C File System File Functions

EXPERIMENT 1.2

Propose of the experiment

- Continue from previous experiment to be familiar with CCS environment
- Write a C language file input / output (CIO) program to read in a data file and write out a data file
- Understand how binary data will be handled by CCS differently than the ASCII text data
- The CIO feature introduced in this experiment will be used in future experiments

Start CCS

(Example: Code Composer Studio Version 5)





Create workspace

(Example: C:\User\DSP_Experiment\Ch1\Exp1.2)

🐨 Workspace	Launcher		X
Select a wo	orkspace		
Code Comp Choose a wo	oser Studio stores your projects in a folder called a workspace. orkspace folder to use for this session.		
Workspace:	C:\Users\DSP_Experiments\Ch1\Exp1.2	Browse	
🔲 Use this a	s the default and do not ask again OK	Cance	el

Go to CCS



Create a new project

(File -> New -> CCS Project)

1	CS Edit - Code Composer Studio		
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate S	rch <u>Project Tools R</u> un Scripts <u>W</u> indow <u>H</u> elp	
	New Alt+Shift+N	😰 CCS Project 🔹 🗘 🤿	r 🗈 📴 CCS Edit
	Open File	Project	□ 🗄 Outline 🛛 🔍 🗖 🗖
	Close Ctrl+W Close All Ctrl+Shift+W	Source File Header File Class	An outline is not available.
	Save Ctrl+S Save As Save All Ctrl+Shift+S Revert	Image: State of the state	
	Move	Image: State Stat	
6 9	Rename F2 Refresh F5 Convert Line Delimiters To F5	Ctrl+N	
Ð	Print Ctrl+P		
	Switch Workspace Restart		
2 2	Import Export		
	Properties Alt+Enter		
	Exit		
4			
□	Licensed		

Create a new project name

(Example: fileIO)

Eile Edit View Source CCS Project	
Create a new CCS Project.	CCS Edit
Ec C/C++ Projects ⊠ ine	x - D
□ □ □ □ □ □ □ □ □ □	is not available.
▼ Use default location	
Location: C:\Users\DSP_Experiments\Ch1\Exp1.2\fileIO Browse	
Cancel	
□* Licensed	

Select C5500 as the new project

💎 New CCS Project								
Select a type of project								
Select the platform and configurations you wish to deploy on								
Project Type C5500								
Configurations:								
📝 🛞 Debug	Select All							
Release	Deselect All							
? < <u>Back</u> <u>Next</u> > <u>Finish</u>	Cancel							

Go to next

(We do not need any additional settings)

🐨 New CCS Project	
Additional Project Settings	
Define the inter-project dependencies, if any.	
Projects C/C++ Indexer	
Referenced C/C++ Projects	
(?) < <u>B</u> ack <u>Next</u> <u>Finish</u>	Cancel

Set up the project

(Executable output, Device C5505, Library rts55x.lib)

😯 New CCS Project								
Project Settings								
Select the project settings.								
Output type Executable			•					
Project settings								
Device Variant:	<select filter=""> TMS320C5505</select>	•	More					
Device Endianness:	big	•						
Code Generation tools:	TI v4.3.8	•	More					
Output Format:	legacy COFF	•						
Linker Command File:		•	B <u>r</u> owse					
Runtime Support Library:	rts55x.lib	•	Browse					
	\smile							
? < E	ack <u>N</u> ext > <u>Einish</u>		Cancel					

Project: fileIO

😳 CCS Edit - Code Composer Studio	and the second s	
<u>File Edit View Source Refactor</u>	<u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>T</u> ools <u>R</u> un Scripts <u>W</u> indow	<u>H</u> elp
	I to	🅸 ▼ 🔗 ▼ 🗏 🗊 🗈 📴 CCS Edit
		⊖ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □
		An outline is not available.
□ [◆] Licensed	/fileIO	

Copy files to workspace

- Copy the following files cam with the book to workspace (folder fileIO)
 - Ink.cmd

- fileIO.c
- tistdtypes.h
- C55DSPUSBStickAudioTest.pcm
- Create a folder under Exp1.2 and name the folder "output"



CCS project

(The fileIO.c and lnk.cmd are included in the project)

CCS Edit - Code Composer Studio		
<u>File Edit View Source Refactor Navigate</u>	Se <u>a</u> rch <u>P</u> roject <u>T</u> ools <u>R</u> un Scripts <u>W</u> indow <u>H</u> elp	
<mark>□1 ▼ </mark>		1 🗈 🔂 CCS Edit
□ C/C++ Projects	- E	🗄 Outline 🛛 🔍 🗖 🗖
		An outline is not available.
A 🛱 fileIO		
▶ ▶ ■ Incluses ▶ ■ fileIO.c		
🕨 🖻 tistdtypes.h		
C55DSPUSBStickAudioTest.pcm		
lø Ink.cmd		
Licensed	/fileIO	

Setup build environment

- Right click on fileIO then select Property
- Select and expand C/C++ Build option
- Select Settings, then Runtime Options
- Set type size to 16 and memory model to large

Set target configuration

- Right click on Project->New->Target
 Configuration File
- Create a target configuration file name, fileIO.ccxml
- SelectTexas Instruments XDS100v2 USB Emulator
- Check the box for USBSTK5505
- Save the configuration

Launch target

- From Target Configuration window
- Open Project and right click on fileIO.ccxml
- Select Launch target configuration
- In Debug window, right click on Texas Instruments XDS100v2 USB Emulator_0/C55xx
- Select Connect Target to launch the target
- You shall see target reset and configured automatically

Build the project

(Project->Build All)

😚 CCS Edit - fileIO/filIO.ccxml - Code Com	poser Studio		
File Edit View Navigate Search Pro	ject Tools Run Scripts Window Help	_	
	Open Project Close Project	🎄 ▾ 🛛 🛷 ▾	🗐 🗊 📑 🎭 CCS Debug [>>
E C/C++ Projects ⊠ 🖬	Build All Ctrl+B		🗄 O 🖹 T 🎬 Di 🛛 🗖
수 수 🗟 🗖 💲	Build Configurations		No debug context ~
🖾 fileIO	Build Project		Enter location here 👻
🗱 Binaries	Build Working Set		
🔊 Includes	Clean	but the target.	
🔁 Debug	Build Automatically	Emulator 🗏	
C fileIO.c	Import Existing CCS/CCE Eclipse Project		
C55DSPLISBStickAudioTest r	Import Legacy CCSv3.3 Project		
fillO cccml [Active/Default]	Add Files		
link.cmd	Link Files	28335	
	Properties		
	Experimenter's Kit - Piccolo Experimenter's Kit - Piccolo OSK5912 Cosole Source C-Build [fileIO] V S S S C-Build [fileIO] V S S S C-Build [fileIO] V S S S S C-Build [fileIO]	F28027 F28035	
∎* Licensed	ke\gmake -k all gmake: Nothing to be done for `all'.	•	

Load program fileI0.out

- From CCS Run->Load->Load Program
- Blows and select the executable file fileiO.out from C:\Users\DSP_Experiments\Ch1\Exp1.2\fileI O\fileIO.out
- Click Open and then OK to load the program

Once the program is loaded

(Program counter is at entry point of function main())

💱 CCS De	bug - fileIO/fileIO.c - Code Composer St	tudio									X
File Edit	View Source Refactor Navigate	Search Project	Tools Run	Scripts \	Window H	lelp					
1 📬 🗖 🗔	I 🗈 📄 🔳 🍬 🖉 🔻 🕅 🔯	• • • •	· 🤌 🖋 🔻	• 🗐 🎤 😜	图 • 6	• 🐤 🔶 •	• 🜩 •		🖹 🎭 C	CS Deb	ug [»
ᅓ Debug	X		🛛 🕬= Variable	es 🖾 🕼 Ex	pressions		🗣 Brea	akpoints	; 🛙 🗌		
🎽 🕩 I	00 🔳 🕅 🔊 👁 🔊 🖉 🔤 🕅	襐 ▾ 🕹 🖗 ♡		<u>∦</u> ⊂ \$	🖻 🕸 🕯	*** 🛛					
🔺 🕄 filIC	D.ccxml [Code Composer Studio - Devic	e Debugging]	Name	Туре	Value	Location	8 -	× ¾	i - 🧐	8	
4 🖉	Texas Instruments XDS100v2 USB Emula	tor_0/C55xx (Susp	• fp1	struct un	0x5192B	0x00001	₽ ₽	\$∎¢			
	main() at fileIO.c:24 0x000234B1	-	♦ fp2	struct un	0xF18A	0x00001	Ide	Na	Con	Cou	Acti
	args_main() at args_main.c:25 0x000.	243F0 (_args_mai	(×)= j	unsigne	2393348	0x00001		Brea		0 (0)	Rem
•	III	4									
🖹 filIO.cc	xml 🚺 fileIO.c 🛛 🖻 tistdtypes.h	C 0xff6adc				- 8	🕄 Targ	et Co	🗳 Cor	nsole 🛛	
23 voi	d main()						C-Build	[fileIO]			
≥24 {							₽ 0(€ } ∏:	a: 🖍		
25	FILE *fp1, *fp2;	// file poi	Inters		1		er 🗉	▼ ■ 1			
20	01nt32 1;	// unsigned	i long in	iteger u	sed as a	a co					
28	printf("Exp. 1.2 file	IO(n");					****	Build	of		
29							confi	gurat	ion I)ebug	
30	<pre>fp1 = fopen("\\C55DSPUS</pre>	BStickAudio	[est.pcm"	', " <u>rb</u> ")	;		for p	rojec	t fil	.eIO	
31	fp2 = fopen(" .\\\\o	utput\\C55DS	SPUSBStic	kAudioT	est.wav	", " _	****				
32	if (from $-$ NULLE)	// chock if	the inn	ut file	ouista	_	C·\+i	\ccsv	5\11t i	1s\m	nak
34	((IDI NOLL)	// CHECK II	. une inp	Juc IIIe	EXISUS		e\ama	ike -k	all	.10 (gi	licent
35	printf("Failed to ope	n input file	e 'C55DSE	USBStic	kAudioTe	est.	gmake	: Not	hing	to be	э
36	<pre>exit(0);</pre>	-					done	for `	all'.		
37	}										
38	f> (f2) 44 0) -										
40 i=0:				TTE bol	nu 44 D	ytes					
41	while (fread(ch, sizeof(U	int8), SIZE,	fp1) ==	= SIZE)	// read	SIZ 🔻					
•	III III					•					-
□◆	Licensed	N	/ritable	Sma	ert						

Use Resume Run the program

(CCS console displays experiment messages)

CCS Debug - fileIO/fileIO.c - Code Composer Studio									
<u>F</u> ile Edit <u>V</u> iew <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>T</u> ools <u>R</u> un Scripts <u>W</u> indow <u>H</u> elp									
🗂 🕶 🔄 📾 🐘 🚇 🔻 🔊 🛇 🦞 🗸 🖏 🗮 🐲 🖉 🥓 🖛 ಶ 🖋 🗸 👘 🖓 🗸 👘									
≫ Debug ⊠ □ ∞ Varia ⊠ ∰ Expr □ □ ₯ Help ⊑ Console ⊠ □ □									
🍇 🕨 🗉 🗧 🕴 🤽 👁 א 👁 🕫 🚁 🛼 🔅 🕸 👻 🍓 🗸 🕹 🚳 🏹 🌆 🐗 🗖 🔞 🖉 💥 🍇 🏹 fillo	D.ccxml:CIO								
💱 filIO.ccxml [Code Composer Studio - Device Debugging] Name Type									
Pexas Instruments XDS100v2 USB Emulator_0/C55xx (Suspended)	p. completed								
■ abort() at exit.c:60 0x000245A2 (abort has only skeletal debug	p. 1.2 file IO								
204	48								
	72								
۲ · · · · · · · · · · · · · · · · · · ·	96								
FillO.ccxml I tistdtypes h I fileIO.c X I abort() at /sim/sds12/scratch/a0322878 rt	20								
	68								
23 void main() 81	92								
25 FILE *fp1,*fp2; // file pointers 92:	16								
26 Uint32 i; // unsigned long	240								
27	288								
28 printf("Exp. 1.2 file IO\n");	312								
143	336 🗉								
31 fp2 = fopen(".\\\\output\\C55DSPUSBStickAudioTest.wav" = 15	360								
32	384								
33 if (fp1 == NULL) // check if the : 18	432								
34 { 35 printf("Failed to open input file 'C55DSPUSEStickAudioTe	456								
36 exit(0);	480								
37 }	504								
38 223	552								
39 fseek (fp2, 44, 0); // advance output 24	576 _								
	enn								
□ [◆] Licensed Writable Sma…ert									

Verify the program result

- After running the experiment, your program shall create a WAV file named as "C55DSPUSBStickAudioTest.wav" in the output folder that you have created
- Play this WAV file using a player such as Windows Media Player. You shall hear the male voice saying: "C55 DSP USB Stick Audio Test".

New experiment assignments

Write a C program that will

- Read the input data file, C55DSPUSBStickAudioTest.pcm, as Exp1.2
- Open a file to write the data in ASCII integer, the name of the output file can be "C55DSPUSBStickAudioTest.xls"

hint: use fprintf() function to replace fwrite()

- Convert every two -byte input data values into an integer (16-bit) number as:
 o = ch[j] | (ch[j+1]<<8); // where j = 0, 2,
 - and write input data in ASCII integer to output file
- Build and run the program to generate your output file
- using Microsoft Excel to open the output file, select the data column to plot the data to view the waveform of "C55DSPUSBStickAudioTest", see next page
- Write a C program that will
 - Read the input file "C55DSPUSBStickAudioTest.xls" created in above experiment and generated a WAV file output
 - Play the WAV file on a computer and listen to the audio.
 - Q1: Does it sound the same as the WAV file output obtained from the experiment given in section 1.5.2 (Exp1.2)?

The new experiment result

(The waveform of the audio C55 DSP USB Stick Audio Test)

	_ •) • (*	ッ・ 🔍 - 🗧 C55DSPUSBStickAudioTest.xls - Microsoft Excel			Chart	: Tools		×			
Fil	e Home	Insert Page Lay	ou Form	ulas Data	Review View	Universal D	Design Lay	yout Format	▽ 🕜 ⊏	- 6	23
	Chart 1	• (*	f_x								~
	А	В	С	D	E	F	G	Н	I		
1	20	(i				3333					
2	28	25000 -									
3	36	20000 -									_
4	39	15000		1	1		l i				_
5	55	15000		h							_
6	57	10000 -									-
7	56	5000 -									-
8	66	0 -				T T					_
9	62	-5000 -		156 941 126	311 366 381	321 326 306	761 761 246	731 216	Series1		_
10	63	10000	й Г	4 22 2 2	x 11 11 10 x	116: 19: 19:	23, 23, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25	28.2			_
11	74	-10000 -									-
12	60	-15000 -				╫╢─╢─		 			-
13	49	-20000 -	 		╀╴╀╴┼	_ [[-
14	58	-25000 -			i i r	P				_	-
15	61	-30000 -									-
16	58					9999				4	-
1/	48										
	► ► <u>C55</u>	DSPUSBStick	AudioTe	st 🔏							
Read	У	Averag	e: 0.57169	3157 Cou	int: 29696 Sur	m: 16977		0% 🕒		(±	

Programming quick review 1

- This experiment used a few more C file IO functions, such as *fopen()*, *fread()*, *fwrite()*, and *fclose()*. These functions are defined in the header file stdio.h.
- The *fopen()* and *fclose()* are used to open and close the files used in the experiment. The *fread()* and *fwrite()* are used to read data and write data from and to the input and output devices.
- The *fopen()* function has two arguments, the first argument is a character string for the name of the file and the second argument is also a character string telling the function how to use the file opened. The second argument "rb" and "wb" used in the fileIO experiment are for reading binary data in and writing binary data out.
- For any file opened, it must be properly closed after the operation. This can be done by the *fclose()* function.
- The *fread()* and *fwrite()* functions have 4 arguments. The *fread* reads from stream (4th argument) into the array (1st argument), the number of data (3rd argument) with the size of (2nd argument). The *fwrite* writes data in the array (1st argument) to the stream (4th argument), the number of data (3rd argument) with the size of (2nd argument).
- The *fseek()* and *rewind()* functions are used to set the starting point of the files for writing the output data. These functions help manage the file access.

Programming quick review 2

- Clanguage has many different data types.
- To improve portability of the programs written for one type of computer devices to another, we use a unique type define header file, tistdtypes.h, to specify the data type to avoid any ambiguity.
- The following data types are used in this experiment
 - Uint8 (unsigned 8-bit integer data, unsigned char)
 - Uint32 (unsigned 32-bit integer data, unsigned long)
 - FILE * (file pointer)

Special notice: *fread()* and *fwrite()* functions in CCS these functions will only read or write byte size (8-bit) binary data during memory access. When the *fread()* function reads a 16-bit data into a data memory location, for example 0x1234, it reads 0x34 first then 0x12, and place them into memory ch[0]=0x34 and ch[1]=0x12. When *fwrite()* function writes data 0x1234. It only writes 0x34 but not write 0x12. So it is important to convert each 16-bit data into two 8-bit data and read or write two bytes for experiment.



 C Programming Language (2nd Edition), by Brian Kernighan and Dennis Ritchie, Prentice Hall