-Lab1/ELEX7820-Lab1-main.c - Code Composer Studio						
🗈 • 🖃 🔞 🖉 🕸 • 📳 % 🚇 • 🔊 🔊 🗳 • 🛷 •				😭 🎭 CCS Debug 🕞 (🔌		
🎋 Debug 🛛 🥂 🔌 🦓 🕪 💷 🔳 🔍 👁 🐟 🖈 🍐 🛸 🅹	🛛 😓 🖓 🖓 🖓 🖓 🖓 🖓	Expressions	💩 Breakpoints 🛛 🏝 🏘	3 🕸 🖇 💥 🔂 🔂 🖓 🗖		
 ▲ SELEX7820-Labl [Code Composer Studio - Device Debugging] ▲ P Texas Instruments XDS100/1 USB Emulator/C28xx (Suspended - SW Breakp	oint) (v)= LED_Prd (v)= LED_Prd_ctr (v)= LED_Prd_ctr	Type int int	Value 0 -32391	Location 0x00000403@Data 0x00000402@Data		
ELEX7820-Lab1-main.c 🛛						
<pre>1// Filename: ELEX7820-Labl-main.c 2// 3// Description: This program blinks LED3 (GPI034) on the 4// at a rate determined by Con Timer 0 and 1 6// Version: 1.0 7// 8// Target: TM5320F28027 9// 10// Author: David Romalo. 11 12//include files: 13 #include TSP2802x_Device.h" 14 15//function prototypes: 16/videclare global variables: 19 17 17//declare global variables: 19 20//main code starts here: 21 void main(void) 22</pre>	<u>Piccola</u> ControlStick the variable LED_Prd.					
<pre>23 //declare main variables: 24 int16 LED_Prd; 25 int16 LED_Prd_ctr; 26 27 //initialization: 28 DeviceInit(); //initialize processor 29 4</pre>						
🕫 🖹 Licensed : 73 🛙						

The Program Counter has been set to point to the beginning of main.

Start the program executing with the Resume button. A blue LED on the LaunchPad should flash continuously.

At what rate does the LED turn on and off? What is the duty cycle?

Does setting GPIO0 to a logical "1" output cause the LED to turn on or off? Why?

CCS Debug - ELEX7820-Lab1/ELEX7820-Lab1-main.c - Code Composer Studio	amon Chapter	same AaB	added assess to	
<u>File Edit View Project Tools Run Scripts Window Help</u>				
📑 🕶 🔛 🐘 🖛 두 🔜 % 🚜 🕶 🔊 🖄 💣 🕶 🛷 🕶				🖹 🎭 CCS Debug 🕞 (»
🎋 Debug 🛛 🦉 🦄 🕪 💷 🔳 🔍 👁 🤄 🔿 🕀 🌸 🗸 🔞 🏹 🗖 🗖	🔲 🕬= Variables 🛛 🖓 Exp	ressions 1919 Registers	Breakpoints 👘 🕸	
A 🖏 ELEX7820-Lab1 [Code Composer Studio - Device Debugging]	Name	Туре	Value	Location
Texas Instruments XDS100v1 USB Emulator/C28xx (Suspended)	(x)= LED_Prd	int	500	0x00000405@Data
main() at ELEX/820-Lab1-main.c:38 0x008180 args main() at args main c:92 0x008182 (args main has only skeletal debug info)	(x)= LED_Prd_ctr	int	409	0x00000404@Data
BLEX7820-Lab1-main.c ⊠				
25 int16 LED_Prd_ctr;				*
26 27 //initialization:				
<pre>28 DeviceInit(); //initialize processor</pre>				
<pre>29 30 CpuTimer0Regs.PRD.all = mSec1;</pre>				
31				
33 LED_Prd = 500; //Dink once every ? <u>msec</u>				
34				
36 for(;;)				
37 { 38 if(Coutimer@Pers TCP bit TTE == 1)				
39 {				
40 CpuTimer0Regs.TCR.bit.TIF = 1; //clear flag				
42 {				
43 LED_Prd_ctr++; //increment counter				
45 else				E -
46 { 47 GDioDataPegs GPBT056LE bit GPT034 = 1: //toggle line				
48 LED_Prd_ctr = 0; //reset counter				
49 } //endifelse				
51 } //endfor				
52 } //endmain				-
				4
📑 🖹 Licensed : 73 E				

Try modifying the values of variables by editing them in the Variables window.

Example:

- 1. Suspend
- 2. Modify value of "LED_Prd"
- 3. Resume

Edit <u>V</u> iew <u>P</u> roject <u>T</u> ools <u>R</u> un Scripts <u>W</u> indow <u>H</u> elp			
• 🔛 🔞 🖉 🔅 • 🔜 🍬 😃 • 😿 😂 • 🛛 🛷 •			😭 🎭 CCS Debug 📑 🤇
Debug 🛛 🔰 🙀 🕪 💷 🔳 🔍 👁 🔍 👁 📌 🍪 🖗 🍸 🗖 🗖	🗱 Variables 👬 Registers 🔀 🛱	Expressions	🗄 🌆 🖻 👫 📑 🏟 🌣 🖓 👘
🖏 ELEX7820-Lab1 [Code Composer Studio - Device Debugging]	Name	Value	Description
Texas Instruments XDS100v1 USB Emulator/C28xx (Suspended - SW Breakpoint)	A M CPUTIMER		
main() at ELEX7820-Lab1-main.c:22 0x00817B	1919 TIMEROTIM	0xFEF2	CPU-Timer 0, Counter Register [Memory Mapped]
_args_main() at args_main.c:92 0x0081B2 (_args_main has only skeletal debug info)	1919 TIMEROTIMH	0x0000	CPU-Timer 0, Counter Register High [Memory Mapped]
	1919 TIMEROPRD	0xFFFF	CPU-Timer 0, Period Register [Memory Mapped]
	1919 TIMEROPRDH	0x0000	CPU-Timer 0, Period Register High [Memory Mapped]
	1919 TIMEROTCR	0x0001	CPU-Timer 0, Control Register [Memory Mapped]
	1919 TIMEROTPR	0x0000	CPU-Timer 0, Prescale Register [Memory Mapped]
	1919 TIMEROTPRH	0x0000	CPU-Timer 0, Prescale Register High [Memory Mapped]
	1919 TIMER1TIM	0xFEF2	CPU-Timer 1, Counter Register [Memory Mapped]
	1919 TIMER1TIMH	0x0000	CPU-Timer 1, Counter Register High [Memory Mapped]
	IIII TIMER1PRD	0xFFFF	CPU-Timer 1, Period Register [Memory Mapped]
	IIII TIMER1PRDH	0x0000	CPU-Timer 1, Period Register High [Memory Mapped]
	1919 TIMERITCR	0x0001	CPU-Timer 1, Control Register [Memory Mapped]
	1919 TIMER1TPR	0x0000	CPU-Timer 1, Prescale Register [Memory Mapped]
	1919 TIMER1TPRH	0x0000	CPU-Timer 1, Prescale Register High [Memory Mapped]
	1919 TIMER2TIM	0xFEF2	CPU-Timer 2, Counter Register [Memory Mapped]
	1919 TIMER2TIMH	0x0000	CPU-Timer 2, Counter Register High [Memory Mapped]
	1919 TIMER2PRD	0xFFFF	CPU-Timer 2, Period Register [Memory Mapped]
	1919 TIMER2PRDH	0x0000	CPU-Timer 2, Period Register High [Memory Mapped]
	1919 TIMER2TCR	0x0001	CPU-Timer 2, Control Register [Memory Mapped]
	IIII TIMER2TPR	0x0000	CPU-Timer 2, Prescale Register [Memory Mapped]
	1919 TIMER2TPRH	0x0000	CPU-Timer 2, Prescale Register High [Memory Mapped]
	DEVEMU		
	SYSPWRCTRL		
	⊳ 👬 eCAP1		
	▷ M COMP1		
	▷ A COMP2		
	⊳ 👬 ePWM1		
	⊳ m ePWM2		
	▷ m ePWM3		
	⊳ m ePWM4		
	FLASH		
	•		III +

CPU Timer and GPIO Registers in Registers Window

Edit <u>V</u> iew <u>Project</u> <u>Tools R</u> un Scripts <u>W</u> indow <u>H</u> eip			
▾ 🗑 😰 🔅 ▾ 📳 🇞 😃 ▾ 🗞 📽 ▾ 🛷 ▾			🔛 🎭 CCS Debug 😨 🤇
ebug 🛛 🛛 🙀 🕪 💷 📕 🔍 👁 🔍 👁 📌 🍐 🧇 🏹 🗖	🙁= Variables 🔤 👯	Expressions	🖢 🕶 🗖 🔂 👘 🗂 🖆 🔅 🗸 🗖
ELEX7820-Lab1 [Code Composer Studio - Device Debugging]	Name	Value	Description
Texas Instruments XDS100v1 USB Emulator/C28xx (Suspended - SW Breakpoint)	A MA GPIO		
main() at ELEX7820-Lab1-main.c:22 0x00817B	Bill GPACTRL	0x00000000	GPIO A Control Register (GPIO0 to 31) [Memory Mapped]
_args_main() at args_main.c:92 0x0081B2 (_args_main has only skeletal debug info)	IIII GPAQSEL1	0x00000000	GPIO A Qualifier Select 1 Register (GPIO0 to 15) [Memory M
	IIII GPAOSEL2	0x00000000	GPIO A Qualifier Select 2 Register (GPIO16 to 31) [Memory M
	IIII GPAMUX1	0x00000000	GPIO A MUX 1 Register (GPIO0 to 15) [Memory Mapped]
	IIII GPAMUX2	0x00000000	GPIO A MUX 2 Register (GPIO16 to 31) [Memory Mapped]
	IIII GPADIR	0x00000000	GPIO A Direction Register (GPIO0 to 31) [Memory Mapped]
	1010 GPAPUD	0x00000FFF	GPIO A Pull Up Disable Register (GPIO0 to 31) [Memory Mag
	1111 GPBCTRL	0x00000000	GPIO B Control Register (GPIO32 to 38) [Memory Mapped]
	IIII GPBQSEL1	0x00000000	GPIO B Qualifier Select 1 Register (GPIO32 to 38) [Memory N
	IIII GPBMUX1	0x00000000	GPIO B MUX 1 Register (GPIO32 to 38) [Memory Mapped]
	888 GPBDIR	0x00000000	GPIO B Direction Register (GPIO32 to 38) [Memory Mapped]
	1888 GPBPUD	0x00000000	GPIO B Pull Up Disable Register (GPIO32 to 38) [Memory Ma
	IIII AIOMUX1	0xAAAAAAA	Analog I/O MUX1 Register [Memory Mapped]
	1919 AIODIR	0x00000000	Analog I/O Direction Register [Memory Mapped]
	IIII GPADAT	0x300F10F5	GPIO A Data Register (GPIO0 to 31) [Memory Mapped]
	3888 GPASET	0x00000000	GPIO A Data Set Register (GPIO0 to 31) [Memory Mapped]
	1888 GPACLEAR	0x00000000	GPIO A Data Clear Register (GPIO0 to 31) [Memory Mapped]
	1818 GPATOGGLE	0x00000000	GPIO A Data Toggle Register (GPIO0 to 31) [Memory Mappe
	1919 GPBDAT	0x000006F	GPIO B Data Register (GPIO32 to 38) [Memory Mapped]
	IIII GPBSET	0x00000000	GPIO B Data Set Register (GPIO32 to 38) [Memory Mapped]
	3888 GPBCLEAR	0x00000000	GPIO B Data Clear Register (GPIO32 to 38) [Memory Mapped
	1888 GPBTOGGLE	0x00000000	GPIO B Data Toggle Register (GPIO32 to 38) [Memory Mapp
	1111 AIODAT	0x00000000	Analog I/O Data Register [Memory Mapped]
	1888 AIOSET	0x00000000	Analog I/O Data Set Register [Memory Mapped]
	AIOCLEAR	0x00000000	Analog I/O Data Clear Register [Memory Mapped]
	IIII AIOTOGGLE	0x00000000	Analog I/O Data Toggle Register [Memory Mapped]
	IIII GPIOXINT1SEL	0x0000	XINT1 GPIO Input Select Register (GPIO0 to 31) [Memory Ma
	1919 GPIOXINT2SEL	0x0000	XINT2 GPIO Input Select Register (GPIO0 to 31) [Memory Ma
	1919 GPIOXINT3SEL	0x0000	XINT3 GPIO Input Select Register (GPIO0 to 31) [Memory Ma
	3989 GPIOLPMSEL	0x00000000	LPM GPIO Select Register (GPIO0 to 31) [Memory Mapped]
	▷ 6101 I2CA		
	•		П

Understand how the timer works by perusing "Operation of CpuTimer0.pdf" and "sprufn3c excerpt pp64-68 - CPU Timers.pdf".

Study "Peripheral Structure Hierarchy - e.g. CpuTimer0.pdf" and follow (trace through) the hierarchy of the **struct**s and **union**s that define the registers of Timer 0.

Modify the main program by adding code to do the following (retain the flashing LED code):

- Configure a GPIO pin of your choice (other than the LED one) as an output and initialize it to 0. (Choose one that is accessible on the LaunchPad board headers.)
- Write C code for an N-tap FIR filter¹ that processes non-zero² dummy data once every LED period.
- Set your GPIO pin before the FIR filter code and clear it afterwards.
 When you type in the struct/union name of the new GPIO line, use the "look-ahead" context feature.
- 4. Observe your GPIO pin on a scope to determine a rough upper bound on the processing time for an N-tap FIR filter and hence a lower bound on the corresponding sampling rate for three different values of N = 10, 50, 100. (It may be handy to trigger on the (slower) LED waveform.)
- 5. Take a scope snapshot of the pulse for N = 50 for both array and pointer code.

How many core clock cycles are needed to do N taps? Is this what you expected? Explain.

Submit to D2L a zip file named "LastnameFirstnameELEX7820Lab1.zip" that contains:

- A Word doc that includes your observations and answers to any questions that are posed in this handout or on the whiteboard, plus scope snapshot(s). (In your document that you submit, repeat the question above each answer and list them in the same order as presented in this handout.)
- Your C code file(s) for any code that you modified (*do not submit files that you did not modify*).

¹Just do an N-point dot product, no need to do circular buffering of the data.

² Your dummy data need to be non-zero: If the data are zero, the compiler will optimize out the calculations and you will not measure a valid processing time.